Robust Fine-Grained Image Classification (PhD)

A THESIS SUBMITTED TO
THE SCIENCE AND ENGINEERING FACULTY
OF QUEENSLAND UNIVERSITY OF TECHNOLOGY
IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Zong-Yuan Ge

School of Electrical Engineering and Computer Science
Science and Engineering Faculty
Queensland University of Technology

September 2016

Copyright in Relation to This Thesis

Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signature:

Date:

To Tong

Abstract

The general object classification task distinguishes very different object categories, such as a house and a bird. In contrast, fine-grained image classification aims to answer the question of given a bird image: which bird species is it? In a more specific way, it is about species and sub-category classification.

This is a challenging task for two reasons. Firstly, some classes (species) from the same category, such as fish, have a very similar appearance leading to low inter-class variation. Secondly, a high degree of variability is prone to occur even within the same class due to large pose, lighting, and illumination variations in the natural environment. To deal with these challenges, much of the work has proposed parts-based modelling to explicitly or implicitly find local parts and attributes to locate subtle differences in appearance across species.

This thesis explores methods to improve fine-grained classification. Firstly, we present a novel method to deal with intra-class variability by extending the idea of inter-session variability modelling (ISV), used for face recognition, to the fine-grained classification task. We extend ISV by modelling local variations (local ISV) and empirically demonstrate that this considerably improves performance. Next, we introduce an automatic subset pre-clustering framework which allows us to learn discriminative features for each subset (subset feature learning). Subset feature learning allows us to learn features specific for each subset. This leads to considerable improvements in performance, however, its performance is limited by its ability to select the correct subset at test time. To overcome this limitation we present a mixture of deep convolutional neural networks (MixDCNNs) which probabilistically assigns each sample to a subset. Finally, we explore the usage of both spatial and temporal information and demonstrate the potential gains that can be made for the task of fine-grained bird classification.

Table of Contents

Keywords

Deep Convolutional Neural Network, Fine-Grained Classification, Video Classification, Gaussian Mixture Models, Feature Engineering, Inter-Session Variability Modelling

Acknowledgments

I would like to give my sincere appreciation to my supervisors Dr. Christopher McCool, Senior Scientist Conrad Sanderson and Professor Peter Corke. It has been an honour for me to working with them. First, I would like to thank Chris for guiding me to the field of computer vision. He has taught me how to conduct a good research project and gave me much support. Most importantly, he showed me the way how to write a good research paper. Next, I would like to thank Conrad for fruitful discussions and for providing high-level research ideas. I would also like to thank Peter for being so patient with me, and giving me advice on life and careers. My three supervisors fight tirelessly to push scientific boundaries. They will be my research idols for the rest of my life.

It has been a great journey working with many talented people in our centre: Centre Chief Operating Officer (COO) Sue Keay, Associate Professor Ben Upcroft, Alex Bewley, Zetao Chen, Gavin Suddrey, Fangyi Zhang and Kanes Anantharajah. I would like to thank Professor Chunhua Shen for inviting me to visit University of Adelaide. During my PhD, I did an amazing summer internship at DeepGlint, Beijing. I would like to thank to my internship mentors Dr. Yong Zhao and Dr. Peng Ding.

Last, I want to dedicate all the works to Tong and my parents for their love and support through my PhD journey. I am glad I spent three years here at Cyphy/ACRV lab, I regard this as my second home in Australia.

Acronyms

BoV Bag of Visual Words

CLEF Conference and Labs of the Evaluation Forum

CNN Convolutional Neural Network

CONV Convolutional

CRF Conditional Random Field

CUB Caltech-UCSD Birds

CVPR Computer Vision and Pattern Recognition

CVPRW Computer Vision and Pattern Recognition Workshop

DCNN Deep Convolutional Neural Network

DeCAF Deep Convolutional Activation Feature

DNN Deep Neural Network

DPD Deformable Part Descriptor

DPM Deformable Part Model

DoG Difference of Gaussian

ECCV European Conference on Computer Vision

FC Fully-connected

FK Fisher Kernel

FV Fisher Vector

HoG Histogram of Oriented Gradients

ICIP International Conference on Image Processing

ISV Inter-Session Variability modelling

LBP Local Binary Pattern

LDA Linear Discriminant Analysis

MixDCNN Mixture Deep Convoultional Neural Network

POOF Part-based one-vs-one Features

RNN Recurrent Neural Network

SIFT Scale Invariant Feature Transform

SURF Speeded-up Robust Features

SVM Support Vector Machine

WACV Winter Conference on Applications of Computer Vision

Chapter 1

Introduction

1.1 Overview

Nowadays, countless multi-media resources are being uploaded by people around the world to the Internet everyday. There were about 880 billion images being taken and uploaded in 2014. It brings us a major challenge to analyse and understand all these images. Image classification, specifically object classification, serves as an automatic way to interpret, understand and process images. Object classification has been a major focus of research in the computer vision and machine learning communities in the last decade [Fergus et al., 2003, Krizhevsky et al., 2012, Perronnin et al., 2010]. It focuses on identifying objects in images. For example, an image showing a persian cat is classified with a cat label. Many real-world applications based on object recognition have been developed for the purpose of automatic image tagging, image captioning and user interest analysis [Chen et al., 2013, Karpathy and Fei-Fei, 2015].

General object classification is limited in its ability to understand image content at a deeper level. For example, answering the question of whether a bird is presented in the image is easy, but to tell which bird species is presented is impossible using a general objection classification system. Because constructing an object classification system to recognise bird species requires considerable domain expertise to design a classifier that transforms the raw pixel values of an image into a representation which could detect and classify a specific pattern between several visually similar bird species. Recently, there has been an increasing interest in the research of sub-category classification, also known as fine-grained classification. Fine-grained classification is a relatively new field and serves as a sub-field for object classification research.

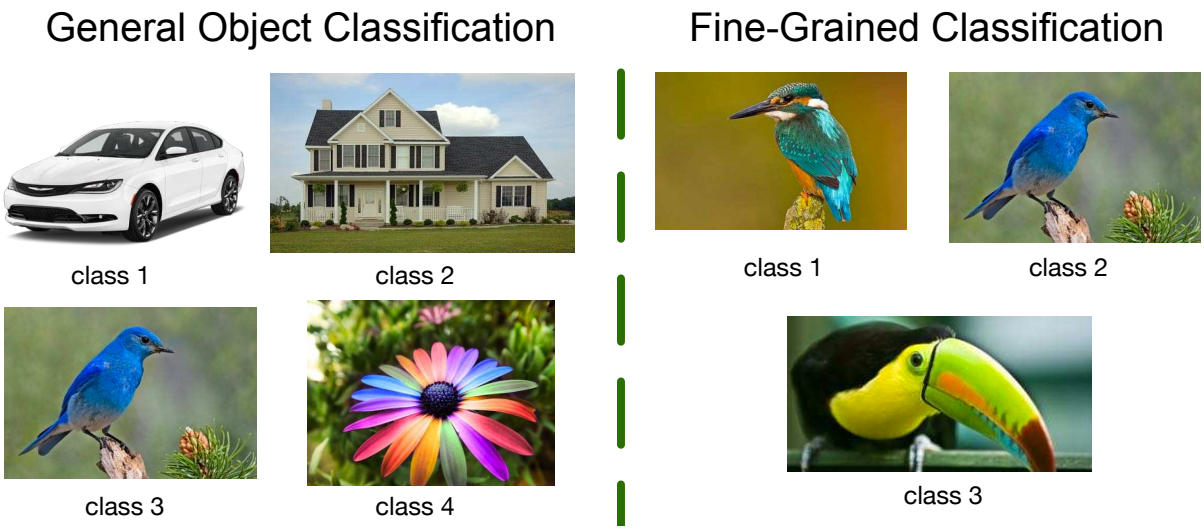


Figure 1.1: Figure shows the concept of general object classification versus fine-grained classification. The general object classification usually refers to distinguishing very different object categories such as a car category and a house category. In fine-grained classification problem, all classes belong to the same basic category, but are different bird species.

Distinct from general object classification, which aims to find the correct overall category such as a bird, dog or plant, fine-grained image classification aims to identify the particular sub-category [Belhumeur et al., 2008, Kumar et al., 2012, Liu et al., 2012, Parkhi et al., 2012] One typical example for fine-grained classification is bird classification. The difference between object classification and fine-grained classification is illustrated in Figure. 1.1.

The objective of fine-grained object classification is to identify what sub-category (species) is present. It enables human beings to further extend image and video understanding by providing greater detailed information about the objects present in the image or video. For instance, video cameras embedded with a bird classification algorithm could be used to recognise a rare bird species as well as endangered species. Another example is a food classification system installed in mobile phones can help obese people to calculate and control calorie consumption.

While general image classification has progressed at a rapid pace in the past few years, it is still a challenging task to perform accurate fine-grained classification of object sub-categories. There are three aspects that make fine-grained image classification a challenging computer vision problem, which are illustrated by taking bird species as an example, see Figure. 1.2. The first challenge is the large variations in pose, illumination, and environments within the same species (intra-class variation). Birds usually live in outdoor environments across various

Figure 1.2: Example images from the bird dataset which exhibit large intra-class variations and low inter-class variations. Each column represents a unique class.

habitats such as tropical forests, coastline and urban areas. Therefore, photos are taken under different scenarios with day-time and night-time light changes. The second challenge is the subtle differences between some bird species (inter-class variation). Some bird species have identical shapes. Sometimes they even share very strong colour and texture similarity. The third challenge is the limited number of annotated images available for each species. It is very difficult for humans without expert knowledge of birds to annotate ground-truth labels for images and videos.

1.2 Research Questions

The research objective of this thesis is to investigate a general and robust fine-grained classification system to categorise different sub-categories in challenging scenarios. This is critical for establishing a more detailed understanding of the visual world. The main question this thesis aims to answer is: "*How can images or videos of sub-categories in challenging scenarios be robustly classified?*" This broad question can be broken into three smaller sub-questions.

- "Can we model different instances of the same class under various environments (large intra-class variations)?"

For fine-grained classification, the same class can differ considerably in appearance (see Figure. 1.2). In the field of biometrics, such as face and speaker verification, probabilistic models such as inter-session variability modelling have been proposed and obtained good results. We explore these techniques for fine-grained fish and food recognition tasks. In visual biometrics, there is a strong assumption that all images are well-controlled regarding lighting, distance from camera, viewpoint etc. This remains a challenge in fine-grained classification because the images are taken in uncontrolled environments where nuisance from background noise and motion blur is presented. To deal with those challenges, we extend the session variability modelling into a local region based model where subtle session variations in local regions are better modelled.

- "Can we learn robust and discriminative features in order to classify fine-grained classes which have small inter-class variations?"

  Fine-grained tasks are challenging due to the subtlety of their class differences. In order to distinguish visually similar classes, it is important to generate discriminative feature representations for different classes. Pre-clustering similar classes into one subset and learning subset-specific features for each subset can substantially improve the capacity of feature representation and make it possible to learn more discriminative features in order to distinguish classes which have high visual similarities.

- "Can we exploit temporal information available in videos to improve robustness of fine-grained classification?"

  Prior work treats the fine-grained classification task as a still-image classification problem and ignores the large number of videos available of different fine-grained classes. The videos are a rich resource in terms of complementary temporal information and extra training samples regarding different poses and viewpoints. We examine these questions by evaluating multiple DCNN architectures that each take a different approach to combining information across the time domain. We apply the bilinear DCNN in a novel manner to jointly exploit spatial and temporal information. In brief, we aim to investigate information and decision fusion techniques of different sources based on the assumption that multi-classification systems complement each other.

1.3 Contributions

The main contributions of this thesis are summarised as follows:

1. We apply inter-session variability (ISV) modelling to fine-grained classification of two datasets: fish swimming in a natural environment, and different types of food on plates in a table setting. We demonstrate that the proposed system can achieve better performance compared to traditional previous approaches. We then propose a novel extension to ISV which is called local ISV, so that local region based inter-session variations could be modelled. We then introduce deep convolutional neural network (DCNN) to generate low-dimensional feature representations in conjunction with the local ISV model (see Chapter 3).

2. We propose a novel hierarchical learning framework, which operates in a fully automatic manner and can be used to learn discriminative subset-based classifiers and features for the fine-grained classification problem. Unsupervised pre-clustering is performed to split visually similar classes into subsets and subset-specific features are then learnt and classifiers for each subset (see Chapter 4).

3. Leveraging the previous work in subset feature learning, we propose a model that can probabilistically combine multiple DCNNs where each DCNN has been trained on a subset. To do so, a novel mixture of DCNNs is proposed (MixDCNN) which allows us to jointly train an end-to-end network. It obviates the performance loss of two stage hierarchical system by making the final classification decision summed up from each DCNN component weighted proportionally to the confidence of its decision (see Chapter 5).

4. We introduce the problem of video-based fine-grained object classification, and explore several methods to exploit the temporal information. A corresponding new dataset is proposed to evaluate the proposed bilinear DCNN, which extracts local co-occurrences by combining information from the convolutional layers of spatial and temporal DCNNs (see Chapter 6).

1.4 Outline

Much of the work in this thesis has been peer reviewed and published as conferences papers. The outline of the thesis is as follows:

Chapter 2 presents an overview of prior work in object and fine-grained classification. It examines contributions to the core aspects of fine-grained classification problem: supervised and unsupervised parts modelling, feature engineering, transfer learning, and lastly reviews, video classification and lastly reviews the current datasets being evaluated for fine-grained classification.

Chapter 3 introduces local region based inter-session variability (ISV) modelling using deep convolutional neural networks (DCNNs). Two contributions are made. First we introduce the concept of local inter-session variability modelling by partitioning each image into N by N regions $(R_1, ..., R_{N^2})$ and learn a separate ISV model for each local region R_i. Second, we introduce bottle-neck features for DCNNs so that a low-dimensional DCNN representation can be used in conjunction with the ISV model; the DCNN features are usually high dimensional $D = 4096$ and we show that this can be reduced to $D = 128$ dimensions using the proposed bottle-neck features. We then demonstrate the efficacy and effect of this technique on a challenging real-world fish dataset which includes images taken underwater. We also use it on a database of food images taken by mobile devices, providing significant real-world session variations.

Having discussed the importance of local modelling approaches in two related applications, **Chapter 4** discusses the potential of using hierarchical clustering for fine-grained classification. In the first part of Chapter 4 we present a novel method for fine-grained image classification for bird species based on a hierarchical structure. Our automatically generated hierarchical system is inspired by the idea of forming a similarity tree where classes with strong visual correlations are grouped into subsets. An expert local classifier with strong discriminative power to distinguish visually similar classes is then learnt for each subset. In the second part we propose a learning system which learns deep convolutional neural network (DCNN) features specific to each subset to learn a more discriminative feature representation.

Chapter 5 presents a novel deep convolutional neural network (DCNN) system for fine-grained image classification, called a mixture of DCNNs (MixDCNN). We provide a formulation to perform joint end-to-end training of multiple DCNNs simultaneously. The output from

each of the DCNNs is combined to form a single classification decision. This is in contrast to subset feature learning in Chapter 4, where each expert is used only for feature extraction. We evaluate this proposed MixDCNN system on bird and plant datasets. It outperforms previous state-of-the-art subset feature learning system and general ensemble DCNNs.

In **Chapter 6**, we present the novel task of video-based fine-grained object classification, and perform a systematic study of several recent deep convolutional neural network (DCNN) based approaches, which we specifically adapt to the task. We evaluate three-dimensional DCNNs, two-stream DCNNs and bilinear DCNNs. Two forms of the two-stream approach are used, where spatial and temporal data from two independent DCNNs are fused either via early fusion (combination of the fully-connected layers) and late fusion (concatenation of the softmax outputs of the DCNNs). For bilinear DCNNs, information from the convolutional layers of the spatial and temporal DCNNs is combined via local co-occurrences. We then fuse the bilinear DCNN and early fusion of the two streams to combine the spatial and temporal information at the local and global level (Spatio-Temporal Co-occurrence). These algorithms are evaluated on the new and challenging video dataset of birds which we have developed and released.

1.5 Publications

1. K. Anatharajah, **Z. Ge**, C. McCool, S. Denman, C. Fookes, P. Corke, D. Tjondronegoro and S. Sridharan, Local Inter-Session Variability Modelling for Object Classification. (Published) IEEE Winter Conference on Applications of Computer Vision (WACV), Steamboat Springs CO, 2014.

2. Z. Chen, S. Lowry, A. Jacobson, **Z. Ge**, M. Milford, Distance Metric Learning for Feature-Agnostic Place Recognition (Published) IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Hamburg, Germany, 2015.

3. **Z. Ge**, C. McCool, C. Sanderson and P. Corke, Modelling Local Deep Convolutional Neural Network Features to Improve Fine-Grained Image Classification. (Published) IEEE International Conference on Image Processing (ICIP), Quebec City, Canada, 2015.

4. **Z. Ge**, C. McCool, C. Sanderson, A. Bewley, Z. Chen and P. Corke, Fine-Grained Bird Species Recognition via Hierarchical Subset Learning. (Published) IEEE International Conference on Image Processing (ICIP), Quebec City, Canada, 2015.

5. **Z. Ge**, C. McCool, C. Sanderson and P. Corke, Subset Feature Learning for Fine-Grained Category Classification. (Published) Computer Vision and Pattern Recognition Deep Vision Workshop (CVPRW), Boston, 2015.

6. **Z. Ge**, C. McCool, C. Sanderson and P. Corke, Content Specific Feature Learning for Fine-Grained Plant Classification. (Published) Labs of Evaluation Forum (CLEF), working notes, Toulouse, France, 2015.

7. **Z. Ge**, A. Bewley, C. McCool, C. Sanderson, B. Upcroft and P. Corke, Fine-Grained Classification via Mixture of Deep Convolutional Neural Networks, (Published) IEEE Winter Conference on Applications of Computer Vision (WACV), 2016.

8. A. Bewley, **Z. Ge**, L. Ott, F. Ramos, B. Upcroft, Simple Online and Realtime Tracking (Under Review) IEEE International Conference on Image Processing (ICIP), Arizona, USA, 2016.

9. **Z. Ge**, C. Mccool, C. Sanderson, P.Wang, L.Liu, I.Reid and P. Corke, Fine-Grained Video Classification via Spatial and Temporal Encoding, (Accepted) International Conference

on Digital Image Computing Techniques and Applications (DICTA), Gold Coast, Australia, 2016.

Chapter 2

Literature Review

Object classification and fine-grained classification differ significantly. For general object classification problems, category differences are salient because the object types and attributes are distinct from each other. For instance, to distinguish a car from a house, it is easy to locate the differences in multiple aspects such as texture and shape. By contrast, for fine-grained classification the differences between classes are visible due to subtle changes in terms of colour, texture and shape because they belong to the same category (such as bird species). In this chapter we will present an overview of related work to object classification and fine-grained classification.

Object classification has been a major focus of research in the computer vision and machine learning communities in the last decade, and considerable progress has been made [Fergus et al., 2003, Gehler and Nowozin, 2009, Krizhevsky et al., 2012, Mutch and Lowe, 2008, Perronnin et al., 2010, Ponce et al., 2006]. A variety of techniques have been proposed and have achieved impressive results on some popular datasets such as the PASCAL VOC dataset [Everingham et al., 2010] and Caltech 256 [Griffin et al., 2007]. These techniques include feature encoding methods such Fisher vectors (FV) [Perronnin et al., 2010] and Histogram Encoding [Chatfield et al., 2011] through to the more recent advent of deep learning [Krizhevsky et al., 2012].

A sub-field of object classification called fine-grained classification has made great progress in recent years [Kumar et al., 2012, Parkhi et al., 2012, Wah et al., 2011b]. The prior work in fine-grained classification can be roughly divided into two tracks. The first is to localise the discriminative object parts in the image to compensate for nuisance variations such as

pose.Many parts-based methods with geometric constraints have been proposed for bird classification [Zhang et al., 2014], cars [Krause et al., 2014], and dogs [Parkhi et al., 2012]. Some of the works explicitly use parts annotations from the dataset to train a strongly supervised parts detector [Chai et al., 2013a, Krause et al., 2014, Zhang et al., 2014, 2013b] to reduce the effect of the nuisance variations such as pose and viewpoint. However, these approaches often require not only ground-truth bounding boxes of the bird's (or other fine-grained object's) location but also annotations which provide the location of interest parts. Labelling parts for hundreds or thousands of fine-grained domains is laborious and cost-prohibitive. It is an interesting research direction to free the algorithm from detailed annotations. Recent work has examined ways to alleviate this problem by exploring methods to derive weakly supervised or unsupervised parts detection models [Goring et al., 2013, Jaderberg et al., 2015, Krause et al., 2015b, Lin et al., 2015].

The second track is to derive discriminative and robust features. Classic hand-crafted feature descriptors such as the Scale Invariant Feature Transform (SIFT) [Lowe, 2004a], Histogram of Oriented Gradients (HoG) [Dalal and Triggs, 2005a], and Color Histogram [Van De Weijer et al., 2009] which take advantage of color, texture and edge information have been successfully translated from general object classification to the fine-grained classification domain [Duan et al., 2012, Yao et al., 2011]. Others methods such as the Part-based One-vs-One Features (POOFs) [Berg and Belhumeur, 2013] focus on modelling corresponding parts activation, and have been derived specifically for fine-grained classification. More recently, deep convolutional neural network (DCNN) approaches for general object classification have been transferred to achieve state-of-the-art performance for fine-grained classification by applying transfer learning [Krause et al., 2015b, Xu et al., 2015, Zhang et al., 2015]. Below we describe the prior work within these two tracks and then summarise progress that has been made in video classification, an area that is explored within this thesis. We then end with a review of datasets relevant for fine-grained classification.

2.1 Parts-based Modelling

Many of the categories, such as animals and flowers, that fine-grained image classification is applied to are highly deformable. This has led researchers to examine the potential to localise the relatively rigid parts prior to performing classification, because this may ameliorate the

negative effects caused by pose and viewpoint variations. Methods that perform parts-based modelling can be split into two categories: (i) supervised approaches which learn to recognise the parts from an annotated dataset and (ii) unsupervised approaches which attempt to learn consistent parts from a given dataset.

2.1.1 Supervised Parts Modelling

Supervised parts modelling refers to a setting where parts labels or keypoints are explicitly provided when training a model. We review a keypoint-based model where the explicit location of each annotation (such as beak, right eye, left eye of bird) is used to either train a poselets model [Bourdev and Malik, 2009] or keypoint-based segmentation [Xie et al., 2013]. Then we move onto a parts-based model which focuses on patch-based parts modelling [Zhang et al., 2013b]. Lastly, we briefly mention the "human in the loop" method where users are required to give feedback to the computer during the model training and testing process [Wah et al., 2011a].

Keypoint-Based Model

Keypoint-based methods like poselets [Bourdev and Malik, 2009] are helpful to localise discriminative parts of objects. Poselet is a pose estimation method based on the correspondence of configuration in addition to appearance of the object parts. The key idea is to define parts that are tightly clustered both in configuration space (which can be parametrized by the locations of various keypoints) and in appearance space (can be parametrized by pixel values in an image patch). The poselets are created by a search procedure. A patch is randomly chosen in the image of a randomly picked object (the seed of the poselet), and other examples are found by searching in images of other objects for a patch where the configuration of keypoints is similar to that in the seed. Given a set of examples of a poselet, which are, by construction, tightly clustered in configuration space, HoG features are computed for each of the associated image patches. These are used as positive examples for training a linear support vector machine (SVM). At test time, a multi-scale sliding window is used to find strong activations of the different poselet filters. Such an approach was implemented by Farrell et al. [2011] for birds as birdlets. Their model associates the underlying image patterns with volumetric part locations. Birds in images are then modelled by a configuration of volumetric parts. This work was later extended by Zhang et al. [2012], with the contribution that the parts model was learnt with fewer representational

assumptions. This method is able to compensate for variations in pose and different camera viewing angles by using the ensemble of responses of specific pose-keypoint configurations.

Transferring parts or keypoints to novel datasets is another active research field in computer vision. One example is exemplar-based method proposed by Liu et al. [2012]. Belhumeur et al. [2011] on localising fiducial points in human faces, Liu et al. [2012] predict accurate locations of dog eyes and noses by learning exemplar-based geometric and appearance models from the dog training dataset. However, this method is parametric-based and is sensitive to novel samples. Goring et al. [2013] proposed a non-parametric parts transfer method. The method is very simple but has strong non-linearity. To locate parts of the test sample, first the similar overall layout of the object of interest is found using the training dataset with HoG as features. After that, the parts locations are obtained from the annotations of K training images, which are scaled proportionally to the bounding box of the test image. This non-parametric parts model can alleviate the high variation in part positions that arises from the large number of different poses of objects in a limited number of images. The advantage of non-parametric methods allows for coping with high degree of pose and view variations in unseen images where traditional detection models like deformable part model (DPM) [Felzenszwalb et al., 2010] and exemplar-based method fail. This is valuable for fine-grained problems because intra-class variations are extremely high.

Deformable Models

Another widespread object detection approach to performing parts localisation is the deformable model approach. An example of this is the DPM [Felzenszwalb et al., 2010] which is an object detection system based on a set of multi-scale individual part models, see Figure. 2.1. This concept can be described by the equation:

$$E^{DPM}(I, model) = F^{root} \cdot \Phi(I, p_0) + \sum_{n=1}^{N} E_{part}(p_n, I, model) \tag{2.1}$$

$$E_{part}(p_i, I, model) = F_n^{model} \cdot \Phi(I, p_n) - \Phi_d(dx_n, dy_n) \tag{2.2}$$

where E or E^{DPM} or E_{part} can be interpreted as the matching score between a trained model and a given image I. The trained root model and the n-th part model are described by F_{root} and F_n^{model}. The feature vector at location p_n in an image is defined as $\Phi(I, p_n)$. To penalize the

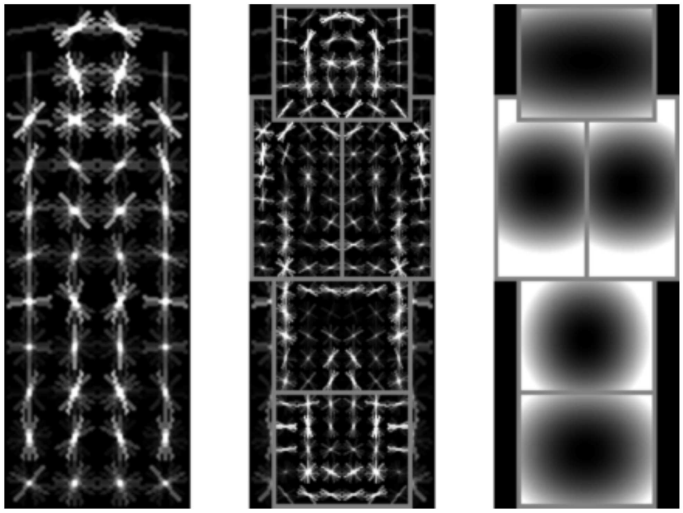


Figure 2.1: Figure shows the concept of DPM model. The model is defined by a coarse root filter and several higher resolution part filters. Felzenszwalb et al. [2010]

atypical geometric configurations, $\Phi_d(dx, dy)$ is a quadratic function to calculate the relative location of the part and the root.

Several early works on parts-based fine-grained classification adapted this approach to their problem [Chai et al., 2013a, Parkhi et al., 2012, Zhang et al., 2013b]. Parkhi et al. [2012] used DPM to localize the heads of cats and dogs to create the head mask. This methodology is relatively effective when objects have limited pose variation. Chai et al. [2013a] demonstrated that the synergy between segmentation and DPM-based detection can be leveraged to create one framework to alleviate the background noise and large pose variations, which can be beneficial to fine-grained image classification. However, an issue with the DPM approach is that the distance term is a Gaussian-based model for the part locations. Goring et al. [2013] analysed this and concluded that the distribution of parts is not Gaussian for CUB-200-11 [Wah et al., 2011b], a frequently used fine-grained bird database. Thus, a simple Gaussian distribution does not have enough capacity to model the pose variations.

Zhang et al. [2013b] proposed the deformable part descriptors (DPD) based on DPM. It applies semantic pooling on a weakly-supervised DPM based on weights learnt from training data. This process avoids using the hard assignment of a distance term for each detected part. Therefore, the DPD enables pooling across pose and parts without following a Gaussian distribution, which facilitates tasks such as fine-grained recognition. The overview of the method can be seen in Figure. 2.2.

More recently, deep learning has made considerable progress in detection [Girshick et al.,

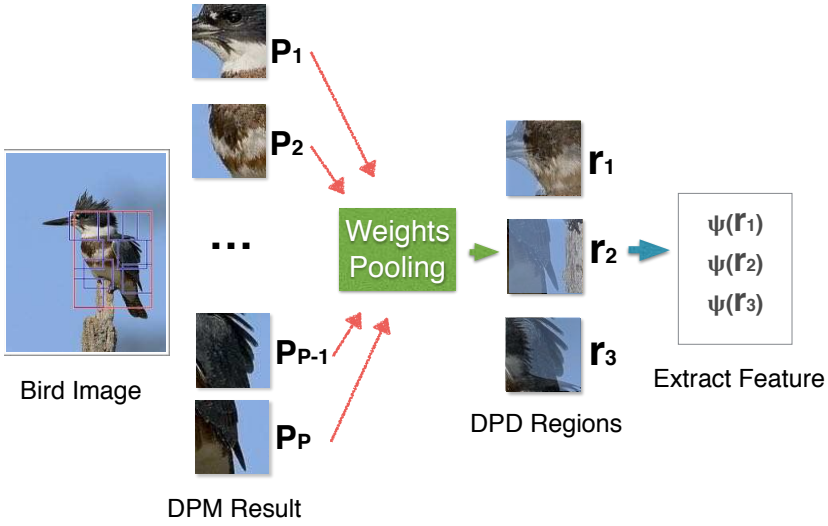


Figure 2.2: The DPD [Zhang et al., 2013b] pose-normalised descriptor is generated by pooling the result of the DPM part localisation result. These results are pooled into regions which are then concatenated together to form a single feature vector.

2014, Sermanet et al., 2013], and this has been translated to the fine-grained domain. An example of this is the region-based deep convolution neural network (R-CNN) framework. This is a two-stage framework where the first stage provides regional proposals by using a method such as edge boxes [Zitnick and Dollár, 2014] or selective search [Van de Sande et al., 2011]. In the second stage features are extracted from each region proposal using a deep convolutional neural network. These are then classified using a multi-class SVM. Zhang et al. [2014] adapted the R-CNN approach to fine-grained bird classification by learning a deep representations of parts with extra geometric constraints to improve the accuracy of bird and semantic parts detection. However, errors are likely to accumulate in the first regional proposal stage, leading to overall performance loss. To address this issues, Zhang et al. [2015] proposed an end-to-end deep convolutional network (DCNN) to perform parts detection and classification simultaneously by using a spatially fine-grained detection model.

Human in the loop

Other works show impressive results with " human in the loop " to assist finding discriminative parts or keypoints for fine-grained image classification. Wah et al. [2011a] proposed an approach that relied on human assistance to give binary answers to the predicted interesting locations of an object. These responses were used to generate a discriminative feature descriptor from those points to increase the classification accuracy. Deng et al. [2013] proposed an

interactive game requesting the user to find the most discriminative parts to help boost the performance of the classifier, and a method proposed by Parikh and Grauman [2011] discovers discriminative image parts by machine learning at the first stage and then asks users to manually correct and name them. A part-to-part based pairwise comparison mechanism is then applied to boost the classification accuracy. However, these methods are always restricted to small datasets since human input is time-consuming and requires expert knowledge. Nevertheless, it remains an open question in fine-grained image classification whether it is more critical to accurately localise corresponding locations over object instances or simply have the ability to capture detailed information [Gavves et al., 2013].

2.1.2 Unsupervised Parts Modelling

The previous methods for parts modelling require laborious manual annotation of images. By contrast unsupervised parts modelling provides a more realistic setting for real applications. Gavves et al. [2013] demonstrated a region partition method to describe parts. Their work showed that performing accurate localisation of parts was unnecessary as simply dividing the detected foreground image into a grid of regions provided similar results. The core idea of this method is that fine-grained classes such as birds normally share considerable shape and appearance similarities. Therefore, exterior shape can be used as a strong signal to locate relative part, Figure 2.3 provides two examples of this. From the segmentation result of GrabCut [Rother et al., 2004], unsupervised parts localisation can be calculated with the following:

$$\bar{x_s} + e_j\sqrt{\lambda_j} \tag{2.3}$$

where x_s is the average location of the segmentation pixels and λj and e_j are the j-th eigenvalue and eigenvector of the $(x_s - \bar{x_s})^T(x_s - \bar{x_s})$ covariance matrix. This model approximates the shape of the object as an ellipse and this ellipse should follow the "spine" of the object. The the three rough parts, head, body, and tail are then roughly segmented.

Duan et al. [2012] proposed a CRF approach to automatically find discriminative body parts of animals and their support regions by employing a latent CRF model to discover candidate parts. Yang et al. [2012] provide an unsupervised approach to localise parts of the bird by using template learning and matching. In this approach, a template represents a shape and texture pattern and the relationship between two patterns is captured by the relationship between

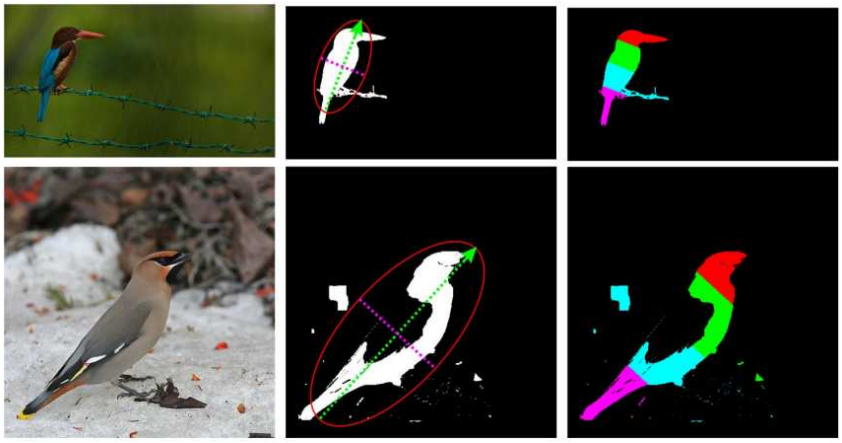


Figure 2.3: A rough shape is aligned in the middle column pictures. Then based on the alignment results, the shapes are split in the right column pictures equally along the principal axis to get consistent regions. The red and purple regions represent head and tail respectively [Gavves et al., 2013].

templates. This reflects the probability of co-occurrence in the same image. Krause et al. [2015a] achieve good performance on fine-grained bird and car classification without using any specific parts labels. They propose a method applying co-segmentation to perform pose and parts alignment. Current state-of-the-art of fine-grained bird classification performance without using parts are proposed by Jaderberg et al. [2015]. The method is called "spatial transformer network". It allows the spatial manipulation of data on the existing convolutional neural network with a differentiable module inserted. By properly modifying the localisation network, it can localise the discriminative parts of the fine-grained object. It guarantees the discriminability of the parts detected by driving an end-to-end learning of transformations.

2.1.3 Summary

The motivation to use local parts information is that some fine-grained classes often share the same parts such as wings, legs and heads for bird species.

Many methods have been proposed to make use of parts-based information which have already been annotated, a supervised setting (see Sec.2.1.1). These methods assume that this prior knowledge provides crucial information to discover corresponding patches of the image which are discriminative for the fine-grained classes.

The first concern with using parts annotation for large-scale recognition is that it requires

considerable time and effort to annotate. This motivated the works described in the unsupervised parts annotation (Sec.2.1.2). The methods described in that section overcome this issue by either only using annotation information at the training stage or deriving methods that require minimal annotation information (just a bounding box). These unsupervised approaches are promising due to the potential to enable widespread deployment.

The second concern is that the annotated parts may not contain the most discriminative information to distinguish fine-grained classes. For example, to distinguish an Africa crow and an America crow, the texture from local parts does not give much useful information. Instead, the shape of the whole bird should be considered. This case leads to one opinion that different birds may need different features to describe them in order to get the best accuracy for the classification.

2.2 Feature Engineering

Several different feature descriptors proposed for general image classification have been directly applied for fine-grained image classification in some pioneer works. Most of these are the classic hand-engineered features that have been used for general object recognitions, including: Scale Invariant Feature Transform (SIFT) [Lowe, 2004b], Speeded-up Robust Features (SURF) [Bay et al., 2006], local binary pattern (LBP) [Ojala et al., 2002], and Histogram of Oriented Gradients (HoG) [Dalal and Triggs, 2005b]. Later classic features were found to be suffering from the problem of losing subtle differences between inter-class variations. To generate more discriminative feature representations, feature encoding methods such as POOF proposed by Berg and Belhumeur [2013] are implemented to perform fine-grained bird classification. Recent advances in deep learning have led to state-of-the-art results for large-scale object classification [Krizhevsky et al., 2012]. These advances coupled with developments in transfer learning [Donahue et al., 2014] have led to the applicability of these features to fine-grained classification problems.

2.2.1 Hand-crafted Features

Hand-crafted features refer to manually designed features to extract global or local information from the image and generate effective representations. There are many manually designed

features in computer vision for general object classification. SIFT was proposed by Lowe [2004b]. It is a descriptor that accumulates the statistics of gradients of a circular image patch. Each image patch is partitioned into several local regions where the histogram of the orientations of the gradient is formed in each of the regions. The descriptor is formed with the concatenation of all the histograms. The HoG feature [Dalal and Triggs, 2005b] is similar to SIFT, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast to normalize the feature vector. LBP [Ojala et al., 2002] was originally designed for text recognition. It is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighbourhood of each pixel and considering the result as a binary number. Numerous hand-crafted features have been directly or indirectly implemented for fine-grained image classification. Hand-crafted features such as SIFT and HoG are most commonly used for fine-grained image classification [Brown et al., 2011, Hariharan et al., 2012, Yao et al., 2012]. One of the major challenges to directly apply hand-crafted features in fine-grained tasks is that traditional feature descriptors tend to encode the global salient differences between two categories, but it sometimes fails to catch the subtle differences between two fine-grained classes.

2.2.2 Feature Encodings

In order to enhance the standard histogram of quantised local features and retain more information about the original image features, several feature encoding techniques have been proposed for general object classification in the last decade [Perronnin et al., 2010, Sánchez et al., 2013]. Encoded features such as bags of visual words (BoV) [Csurka et al., 2004] and Fisher vector (FV) encoding [Perronnin et al., 2010] are widely used in some fine-grained tasks [Gavves et al., 2013, Gosselin et al., 2013, Philippe and Naila, 2012]. The typical feature encoding system is composed of the following two steps:

1. Extract local features, such as SIFT and HoG, from images and videos.

2. Summarise the set of local features such as through vector quantisation.

Below we describe the commonly used encoding methods for fine-grained classification: hard-coded BoV, soft-coded FV, and Part-based One-vs-One Features (POOFs). We also present inter-session variability modelling (ISV) which is a Gaussian based probabilistic modelling,

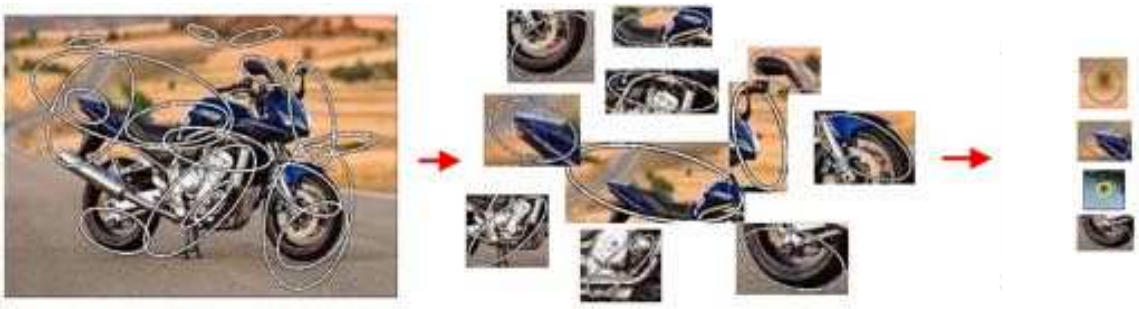


Figure 2.4: BoV feature encoding.

similar to FV, that is often used in face and speaker recognition [Vogt and Sridharan, 2008, Wallace et al., 2011].

Bags of Visual Words

BoV, as the name suggests, extracts a set of local patches and quantises the local descriptors into a finite set of elements to form a histogram. This method extracts and encodes a set of local descriptors, such as SIFT descriptors [Lowe, 2004b]. It assigns each descriptor to the closest entry in a codebook learned offline by clustering all local descriptors with k-means. This procedure is described in Figure. 2.4. The BoV method has been extended to include soft assignment [Philbin et al., 2008] and to use spatial pyramids so that multi-scale and spatial information can be captured [Lazebnik et al., 2006].

Fisher Vector

FV encoding summarises all of the features using the first and second order differences. It comes from the Fisher Kernel (FK) [Jaakkola et al., 1999]. In brief, it consists of characterising an image sample by its deviation which is measured by the sample log-likelihood with respect to the model parameters from the generative model. In comparison to BoV the codebook is represented by a mixture of Gaussians. A GMM is commonly used as a "probabilistic visual vocabulary". The FV representation has many advantages over the popular BoV framework. Firstly, BoV is a special case of FV where all clusters share the same weights. Secondly, FV can be trained more quickly and on smaller vocabularies with no performance loss. Finally, it works well with linear classifiers which are very efficient to learn [Bottou and Bousquet,

Figure 2.5: Images from the SCface [Grgic et al., 2011] database. Significant variations are present between images.

2011]. Chai et al. [2012] extracted features based on classification-oriented encodings and Fisher vectors. Gavves et al. [2013] implemented Fisher vectors not only globally, but also on localised appearance descriptors. Gavves et al. [2013], the state-of-the-art performance is achieved by using FV with colour SIFT features [Van De Sande et al., 2010].

Gaussian Mixture based Session Modelling

While FV uses a GMM as a visual vocabulary to represent difference classes, session variation modelling aims to model why different instances of the same class (object) appear differently. Session variation within each class caused by pose and illumination variation has been a constant issue for classifiers in computer vision. It causes one instance of a class to look different to another image of the same class. Causes of session variation include: appearance, illumination and pose variations. Example images with some of these variations are in Figure 2.5. In speaker authentication, various microphones and noisy transmission channels can cause the variation.

A number of techniques have been proposed to compensate for various aspects of session variability in the verification process. Some early work on speech verification [Wand and Schultz, 2011] focus on Gaussian Mixture Model (GMM) based models to model the effect of session differences and suppress session variation. The GMMs represent each observation as the combination of a session-independent speaker model with an additional session-dependent

Figure 2.6: This picture shows two references species' images are aligned according to chosen parts fixed locations [Berg and Belhumeur, 2013].

offset of the model means. The formulation of a GMM based session variation modelling can be represented as follows:

$$S = m + Ux_{ij} + Dz_i \tag{2.4}$$

Super-vector m is the concatenation of the GMM component mean vectors while x_{ij} is a low-dimensional representation of the variability of class i with instance j. And U is the low-rank transformation matrix from the constrained session variability sub-space. D is a diagonal matrix that incorporates the relevance factor, and z_i is a latent variable with norm distribution. Ideally, we would like a session variation modelling algorithm that can accurately discern the session-independent speaker. Recently, inter-session variability modelling (ISV) and joint factor analysis (JFA) are the two most successful techniques in session variation modelling [McCool et al., 2013, Vogt and Sridharan, 2008, Wallace et al., 2011]. ISV and JFA have been applied successfully to both speaker Vogt and Sridharan [2008] and face verification [McCool et al., 2013]. ISV aims to suppress session variation by explicitly modelling and removing within-client variation using a low-dimensional subspace while JFA also considers the between-client variation.

Part-based One-vs-One Features

Berg and Belhumeur [2013] proposed a framework to learn a large set of discriminative intermediate-level features called Part-based One-vs-One Features (POOFs) specialised for a set of parts for fine-grained classification. It is a fully automatic way to learn POOF based features on any reference dataset. Berg and Belhumeur [2013] randomly train pair-wise classifiers by choosing reference pairs from the data set with parts alignment as shown in Figure 2.6. Given parts locations of labelled images, a POOF is defined to specify two classes. All training samples of

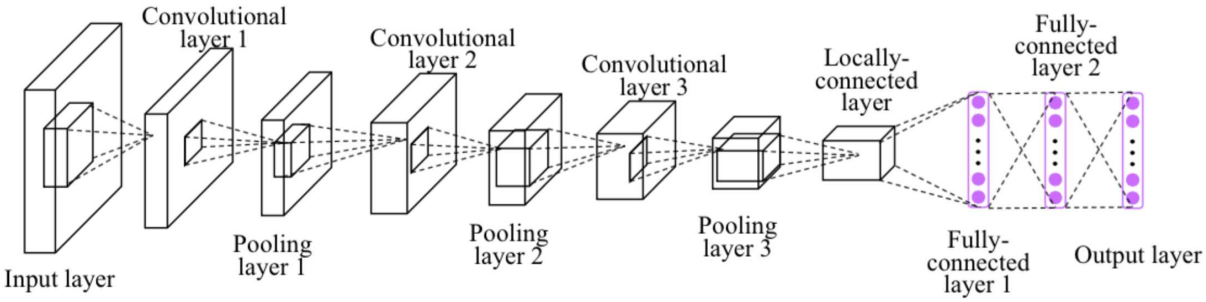


Figure 2.7: Typical CNN architecture with CONV, FC and POOL layers

two classes are aligned to fix locations of two chosen parts. Small cells with multiple scales are generated where base features are extracted. The maximal connected components contiguous to the chosen parts are selected by using a linear classifier. The one-vs-one POOF feature is extracted based on the base feature values from the support region.

2.2.3 Deep Networks

A recent trend in computer vision has been to learn features directly from datasets by applying a class of techniques known as deep learning. An example of this is deep convolutional neural networks (DCNNs), see Figure. 2.7. DCNNs were initially proposed by Le Cun et al. [1990] to recognize handwritten notes. It attempts to model high-level abstractions in data by using architectures composed of multiple non-linear transformations. A DCNN is a type of feed-forward artificial neural network with individual neurons tiled in such a way that they respond to overlapping regions in the visual field. CNNs carry out sub-sampling of images so that computing time can be reduced. At each convolutional layer, feature maps from the previous layers are convolved with learnable filters, which then go through a transaction function to form new feature maps. An example of different feature maps in a CNN is shown in Figure. 2.8. Each newly generated feature map can be viewed as a combination of multiple input maps.

Hinton et al. [2014], Krizhevsky et al. [2012] have shown that deep or layered compositional architectures are able to capture salient aspects of given images through discovery of salient clusters, parts and mid-level features. Krizhevsky et al. [2012] reached state-of-the-art performance in the ImageNet Large Scale Recognition Challenge (ILSVRC) in 2012 by using DCNN. Such models are able to perform much better than some traditional hand-crafted features [Le et al., 2011] by an absolute improvement of 10% on the classification track. It

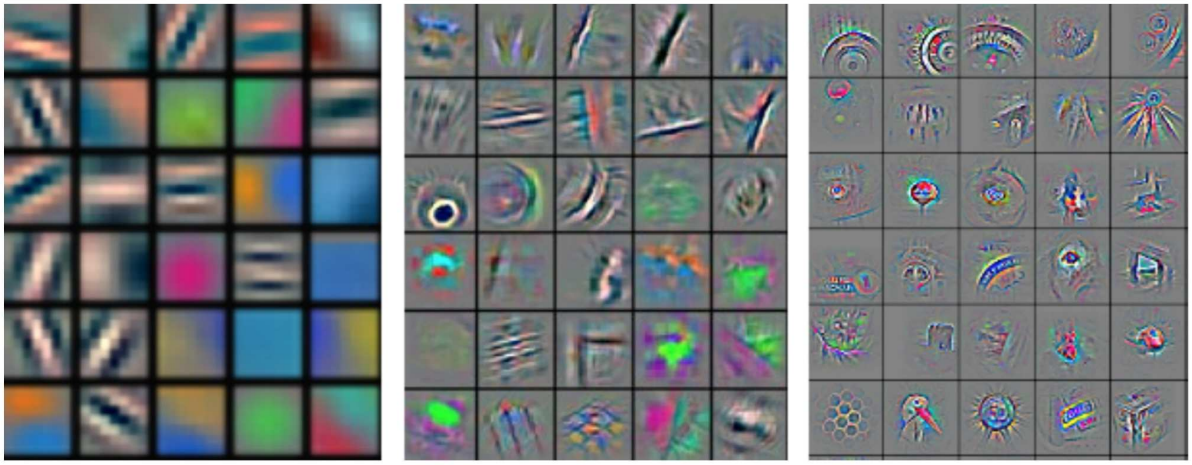


Figure 2.8: Visualisation of feature kernels. The results are produced by using the deconvolutional network approach proposed by Zeiler and Fergus [2013].

is believed that conventional hand-crafted features are limited in their ability to learn multiple levels of abstraction. Donahue et al. [2014] tested a deep convolutional activation feature for generic visual recognition (DeCAF) on Zhang et al. [2013a] DPD results and achieved state-of-the-art performance on CUB200-2011 with 64.96% mean accuracy. In the following sections first some typical layers in DCNNs will be described, case studies of various recent proposed network architectures will be given the relevant details after.

Neural Network Layers

A typical deep learning architecture contains a stack of modules, an example of this is shown in Figure 2.7. Often, each layer has a non-linear function applied to the output. The following paragraphs describe the various neural network layers, commonly used for classification tasks, and their specific functions.

Convolutional Layer: The main purpose of a convolutional layer (CONV) is to identify the local correlations of features from the previous layer. Each unit in a convolutional layer contains a set of weights and is densely connected to local patches of the previous layer. Weights from units are learnt through back-propagation training and are regarded as a filter bank. In many applied networks, outputs of a convolutional layer are fed into a pooling layer followed by a non-linear activation function such as a rectified linear unit (ReLU) [Glorot et al., 2011a] to increase non-linearity and regularisation.

Pooling: To further reduce the number of weights produced by convolutional layers, an additional sub-sampling layer often applied and is referred to as a pooling function [Le et al., 2011]. Pooling functions (POOL) such as max-pooling or average-pooling are very useful when generating statistical features over a small region. The reason for this is that images tend to have the property of stationarity, that is, features that are applicable in one subregion are likely to be applicable in other subregions.

Fully-connected Layer: A fully-connected (FC) layer is a typical layer to form a neural network. A single FC layer has full connections to every single neurons from the previous layer, thus more weights are retained in this type of layer. The activations can be computed with a matrix multiplication followed by a bias offset. A FC layer can be easily converted into a convolutional layer through treating every neuron in the FC layer as a 1×1 feature kernel. This is because both layers compute dot products, so their functional form is identical.

Softmax Layer: To calculate the score (probability) of each class in the neural network, a Softmax layer is normally attached at the end as a classifier. The Softmax classifier is trained to minimize the cross-entropy between the predicted class probabilities. The cross-entropy of a Softmax layer is given as follows:

$$-\,log(\frac{\exp^{f_{y_i}}}{\sum_j \exp^{f_j}}) \tag{2.5}$$

where y_i is the corresponding class label and f_j is the j-th element of the vector of class scores. The function in the bracket is the Softmax function and gives greater emphasis to the class that achieved the highest score as well as ensuring that the probabilities sum to 1.

Network Architecture Case Studies

In this section we briefly introduce a few popular DCNNs architectures. These either produced state-of-the-art results in the ImageNet challenge or are representative works demonstrating recent advances for DCNNs.

AlexNet: This refers to the network described by Krizhevsky et al. [2012], and achieved sate-of-the-art performance in the 2012 ImageNet classification challenge (ILSVRC). The authors trained a deep CNN consisting of eight layers (five CONV layers and three FC layers) using the ImageNet dataset [Deng et al., 2009a]. On the basis of an optimized parallel training process,

the CNN, consisting of more than one billion parameters, took approximately one week to learn the ImageNet.

GoogLeNet: This structure was proposed by Szegedy et al. [2014] and was the winner of ILSVRC 2014. The network is much deeper than the AlexNet containing 22 layers. The main idea behind the GoogLeNet is the inception layer which combines information from multiple scales and significantly reduces the number of parameters by using 1 by 1 CONV layers in the network. Apart from that, GoogLeNet is attached to three loss layers to propagate loss from the early, middle and late layers of the network which is able to alleviate the issue of dying gradients during back-propagation training.

VGGNet: The main contribution of VGGNet is to show that using fixed small filter size (3 by 3) from the beginning to end can perform better than large kernel size networks such as AlexNet and GoogLeNet. Additionally, it was found that VGGNet shows good generalization in multiple transfer learning tasks [Long et al., 2015].

ResNet: Residual Network [He et al., 2015] is the latest winner of ILSVRC 2015. This high performance model consists of 152 layers and uses batch normalization to compensate irrelevant variations in every layer. The most important contribution of this architecture is the special training method where shortcut connections are applied to several local layers to fit a residual mapping. This technique allows easy optimisation even when more than one hundred layers are presented.

Transfer Learning

A deep learning framework usually needs huge amounts of data to train in order for the cost function to converge to a good local minimum point and avoid overfitting on the training set. For some tasks such as fine-grained classification where the size of the dataset is significantly smaller than the ImageNet dataset, a process known as transfer learning [Donahue et al., 2014, Glorot et al., 2011b] can be used as a powerful tool to enable training a large target network without overfitting.

A typical way to perform transfer learning or domain adaptation is to train a network from a general large-scale dataset, in which it is believed that features learned are fairly general, and then fine-tuning the network parameters on the target dataset. The reason why transfer learning

works well on DCNN is that a DCNN is used to discover intermediate representations built in a hierarchical manner, which means the learnt low-level or mid-level features are likely to be quite general and so can be used to initialize other deep neural networks. Recent studies have taken advantage of this fact to obtain state-of-the-art results in fine-grained classification and a few other applications [Donahue et al., 2014, Zeiler and Fergus, 2013]. Donahue et al. [2014] trained Deep Convolutional Activation Features (DeCAF) in ImageNet and achieved significantly better results in general object recognition on Caltech-101 [Fei-Fei et al., 2007], domain adaptation on Amazon dataset [Saenko et al., 2010], fine-grained recognition on CUB-200, and scene recognition on SUN-397 [Xiao et al., 2010], compared to traditional non-DCNN based methods.

Since transfer learning from a general dataset has proven to be an effective and efficient method for various tasks, data expansion on the target dataset would add more discrimination. Some researchers have started to use additional help from the Internet as an useful way to expand their training dataset [Krause et al., 2015b, Xie et al., 2015, Xu et al., 2015]. Xie et al. [2015] proposed a method to extend the fine-grained vehicle dataset with external vehicle data annotated by some hyper-classes. The performance has been further improved on their fine-grained car classification with extra guidance to the learning process by exploring the relationship between the original fine-grained vehicle class and the new hyper-class. Krause et al. [2015b] further showed that training the DCNN-based model on publicly available noisy bird images from the web with an active learning system and achieved state-of-the-art performance on CUB200-2011 dataset. The performance of using extra 100,000 images has reached about 90% mean accuracy on the popular CUB200-2011 bird dataset, while using only 5,794 training images resulted in 80.1% mean accuracy .

2.2.4 Summary

According to the recent literature of feature learning for fine-grained classification, there is no clear line for features being used between general and fine-grained classification. In the early stages hand-crafted feature descriptors such as SIFT and HoG were being used directly for fine-grained image classification. Later on feature encoding methods replaced traditional hand-crafted features and achieved much better performance. There were no specific features designed for the fine-grained problem as it was treated as a texture recognition problem.

Compared to feature encoding with three steps of extraction, encoding and pooling, the recent success of DCNN enables joint optimization of the whole pipeline, leading to significantly higher recognition accuracy in many object recognition tasks. Transfer learning is normally conducted with the DCNN for the fine-grained task, however, the DCNN filters are initially learned from general image classification dataset.

It seems likely many of the filters learned for the general image classification are not helpful to locate nuances in local parts since those filters tend to capture, for example, shape and repetitive patterns which are useful to distinguish general classes.

Another problem of using a pre-trained DCNN as a feature extractor is that fully-connected layer is adapted as a pooling and encoding mechanism, resulting in high dimensional feature vector and losing of spatial information, which is important to find local subtle differences for various fine-grained classes.

2.3 Video Classification

Video classification has been studied for decades in the computer vision community. Various problems have been explored such as action recognition and video retrieval [Bendersky et al., 2014, Blank et al., 2005, Schüldt et al., 2004]. We divide existing video classification technologies into two tracks: those that describe videos by conventional hand-crafted features; and those that describe videos by DCNN-based features.

2.3.1 Conventional Features

Traditional video classification is a successful area in obtaining global descriptors that encode both motion and appearance information. There are normally three steps to perform a video classification which are feature extraction, feature encoding and classification. The first step is to either densely or sparsely extract and aggregate features from local appearance and motion using hand-crafted features [Liu et al., 2009, Sivic and Zisserman, 2003, Wang et al., 2009]. Several features such as SIFT, SIFT-3D [Scovanner et al., 2007], HoG, HoG-3D [Klaser et al., 2008] or Histogram of Optical Flow (HoF) [Chaudhry et al., 2009], can be used as a dense feature representation at this stage. Dense features require heavy computational cost and can not easily used for real-time applications. One solution to this is to use sparse feature for video

description. Sparse feature descriptors such as spatio-temporal interest points (STIPs) proposed by Willems et al. [2008] and which is an extension of Harris corner detector is applied in some video applications. Wang et al. [2013] proposed dense trajectories with hand-crafted features and achieved good and fast performance for behaviour recognition. Then they improved their work by showing that motion signals can be handled separately from the spatial signal [Wang and Schmid, 2013]. In the next step, extracted features are encoded into a fixed-sized video-level description. One of the popular encoding methods is through BoW. In video classification, BoW is used to learn a dictionary and accumulate the visual words into histograms of varying spatio-temporal information. Other encoding methods such as FV (introduced in the previous section) can also be applied for action recognition. The final step in the conventional model is to train a multi-class SVM for the classification task.

2.3.2 Deep Convolutional Based Model

In the previous section we introduced DCNN and their ability to automatically learn complex features using a hierarchy of kernels and pooling operations, have proven highly successful at still image classification problems from the small dataset PASCAL-VOC [Everingham et al., 2010] to the large scale dataset ImageNet [Deng et al., 2009b]. Some work attempts to use CNN to encode both global and motion information for video classification. To transfer DCNNs from image classification to video classification task, we need to understand the difference between them under a DCNN framework. The easiest way to implement the change to video classification is by extracting image-based or motion features from each frame and then pooling all information across time to make video-level predictions. Karpathy et al. [2014] applied a DCNN to extract features from every single frame and demonstrated strong performance over several traditional video classification methods (see Figure. 2.9). However, in this work the pooling results of multiple frames is only marginally better than the single frame based method which implies that learning motion features using DCNN is difficult. By contrast, Simonyan and Zisserman [2014] incorporated motion information from optical flow input with fixed inference time. Tran et al. [2015] employed 3D based convolutional kernel video classification. The performance is superior compared to methods using traditional hand-crafted features. Encouraged by those recent achievements, in this section we will briefly review two related CNN-based methods for video classification.

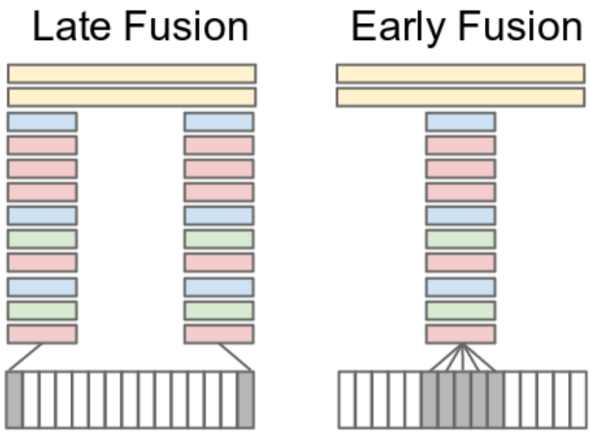


Figure 2.9: Figure shows early fusion (right) and late fusion (left) approaches to fuse information over temporal dimension. Red bars represent convolutional layers, green represents normalisation layers and blue represents pooling layers. [Karpathy et al., 2014]

Two-Stream Network

Simonyan and Zisserman [2014] proposed the two-stream network for action recognition. It consists of two independent spatial and temporal convolutional networks. The architecture of the network can be seen in Figure. 2.10. The spatial CNN operates the same as a still image classifier, making predictions on individual frames. It has proved that background and context information is useful to recognise various actions since some actions have high correlations with certain objects. The temporal CNN takes input from a stacked optical flow maps between consecutive frames. The optical flow is able to explicitly describe the motion difference between frame with intensity. The classification result is obtained through a late fusion of two softmax outputs of the independent networks. The downside of this method is in the restriction of the number of inference frames for prediction. This results in similar performance when compared to a single frame based CNN method.

3-Dimensional Convolutional Network

The deep 3-dimensional convolutional network (C3D) approach was proposed by Tran et al. [2015] for action recognition. The network structure is similar to 2D CNN architecture except all 2D convolution and pooling operations are replaced by 3D receptive fields. It utilises 3-dimensional convolutional kernels to model multiple frames of information simultaneously. In contrast to optical flow features where temporal information is explicitly modelled, this approach implicitly models the information within the DCNN structure. The C3D network

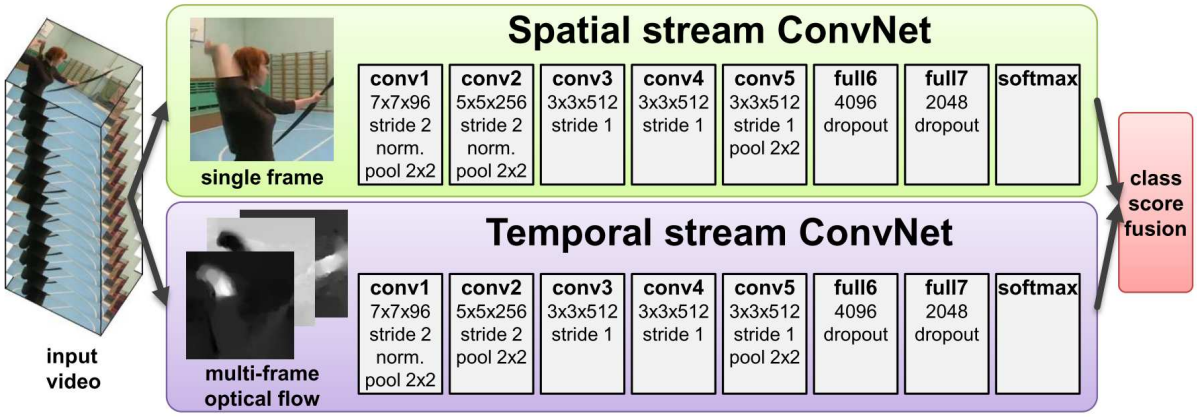


Figure 2.10: Two-stream network proposed by Simonyan and Zisserman [2014]. The architecture and parameter setting of spatial and temporal network are the same.

is claimed to have two advantages over two-stream network: First, generic features can be extracted from the network and applied to various tasks such as action classification, sports classification, and scene recognition without restriction of the number of frames. Second, it provides superior performance on action classification in a compact form with low feature dimensionality.

2.3.3 Summary

Recent work for action recognition has been dominated by the use of DCNNs, several methods simply stack consecutive video frames into the 2D image-based DCNN to exploit the temporal information. Such an approach assumes that the DCNN is able to learn the spatio-temporal information and to assist with this they use pooling to act as a high-level summarisation layer to capture the movement of the object based on appearances difference in consecutive frames. However, this method does not give superior performance improvement as stacking the DCNN is not able to take full advantage of temporal information by just stacking the images.

In order to better use the motion information, the two-stream network decomposes video frames into spatial and temporal DCNNs by using raw RGB pixels and optical flow frames. Each stream is learned separately through two different DCNN component, the final classification is performed by combing the softmax scores from two networks. The optical flow component can be treated as a mechanism to force the motion information into the learning process. One of the major concerns is that it is very hard to optimise the number of horizontally and vertically flow fields to be fed into the DCNN. Also, the two-stream method lacks a good

explanation of why using the softmax to fuse the information from two networks.

The C3D method doesn't limit itself to fuse the spatial and temporal information at the last decision layer. Instead, it embeds temporal information into the network by using filter kernels with size 3x3x3 supporting maximum 16 consecutive frames. All filters operate through space and time simultaneously. However, this network can be visualised as a 2D spatial network with a 1D temporal convolution, the temporal convolutional is performed at higher layers of the network [Sun et al., 2015]. There is also a strong doubt that C3D can not be controlled to learn both long term and short term temporal information as the number of channels for each layer is a fixed number ($L = 16$).

2.4 Fine-grained Datasets

One of the reasons why fine-grained classification is a difficult problem is levels of annotations (from the bounding box of the bird location in the image to the location of different parts) in the dataset.

Fortunately, the rapid development of the computer vision data collection community provides several tools such as online crowd-sourcing technologies [Deng et al., 2009b, 2013] and advanced methodologies [Van Horn et al., 2015a,b] to ease and accelerate the collection of large-scale datasets. Datasets collected by those tools accelerate the progress in various object recognition tasks [Deng et al., 2009b, Everingham et al., 2010, Soomro et al., 2012]. In 2012, a large-scale image recognition contest was hosted by Stanford University. Despite the fact that the winning method can be programmed and trained to recognize various categories of objects in an image, the competition is made possible by a set of more than 14 million images collected by the Machine Vision Group of Stanford University. For training and competition purposes, a subset of the ImageNet containing only 1000 categories is used. Overall, there are approximately 1.4 million images for training and testing and each category has roughly 1000 images. Such a huge dataset enables sufficient training of computer vision models, which consequently leads to greater accuracy of image recognition.

The whole ImageNet dataset structure is built based on WordNet hierarchy. However, we should note that the existing 1,000 classes do not belong to the same parent node, implying that subcategories are not considered. This makes it difficult to use as a fine-grained dataset. Luckly,

Figure 2.11: Parts annotation on CUB-200-11.

we have seen rapid growth from the fine-grained community in releasing various fine-grained datasets such as bird datasets: CUB-200 [Wah et al., 2011b].

CUB-200-2012 [Wah et al., 2011b] contains 200 bird species from north America and each species is organized by scientific classification under order, family, genus, and species. 11,788 images which results in about 60 images for each class. Images are downloaded from Flickr image search and annotated via Amazon Mechanical Turk. It is a challenging dataset because large pose variations are presented in each category. Each image comes with an annotated bounding box around the bird, as well as annotations for many constituent parts of the object. Overall, 15 parts are annotated in each image including: back, beak, belly, breast, crown, forehead, left/right eye, left/right leg, left/right wing, nape, tail, and throat. All parts and bounding box are annotated by pixel location and visibility in each image.

Chapter 3

Inter-Session Variation Modelling

A challenge for fine-grained classification is to correctly identify a class despite large intra-class variations due to pose and environmental variations. The first two publications of this thesis explore the first research question "Can we model different instances of the same class under various environments (large intra-class variations)?".

In the first publication "Local Inter-Session Variability Modelling for Object Classification", we introduce inter-session variability modelling (ISV) for fine-grained classification. ISV aims to suppress session variation by explicitly modelling and removing intra-class variation using a low-dimensional subspace. It has been applied successfully to both speaker and face verification [McCool et al., 2013, Wallace et al., 2011]. We extend this GMM-based method by modelling local session variations. This is achieved by dividing an image into local regions and each region is modelled independently. Local region ISV allows us to re-enforce spatial constraints that were previously being discarded. The proposed method demonstrates improved performance over the ISV for fine-grained classification of fish and face images.

In the second paper "Modelling Local Deep Convolutional Neural Network Features to Improve Fine-grained Image Classification", we explore the potential of learning local features, using deep convolutional neural networks (DCNNs) to extract features from uniformly partitioned patches in the image. DCNN-based features have been shown to considerably improve the object recognition performance in various computer vision tasks [Donahue et al., 2014, Zeiler and Fergus, 2013]. However, DCNN features are high dimensional representations (4096) compared to traditional features such as SIFT which has 128 dimensions [Lowe, 2004a], making it difficult to use them as a local feature for stochastic models. Therefore, we propose

to reduce the dimensionality of DCNN features through layer-restricted re-training. We show that this novel DCNN-based local feature has superior performance over 2D-DCT features for fine-grained classification of fish and food.

To evaluate our proposed methods, we present a new challenging fine-grained database of fish with 3,960 images collected from 468 species. This data consists of real-world images of fish captured in conditions defined as "controlled", "out-of-the- water" and "in-situ". More details can be found in the first publication of this chapter.

"Local Inter-Session Variability Modelling for Object Classification" has been published at the Winter Conference on Applications of Computer Vision, 2014, and "Modelling Local Deep Convolutional Neural Network Features to Improve Fine-grained Image Classification" was presented at the International Conference on Image Processing, 2015.

Local Inter-Session Variability Modelling for Object Classification

Kaneswaran Anantharajah
QUT, SAIVT and MILAB

ZongYuan Ge
QUT, CyPhy Lab.

Chris McCool
NICTA, Brisbane

Simon Denman
QUT, SAIVT

Clinton Fookes
QUT, SAIVT

Peter Corke
QUT, CyPhy Lab.

Dian Tjondronegoro
QUT, MILAB

Sridha Sridharan
QUT, SAIVT

Abstract

Object classification is plagued by the issue of session variation. Session variation describes any variation that makes one instance of an object look different to another, for instance due to pose or illumination variation. Recent work in the challenging task of face verification has shown that session variability modelling provides a mechanism to overcome some of these limitations. However, for computer vision purposes, it has only been applied in the limited setting of face verification.

In this paper we propose a local region based inter-session variability (ISV) modelling approach, and apply it to challenging real-world data. We propose a region based session variability modelling approach so that local session variations can be modelled, termed Local ISV. We then demonstrate the efficacy of this technique on a challenging real-world fish image database which includes images taken underwater, providing significant real-world session variations. This Local ISV approach provides a relative performance improvement of, on average, 23% *on the challenging MOBIO, Multi-PIE and SCface face databases. It also provides a relative performance improvement of* 35% *on our challenging fish image dataset.*

1. Introduction

Object classification is a challenging problem due to variations in the appearance of the objects and the environment in which they appear. One of the best known and most well investigated object classification problems is that of face recognition, where variations in subject pose and lighting present significant challenges [6]. A recent state-of-the-art face recognition approach uses session variability modelling [12] to provide a general model that describes the differences that occur between instances of the same class, whether that be from pose, illumination or expression variation. This session variability modelling approach is applied in the context of a free-parts model [16], which discards potentially useful spatial relationships.

The free-parts approach described in [16] divides the face into blocks and each block is considered to be a independent observation of the same object (the face). The distribution of these blocks is described by a Gaussian mixture model (GMM) and has been investigated by several researchers [16, 9, 10, 19]. Lucey and Chen [9] showed that a relevance adaptation approach, similar to the one used for speaker authentication [14], could be used to quickly obtain client (class) specific GMMs by using a universal background model (UBM). Furthermore, Lucey and Chen showed that adding spatial constraints to this free-parts approach could yield state-of-the-art face recognition performance on the BANCA dataset [13]. Sanderson et al. [15] proposed a multi-region probabilistic histogram (MRH) approach which used the free-parts approach as its basis but incorporates spatial constraints and also makes several simplifications for efficiency purposes. This efficient method provided state-of-the-art performance on the labeled faces in the wild (LFW) dataset [1].

Recently in [18, 12] the GMM free-parts (GMM-FP) model was extended to include an inter-session variability (ISV) modelling component. ISV learns a sub-space which models the differences in instances of the same object (the face). Such an approach was initially proposed to cope with similar problems in speaker authentication [17]. This model of session variability is used to estimate session variations in order to suppress, or account, for them. Using this model yielded state-of-the-art performance on several well known face datasets such as MOBIO [11] and Multi-PIE [6]. Despite this state-of-the-art performance, this approach has an obvious limitation as it does not enforce any spatial relationships between the blocks (observations), which discards spatial information which would help to disambiguate between the classes. Furthermore, its general applicability to vision problems has not been shown as it has only ever been applied to face recognition.

Contributions: In this paper we propose a local inter-

[1] http://itee.uq.edu.au/ conrad/lfwcrop/

session variability modelling approach that enforces local spatial relationships that were previously discarded. This approach is similar to [15] which adopts a multi-region probabilistic histogram approach. However, rather than using a probabilistic histogram that uses the zeroth order statistics of a GMM [15], we apply this to the GMM-FP and ISV approaches which, as has been shown in [12], uses the zeroth and first order statistics which provide a better approximation of the underlying data. We also apply, for the first time, the ISV model to the broader problem of object classification to examine the general applicability of this technique. To do this we use a large fish image dataset that contains challenging real-world images consisting of fish images captured in conditions ranging from controlled with a constant background and illumination, through to underwater imagery of fish in their natural habitat with significant illumination and pose variations.

We show that introducing spatial constraints leads to state-of-the-art performance for face and fish image classification. Spatial constraints are introduced by dividing the images into R regions and learning a model specific to each region. This allows us to locally model session variability and capture local identity information. For face recognition this Local ISV approach provides an average relative improvement of 23% for the MOBIO [11], Multi-PIE [6] and SCface [5] databases over the existing state-of-the-art. For fish classification, we show that using Local ISV provides a relative performance improvement of 35%.

Finally, we examine the sensitivity of the Local ISV approach to real-world problems such as errors in face localisation. Using the real-world MOBIO database, which consists of face images captured from a mobile phone, we introduce noise to the manually annotated landmarks to simulate misalignment, a problem often encountered in practical applications [7]. Empirically we show that the Local ISV approach is more sensitive to this misalignment, but still provides superior performance when the noise in the position of the landmarks is less than 20% of the inter-eye distance.

The remainder of the paper is organized as follows. An overview of existing work is presented in Section 2; the proposed region based GMM and ISV based face authentication frame works are explained in Section 3. Databases and protocols used in the experiments are presented in Section 4. In Section 5, we present the experimental results using our novel fish image database and three face databases. We conclude the paper in Section 6.

2. Prior work

2.1. GMM Free-Parts for Face Verification

Several researchers have examined the use of the GMM-FP framework to perform face verification [16, 9, 19]. Introduced in [16], this approach divides the image (the face) into N overlapping blocks which are considered to be independent observations of the same underling signal (the face), $\boldsymbol{O}$. From each block a 2D-DCT feature vector of dimension M is obtained to compactly represent each block, such that the n-th block yields the feature vector $\boldsymbol{o}_n$. Thus the j-th image of the i-th client yields the set of n observations $\boldsymbol{O}_{i,j} = [\boldsymbol{o}_{i,j,1}, \ldots, \boldsymbol{o}_{i,j,n}]$. The distribution of these feature vectors is then modelled using a GMM,

$$Pr\left(\boldsymbol{O} \mid \boldsymbol{\theta}\right) = \prod_{n=1}^{N} \sum_{c=1}^{C} \omega_c \mathcal{N}\left[\boldsymbol{o}_n \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\right], \qquad (1)$$

where C is the number of components for the GMM, ω_c is the weight for component c, $\boldsymbol{\mu}_c$ is the mean for component c, and $\boldsymbol{\Sigma}_c$ is the covariance matrix (usually considered to be diagonal) for component c.

In order to overcome the limited number of samples per client, i, mean-only relevance MAP adaptation [9] is used to enroll the client (class). Originally proposed for speaker authentication [14], mean-only relevance MAP adaptation takes a prior model, usually referred to as a universal background model (UBM) GMM, and performs MAP adaptation on the means using the observations of the i-th client, $\boldsymbol{O}_i$, to obtain a model for the client. Since only the mean vectors change, it has been shown [17] that this can be written as,

$$\boldsymbol{s}_i = \boldsymbol{m} + \boldsymbol{D}\boldsymbol{z}_i, \qquad (2)$$

where $\boldsymbol{s}_i$ is the mean super-vector for the i-th client, $\boldsymbol{m}$ is the mean super-vector of the UBM GMM (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 [17] and ensures the result is equivalent to mean-only relevance MAP adaptation.

To evaluate the likelihood that image t, described by a set of observations $\boldsymbol{O}_t$, was produced by client i a log-likelihood ratio is used. In this case the positive class is given by the claimed identity i and the negative class is represented by the UBM GMM. Thus, the log-likelihood ratio is,

$$h\left(\boldsymbol{O}_t, \boldsymbol{s}_i\right) = \log\left[p\left(\boldsymbol{O}_t \mid \boldsymbol{s}_i\right)\right] - \log\left[p\left(\boldsymbol{O}_t \mid \boldsymbol{m}\right)\right]. \qquad (3)$$

It was shown in [19] that this could be efficiently calculated using the linear scoring approximation [4] leading to,

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

where the diagonal matrix $\boldsymbol{\Sigma}$ is formed by concatenating the diagonals of the UBM covariance matrices and $\boldsymbol{f}_{t|\boldsymbol{m}}$ is the super-vector of mean normalised first order statistics as given in [12]. A decision threshold, τ, is applied to this score to decide if the observations were generated by the model, $\boldsymbol{s}_i$. Image, $\boldsymbol{O}_t$, is classified as being of client i if and only if $h_{linear}\left(\boldsymbol{O}_t, \boldsymbol{s}_i\right) \geq \tau$.

Super-vector notation is a way of compactly representing data for a GMM. It is particularly useful when we consider mean-only relevance MAP adaptation as the only part of the model that changes is the means. Since the weights, $[\omega_1, \ldots, \omega_C]$, and variances, $[\Sigma_1, \ldots, \Sigma_C]$, are fixed each model can be described by the concatenation of their means to form a single super-vector $\boldsymbol{a} = [\boldsymbol{\mu}_1^T, \ldots, \boldsymbol{\mu}_C^T]^T$. More details for this notation can be found in [12].

2.2. Inter Session Variability Modelling

Inter-session variability modelling (ISV) has been applied successfully to speaker [17] and face verification [12]. ISV aims to model and suppress session variation, that is variation that makes one image of the same class look different to another image of the same class. For face recognition this is often considered to be illumination, pose or expression variation. At enrollment time session variation is suppressed by jointly estimating a latent session variable along with a latent identity variable, the latent session variable is then discarded. When scoring, an estimate of the latent session variable, $\boldsymbol{x}_t$, is obtained from the test samples, $\boldsymbol{O}_t$. This estimate, $\boldsymbol{x}_t$, is then used to offset the models so that the likelihood function now takes into account the session variation (noise), of the test samples; see [12] Section 3.5 for more details.

Enrolling a client for ISV consists of MAP adaptation, similar to mean-only relevance MAP adaptation. The difference is that a sub-space, $\boldsymbol{U}$, is introduced to model session variation and so restricts the movement for relevance adaptation such that the model for the j-th image of the i-th client (class) is,

$$\boldsymbol{u}_{i,j} = \boldsymbol{m} + \boldsymbol{U}\boldsymbol{x}_{i,j} + \boldsymbol{D}\boldsymbol{z}_i, \tag{5}$$

where $\boldsymbol{x}_{i,j}$ is the latent session variable and is assumed to be normally distributed. In this way each image is considered to have been produced with its own session variation; for instance due to pose or illumination variation. As previously mentioned when performing enrollment the session varying part ($\boldsymbol{U}\boldsymbol{x}_{i,j}$) is discarded and only those parts pertaining to identity are retained. Thus, the ISV client model is given by,

$$\boldsymbol{s}_{ISV,i} = \boldsymbol{m} + \boldsymbol{D}\boldsymbol{z}_i. \tag{6}$$

This should not be confused with mean-only relevance MAP adaptation (see Equation 2) as the latent variables $\boldsymbol{x}_{i,j}$ and $\boldsymbol{z}_i$ are jointly estimated for ISV.

Scoring with ISV is performed by first estimating the latent session variable, $\boldsymbol{x}_t$, for the test sample $\boldsymbol{O}_t$. This is then used to offset the client model ($\boldsymbol{s}_{ISV,i}$) and the UBM ($\boldsymbol{m}$) so that the log-likelihood is estimated in the session conditions of the test samples. This provides a mechanism to compensated for session variation. When used in the context of linear scoring, this leads to the following log-likelihood ratio (LLR),

$$\begin{aligned} h_{ISV}\left(\boldsymbol{O}_t, \boldsymbol{s}_{ISV,i}\right) = &\left(\boldsymbol{s}_{ISV,i} - \boldsymbol{m}\right)^T \boldsymbol{\Sigma}^{-1} \\ &\left(\boldsymbol{f}_{t|\boldsymbol{m}} - \boldsymbol{N}_t \boldsymbol{U} \boldsymbol{x}_{t|UBM}\right), \end{aligned} \tag{7}$$

where $\boldsymbol{N}_t$ is the zeroth order statistics for the test sample in a block diagonal matrix as defined in Equation 11 of [12].

3. Proposed approach

We propose to overcome one of the major limitations of the ISV approach to image classification by dividing an image into local regions. Doing this allows us to re-enforce spatial constraints that were previously being discarded. To properly evaluate the local ISV approach we also have to evaluate the local GMM-FP approach to ensure that locally modelling session variability is not being boosted solely by being able to extract local class specific information.

The approach is similar to work conducted in [15] where a probabilistic histogram for local regions was formed using a GMM, termed a multi-region probabilistic histogram (MRH). This MRH approach collates the zeroth order statistics, the occupation probabilities, of a GMM to perform classification. By contrast, we propose to apply local region decomposition to the ISV approach due to their state-of-the-art performance when used globally in [12]. These techniques collate the zeroth and first order statistics of a GMM to perform classification, furthermore, ISV provides an additional constraint to the MAP equations to suppress session variations (noise).

3.1. Local GMM Free-Parts Approach

We propose an extension to the GMM-FP approach whereby the input images are divided into a set of R regions and each region is modelled independently. This approach, termed Local GMM-FP, allows us to derive local descriptions of the identity variation. Similar to the GMM-FP approach, the proposed Local GMM-FP technique divides each region into a set of overlapping blocks from which DCT features are extracted. A local GMM UBM is then learnt for each specific region M_r, $\boldsymbol{m}_r$, and local models of the identity are then obtained using region specific mean-only relevance MAP adaptation,

$$\boldsymbol{s}_{r,i} = \boldsymbol{m}_r + \boldsymbol{D}_r \boldsymbol{z}_{r,i}, \tag{8}$$

where $\boldsymbol{s}_{r,i}$ is the i-th client model corresponding to region r, $\boldsymbol{z}_{r,i}$ is a normally distributed latent variable for region r, and $\boldsymbol{D}_r$ is a diagonal matrix that incorporates the relevance factor and the covariance matrix [17] as per Section 2.1.

The t-th image is compared to the i-th client model in a region specific manner. Thus the observations from the r-th region of t-th image, $\boldsymbol{O}_{r,t}$, are compared to the i-th client's

model for the r-th region, $s_{r,i}$. Thus the LLR becomes region specific,

$$h_{linear}\left(\boldsymbol{O}_{r,t}, \boldsymbol{s}_{r,i}\right) = \left(\boldsymbol{s}_{r,i} - \boldsymbol{m}_r\right)^T \boldsymbol{\Sigma}_r^{-1} \boldsymbol{f}_{r,t|m_r}, \quad (9)$$

where $\boldsymbol{\Sigma}_r$ is the covariance matrix for the r-th region and $f_{r,t|m_r}$ is the mean normalised first order statistics for the r-th region. Subsequently, all region specific scores are summed and compared to the threshold, τ.

3.2. Local Inter-Session Variability Modelling

In this section we propose to apply ISV to local regions so that we can locally model session variability and capture local identity information. We apply a similar concept to Section 3.1 of dividing the image into R regions and again perform MAP adaptation for each region independently. Thus for the j-th image of the i-th client in the r-th region we obtain the model,

$$\boldsymbol{u}_{r,i,j} = \boldsymbol{m}_r + \boldsymbol{U}_r \boldsymbol{x}_{r,i,j} + \boldsymbol{D}_r \boldsymbol{z}_{r,i}. \quad (10)$$

A region specific ISV client model, $\boldsymbol{s}_{ISV,r,i}$, is formed by,

$$\boldsymbol{s}_{ISV,r,i} = \boldsymbol{m}_r + \boldsymbol{D}_r \boldsymbol{z}_{r,i}. \quad (11)$$

During the evaluation process, the region specific latent session variable $\boldsymbol{x}_{r,i}$ is estimated for $\boldsymbol{O}_{r,i}$ using the r-th region from the i-th client model. Then, session variation is compensated for by adding this estimated session offset to $\boldsymbol{s}_{ISV,r,i}$ prior to scoring.

4. Database and Evaluation Protocols

4.1. Fish Image Set

To evaluate the new ISV approach in the broader object classification domain we introduce a new, large fish image dataset consisting of $3,960$ images collected from 468 species. This data consists of real-world images of fish captured in conditions defined as "controlled", "out-of-the-water" and "in-situ". The "controlled" images consist of fish specimens, with their fins spread, taken against a constant background with controlled illumination, see Figure 2 (a) and (b). The "in-situ" images are underwater images of fish in their natural habitat and so there is no control over background or illumination, in addition there is the challenge of the unique underwater imaging environment, see Figure 2 (c) and (d). The "out-of-the-water" images consist of fish specimens, taken out of the water with a varying background and limited control over the illumination conditions, see Figure 2 (e) and (f).

There are two main difficulties when performing classification on the fish imagery. The first is that, in many cases, different species are visually similar, as shown Figure 1 (a)-(d) where it can be seen that four species are visually similar. The second is that there is a high degree of variability in the image quality and environmental conditions, see Figure 2 for example images [2] for some example images.

Approximately half of the images have been captured in the "controlled" condition, where the image of the fish has been captured out-of-the-water with a controlled background. The "in-situ" condition consists of images taken underwater with no control over the background and with significant pose and illumination variations. Approximately one third of the data was captured in this manner. Finally, the remaining images are captured "out-of-the-water", but without a controlled background and may contain some minor pose variation.

Evaluation Protocol: An evaluation protocol, similar to [11] and [3], has been developed for experiments on this dataset. We define three sets of data by splitting the data, based upon species (class), into a training set (*train*) to learn/derive models; a development set (*dev*) to determine the optimal parameters for our models; and an evaluation set (*eval*) to measure the final system performance.

Two protocols are defined to evaluate the system performance when high quality ("controlled") and low quality ("in-situ") data is used to enrol classes. Protocol 1a uses one enrollment image per species from the "controlled" data. Protocol 1b uses one enrollment image per species from the "in-situ" data. For both protocols, the same test imagery (a mix of "controlled", "in-situ" and "out-of-the-water" images) is used. The *train* set consists of $1,296$ images from 169 species, and can be used to learn or derive models for principal component analysis, probabilistic linear discriminant analysis, or for learning the UBM GMM [3]. The *dev* set consists of 958 images from 93 species, and the *eval* set consists of 963 images from 98 species. For these two protocols the *dev* and *eval* partitions consist of the sub-set of species for which we have at least three images, with at least one "controlled" and one "in-situ" image.

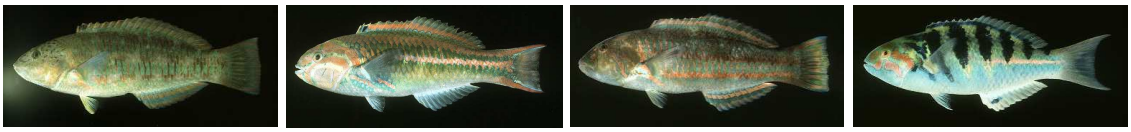
We evaluate system performance by measuring the Rank-n identification rate, using manually annotated bounding boxes.

Rank-n refers to the percentage of queries for which the correct result in within the top n matches. We measure performance at $n = 1$, $n = 5$ and $n = 10$. The bounding boxes were obtained by inscribing a region around the body of each fish, an extra 3% margin was added to avoid losing edge information, example bounding boxes are shown in Figure 2. The new fish database which has been presented will be made publicly available[4].

[2] images (a) and (c) in the Figure 2 are from Australian National Fish Collection CSIRO, (b) is taken by G. Edgar, and (d) is taken by Dennis King

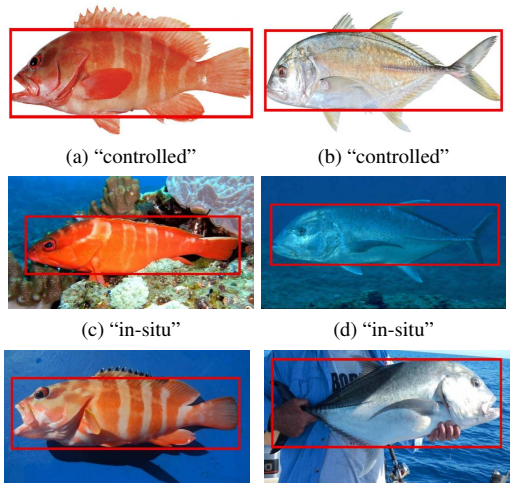
[3] to train ISV there we only use the 155 classes that have more than one image per species

[4] see http://tiny.cc/fishdataset for details



(a) Thalassoma Trilobatum (b) Thalassoma Quinquevittatum (c) Thalassoma Purporeum (d) Thalassoma Hardwicke

Figure 1: Example images of four different fish species, all which have similar visual appearance despite being distinct species. (Images taken by J.E. Randall)



(a) "controlled" (b) "controlled"

(c) "in-situ" (d) "in-situ"

(e) "out-of-the-water" (f) "out-of-the-water"

Figure 2: Example images of two different fish species captured under the three different capture conditions (from top to bottom): "controlled", "in-situ" and "out-of-the-water". Significant variation in appearance due to the changed imaging conditions (session variation) is evident. Ground truth bounding boxes are shown in red.

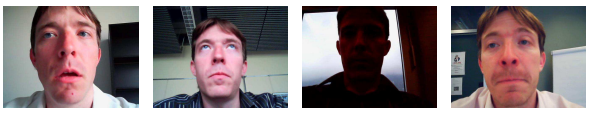


Figure 3: Example images from the MOBIO [11] database.

Figure 4: Example images from the Multi-PIE [6] database.

Figure 5: Example images from the SCface [5] database.

4.2. Face Databases

Three face databases are used to evaluate the proposed approach: MOBIO [11], Multi-PIE [6], and SCface [5]. Face verification is still a challenging classification problem and we want to compare the proposed approach to the current state-of-the-art. The MOBIO and Multi-PIE databases contain pose and illumination variations, while MOBIO and SCface contain images captured with different sensors. SCface also contains variations in the resolution of the captured images.

When performing evaluations on each database we use the well defined protocols that provide dedicated *train*, *dev* and *eval* sets. In each case approximately one third of the data is used for each set. The *train*, *dev* and *eval* datasets are used in the same manner as outlined in Section 4.1. For all three databases we use manually annotated eye locations and examples images are provided in Figures 3, 4 and 5 for the MOBIO, Multi-PIE and SCface databases respectively. More details on the protocols for the MOBIO and SCface databases are given in [18], and for the Multi-PIE database in [3].

System performance is presented in terms of equal error rate (EER) and half total error rate (HTER) [11]. EER is used for the development set and is the point at which the false alarm rate equals the false rejection rate (a smaller number is better). The threshold, τ, derived from the EER on the development set is then used on the evaluation set to obtain the HTER (the average of the false alarm rate and false rejection rate) to present the final system performance (a smaller number is better). Linear scoring and ZT-Normalisation are used for all evaluated systems, as it has previously been shown to be effective for face recognition [19].

4.3. Impact of Face Localisation Error

An issue for any real world face verification system is it's robustness to face mis-alignment; that is, the performance degradation when the face image is not extracted per-

System	Protocol 1a		Protocol 1b	
	Dev	Eval	Dev	Eval
PCA+PLDA	23.8	23.8	16.4	17.9
RBF-SVM (HoG)	31.8	31.4	24.2	25.5
GMM-FP	29.5	32.6	25.2	28.0
Local GMM-FP	37.4	43.0	34.6	40.2
ISV	34.9	37.8	30.9	33.5
Local ISV	**43.1**	**49.3**	**40.8**	**46.7**

Table 1: Fish Identification Results. Rank-1 identification rate results are given, and the best performing system is shown in **bold**.

fectly (based on the eye positions). Therefore, we evaluate the robustness of our proposed approach to errors in misalignment by introducing noise into the manually annotated landmarks. We choose the MOBIO database for this evaluation, and add uniform random noise equal to 2%, 5%, 10% and 20% of the average inter-eye distance (119 pixels for the MOBIO database). The new landmark points which have been used in this experiment are publicly available [5].

5. Experiments

The proposed techniques have been implemented using the the freely available signal processing and machine learning tool box, BOB [1].

5.1. Evaluation on Fish Image Set

The images are cropped with an extra margin of 3% added to the ground truth bounding boxes. Images are then converted to gray-scale and resized to 160×64 pixels. DCT features are extracted exhaustively using a block size of 20×20 with $M = 65$. Mean and standard deviation is applied to each block, as such the zeroth DCT coefficient is discarded. GMM based approaches use 512 components, for the sub-space size is set to 64 for Protocol 1a and 32 for Protocol 1b. For the local approaches the optimal region size was found to be 4×4.

The fish image dataset is a new dataset and so in addition to the proposed approaches we also present several baseline systems. The baseline systems used in this work are probabilistic linear discriminant analysis (PLDA) which achieves state-of-the-art performance for face recognition [8], and a support vector machine (SVM) approach similar to that used for classifying pedestrians [2]. For both the PLDA and SVM approaches we used the gray-scale images which have been resized to 160×64 pixels. For PLDA we apply dimensionality using principal component analysis (PCA) as this showed improved performance, this is termed PCA+PLDA.

[5] visit https://wiki.qut.edu.au/display/saivt/Noisy+MOBIO+Landmarks for details

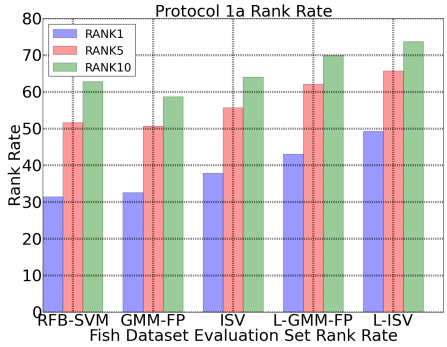


Figure 6: Rank-1, Rank-5 and Rank-10 identification rates for Protocol 1a on the evaluation set.

For the SVM approach we use a histogram of oriented gradients as the feature and a radial basis function as this provides superior performance over a linear SVM, referred to as RBF-SVM.

Results presented in Table 1 show that the Local ISV approach outperforms all other approaches. The standard ISV approach clearly outperforms the RBF-SVM and GMM-FP approaches, and the Local ISV approach provides a relative performance gain of 35% when compared to ISV. The next best system is the Local GMM-FP approach which provides a relative performance gain of 38% when compared to GMM-FP. The Rank-5 and Rank-10 identification results, in Figures 6 and 7, show that Local ISV and Local GMM-FP provide consistently improved performance.

A general trend for all of the classifiers is that Protocol 1a provides better performance than Protocol 1b. The average relative performance difference for all classifiers between Protocol 1a and Protocol 1b is 13%. This is likely due to the fact that for Protocol 1a the enrollment data consists of a "controlled" image, compared to Protocol 1b which uses an "in-situ" image. This demonstrates the importance of having high quality enrollment data with which to generate a model, even when session variability modelling is used.

5.2. Evaluation on Face Verification Databases

When extracting the DCT features we use a block size of 12×12 with $M = 44$ for the MOBIO and Multi-PIE databases. For the SCface database, we used a block size of 20×20 with $M = 65$. These optimal block and feature sizes were taken from [19].

We evaluated the proposed local face verification approach on three databases as outlined in Section 4.2. Our proposed technique is compared to three baseline techniques: MRH, GMM-FP and ISV. In this experiment UBMs are trained with 512 components for MOBIO and Multi-PIE and 256 components for SCface. In the ISV and Local ISV approaches a sub-space of 40 components is used for MO-

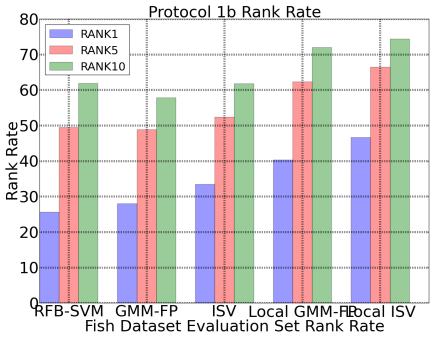


Figure 7: Rank-1, Rank-5 and Rank-10 identification rates for Protocol 1b on the evaluation set.

BIO and SCface, and 80 components is used for Multi-PIE. For the Local GMM-FP approach we use a region size of 4×4 for MOBIO and Multi-PIE, and 1×2 for SCface. For the Local ISV approach, we use region sizes of 4×4 for MOBIO, 2×2 for Multi-PIE and 2×2 for SCface.

Table 2 shows the performance of the proposed approaches and the baselines. It was found that the Local ISV approach performs best in all cases except for the SCface evaluation dataset, which obtains best performance using the ISV system. The Local ISV modelling technique marginally improves the verification performance in the *dev* set and marginally decreases the performance in the *eval*. This marginal performance degradation is likely due to the large block size used (20×20) in conjunction with many images in the SCface database being up-sampled to have an inter-eye distance of 33 pixels. The Local ISV system provides an average relative performance improvement of 32% for the MOBIO and Multi-PIE databases. We also note that the Local GMM-FP system consistently outperforms the GMM-FP system on all three databases, with an average relative improvement of 18%, further demonstrating the value of a region based approach. The Local ISV approach outperforms the Local GMM-FP system on all three databases, and demonstrates the value in modelling session variability and capturing identity information locally.

5.3. Evaluation of Face Verification Performance in the Presence of Localisation Error

The performance of face verification in the presence of localisation noise is evaluated as outlined in Section 4.3. Figures 8 and 9 show the half total error rate (HTER) of the Local GMM-FP and Local ISV face verification systems and their respective baselines (GMM-FP and ISV) in the presence of increasing levels of face localisation noise on the MOBIO database. The same systems configurations as those in Section 5.2 are used. We evaluate performance at five different noise levels: no noise; and with localisation

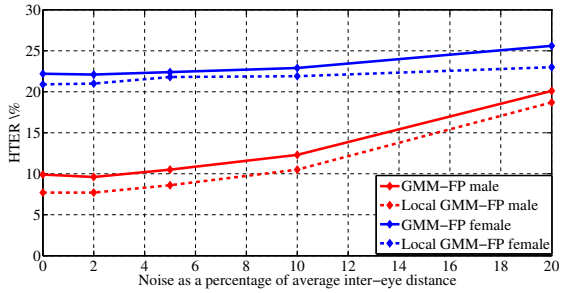


Figure 8: Performance of the Local GMM-FP and GMM-FP face verification systems in the presence of face localisation noise on MOBIO database evaluation set.

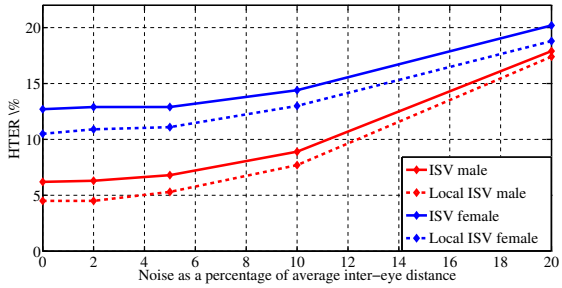


Figure 9: Performance of the Local ISV and ISV face verification systems in the presence of face localisation noise on MOBIO database evaluation set.

error of up to 2%, 5%, 10% and 20% of the average inter-eye distance.

For both the proposed and baseline systems, system performance degrades as noise increases. At levels of noise up to 20% of the average inter-eye distance the proposed approaches outperform their baselines. However, as noise is increased above 10%, the proposed performance of all systems degrades considerably (see Figure 8).

This increased degradation is likely caused by the nature of the region based systems. At high levels of noise and with small region sizes, the locations of the regions relative to the face changes significantly. Thus the assumption that corresponding regions between the client model and probe image are modelling the same portion of the face is increasingly likely to be violated as noise increases. However this effect could be mitigated by using fewer regions (i.e. 2×2 rather than 4×4), which would incur a small drop in performance under ideal conditions, but offer greater invariance to localisation errors.

6. Conclusions and Future Work

This works shows that state-of-the-art performance can be obtained for fish and face image classification through a region based, Local ISV modelling technique. This ap-

System	MOBIO (female)		MOBIO (male)		SCface		Multi-PIE	
	Dev	Eval	Dev	Eval	Dev	Eval	Dev	Eval
MRH [12]	14.5	21.9	13.6	13.0	28.3	30.3	4.8	6.2
GMM-FP	11.5	22.2	7.5	9.9	16.7	16.3	3.1	3.8
Local GMM-FP	10.3	20.9	4.8	7.7	15.7	15.9	1.1	2.3
ISV	6.7	12.7	4.1	6.2	13.6	**12.8**	1.6	2.2
Local ISV	**5.2**	**10.5**	**2.5**	**4.5**	**12.0**	13.4	**0.6**	**1.1**

Table 2: Face Verification Results. The MRH results are taken from [12]. Results for the *Dev* data are equal error rates, while results for the *Eval* data are half total error rates. The best performing systems are shown in **bold**.

proach allows noise (in the form of session variation) to be modelled locally, while also capturing local identity information. For the first time, we have applied the ISV model to challenging natural world images of fish to examine the broad applicability of this technique to the more general object classification domain, and have shown that the Local ISV approach outperforms the standard ISV by 35%. In the face verification task, the Local ISV technique outperforms the standard ISV technique by an average of 32% for the MOBIO database and Multi-PIE unmatched illumination data set. We have shown that the Local GMM-FP system also consistently outperforms the GMM-FP system on all three face databases with an average relative improvement of 18%, further demonstrating the value of a region based approach.

In addition to this, we have evaluated the real-world applicability of the Local ISV approach to face verification in the presence of face localisation error. It has been shown that Local ISV outperforms baseline systems at noise levels of up to 20% of the average inter-eye distance. Future work will consider the selection of weights for combining the region based models, and will investigate approaches to incorporate features such as colour into the models, which may be of particular use for classification of natural images.

Acknowledgements

NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program. This research was supported in part by a grant form Cooperative Research Centre for Smart Services - (CRC-SS) and by an Australian Research Council (ARC) Discovery grant DP110100827.

References

[1] A. Anjos, L. E. Shafey, R. Wallace, et al. 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.

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, volume 1, pages 886–893 vol. 1, 2005.

[3] L. El-Shafey, C. McCool, and S. Marcel. A scalable formulation of probabilistic linear discriminant analysis: Applied to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

[4] O. Glembek, L. Burget, N. Dehak, et al. Comparison of scoring methods used in speaker recognition with joint factor analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2009. ICASSP 2009.*, pages 4057–4060, 2009.

[5] M. Grgic, K. Delac, and S. Grgic. Scface — surveillance cameras face database. *Multimedia Tools Appl.*, 51(3):863–879, Feb. 2011.

[6] R. Gross, I. Matthews, J. Cohn, et al. Multi-pie. *Image Vision Comput.*, 28(5):807–813, May 2010.

[7] G. B. Huang, M. Mattar, H. Lee, et al. Learning to align from scratch. In *Neural Information Processing Systems*, 2012.

[8] P. Li, Y. Fu, U. Mohammed, et al. Probabilistic models for inference about identity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):144–157, 2012.

[9] S. Lucey and T. Chen. A GMM parts based face representation for improved verification through relevance adaptation. In *Proc. 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–855–II–861 Vol.2, 2004.

[10] C. McCool, V. Chandran, S. Sridharan, et al. 3D Face Verification using a Free-Parts Approach. *Pattern Recognition Letters*, 29:1190–1196, 2008.

[11] C. McCool, S. Marcel, A. Hadid, et al. Bi-modal person recognition on a mobile phone: using mobile phone data. In *IEEE ICME Workshop on Hot Topics in Mobile Multimedia*, 2012.

[12] C. McCool, R. Wallace, M. McLaren, et al. Session variability modelling for face authentication. *IET Biometrics*, 2:117–129(12), September 2013.

[13] K. Messer, J. Kittler, M. Sadeghi, et al. Face authentication test on the banca database. In *International Conference on Pattern Recognition*, pages 523–532, 2004.

[14] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10:2000, 2000.

[15] C. Sanderson and B. Lovell. Multi-region probabilistic histograms for robust and scalable identity inference. In *Advances in Biometrics*, volume 5558 of *Lecture Notes in Computer Science*, pages 199–208. Springer Berlin Heidelberg, 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] R. Vogt and S. Sridharan. Explicit modelling of session variability for speaker verification. *Computer, Speech and Language*, 22:17–38, 2008.

[18] R. Wallace, M. McLaren, C. McCool, et al. Inter-session variability modelling and joint factor analysis for face authentication. In *International Joint Conference on Biometrics*, 2011.

[19] R. Wallace, M. McLaren, C. McCool, et al. Cross-pollination of normalisation techniques from speaker to face authentication using

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

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

[†]Australian Centre for Robotic Vision, Brisbane, Australia
[‡]Queensland University of Technology, Brisbane, QLD 4000, Australia
[◇]NICTA, PO Box 10522, Adelaide St, Brisbane, QLD 4001, 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].

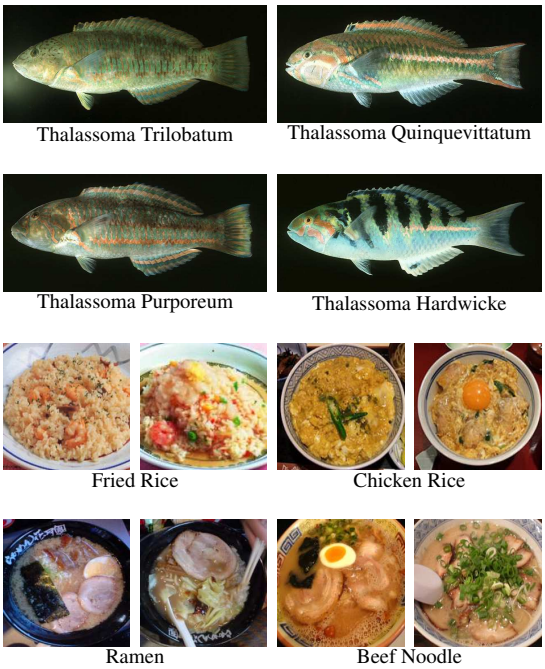


Fig. 1: First two rows show example images of four fish species, which have low inter-class variation: similar visual appearance despite being distinct species. (Images taken by J.E. Randall). The last two rows show images of four food dishes, with each dish type having high intra-class variation.

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 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.

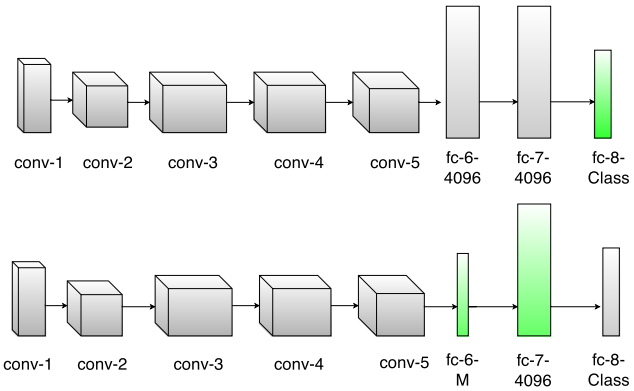


Fig. 2: Modifying and retraining the deep CNN through a 2 step procedure. For each step we have shaded in green the parts of the network that are changed and retrained. First step: the highlighted fc-8 layer is modified to have only as many outputs as the number of dataset specific classes. The layer 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.

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.

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**.

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;

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

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**

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.

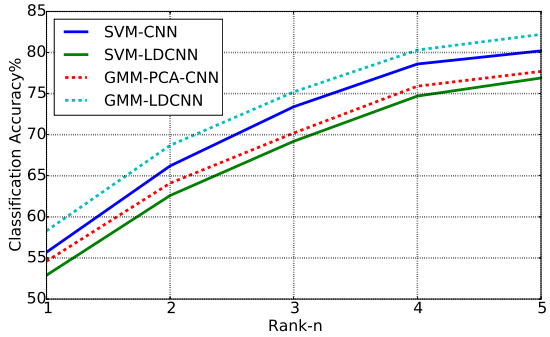


Fig. 3: Rank-n classification accuracy on the UEC FOOD-100 dataset [12].

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. 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.

Chapter 4

Hierarchical Reasoning for Fine-Grained Classification

As discussed in chapter 3, the classes are often similar in terms of shape, colour and texture because they belong to the same overarching category (eg.birds). In the previous chapter, the proposed local ISV algorithm is able to distinguish similar looking sub-categories (fish) by modelling local parts. However, it assumes that all images are well aligned with minimal pose and viewpoint variation. Furthermore, the capacity of the session variation modelling method heavily relies on the number of Gaussians in each class model and it is scale-variant because of the fixed size of the local patches being extracted from images. Unfortunately, such an approach is difficult to translate to other fine-grained problems when the objects' photos are taken in natural environments with large pose and scale variations. One example is bird classification where bird images are taken with various poses such as flying, walking and swimming. In this chapter we explore the second research question "Can we learn robust and discriminative features in order to classify fine-grained classes which have small inter-class variations?" .

This chapter looks at two aspects. The first aspect is to divide the classes into K subsets of visually similar classes; an expert classifier is then learnt for each subset. The second aspect uses the same subset of visually similar classes and, instead of learning an expert classifier, learns discriminative features for each subset using DCNNs. The Both approaches can be applied on top of any explicit parts modelling methods such as DPD [Zhang et al., 2013a]. This subset pre-clustering method can be illustrated as a two layer hierarchical structure. Each subset serves as a node and each specific class is a leaf.

Our proposed hierarchical structure operates in a fully automatic manner and can be used for various fine-grained classification tasks. On the challenging CUB200-2011 bird dataset, we show that considerable performance improvements can be achieved with our proposed approach. The mean accuracy increases from 60.5% following the baseline global approach of Donahue et al. [2014], to 71.4% for the hierarchical classifier approach when ground truth cluster labels are used. The fully-automatic system can achieve an accuracy of 68.6%. This is a substantial performance improvement and highlights the potential benefits that are possible when an hierarchical approach is used. It is important to note that without using parts information, we still achieved impressive results compared to those methods using a parts-based model [Berg and Belhumeur, 2013, Chai et al., 2013b, Donahue et al., 2014]. The hierarchical feature learning approach reaches 77.5% on CUB200-2011. We also applied to the plant classification problem on the PlantCLEF dataset. Our approach won second place in the PlantCLEF 2015 competition.

These two approaches provide considerable improvements in performance but their accuracy is limited by the accuracy of assigning a class to its correct subset. In the next chapter we extend this work by probabilistically assigning the responsibility of producing each sample.

"Fine-Grained Bird Species Recognition via Hierarchical Subset Learning" was presented at the 2015 International Conference on Image Processing, "Subset Feature Learning for Fine-Grained Category Classification" was presented at 2015 Computer Vision and Pattern Recognition Deep Vision Workshop and "Content Specific Feature Learning for Fine-Grained Plant Classification" was published as a working note at the 2015 International Conference and Labs of Evaluation Forum.

FINE-GRAINED BIRD SPECIES RECOGNITION VIA HIERARCHICAL SUBSET LEARNING

ZongYuan Ge[†‡], *Chris McCool*[†‡], *Conrad Sanderson*[◇], *Alex Bewley*[‡], *Zetao Chen*[†‡], *Peter Corke*[†‡]

[†]Australian Centre for Robotic Vision, Brisbane, Australia
[‡]Queensland University of Technology, Brisbane, QLD 4000, Australia
[◇]NICTA, PO Box 10522, Adelaide St, Brisbane, QLD 4001, Australia

ABSTRACT

We propose a novel method to improve fine-grained bird species classification based on hierarchical subset learning. We first form a similarity tree where classes with strong visual correlations are grouped into subsets. An expert local classifier with strong discriminative power to distinguish visually similar classes is then learnt for each subset. On the challenging Caltech200-2011 bird dataset we show that using the hierarchical approach with features derived from a deep convolutional neural network leads to the average accuracy improving from 64.5% to 72.7%, a relative improvement of 12.7%.

***Index Terms*—** fine-grained classification, subset clustering

1. INTRODUCTION

Fine-grained image classification is a challenging computer vision problem. Distinct from general object classification which aims to find the correct overall category such as a bird or dog, fine-grained image classification aims to identify the particular sub-category of a given category [1, 13, 14]. As an example, for an overall category of *bird* we wish to discriminate between various sub-categories with similar appearance, as shown in Fig. 1. In fact, bird classification is an area of particular interest within fine-grained image classification [3, 5, 7, 8].

Recent work in bird classification has concentrated on the issues of pose and view-point variation by finding local parts or extracting normalised features. Several authors have examined ways in which locating the parts of the birds (and other animals) can be used to improve classification [4, 5, 14]. Extracting pose-normalised features has been another popular approach [18] and is the basis for the deep convolutional bird classification system of Donahue et al. [6].

Aside from the issue of pose and view-point changes, a major challenge for any fine-grained classification approach is how to distinguish between classes that have high visual correlations. In Fig. 1 it can be seen that the *hooded oriole* and *baltimore oriole* species are visually very similar, but can be easily differentiated from the *black throate* species. This visual similarity was exploited by Berg and Belhumeur [2] to build a similarity tree that divides visually similar classes into subsets, which in turn was used to help derive a visual field guide. However, the application of the similarity tree to automatic classification for bird images has not been explored.

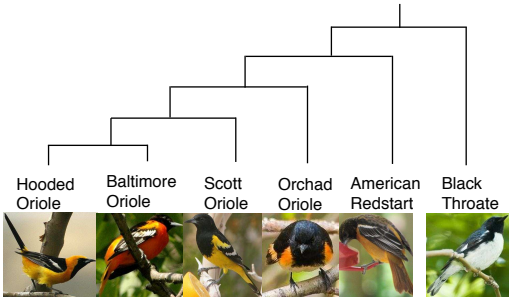


Fig. 1: One subset of the similarity tree of Berg and Belhumeur [2], built from the visual similarity matrix based on part-based one-vs-one features [3]. Species from the same node (eg. oriole) appear very similar to each other in terms of overall color and texture.

Inspired by the similarity tree of Berg and Belhumeur, we propose a hierarchical approach for fine-grained image classification. Our hierarchical approach begins by clustering visually similar classes before learning separate expert local classifiers which focus on discriminating the similar classes.

As a baseline for bird classification, we use the recently proposed deep convolutional feature approach of Donahue et al. [6]. This approach first performs part detection and pose normalisation, followed by extracting local features. The part detection and pose normalisation is achieved by using the deformable part descriptors model [18] on local parts which have been extracted using a pre-trained deep convolutional neural network (DCNN) learned from ImageNet [12]. Features obtained from the 6-th layer (fc-6) of the DCNN are used which are then classified using a linear regression approach.

The paper is continued as follows. In Section 2 we present our proposed hierarchical classification system in detail. Section 3 is devoted to a comparative evaluation with several recent methods on the task of fine-grained bird classification. Conclusions and possible future avenues of research are given in Section 4.

2. PROPOSED HIERARCHICAL CLASSIFICATION

Our proposed approach to hierarchical fine-grained image classification consists of two steps. First, the system performs a coarse classification to assign the test sample to the most likely subset k using a *subset selector*. Each subset consists of visually similar species; the subsets are automatically generated using a similarity tree. Secondly, if the confidence of the *subset selector* is sufficiently high, for each chosen subset k, fine-grained classification is performed using a local classifier $LocalSVM_k$. Each $LocalSVM_k$ has been trained to differentiate between the visually similar species belonging to this subset. If the confidence is low, a one-vs-all *GlobalSVM* classifier is used. An overview of the system can be seen in Fig. 2. The details of each component are explained in the following subsections.

2.1. Automatically Obtaining the Similarity Tree

There are two main issues with using the similarity tree of Berg and Belhumeur [2] to derive our hierarchical structure. First, it has a deep hierarchical structure of up to 17 layers and in this work we wish to explore the potential for a shallow structure of just 2 layers. Second, we want to generate the hierarchical structure in a fully automatic manner. In contrast, the similarity tree in [2] is learned from features obtained from manual part annotation which may not always be possible or desirable.

Our aim is to derive a similarity tree that groups all of the J_i samples of class i to the same subset (cluster), as well as grouping together similar classes. To do this we first obtain discriminant features by applying linear discriminant analysis (LDA) [15] to DCNN-based features (see Section 3 for more details). We use discriminant features as they will aid in having samples from the same class being assigned to the same subset (cluster). Using these discriminant features we then learn the similarity tree by performing k-means clustering.

An issue with this automatically derived similarity tree is that not all of the samples from a class are assigned to just one cluster (subset). To deal with this issue we use the result of k-means as an initial split of classes into subsets. We then determine the subset s_k which contains the majority of its samples for each class i and declare this as being the subset responsible for that class. Using this assignment of classes to subsets, we then learn a discriminative *subset selector* so that we can more accurately assign a sample to its correct subset.

2.2. Subset Selectors

We train a discriminative subset selector to minimise the number of mis-assignments of species to its subset. The k-th subset is assigned I_k classes, and so the subset selector $Selector_k$ is trained to correctly assign all the samples from these I_k classes. The positive samples to train the subset selector consist of all the training samples for the I_k classes and the negative samples are the remaining training samples.

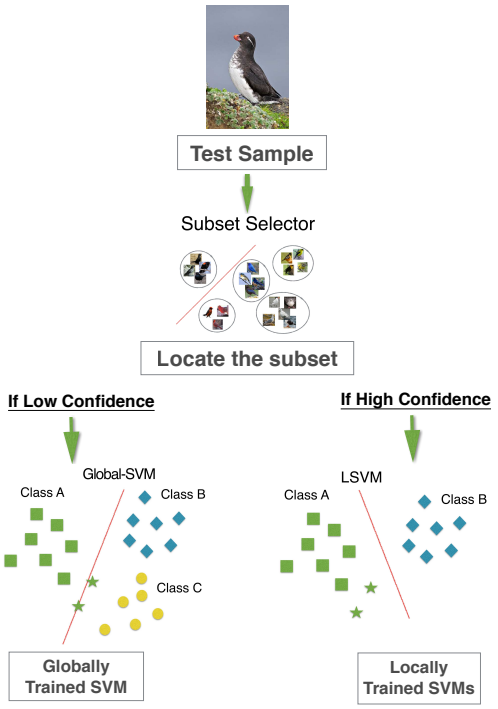


Fig. 2: An overview of the proposed hybrid system (the green stars are test samples for class A). A test image is first coarsely classified into a subset, and receives a confidence on the classification. If the confidence is higher than a predefined threshold, a local classifier *LocalSVM* specific to the chosen subset is used to make the final decision. Otherwise, a one-vs-all SVM (termed *GlobalSVM*) is used to make the decision.

In total, K subset selectors $Selector_{1..K}$ are trained, one for each subset of the hierarchical structure. These subset selectors are trained using a probabilistic SVM as this provides the probability that a sample belongs to a particular subset. This allows us to mitigate potential errors by incorporating this knowledge in the next step.

2.3. Local Expert Classifier Learning

Let $S = \{s_k\}_{k=1}^{K}$ denote the K subsets learned by the hierarchical clustering. An expert classifier (SVM) is then learned for each subset s_k which we term $LocalSVM_k$. Each $LocalSVM_k$ is a linear multi-class SVM. This is different to the classical one-versus-all approach because only the I_k classes assigned to the subset are used to train each *LocalSVM*.

2.4. Hybrid Decision System

The accuracy of the proposed system is dependent on the accuracy of the assignment of a test sample to the correct subset of our hierarchy. If the wrong subset is chosen then we have no way to recover and a mis-classification will occur. To alleviate this issue, we present a hybrid decision system which makes use of the classical global classifier, *GlobalSVM*, as well as our local classifier, *LocalSVM*.

Our hybrid decision system makes use of the probability from the subset selector to combine *GlobalSVM* and the *LocalSVM*. It uses the locally trained classifier ($LocalSVM_k$) only when the confidence of the subset selector is greater than a pre-defined threshold τ. In all other cases the classical *GlobalSVM* trained with all birds species is used to make the classification decision.

3. EXPERIMENTS

We evaluate our approach on the Caltech birds dataset (CUB200-2011) [17]. It contains 11,788 images from 200 bird species in North America. Each species has approximately 30 images for training and 30 for testing. Each image comes with an annotated bounding box around the object of interest (the bird), as well as annotations for many constituent parts of the object.

The feature vectors that we use throughout our experiments are the DCNN features (DeCAF) trained from ImageNet [12]. We fine-tune these features, using Caffe [10], for the task of bird classification by replacing the final output layer (for the $1,000$ classes of ImageNet) with a 200 class layer for bird species. We then retrain the entire network using the training samples for the 200 bird classes with a learning rate of 0.01[1].

The experiments are divided into two parts: **(i)** performance of the proposed hierarchical approach for varying number of subsets, and **(ii)** performance comparison of the proposed system against several recent algorithms. Based on preliminary experiments, the threshold for confidence of the subset selector is set to $\tau = 0.98$ for all experiments.

We first evaluate the performance of the proposed system by varying the number of subsets $K = [2, 3, \ldots, 25]$. The results are presented in Fig. 3, along with the performance of the baseline system DPD-DeCAF [6]. The performance of the proposed system generally increases until $K = 8$, reaching 72.7%. For higher values of K (ie. more subsets), the performance tends to decrease in a non-monotonic manner, indicating that relatively large values of K are not necessarily helpful. A visualisation of the classes assigned to each subset is given in Fig. 4.

Comparisons against other methods are shown in Tables 1 and 2. In Table 1 parts annotations are exploited, while in Table 2 only bounding boxes are used. It can be seen that in Table 1 the proposed method (using the optimal $K = 8$) leads to a relative performance improvement of 12.7% over the baseline DPD-DeCAF system. When ground-truth labels are used for the subset selector, the proposed system can increase its performance from 72.7% to 78.6%. This indicates that if the performance of the subset selector can be improved, we can further improve the performance of the overall system.

In Table 2, where only bounding boxes are used instead of parts annotations, the best performance by the proposed method is obtained at $K = 14$. The proposed method achieves an accuracy of 69.2% compared to 61.8% obtained by a convolutional neural network method presented in [16], resulting in a relative performance improvement of 12.0%.

[1] This rate decreases by a factor of 10 every $5,000$ iterations for a total of $20,000$ iterations.

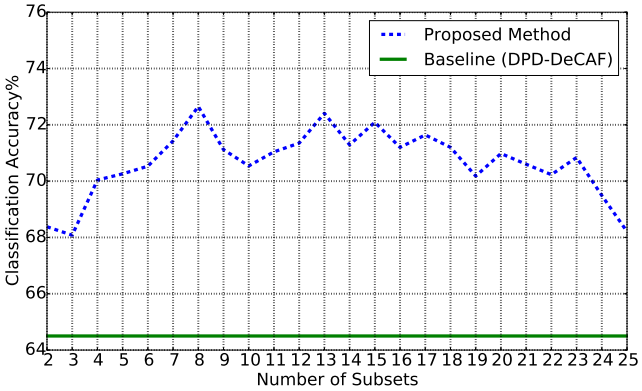


Fig. 3: Performance of the proposed method on the Caltech-UCSD CUB200-2011 bird dataset, while exploting part annotations. The number of subsets (K) is varied from 2 to 25. The subsets are selected automatically. Performance of the baseline system DPD-DeCAF [6] is also shown.

Table 1: Accuracy of various systems on the Caltech-UCSD CUB200-2011 bird dataset, exploiting part annotations.

Method	Accuracy
Pooling feature learning [11]	38.9%
Symbiotic Model [5]	59.4%
POOF [3]	56.9%
Part transfer [9]	57.8%
DPD-DeCAF [6]	64.5%
Proposed method (automatic subsets, K=8)	**72.7%**
Proposed method (ground truth subsets, K=8)	78.6%

Table 2: As per Table 1, but instead of using part annotations, only bounding box information is used.

Method	Accuracy
Bounding Box [16]	53.3%
Bounding Box-aug [16]	61.8%
Proposed method (automatic subsets, K=14)	**68.6%**

Fig. 4: Example images of 10 classes for each of the subsets for the best performing system ($K = 8$). It can be seen that the classes assigned to each subset are visually similar.

4. CONCLUSION

In this paper, we have introduced a novel direction to tackle the problem of fine-grained classification. We have proposed the use of a hierarchical classifier so that classes that have high visual correlations are grouped together into the same subsets. An expert classifier is then learnt for each subset.

The novel hybrid hierarchical classification system yields performance improvements over the recent deep convolutional neural network system proposed in [6]. This hybrid approach combines the classical *GlobalSVM* classification approach with a novel *LocalSVM* classification approach. Evaluations on the challenging CUB200-2011 dataset [17] show that classification accuracy for a fully automatic system can be increased from 64.5% to 72.7%, a relative improvement of 12.7%.

Future work will examine ways to close the gap between the performance of the automatic system and the performance of the ground truth system. The ground truth (assigning all test samples to their correct subset) achieves a classification accuracy of 78.6%, which is considerably better than the 72.7% of the fully automatic system. This implies that performing more accurate assignment of a sample to its subset can yield considerable performance improvements. One possible approach to obtain more accurate assignment would be to learn visual features that best differentiate the subsets rather than all of the classes.

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.

5. REFERENCES

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

[2] T. Berg and P. N. Belhumeur. How do you tell a blackbird from a crow? In *ICCV*, 2013.

[3] T. Berg and P. N. Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, 2013.

[4] S. Branson, P. Perona, and S. Belongie. Strong supervision from weak annotation: Interactive training of deformable part models. In *ICCV*, 2011.

[5] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013.

[6] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *ICML*, 2014.

[7] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011.

[8] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, 2013.

[9] C. Goring, E. Rodner, A. Freytag, and J. Denzler. Nonparametric part transfer for fine-grained recognition. In *CVPR*, 2013.

[10] 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.

[11] Y. Jia, O. Vinyals, and T. Darrell. Pooling-invariant image feature learning. *arXiv:1302.5056*, 2013.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[13] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *ECCV*. 2012.

[14] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur. Dog breed classification using part localization. In *ECCV*. 2012.

[15] C. R. Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203, 1948.

[16] 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.

[17] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. In *Computation & Neural Systems Technical Report, California Institute of Technology*, number CNS-TR-2011-001, 2011.

[18] N. Zhang, R. Farrell, F. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*, 2013.

Subset Feature Learning for Fine-Grained Category 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

Fine-grained categorisation has been a challenging problem due to small inter-class variation, large intra-class variation and low number of training images. We propose a learning system which first clusters visually similar classes and then learns deep convolutional neural network features specific to each subset. Experiments on the popular fine-grained Caltech-UCSD bird dataset show that the proposed method outperforms recent fine-grained categorisation methods under the most difficult setting: no bounding boxes are presented at test time. It achieves a mean accuracy of 77.5%*, compared to the previous best performance of* 73.2%*. We also show that progressive transfer learning allows us to first learn domain-generic features (for bird classification) which can then be adapted to specific set of bird classes, yielding improvements in accuracy.*

1. Introduction

Deep convolutional neural networks (CNNs) have been successful in various computer vision tasks. Deep CNNs have achieved impressive in both general [18, 22, 9] and fine-grained image classification [26, 13]. Recently, deep CNN approaches have been shown to surpass human performance for the task of recognising 1000 classes from the ImageNet dataset [16]. Although deep CNNs can serve as an end-to-end classifier, they have been used by many researchers as a feature extractor for various recognition problem including segmentation [15] and detection [14].

Recently, the task of fine-grained image categorisation has received considerable attention, in particular the task of fine-grained bird classification [26, 3, 7, 10, 12]. Fine-grained image classification is a challenging computer vision problem due to subtle differences in the overall appearance between various classes (low inter-class variation) and large pose and appearance variations in the same class (large intra-class variation).

Much of the work for fine-grained image classification has dealt with the issue of detecting and modelling local parts. Several researchers have examined methods to find local parts and extract normalised features in order to overcome the issues of pose and view-point variation [5, 7, 20, 27, 9]. Aside from the issue of pose and view-point changes, a major challenge for any fine-grained classification approach is how to distinguish between classes that have high visual correlations [3]. Some state-of-the-art pose normalised methods still have considerable difficulty in categorising some visually similar fine-grained classes [26, 6].

To date, there has been limited work which investigates in detail how best to learn deep CNN features for the fine-grained classification problem. Most of the methods used off-the-shelf convolutional neural networks (CNNs) features trained from ImageNet or fine-tuned the pre-trained ImageNet model on the target dataset, then using one fully-connected layer as a feature descriptor [17, 22].

This paper examines in detail how to best learn deep CNN features for fine-grained image classification. In doing so, we propose a novel *subset* learning system which first splits the classes into visually similar subsets and then learns domain-specific features for each subset. We also comprehensively investigate progressive transfer learning and highlight that first learning domain-generic features (for bird classification) using a large dataset and then adapting this to the specific task (target bird dataset) yields considerable performance improvements.

2. Related Work

2.1. Convolutional Neural Networks

Krizhevsky et al. [18] recently achieved impressive performance on the ImageNet recognition task using CNNs, which were initially proposed by LeCun et al. [19] for hand writing digit recognition. Since then CNNs have received considerable attention [22, 14]. The network structure of Krizhevsky et al. [18] remains a popular structure and consists of five convolutional layers ($conv1$ to $conv5$) with two fully-connected layers ($fc6$ and $fc7$) followed by a softmax layer to predict the class label. The network is capable of generating useful feature representations by learning low level features in early convolutional layers and accumulating them to high level semantic features in the latter convolutional layers [25].

978-1-4673-6759-2/15/$31.00 ©2015 IEEE

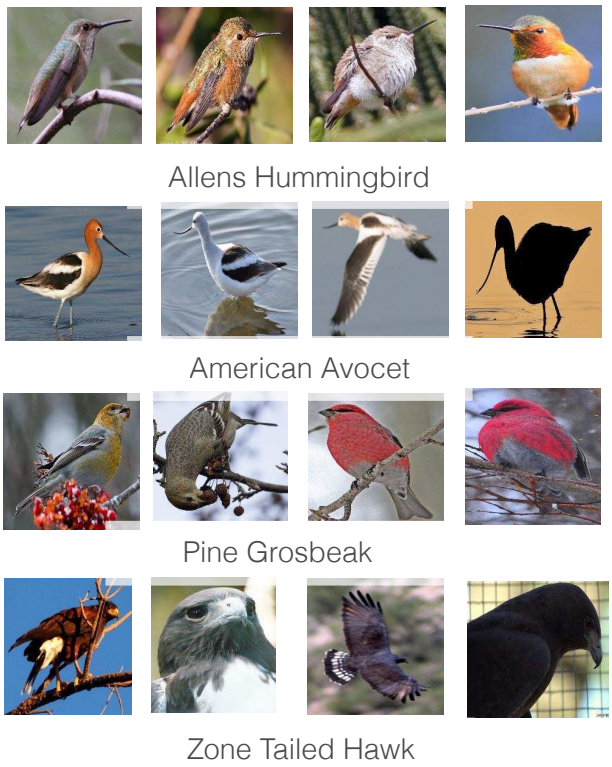


Figure 1. Birdsnap is a very challenging fine-grained bird dataset with sexual as well as age dimorphisms. There are considerable appearance differences between males and females, as well as between young and mature birds. Each row shows images from the same species. For each bird species there are large intra-class variations: pose variation, background variation and appearance variation.

2.2. Features for Fine-grained Classification

Several approaches have been designed to learn feature representations for fine-grained image classification. Berg et al. [3] generated millions of keypoint pairs to learn a set of highly discriminative features. Zhang et al. [27] learned pose normalised features by using the deformable part descriptors model (DPM) [11] on local parts which were extracted using a pre-trained deep CNN. Chen et al. [8] proposed a framework to select the most confident local descriptors for nonlinear function learning using a linear approximation in an embedded higher dimensional space.

The above feature learning schemes are implicitly part-based methods. This means they require the ground truth locations of each part which limits their usefulness in terms of fully automatic deployment.

3. Proposed Method

Our proposed feature learning method consists of two main parts. First, we perform progressive transfer learning to learn a domain-generic convolutional feature extractor (termed ϕ_{GCNN}) from a large-scale dataset of the same domain as the target dataset. Second, we perform subset-specific feature learning from pre-clustered subsets which contain visually similar fine-grained class images. The discriminative convolutional features learned from the subset learning system is termed $DFCNN$, and the related feature extractor is referred as ϕ_{DFCNN}.

For image $\boldsymbol{I}_i$, we apply the $\phi_{GCNN}(\boldsymbol{I}_i)$ and $\phi_{DFCNN}(\boldsymbol{I}_i)$ and combine them to obtain our feature vector to describe the image. For training the classifier, we employ a one-versus-all linear SVM using the final feature representation.

3.1. Progressive Transfer Learning

It is desirable to have as much as data possible in order to avoid overfitting while training a CNN. A typical CNN has millions of parameters which makes it difficult to train when data is limited. Typically fine-grained image datasets are relatively small compared to the ImageNet dataset. To circumvent problems with small datasets, a process known as transfer learning [24] can be applied. Transfer learning has usually been applied by fine-tuning a general network, such as the network of Krizhevsky et al. [18], to a specific task such as bird classification [26]. Recent work by Yosinski et al. [24] found that better accuracy can be achieved if transfer learning is performed using datasets representing the same or related domains.

Inspired by the findings of Yosinski et al. [24], we propose an alternative approach where a generic CNN is progressively adapted to the task at hand. First, a large dataset, which is related to the same domain as the final task, is used to perform transfer learning. This yields a domain-generic feature representation. Second, a smaller dataset which represents the final task at hand is used to adapt the domain-generic features to yield task-specific features. Our experimental results show that progressive transfer learning yields feature representation which lead to consistently improved performance. Furthermore, we will show that the domain-generic features can also be used effectively for the task at hand.

3.2. Subset Specific Feature Learning

Recent parts-based fine-grained methods show relatively good performance on the Caltech-UCSD bird dataset [23]. The methods are good at recognising birds species with distinguishable features with moderate pose variation. However, many mis-classifications occur for birds species that have similar visual appearance.

To address this issue, we propose to pre-cluster visually similar species into subsets and use subset-specific CNNs. Instead of relying on one CNN to handle all possible cases, each CNN focuses on the differences within each subset. In effect, the overall classifier has more parameters, as all

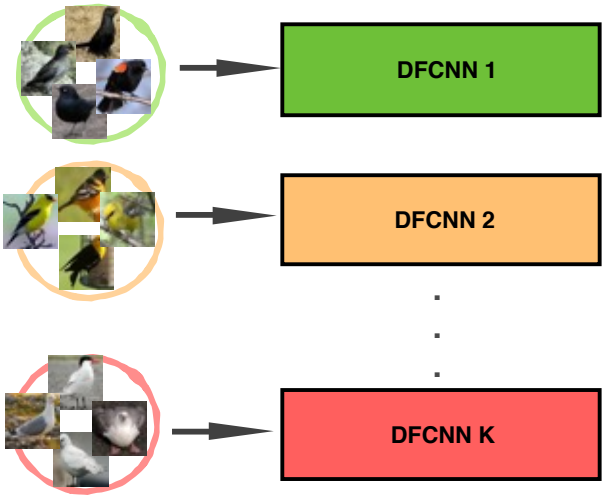


Figure 2. Pre-clustered visually similar images are fed into $DFCNN_{1...K}$ with backpropogation training to learn discriminative features for each subset.

CNNs have the same network architecture. Due to the practical issues such as training time and memory requirements, using separate CNNs dedicated to specific tasks is more practical than having one very large CNN. An overview of this subset learning scheme is shown in Fig. 2.

The above subset feature learning process is initially performed on a large yet related dataset. In particular, we use the large Birdsnap dataset [4] instead of the target Caltech-UCSD dataset [23]. We expect that our learned features are both generalised and discriminative compared to features learned directly on the same size or smaller size target dataset under the same domain.

3.2.1 Pre-clustering

To generate subsets in terms of visually similar images, image representations should focus on colour and texture while being robust to pose and background variations. We investigate three types of features as image representers. Features are obtained from either the 5-th layer $conv5$ or the 6-th layer ($fc6$) of the CNN. These were selected due to their recent use by other researchers to perform object recognition and clustering [9]. We also apply linear discriminant analysis (LDA) [21] to $fc6$ features to reduce their dimensionality. This is done to ameliorate the well known issues of clustering high dimensional data [1]. The subsets are then obtained via k-means clustering.

Examples of clustering results using the three feature types are shown in Fig. 3. The fully connected layer based feature $fc6$ fits our criteria better than clustering using the the convolutional feature $conv5$ that tends to learn shape and pose information, which is undesirable. This particular property can be seen in clusters 1 and 2 in Fig. 3(a) which

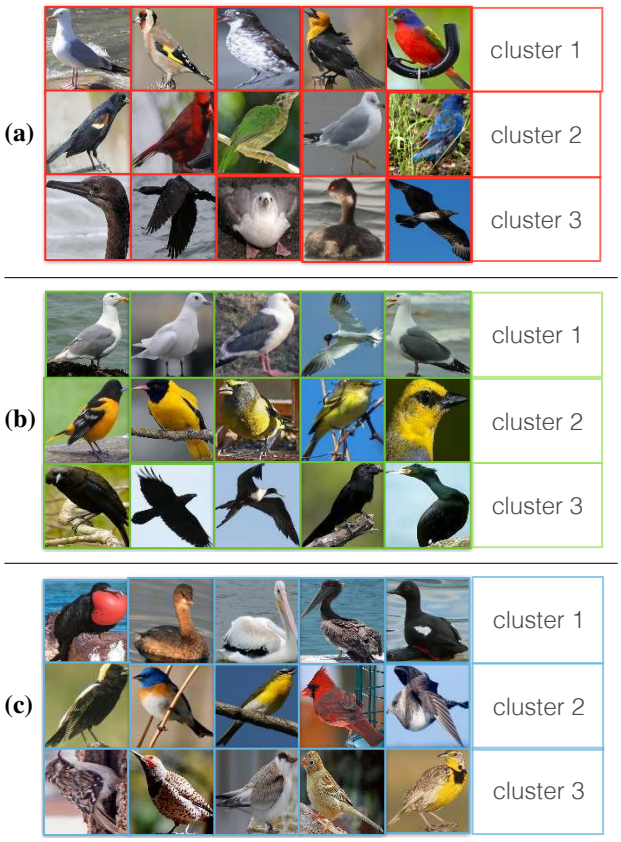


Figure 3. Pre-clustering results using: **(a)** $conv5$ layer features, **(b)** $fc6$ layer features, **(c)** $lda-fc6$ features. Clustering via $conv5$ yields undesirable strong correlations with pose and shape information. Using $fc6$ yields some improvements, but the pose bias is still visibly present. Using $lda - fc6$ provides further clustering improvements in terms of robustness to color and pose variations.

represent right and left pose of birds images while the rest are grouped into cluster 3. We conjecture that this is due to the convolutional based features containing a high degree of spatial information. Using $fc6$ yields some improvements, but the pose bias is still visibly present. Using $lda - fc6$ features provides further clustering improvements in terms of robustness to colour and pose variations.

3.2.2 Subset Feature Learning

A separate CNN is learned for each of the K pre-clustered subsets. The aim is to learn features for each subset that will allow us to more easily differentiate visually similar species. As such, for each subset, we apply transfer learning to the CNN of Krizhevsky et al. [18] (whose structure was described in Section 2). To train the k-th subset ($Subset_k$) we use the N_k images assigned to this subset $\boldsymbol{X}_k = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_k}]$, with their corresponding class labels $\boldsymbol{C}_k = [c_1, \ldots, c_{N_k}]$. The number of outputs in the

associated last fully connected layer $fc8$ is set to the number of classes in each subset. Transfer learning is then applied separately to each network using backpropogation and stochastic gradient descent (SGD). We then take $fc6$ to be the learned subset feature ϕ_{DFCNN_k} for the k-th subset.

3.3. Fine-grained Classification

To predict test labels for an image $\boldsymbol{I}_t$, our classification pipeline combines the $\phi_{GCNN}(\boldsymbol{I}_t)$ feature with the K subset features $\phi_{DFCNN_{1\ldots K}}(\boldsymbol{I}_t)$. A max voting rule is used to retain only the most relevant subset-specific feature. The other $K-1$ features are set to $\mathbf{0}$. See Fig. 4 for a conceptual representation. To balance weights for the domain-generic and subset-specific features, both $GCNN$ and $DFCNN$ features are then $l2$ normalised before combining them into a single feature vector. Using this feature vector, we train a one-versus-all linear SVM in order to make predictions.

3.3.1 Max Voting DFCNN

The final feature representation for image $\boldsymbol{I}$ is the concatenation of generalised features obtained from $\phi_{GCNN}(\boldsymbol{I})$ and the K subsets $\phi_{DFCNN_{1\ldots K}}(\boldsymbol{I})$. However, sometimes an image is more relevant to one subset features than others. For example to extract features for a White Gull image, it is more reasonable to use $DFCNN$ features from the subset which has many relevant white birds.

To emphasise the most relevant $DFCNN$, we first learn a **subset selector** to select the most relevant subset (rank 1) to the image. Max voting is then used to retain the feature from the most relevant subset and the remaining $k-1$ subset features are set to 0. One way to interpret the max voting is to use the **subset selector** to learn a binary vector $\boldsymbol{w}$, where $\sum_{i=1}^{K} \boldsymbol{w}_i = 1$. The final subset feature representation is then $DFCNN = [w_1\phi_{DFCNN_1}(x_i), \ldots, w_k\phi_{DFCNN_K}(x_i)]$. We explore two ways to learn the **subset selector**.

The simplest way of learning the **subset selector** is to use the centroids from the pre-clustering; we refer to this as $Cen_{1\ldots K}$. This provides a simple classifier trained in an unsupervised manner, however, given the importance of this stage we explore the use of a discriminatively trained classifier using a CNN.

Another way to select the most relevant subset is to train a separate CNN based subset selector $SCNN$. Using the output from the pre-clustering as the class labels, we learn a new SCNN by changing the softmax layer $fc8$ to have K outputs. The softmax layer now predicts the probability of the test image belonging to a specific subset $Subset_k$, max voting is then applied to this prediction to choose the most likely subset. As with the previously trained CNNs, the weights of $SCNN$ are trained via backpropogation and SGD using the network of Krizhevsky et al. [18] as the starting point.

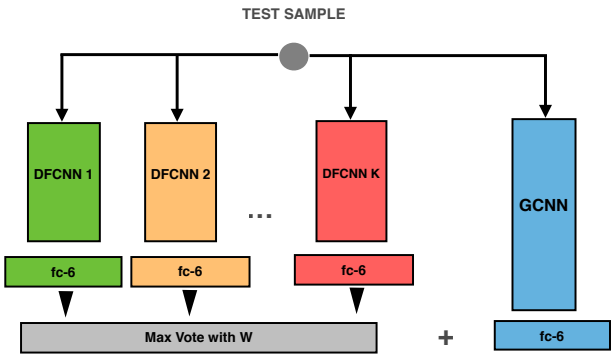


Figure 4. Feature representation of the test image is the concatenated features from both DFCNN with weighting factors and GCNN.

4. Experiments

In this section we present a comparative performance evaluation of our proposed method. We conduct experiments on the Caltech-UCSD dataset [23], which is the most widely used benchmark for fine-grained classification. We train the model using ImageNet [18] and recently released Birdsnap dataset [4].

ImageNet consists of 1000 classes with approximately 1000 images for each class. In total there are approximately 1.2 million training images.

Caltech-UCSD contains 11,788 images across 200 species. Birdsnap contains 500 species of North American birds with 49,829 images. Examples are shown in Fig. 1. Birdsnap is similar in structure to Caltech-UCSD, but has several differences. First, it contains overlapping 134 species and four times the number of images than Caltech-UCSD. Second, there is strong intra-variation within many species due to sexual as well as age dimorphisms. There are considerable appearance differences between males and females, as well as between young and mature birds.

We use the implementation of LDA and k-means from the Bob library [2]. The open-source package Caffe [17] is used to train and extract CNN features. We use $lda-fc6$ layer features to pre-cluster subsets and $fc6$ features for classification.

4.1. Evaluation of Transfer Learning for Domain-Generic Features

The CNN model architecture is identical to the model used by Krizhevsky et al. [18]. We fine-tune the CNN model by using training images from the ground truth bounding box crops of the original images. The resultant cropped images are all resized 227×227. During test time, ground truth bounding box crops of the test images from Caltech-UCSD are used to make predictions.

We conducted 3 sets of experiments for transfer learning:

1. The first experiment used all of the data from Birdsnap (500 species) to perform large-scale progressive feature learning.

2. In the second experiment we removed those species in Birdsnap and Caltech-UCSD that overlapped. This allows us to examine the potential for learning domain features that are not specific to the task at hand.

3. In the third experiment we explored the impact that including the overlapping species has on the transfer learning process.

We use the following acronyms. **IN** represents using weights from the pre-trained ImageNet model. We define **rt** as retraining the network from scratch with random initialised weights. **ft** refers to fine-tuning the network. For example, **IN-CUB-ft** means fine-tuning the ImageNet model weights on the Caltech-UCSD bird dataset. ImageNet dataset is represented as **IN**, while Caltech-UCSD is **CUB**, and Birdsnap is **BS**.

4.1.1 Transfer Learning: Experiment I

In this experiment we used all images (500 species) from Birdsnap to explore large-scale progressive feature learning. We exclude those images that exist in both Birdsnap and the Caltech-UCSD datasets.

The first three rows of Table 1 show the accuracy when the CNNs are trained from scratch. In this setting the **IN-rt** system, the pre-trained network generated by Krizhevsky et al. [18] on ImageNet, performs the best with a mean accuracy of 58.0%. Interestingly, the **BS-rt** system has a considerably higher mean accuracy of 44.8% when compared to **CUB-rt** which has a mean accuracy of 11.4%. We believe that this indicates that the Birdsnap dataset has almost enough data to train a deep CNN from scratch.

Transfer learning offers a way to mitigate the lack of sufficient domain data. As such, we performed transfer learning by fine-tuning the pre-trained CNN. We did this using just the Caltech-UCSD (target) dataset **IN-CUB-ft** or the Birdsnap (domain specific) dataset **IN-BS-ft**.

Somewhat surprisingly, training on the target dataset (**IN-CUB-ft**) provides a lower mean accuracy of 68.3% when compared to using the domain specific dataset (**IN-BS-ft**) which has a mean accuracy of 70.1%. Performing progressive feature learning on the **IN-BS-ft** CNN leads to further improvements achieving a mean accuracy of 70.8% (**IN-BS-ft-CUB-ft**). These two results demonstrate the potential for learning domain-generic features (**IN-BS-ft**) as well as progressive feature learning to perform effective transfer learning (**IN-BS-ft-CUB-ft**) for fine-grained image classification.

Table 1. Mean accuracy of transfer learning on the Caltech-UCSD bird dataset (bounding box annotation provided). Steps represents the number of training stages.

Method	Steps	Mean Accuracy
All species (500)		
IN-rt	1	58.0%
CUB-rt	1	11.4%
BS-rt	1	44.8%
IN-CUB-ft	2	68.3%
IN-BS-ft	2	70.1%
IN-BS-ft-CUB-ft	3	**70.8**%
Non-overlapping species (366)		
IN-BS-ft	2	67.7%
IN-BS-ft-CUB-ft	3	**70.5**%
Overlap (134) + Random (232)		
IN-BS-ft	2	69.5%

An obvious issue that is not addressed in this first experiment is that there are overlapping species in Birdsnap and Caltech-UCSD. To evaluate the impact of this we perform two more experiments.

4.1.2 Transfer Learning: Experiment II

Next we investigate transfer learning features from non-overlapping classes between two bird datasets. We fine-tune the pre-trained CNN using those species from the Birdsnap dataset that do not overlap with Caltech-UCSD. There are 134 species that overlap and so we only use 366 species for this experiment.

As can be seen from the second part of the Table. 1, the result of transfer learning on Birdsnap in this setting is slightly worse with a mean accuracy of 67.7%. However, if we perform progressive feature learning by learning on the target dataset (**IN-BS-ft-CUB-ft**) we obtain a mean accuracy of 70.5%. This is only 0.3% worse than if we used all of the Birdsnap data and demonstrates the effectiveness of progressive feature learning.

4.1.3 Transfer Learning: Experiment III

In this experiment we show the importance of overlapping classes for learning domain-generic features. In order to investigate if the overlapping classes play a key role to learn domain-generic features, we fine-tuned the ImageNet model again with 134 overlapping species and 232 randomly selected unique species from the Birdsnap; this gives us 366 species which is the number of species available in Experiment II. The result shows that overlapping species

are important to learn domain-generic species with a mean accuracy of 69.5%.

4.2. Evaluation of Subset Specific Features

In this set of experiments we evaluate our proposed subset feature learning method on Caltech-UCSD. We use the same evaluation protocol as domain-generic feature learning in the previous section, where the $DFCNN$ is used to extract features from given ground truth bounding box location of the whole bird. We use the acronym **SF** to indicate subset feature learning. Based on initial experiments we set $K = 6$.

Results in Table 2 show that subset feature learning provides considerable improvements. As a baseline, the results from [26] are shown, where the features were fine-tuned on the Caltech-UCSD dataset; this equates to **IN-CUB-ft** in our terminology. Comparing to this baseline, both of our proposed subset feature learning methods, **IN-BS-ft-SF(SCNN)** and **IN-BS-ft-SF(k-means)**, provide considerable improvements with mean accuracies of 72.0% and 70.4% respectively. This demonstrates the effectiveness of our proposed subset feature learning technique, and the importance of the subset selector as the SCNN approach provides an absolute performance improvement of 1.6% when compared to the much simpler k-means approach.

4.3. Comparison with State-of-the-Art

In this section we demonstrate that subset feature learning can achieve state-of-the-art performance for automatic fine-grained bird classification. Recent work in [26] provided state-of-the-art performance on the Caltech-UCSD dataset. This was achieved by crafting a highly accurate parts localisation model which leveraged deep convolutional features computed on bottom-up region proposals based on the RCNN framework [14] . We show that if we use a similar approach but substitute their global feature vector with the feature vector obtained from subset feature learning, then state-of-the-art performance can be achieved.

We present our results under the same setting as [26], where the bird detection bounding box is unknown during test time. This setting is fully automatic and hence more realistic. Since we concentrate on feature learning we use the detection results and parts features from [26], and substitute their global feature vector with the one we learn from subset feature learning.

The results in Table 3 show that our proposed method achieves a mean accuracy of 77.2% when we use domain-generic features and subset-specific features. This is a considerable improvement over the previous state-of-the-art system [26] which achieved a mean accuracy of 73.2%. An extra 0.3% performance is gained when we perform progressive feature learning and fine-tune the CNN model again on the Caltech-UCSD dataset. Qualitative results are shown in Fig. 5 which highlight instances where the previous state-of-the-art methods provides an incorrect class label despite large visual dissimilarities. In contrast, our approach provides the correct class label.

Table 2. Mean accuracy on the Caltech-UCSD bird dataset of subset-specific features learned using subset feature learning. Annotated bounding boxes are used.

Method	Mean Accuracy
Fine-tuned Decaf [26]	68.3%
IN-BS-ft + SF(k-means)	70.4%
IN-BS-ft + SF(SCNN)	**72.0%**

Table 3. Comparison to recent results on the Caltech-UCSD bird dataset. Bounding boxes are not used.

Method	Mean Accuracy
DPD-DeCAF [27]	44.9%
Part-based RCNN with δ^{KP} [26]	73.2%
IN-BS-ft + SF(k-means) with δ^{KP}	76.2%
IN-BS-ft + SF(SCNN) with δ^{KP}	**77.2%**
IN-BS-ft-CUB-ft + SF with δ^{KP}	**77.5%**

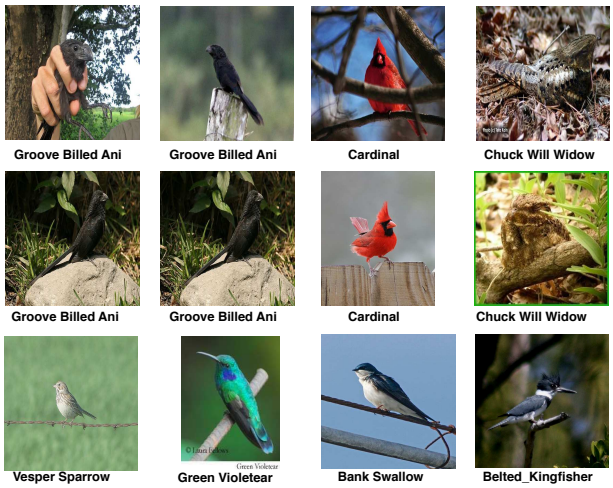


Figure 5. Qualitative comparison between our proposed method and the previous state-of-the-art approach [26] (part-based RCNN with δ^{KP}). The first row shows examples of test images, the second row shows the corresponding predicted classes from our proposed method, and the last row images shows the predictions using [26]. It can be seen that the previous state-of-the-art approach made errors despite the large visual dissimilarities between the test image and the predicted classes. In contrast, the proposed approach provides the correct class labels in these cases.

5. Conclusion

We have proposed a progressive transfer learning system to learn domain-generic features as well as subset learning to learn subset specific features. For progressive transfer

learning, we have shown that it is possible to learn domain-generic features for tasks such as fine-grained image classification. Furthermore, we have shown that progressive transfer learning of these domain-generic features can be performed to learn target set specific features, yielding considerable improvements in accuracy.

Finally, we have presented a subset feature learning system that is able to learn subset-specific features. Using this approach we achieve state-of-the-art performance of 77.5% for fully automatic fine-grained bird image classification, the most difficult setting. We believe our proposed method can be useful not only for fine-grained image classification, but also for improving general object recognition. We will examine this potential in future work.

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.

References

[1] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *International Conference on Very Large Data Bases*, pages 901–909, 2005.

[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 *ACM Conference on Multimedia Systems (ACMMM), Nara, Japan*, 2012.

[3] T. Berg and P. N. Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, 2013.

[4] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *CVPR*, pages 2019–2026, 2014.

[5] S. Branson, P. Perona, and S. Belongie. Strong supervision from weak annotation: Interactive training of deformable part models. In *ICCV*, 2011.

[6] S. Branson, G. Van Horn, S. Belongie, and P. Perona. Bird species categorization using pose normalized deep convolutional nets. *arXiv:1406.2952*, 2014.

[7] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013.

[8] G. Chen, J. Yang, H. Jin, E. Shechtman, J. Brandt, and T. X. Han. Selective pooling vector for fine-grained recognition. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 860–867, 2015.

[9] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *ICML*, 2014.

[10] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011.

[11] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[12] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, 2013.

[13] Z. Ge, C. McCool, C. Sanderson, and P. Corke. Modelling local deep convolutional neural network features to improve fine-grained image classification. *arXiv:1502.07802*, 2015.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv:1311.2524*, 2013.

[15] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, pages 297–312. Springer, 2014.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv:1502.01852*, 2015.

[17] 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:1408.5093*, 2014.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[19] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[20] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur. Dog breed classification using part localization. In *ECCV*. 2012.

[21] C. R. Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203, 1948.

[22] 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.

[23] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. In *Computation & Neural Systems Technical Report, California Institute of Technology*, number CNS-TR-2011-001, 2011.

[24] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pages 3320–3328, 2014.

[25] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. *arXiv:1311.2901*, 2013.

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

[27] N. Zhang, R. Farrell, F. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*, 2013.

Content Specific Feature Learning for Fine-Grained Plant Classification

ZongYuan Ge †, Chris McCool †, Conrad Sanderson *, and Peter Corke †

† Australian Center for Robotic Vision, Queensland University of Technology
* NICTA, Australia
Corresponding author: z.ge@qut.edu.au or c.mccool@qut.edu.au

Abstract. We present the plant classification system submitted by the QUT RV team to the LifeCLEF 2015 plant task. Our system learns a content specific feature for various plant parts such as branch, leaf, fruit, flower and stem. These features are learned using a deep convolutional neural network. Experiments on the LifeCLEF 2015 plant dataset show that the proposed method achieves good performance with a score of 0.633 on the test set.

Keywords: deep convolutional neural network, plant classification, subset feature learning

1 Introduction

Fine-grained image classification has received considerable attention recently with a particular emphasis on classifying various species of birds, dogs and plants [1, 3, 4, 11]. Fine-grained image classification is a challenging computer vision problem due to the small inter-class variation and large intra-class variation. Plant classification is a particularly important domain because of the implications for automating Agriculture as well as enabling robotic agents to detect and measure plant distribution and growth.

To evaluate the current performance of the state-of-the-art vision technology for plant recognition, the Plant Identification Task of the LifeCLEF challenge [5, 7] focuses on distinguishing 1000 herb, tree and fern species. This is an observation-centered task where several images from seven organs of a plant are related to one observation. There are seven organs, referred to as **content** types, and include images of the entire plant, branch, leaf, fruit, flower, stem or a leaf scan.

Inspired by [4], we use a deep convolutional neural network (DCNN) approach and learn a separate DCNN for each content type. We combine the content-specific feature with a generic DCNN feature, which is trained using all of the content types. This approach yields a highly accurate classification system with a score of 0.633 on the test set.

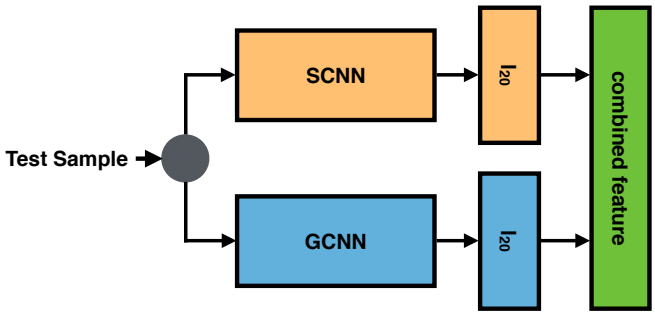


Fig. 1. For each test sample, a domain-generic (GCNN) and subset-specific (SCNN) feature is extracted. These two features are then concatenated to form a combined feature vector.

2 Our Approach

Our proposed system consists of two main parts. First, we perform transfer learning to learn a domain-generic feature termed as ϕ_{GCNN} from all plants images (regardless of content type). Second, we manually cluster the dataset into subsets based on content type and learn a feature specific to each subset (ϕ_{SCNN}). For each image we extract both domain-generic (ϕ_{GCNN}) and subset-specific (ϕ_{SCNN}) features, these features are obtained from layer 20, l_{20}, of the deep network. The two feature vectors are then concatenated to form a single feature vector as shown in Figure 1. These features are then used to learn a multi-class linear SVM. Power and l_2 norm are applied independently for domain-generic feature and content specific feature prior to combining the feature vectors.

2.1 Content Clustering

There are 7 pre-defined content types consisting of images from the *entire plant*, *branch*, *leaf*, *fruit*, *flower*, *stem* or a *leaf scan*. In both the training and testing phases all participants are allowed to use the indicated content.

We make use of the content type to learn a DCNN that is fine-tuned (specialised) for a subset of the content types. However, because there is a limited number of images for each content type, we first group the most visually similar content types toghether. In particular, we define four subsets. The first subset conists of the the *entire plant* and *branch* content types, the second subset consists of the *leaf* and *leaf scan* content types, the third subset contains *fruit* and *flower* content types, and the fourth subset consists of the *stem* only.

2.2 Deep Convolutional Neural Networks as Feature Representation

Krizhevsky et al. [8] recently achieved impressive performance on the ImageNet recognition task using CNNs, which were initially proposed by LeCun et al. [9]

for hand written digit recognition. Since then CNNs have received considerable attention and in the Large-scale ImageNet Challenge 2014 (ILSVRC) the top five results were all produced using CNN-based systems [10].

In this work we fine-tune a general model for the task of plant classification. The base model that we fine-tune is the best performing model from ILSVRC [12], referred to as GoogLeNet. GoogLeNet is a very deep neural network model with 22 layers. It consists primarily of convolutional layers. We use the output of the last convolutional layer l_{20}, after average pooling, to obtain our feature vectors.

2.3 Domain Specific Feature Learning

Transfer learning has usually been applied by fine-tuning a general network, such as the network of Krizhevsky et al. [8], to a specific task such as bird classification [13].

Inspired by the findings of Zhang et al. [13] we learn a domain-generic DCNN for the task of plant classification. This is achieved by applying transfer learning on the parameters of the GoogLeNet model (learned from the large-scale ImageNet dataset) using all of the training data for the plant classification task. This new DCNN provides domain-generic features for the task of plant classification and is referred to as the domain-generic DCNN. The only difference between the pre-trained GoogLeNet model and the domain-generic DCNN is that the number of outputs for the last fully connected layer is changed to be $1,000$ which is the number of training classes available. For each image we can then obtain a domain-generic feature ϕ_{GCNN} from the last convolutional layer l_{20}.

2.4 Subset Feature Learning as Content Specific Feature

A separate DCNN is learned for each of the $K = 4$ pre-defined subsets by fine-tuning the domain-specific model, described in Section 2.3. The aim is to learn features for each subset that will allow us to more easily differentiate visually similar content of plant species. As such, for each subset, we apply fine-tuning to the pre-trained GoogLeNet model. To train the k-th subset ($Subset_k$) we use the N_k images assigned to this subset $\mathbf{X}_k = [\mathbf{x}_1, ..., \mathbf{x}_{N_k}]$, with their corresponding class labels.

The only difference between these models and the pre-trained GoogLeNet model is that the number of outputs for the last fully connected layer, of each model, is set to the number of training classes in each subset. Transfer learning is then applied separately to each network using backpropogation and stochastic gradient descent (SGD). For each image belonging to the k-th subset a subset feature vector ϕ_{SCNN_k} is obtained by taking the output of the last convolutional layer l_{20}.

3 Experiments

In this section we present a comparative performance evaluation of our proposed method on a validation set and the defined test sets. The provided training dataset is split into two sets: roughly 10% of the total training data was used as a validation set and the rest is used for training the models. The split is based on observation id because final testing is also observation-based.

This results in 82,033 training images, including 21,746 for the *branch* and *entire* subset, 32,186 for *fruit* and *flower* subset, 23,234 for the *leaf* and *leaf scan* subset and 4,867 for the *stem* subset. The validation set consists of 9,725 images.

We use Caffe [6] for learning generic and subset specific features. The open-source package LibLinear [2] is used to train the multi-class linears SVMs. The SVM cost parameter C is set to 1 and all images are resized to 224×224.

3.1 Results on Validation Set

First we assess our proposed method on the validation set. We conducted three sets of experiments which examine the effectives of the domain-specific feature vector, the subset feature vector and the combination of these two feature vectors.

The results on the validation set, shown in Table 1, demonstrate that the combination of these two feature vectors provides a considerable performance improvement. The combination of these two feature vectors achieves a mean accuracy of 66.6%. This is an absolute improvement of 6.5 percentage points over the domain-specific feature vector ϕ_{GCNN} which achieves a mean accuracy of 60.1%. By comparison, the subset feature vector ϕ_{SCNN_k} achieves a mean accuracy of only 58.0%. We believe that the subset feature vector performs worse than the domain-specific feature vector because of the limited number of training images for each subset.

Table 1: Mean accuracy on the LifeCLEF 2015 Plant dataset of our proposed method. Annotated content information is used.

Method	**Mean Accuracy**
Domain Specific Feature	60.1%
Content Specific Feature	58.0%
Combined	**66.6%**

3.2 Results on Test Set

In this section, we present our submitted results for the LifeCLEF2015 plant challenge. We submitted three runs:

– RUN1 is the result of using proposed system for classification purpose. Only the rank 1 score is submitted for each observation.
– RUN2 is the image retrieval task where we take the first 5 predictions.
– RUN3 is based on RUN 2 but we perform an additional softmax normalization for the first five predictions.

In Figure 2 we present the overall performance for all of the competitors using the defined score metric. It can be seen that our best performing system is RUN 2 which achieved a score of 0.633. This is slightly worse than SNUMED INFO systems (RUN 4 and RUN 3).

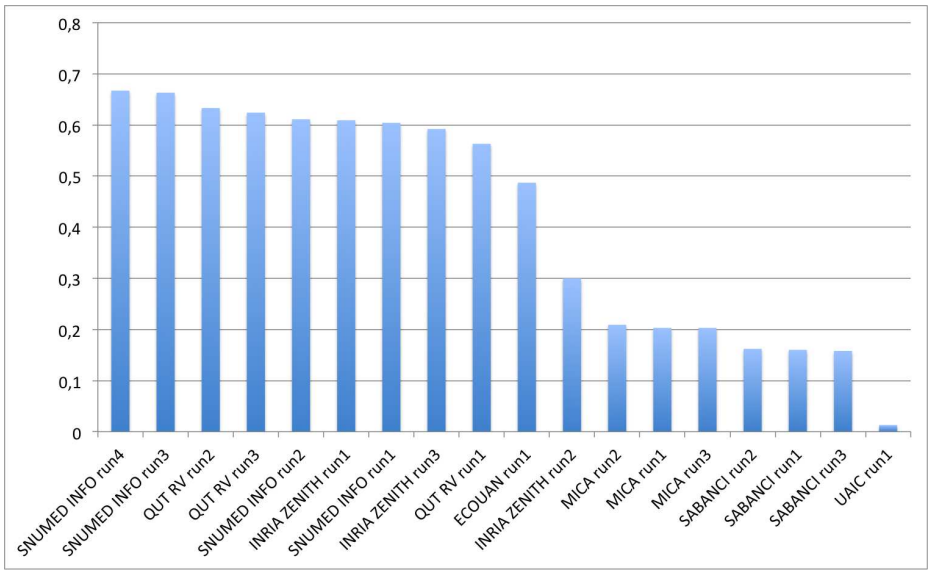


Fig. 2. The results of observation-based for the LifeCLEF Plant Task 2015. Image adapted from the organisers' website.

In Figure 3 we present results for the image-based run. It can be seen that our proposed method provides competitive performance for both the image-based and observation-based metrics. However, we do have a minor performance loss for the image-based result compared to the observation-based result.

4 Conclusions and Future Work

In this paper we presented a domain-specific feature learning and subset-specific feature learning system applied to the plant identification task of LifeCLEF 2015. For domain-specific feature learning, we have shown that it is possible to

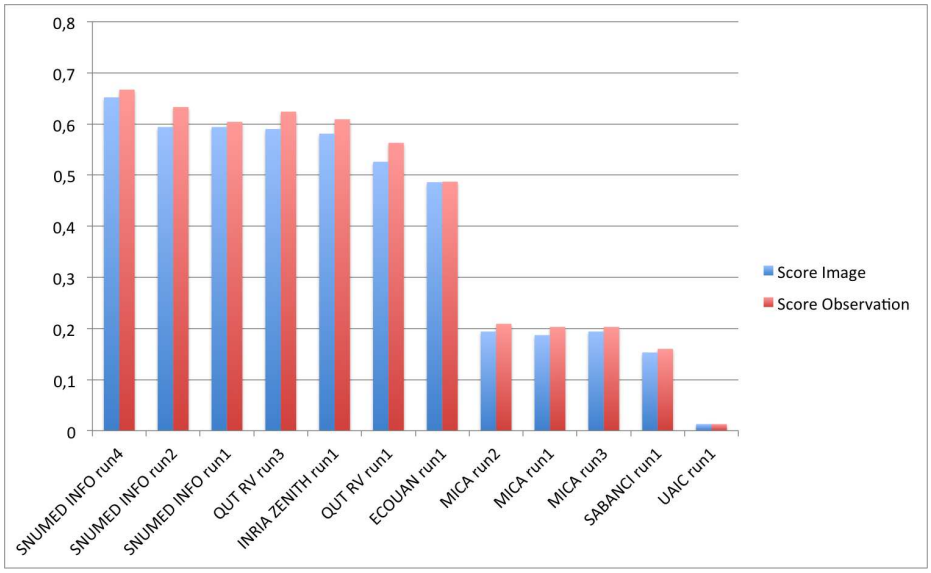


Fig. 3. The results of image-based for the LifeCLEF Plant Task 2015. Image adapted from the organisers' website.

perform transfer learning from a DCNN pre-trained on the larger-scale ImangNet dataset. Furthermore, we have presented a subset feature learning system that is able to learn content specific features. This approach yields highly competitive performance with a score of 0.633 for this year's task.

Acknowledgements

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. We would also like to thank Professor Chunhua Shen and Dr. Lingqiao Liu for the fruitful conversations of this work.

References

1. Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013.
2. Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

3. Efstratios Gavves, Basura Fernando, Cees GM Snoek, Arnold WM Smeulders, and Tinne Tuytelaars. Local alignments for fine-grained categorization. *International Journal of Computer Vision*, pages 1–22, 2014.
4. ZongYuan Ge, Christopher McCool, Conrad Sanderson, and Peter Corke. Subset feature learning for fine-grained classification. *CVPR Workshop on Deep Vision*, 2015.
5. Hervé Goëau, Alexis Joly, and Pierre Bonnet. Lifeclef plant identification task 2015. In *CLEF working notes 2015*, 2015.
6. Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014.
7. Alexis Joly, Henning Müller, Hervé Goëau, Hervé Glotin, Concetto Spampinato, Andreas Rauber, Pierre Bonnet, Willem-Pier Vellinga, and Bob Fisher. Lifeclef 2015: multimedia life species identification challenges. In *Proceedings of CLEF 2015*, 2015.
8. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.
9. Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.
10. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014.
11. Asma Rejeb Sfar, Nozha Boujemaa, and Donald Geman. Confidence sets for fine-grained categorization and plant species identification. *IJCV*, 2014.
12. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv:1409.4842*, 2014.
13. Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based R-CNNs for fine-grained category detection. In *ECCV*, pages 834–849. 2014.

Chapter 5

Fine-Grained Classification via Mixture of Deep Convolutional Neural Networks

---

In the previous chapter, hierarchical learning with DCNN-based feature has shown impressive performance on the fine-grained classification problem. However, the hierarchical system is limited by the accuracy of assigning a class to its correct subset. Another disadvantage is that either a learnt expert SVM classifier or learnt a DCNN feature extraction was needed, so joint training of features and a classifier in a single DCNN framework was not possible.

In contrast to previous techniques, in this chapter we explore a formulation to perform joint end-to-end training of multi DCNNs simultaneously. We introduce a novel system based on a mixture of deep convolutional neural networks (MixDCNNs) that provides state-of-the-art performance on two different fine-grained tasks, birds and plants. The same pre-clustering process as shown in chapter 4 is used to initialise K DCNN parameters. The main difference between the previous method and MixDCNNs is that the classification decision from each component is weighted proportionally to the confidence of its decision, which is termed an occupation probability. This allows us to define a single network (MixDCNN) to perform classification in an end-to-end mechanism, and samples can be re-assigned to the most appropriate expert network during the training process.

Empirical evaluations show that MixDCNN outperforms related approaches such as subset feature learning introduced in the previous chapter, a gated DCNN approach similar to Jacobs et al. [1991], and an ensemble of DCNNs. The results demonstrate performance improvements over three challenging fine-grained datasets including CUB-200-2011 from 80% using a single

DCNN model to 81.1% with MixDCNN, the large-scale bird dataset Birdsnap and the Plant-CLEF dataset with 6.7% and 3.4% absolute percentage improvement respectively over a single model.

The content of this chapter has been published and presented at the 2016 Winter Conference on Applications of Computer Vision under the algorithm track as “Fine-Grained Classification via Mixture of Deep Convolutional Neural Networks”.

Fine-Grained Classification via Mixture of Deep Convolutional Neural Networks

ZongYuan Ge[†‡], Alex Bewley[‡], Christopher McCool[†‡], Peter Corke[†‡], Ben Upcroft[†‡], Conrad Sanderson[*◇]

[†] Australian Centre for Robotic Vision, Brisbane, Australia
[‡] Queensland University of Technology (QUT), Brisbane, Australia
[*] University of Queensland, Brisbane, Australia
[◇] NICTA and Data61, CSIRO, Australia

z.ge@qut.edu.au c.mccool@qut.edu.au

Abstract

We present a novel deep convolutional neural network (DCNN) system for fine-grained image classification, called a mixture of DCNNs (MixDCNN). The fine-grained image classification problem is characterised by large intra-class variations and small inter-class variations. To overcome these problems our proposed MixDCNN system partitions images into K subsets of similar images and learns an expert DCNN for each subset. The output from each of the K DCNNs is combined to form a single classification decision. In contrast to previous techniques, we provide a formulation to perform ***joint*** *end-to-end training of the K DCNNs simultaneously. Extensive experiments, on three datasets using two network structures (AlexNet and GoogLeNet), show that the proposed MixDCNN system consistently outperforms other methods. It provides a relative improvement of 12.7% and achieves state-of-the-art results on two datasets.*

1. Introduction

Fine-grained image classification consists of discriminating between classes in a sub-category of objects, for instance the particular species of bird or dog [2, 5, 8, 9, 23]. This is a very challenging problem due to large intra-class variations (due to pose and appearance changes), as well as small inter-class variation (due to only subtle differences in the overall appearance between classes). See Fig. 1 for examples.

To cope with the above problems, many fine-grained classification methods have performed parts detection [2, 5, 20, 24] in order to decrease the intra-class variation. Recently, an alternative approach was introduced by Ge et al. [13] where the images were first partitioned into K non-overlapping sets and K expert systems were learned. By grouping similar images, the input space is being partitioned so that an expert network can better learn the subtle differences between similar samples. Expert selection was performed by training a dedicated gating network which assigns samples to the most appropriate expert network. This approach has two downsides. Firstly, a separate gating network (subset selector) needs to be trained. Secondly, the expert networks are trained only to extract features, leaving the final classification to be performed by a linear support vector machine (SVM).

Figure 1. Example images from the Birdsnap dataset [3] which exhibits large intra-class variations and low inter-class variations. Each column represents a unique class.

We propose a novel system based on a mixture of deep convolutional neural networks (DCNNs) that provides state-of-the-art performance along with several important properties. Similar to Ge et al. [13], we partition the data into K non-overlapping sets to learn K expert DCNNs. However, unlike [13], the classification decision from the each expert is weighted proportional to the confidence of its decision. This allows us to define a single network (MixDCNN), comprised of K sub-networks (expert DCNNs), that can be trained to perform classification. This is in contrast to [13], where each expert is used just for feature extraction. Our system has similarities to the gated network approach proposed by Jacobs et al. [16], which utilises a separately trained network to select the most appropriate expert network.

The proposed MixDCNN system allows us to jointly train the network, which has two advantages: **(i)** it obviates the need for a separate gating network, and **(ii)** samples can be re-assigned to the most appropriate expert network

during the training process. Empirical evaluations show that this approach outperforms related approaches such as subset feature learning [13], a gated DCNN approach similar to [16], and an ensemble of classifiers.

The paper is continued as follows. In Section 2 we briefly review recent advances in fine-grained classification and overview approaches to learn multiple expert classifiers, particularly within the field of neural networks. In Section 3 we present our proposed MixDCNN approach in detail. Section 4 is devoted to a comparative evaluation against several recent methods on the task of fine-grained classification. Conclusions and possible future avenues of research are given in Section 5.

2. Prior Work

Prior work for fine-grained image classification has concentrated on performing parts detection [2, 5, 20, 24] in order to decrease the intra-class variation. The part-based one-vs-one feature system [2] is an example of this, where parts-based features are progressively selected to improve classification. An alternative is the deformable parts-model which obtains a combined feature from a set of pre-defined parts [24]. Chai et al. [6] proposed a symbiotic model where part localisation is helped by segmentation and, conversely, the segmentation is helped by parts detection. Zhang et al. [24] extract pose-normalised features based on weak semantic annotations to learn cross-component correspondences of various parts.

Recent work has shown the effectiveness of DCNNs for fine-grained image classification, but again, predominantly to perform parts detection. Region proposal methods combined with a DCNN were shown to more accurately localise object parts [23]. Lin et al. [19] showed that a DCNN can be trained to perform both parts localisation and visibility prediction, achieving state-of-the-art results on the CUB dataset [22]. Although the above parts-based approaches are fully automatic at test time, they require a large number of images to be manually annotated in order to train the model.

To remove the need for time-consuming manual annotations, recent work has explored ways to perform fine-grained classification without using part annotations. Zhang et al. [23] and Ge et al. [12] showed that, even without part annotations, DCNNs can provide impressive performance for fine-grained classification tasks. Of particular interest is the approach of Ge et al. [10] which showed that the data can be partitioned into K non-overlapping sets and an expert feature extraction algorithm, utilising DCNNs, can be trained for each of the K sets.

Learning algorithms which construct a set of K classifiers and make decisions by taking a weighted or average of their predictions are often referred to as ensemble methods. A simple ensemble approach called *bagging* has been used to improve the overall performance of a system [4]. Bagging manipulates the training examples to generate multiple hypotheses. In this case, a set of K classifiers is learned using a randomly selected subset of the training data. We use this bagging approach on a set of DCNNs for a baseline method and refer to it as an Ensemble approach (Section 4).

Ensemble approaches, or learning K expert classifiers, has been explored by several researchers within the context of neural networks. In 1991 Jacobs et al. [16] described a gated network structure to learn K expert neural networks and applied it to multi-speaker vowel recognition. The underlying idea is to only allocate a small region of the input space to a particular expert system. This was achieved by having K expert systems (neural networks) which were allocated samples selected by a separate gating network. In [16], the gating network determines the probability that a sample is associated to one of the K expert systems.

More recently, Ge et al. [13] outlined a subset feature learning (Subset FL) approach using K expert DCNNs. The data is partitioned into K non-overlapping sets and for each set an expert DCNN is learned to extract set-specific features. A gating network is then used to extract only the most relevant features from these K DCNNs. Classification is then performed by training an SVM on these features, yielding impressive performance for fine-grained bird and plant classification [11]. An issue with this work is the reliance of an independent gating network $\mathcal{G}$ and the fact that feature extraction and classification are treated as independent steps.

3. Proposed Approach

We propose a novel mixture of DCNNs (MixDCNN) to improve fine-grained image classification by partitioning the data into K non-overlapping sets and learning an expert classifier for each set. This approach has similarities to the gated neural network proposed by Jacobs et al. [16], which has never been applied to DCNNs nor to the fine-grained classification problem. As such, we also outline a gated DCNN (GatedDCNN). An overview of these two approaches is given in Figure 2.

The main idea behind the MixDCNN and GatedDCNN approaches is to learn K expert networks, $[\mathcal{S}_1, \ldots, \mathcal{S}_K]$, which make decisions about a subset of the data. This simplifies the space that is being modelled by each component. Key to both approaches is being able to assign a sample to the appropriate network.

A GatedDCNN assigns samples by learning a separate gating neural network which produces the probability, α_k, that the sample belongs to the k-th network. Learning this gating neural network requires ground truth labels about which sample should be assigned to a particular network, which for our work is an open question. In contrast, a MixDCNN assigns samples based on the confidence of the

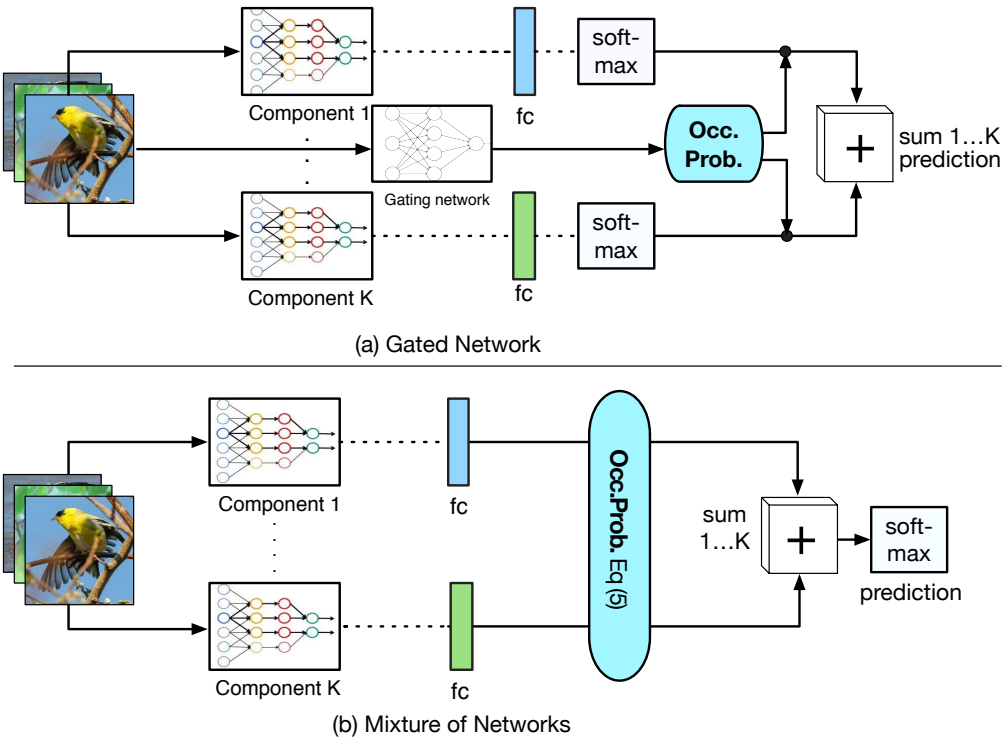


Figure 2. GatedDCNN structure (top) and MixDCNN structure (bottom). The term *Occ. Prob.* refers to occupation probability (responsibility) $\boldsymbol{\alpha}$. In GatedDCNN, the gating network uses the image, the same input as each component (subset networks), to estimate $\boldsymbol{\alpha}$. In contrast, MixDCNN estimates $\boldsymbol{\alpha}$ without the need for an external network.

prediction from each network, which leads us to consider α_k to be the occupation probability of the sample for the k-th network.

Before we describe these two approaches in more detail we define some notation. The output of a DCNN, trained for classification, is an N-dimensional vector $\boldsymbol{z}$ of class predictions, where N indicates the number of classes. These predictions then are normalised by a softmax [18, 21] to give the probability that the sample belongs to the n-th class:

$$c_n = \frac{\exp\{z_n\}}{\sum_{j=1}^{N} \exp\{z_j\}} \tag{1}$$

In the approaches described below, we are most interested in the vector of predictions $\boldsymbol{z}$ prior to applying the softmax.

3.1. GatedDCNN

Inspired by [15, 16], we define a GatedDCNN that consists of K components (DCNNs) and an additional gating network. The overall structure of this network is shown in Fig. 2a. In this arrangement, the k-th DCNN $\mathcal{S}_k$ is given greater responsibility for learning to discriminate subtle differences of the k-th subset of images, while the gating network $\mathcal{G}$ is responsible for associating the image $\boldsymbol{I}$ with the most appropriate component. The gating network $\mathcal{G}$ is a fine-tuned DCNN that is learned using the cross-entropy loss to produce a K-dimensional vector of probabilities $\boldsymbol{\alpha}$. The k-th value denotes the probability that the input image $\boldsymbol{I}$ is associated with the k-th component. We refer to this as an occupation probability.

A fundamental difficulty with training the GatedDCNN is how to provide the T training labels $\boldsymbol{y}$. This label vector is a K-dimensional label vector which indicates which of the K subsets the sample belongs to. To deal with this issue we consider two ways of estimating these labels. The first approach is to initialise the labels $\boldsymbol{y}$ using the partitioning of the training images into K subsets. The gated network $\mathcal{G}$ is then trained using these labels and the K DCNNs (components) are then trained independently so that $\mathcal{S}_k$ is trained exclusively with data from the k-th subset. The second approach is to use the above gated network (and K components) as an initialisation and to iteratively retrain by:

1. Fixing $\mathcal{G}$, and then updating $[\mathcal{S}_1, \ldots, \mathcal{S}_K]$ using the assignments from $\mathcal{G}$.

2. Fixing the K components $[\mathcal{S}_1, \ldots, \mathcal{S}_K]$ and using these to estimate new labels $\boldsymbol{y}$. The network $\mathcal{G}$ is then updated using these new labels.

The labels $\boldsymbol{y}$ estimated in step 2 are obtained by taking

the network which is most confident about its decision. Formally, y_t for the t-th training sample is given by:

$$y_t = \arg\max_{k=1\ldots K} C_{k,t} \tag{2}$$

where $C_{k,t}$ is the best classification result for $\mathcal{S}_k$ using the t-th sample:

$$C_{k,t} = \max_{n=1\ldots N} z_{k,n,t} \tag{3}$$

Classification with the GatedDCNN is performed using a weighted summation of the classification results from the K components:

$$c_n = \sum\nolimits_{k=1}^{K} c_{k,n}\alpha_k \tag{4}$$

where $c_{k,n}$ is the probability of the sample belonging to the n-th class for the k-th component, and α_k is the probability that the sample is assigned to the k-th component $\mathcal{S}_k$.

An issue with the GatedDCNN system is that a separate gating network has to be trained to assign a sample to a particular component $\mathcal{S}_k$. This provides the further complication of having to estimate the labels $\boldsymbol{y}$ in order to train the gating network $\mathcal{G}$. In this paper the first GatedDCNN training approaches provides marginally better performance. In the experiment section, we will report results based on the first approach.

3.2. Mixture of DCNNs (MixDCNN)

We propose a mixture of DCNNs approach where the occupation probabilities $\boldsymbol{\alpha}$ are based on the classification confidence from each component. An advantage of this structure is that we can jointly train the K DCNNs (components) without having to estimate a separate label vector $\boldsymbol{y}$ or train a separate gating network $\mathcal{G}$.

For MixDCNN, the occupation probability for the k-th component is:

$$\alpha_k = \frac{\exp\{C_k\}}{\sum_{c=1}^{K}\exp\{C_c\}} \tag{5}$$

where C_k is given by Eq. (3). This occupation probability gives higher weight to components that are confident about their prediction. The overall structure of this network is shown in Fig. 2b.

Classification is performed by multiplying the output of the final layer from each component by the occupation probability and then summing over the K components:

$$z_n = \sum\nolimits_{k=1}^{K} z_{k,n}\alpha_k \tag{6}$$

This mixes the network outputs together and the probability for each class is then produced by applying the softmax function in Eq. (1). As a consequence our MixDCNN is optimised using the cross-entropy loss[1].

[1]Optimised in a mini-batch Stochastic Gradient Descent framework.

3.3. Differences Between MixDCNN and Ensembles

The aim of the MixDCNN approach is that each component takes greater responsibility for a portion of the data allowing each component to concentrate on samples (or classes) that are more difficult to differentiate. This will allow the MixDCNN to learn subtle differences for similar classes. This is in contrast to an Ensemble approach which randomly excludes a portion of the training data for each DCNN. Therefore, the key difference between the proposed MixDCNN approach and an ensemble of DCNNs (Ensemble) is the use of the occupation probability. For training, this means the MixDCNN approach does not randomly select the data. Instead, each sample is weighted proportional to its relevance to each DCNN $\mathcal{S}_{1,\ldots,K}$. For testing, the MixDCNN approach is able to adaptively calculate the occupation probability for each sample, whereas an Ensemble approach will use pre-defined weights or, more commonly, equal weights.

4. Experiments

4.1. Datasets

We present results on three fine-grained image classification datasets using two network structures. The three datasets are the Caltech-UCSD-2011 (CUB200-2011) [22], Birdsnap [3], and PlantCLEF 2015 [14]. Example images are shown in Figures 1 and 3.

CUB200-2011 is a fine-grained bird classification task with 11,788 images from 200 bird species in North America. This dataset has become a *de facto* standard for the bird classification task. Each species has approximately 30 images for training and 30 for testing. Birdsnap is a much larger bird dataset consisting of 49,829 images from 500 bird species with 47,386 images used for training and 2,443 images used for testing. PlantCLEF 2015 is a large plant classification dataset that has seven content types. To demonstrate the capabilities of the proposed MixDCNN ap-

Figure 3. Examples from CUB-200-2011 and PlantCLEF Flower.

proach for the task of fine-grained classification, we analyse its effectiveness on one content type, Flower. This portion of the dataset consists of 28,705 images from 967 species. We split this data into training and test sets. The training set consists of 25,025 images from 967 species, while the test set has 3,200 images from 801 species.

Both CUB200-2011 and Birdsnap have bounding box annotations around the object of interest. We use this information to extract just the object of interest from the image. PlantCLEF 2015 does not come with bounding box information making it a more challenging dataset.

Prior work [13, 23] has shown the importance of transfer learning for the fine-grained image classification problem. Results have shown that training a DCNN from scratch for either the fine-grained CUB200-2011 or Birdsnap dataset leads to overfitting on the training samples. As such, for all the of our experiments we use pre-trained networks from ImageNet [7] to provide a good initialisation for each DCNN and then perform transfer learning. We consider this to be our baseline and refer to it as **DCNN-tl**. All of our networks are trained using Caffe [17] and partitioning was performed using the Bob toolkit [1].

4.2. Comparative Evaluation

We compare the proposed MixDCNN approach against four other related methods: **(1)** the baseline DCNN-tl, **(2)** an ensemble of K DCNNs, **(3)** an implementation of GatedDCNN, and **(4)** Subset FL [13]. Two network structures considered are the well known AlexNet [18] and the Large Scale Visual Recognition Challenge (ILSVRC) 2014 winner GoogLeNet [21]. AlexNet is a deep network consisting of 8 layers, while ILSVRC has 22 layers[2]. We follow the same procedure as Ge et. al [13] to cluster the data. For the AlexNet structure we use the output of the first fully connected layer as features for clustering. For GoogLeNet we use the output of the last layer, prior to classification, as features. In both cases linear discriminant analysis (LDA) is applied to reduce the dimensionality to $D = 128$. In our initial experiments, we varied D and results showed no impact of that.

The results in Table 1 show that the proposed MixDCNN approach provides consistent improvement regardless of network structure or dataset. MixDCNN provides the best performance for all of the network and dataset combinations, with the exception of the MixDCNN model using the GoogLeNet structure on CUB. It provides an average relative performance improvement of 12.7% over the baseline DCNN-tl approach, excluding CUB.

For the CUB dataset, using multiple expert networks provides limited performance improvement. This is true for all of the methods examined. We attribute this to the fact that CUB200-2011 is a small dataset consisting of just 5,994 training images. This is an order of magnitude fewer samples than other datasets such as Birdsnap. Furthermore, applying transfer learning to GoogLeNet already provides exceptional performance and so minimises the improvement introduced by the MixDCNN framework, or any multi-expert approach.

The proposed MixDCNN method achieves state-of-the-art results on the challenging Birdsnap and PlantCLEF-Flower datasets. For Birdsnap the previous state-of-the-art performance was 48.8% [3]. Applying transfer learning to GoogLeNet already outperforms this prior art with an accuracy of 67.4%. MixDCNN provides a further relative performance improvement of 9.9%. For the PlantCLEF-Flower dataset the baseline performance of DCNN-tl (using GoogLeNet) is 48.7%. MixDCNN provides state-of-the-art performance with a relative performance improvement of 7.0%.

The MixDCNN approach consistently outperforms the Ensemble, GatedDCNN and Subset FL approaches. Interestingly, it provides a considerable improvement over the closely related GatedDCNN approach, with an average relative performance improvement of 9.1%. We attribute this to the ability of the MixDCNN approach to adaptively reassign samples to the most appropriate expert network, in spite of the original partitioning.

In our experiments, component sizes greater than $K = 6$ were not considered as we could not store these in memory on a single GPU[3]. This highlights one of the limitations with this technique as it currently requires all of the networks to be stored on a single GPU; future work should consider how to extend the architecture across multiple GPUs.

5. Conclusion

We have proposed a novel mixture of deep neural networks, termed MixDCNN, which achieves state-of-the-art performance for fine-grained classification. It provides an average relative performance improvement of 12.7% and has been shown to consistently outperform several related methods: subset feature learning, GatedDCNN, and an ensemble of classifiers.

The key advantage of our proposed approach is the use of an occupation probability that weights each sample proportional to its relevance to each DCNN $\mathcal{S}_{1,\dots,K}$. This approach obviates the need for a separate gating function and highlights the importance of being able to adaptively weight samples based on their relevance to a component (DCNN).

Future work will explore alternative methods for initialising the clustering and its impact upon performance. For instance, the impact of grouping images together in terms

[2] To prevent GoogLeNet from over-fitting we use a higher dropout rate equal to 0.5 for the final loss layer, as opposed to the original setting of 0.4.

[3] The GPU used in all our experiments was an Nvidia K40 Tesla with 12 Gb of memory.

Table 1. Comparison of the proposed MixDCNN approach against DCNN-tl, Ensemble, GatedDCNN and Subset FL on three datasets: CUB, BirdSnap and PlantCLEF-Flower. Two network structures are used: AlexNet and GoogLeNet.

		DCNN-tl	**Ensemble**	**GatedDCNN**	**Subset FL**	**MixDCNN**
AlexNet	CUB	68.3%	71.2%	69.2%	72.0%	**73.4**%
	BirdSnap	55.7%	57.2%	57.4%	59.3%	**63.2**%
	PlantCLEF-Flower	29.1%	30.2%	30.2%	31.1%	**35.0**%
GoogLeNet	CUB	80.0%	80.9%	81.0%	**81.2**%	81.1%
	BirdSnap	67.4%	71.4%	70.1%	72.8%	**74.1**%
	PlantCLEF-Flower	48.7%	50.2%	49.7%	51.7%	**52.1**%

of their pose rather than similar visual appearance. Furthermore, we will examine the role of the occupation probability in two ways: **(i)** whether the responsibility for a sample is shared between components, and **(ii)** deeper analysis of how this occupation probability changes during the training process. Additionally, we intend on exploring different methods for computing the occupational probability via alternative aggregation techniques.

Acknowledgements. 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.

References

[1] 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 *ACM International Conference on Multimedia*, pages 1449–1452, 2012.

[2] T. Berg and P. N. Belhumeur. POOF: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, 2013.

[3] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *CVPR*, pages 2019–2026, 2014.

[4] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[5] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013.

[6] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013.

[7] 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.

[8] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011.

[9] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, 2013.

[10] Z. Ge, C. McCool, C. Sanderson, A. Bewley, Z. Chen, and P. Corke. Fine-grained bird species recognition via hierarchical subset learning. In *IEEE International Conference on Image Processing (ICIP)*, pages 561–565, 2015.

[11] Z. Ge, C. McCool, C. Sanderson, and P. Corke. Content specific feature learning for fine-grained plant classification. In *Working Notes of the CLEF 2015 Conference*, 2015.

[12] Z. Ge, C. McCool, C. Sanderson, and P. Corke. Modelling local deep convolutional neural network features to improve fine-grained image classification. In *IEEE Int. Conference on Image Processing (ICIP)*, pages 4112–4116, 2015.

[13] Z. Ge, C. McCool, C. Sanderson, and P. Corke. Subset feature learning for fine-grained classification. In *DeepVision Workshop, Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 46–52, 2015.

[14] H. Goëau, A. Joly, P. Bonnet, S. Selmi, J.-F. Molino, D. Barthélémy, and N. Boujemaa. Lifeclef plant identification task 2014. In *Working Notes for CLEF 2014 Conference*, pages 598–615. CEUR-WS, 2014.

[15] J. B. Hampshire II and A. Waibel. The meta-pi network: Building distributed knowledge representations for robust multisource pattern recognition. *PAMI*, 14(7):751–769, 1992.

[16] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, pages 675–678, 2014.

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

[19] D. Lin, X. Shen, C. Lu, and J. Jia. Deep LAC: Deep localization, alignment and classification for fine-grained recognition. In *CVPR*, pages 1666–1674, 2015.

[20] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur. Dog breed classification using part localization. In *ECCV*. 2012.

[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CVPR*, 2014.

[22] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. In *Computation & Neural Systems Technical Report, California Institute of Technology*, number CNS-TR-2011-001, 2011.

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

[24] N. Zhang, R. Farrell, F. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*, 2013.

Chapter 6

Exploiting Temporal Information for Fine-Grained Object Classification

---

Prior work and the previous three chapters treat the fine-grained classification task as a still-image classification problem and ignores the temporal information available from videos of different fine-grained classes [Anantharajah et al., 2014, Belhumeur et al., 2008, Kumar et al., 2012, Liu et al., 2012, Parkhi et al., 2012].

In this chapter, we introduce the problem of video-based fine-grained object classification, and explore several methods to exploit the temporal information on a new bird video dataset we created. We first present a systematic study on several DCNN-based methods that attempt to exploit temporal information such as 3D ConvNets [Tran et al., 2015], two-stream DCNNs [Simonyan and Zisserman, 2014] and bilinear DCNNs [Lin et al., 2015]. We then propose a novel adaptation of the bilinear DCNN approach for video bird classification and highlight the potential benefits that fine-grained object classification can gain by modelling temporal information. In our proposed method the bilinear DCNN is adapted to extract local co-occurrences by combining information from the convolutional layers of spatial and temporal DCNNs.

We evaluate our method on the new and challenging video dataset of birds which contains several challenges, such as clutter, large variations in scale, camera movement, and considerable pose variations. Experiments show that by using the proposed approach, the performance is improved from 23.1% (using single images) to 41.1%. The best results we obtained surpass all the previous state-of-the-art video classification methods including two-stream DCNN with 38.9%

accuracy and C3D with 38.6%. By incorporating the latest object detection framework [Ren et al., 2015], we can further boost the performance to 53.6%.

The content of this chapter has been submitted to the European Conference on Computer Vision (ECCV) 2016 as "Exploiting Temporal Information for Fine-Grained Object Classification".

Exploiting Temporal Information for Fine-Grained Object Classification

ZongYuan Ge [†‡] ,Christopher McCool [‡†], Conrad Sanderson [*◇] ,Peng Wang [⋆],Lingqiao Liu [star],Chunhua Shen [⋆†],Ian Reid [†⋆], Peter Corke [†‡]

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

Abstract. Fine-grained classification is a relatively new field that has concentrated on using information from a single image, while ignoring the enormous potential of using video data to improve classification. In this work we present the novel task of video-based fine-grained object classification, propose a corresponding new video dataset, and perform a systematic study of several recent deep convolutional neural network (DCNN) based approaches, which we specifically adapt to the task. We evaluate three-dimensional DCNNs, two-stream DCNNs, and bilinear DCNNs. Two forms of the two-stream approach are used, where spatial and temporal data from two independent DCNNs are fused either via early fusion (combination of the fully-connected layers) and late fusion (concatenation of the softmax outputs of the DCNNs). For bilinear DCNNs, information from the convolutional layers of the spatial and temporal DCNNs is combined via local co-occurrences. We then fuse the bilinear DCNN and early fusion of the two-stream to combine the spatial and temporal information at the local and global level (Spatio-Temporal Co-occurrence). Using the new and challenging video dataset of birds, classification performance is improved from 23.1% (using single images) to 41.1% when using the Spatio-Temporal Co-occurrence system. Incorporating automatically detected bounding box location further improves the classification accuracy to 53.6%.

Keywords: fine-grained recognition, video classification, deep learning, deep convolutional neural networks, spatio-temporal information.

1 Introduction

Fine-grained object classification consists of discriminating between classes in a sub-category of objects, for instance the particular species of bird or dog [2, 4, 7, 8, 26]. This is a very challenging problem due to large intra-class variations caused by pose and appearance changes, as well as small inter-class variation due to subtle differences in the overall appearance between classes [1].

Prior work in fine-grained classification has concentrated on learning image-based features to cope with pose variations. Initially such approaches used traditional image-based features such as colour and histograms of gradients [2] while modelling the pose using a range of methods including deformable parts-based approaches [4, 18, 27]. More recently, deep convolutional neural networks (DCNNs) have been used to learn robust features [5], cope with large variations by using a hierarchical model [9], and automatically localise regions of importance [10]. Despite the advances provided by these approaches, prior work treats the fine-grained classification task as a still-image classification problem and ignores complementary temporal information present in videos.

Recent work on neural network based approaches has provided notable results in video-based recognition [13, 21, 23, 25]. Karpathy et al. [13] demonstrated the surprising result that classifying a single frame from a video using a DCNN was sufficient to perform accurate video classification, for broad categories such as activity and sport recognition. Within the action recognition area, Simonyan and Zisserman [21] incorporate optical flow and RGB colour information into two stream networks. Tran et al. [23] apply deep 3D convolutional networks (3D ConvNets) to implicitly learn motion features from raw frames and then aggregate predictions at the video level. Ng et al. [25] employ Long Short-Term Memory cells which are connected to the output of the underlying CNN to achieve notable results on the UCF-101 [22] and Sports 1 million datasets [13]. To date, the above neural network based approaches have not been explored for the task of video-based fine-grained object classification.

Contributions. In this paper, we introduce the problem of video-based fine-grained object classification, propose a corresponding new dataset, and explore several methods to exploit the temporal information. A systematic study is performed comparing several DCNN based approaches which we have specifically adapted to the task, highlighting the potential benefits that fine-grained object classification can gain by modelling temporal information. We evaluate 3D ConvNets [23], two-stream DCNNs [21], and bilinear DCNNs [17]. Two forms of the two-stream approach are used: (i) the originally proposed late-fusion form which concatenates the softmax outputs of two independent spatial and temporal DCNNs, and (ii) our modified form, which performs early-fusion via combination of the fully-connected layers. In contrast to the two forms of the two-stream approach, we adapt the bilinear DCNN to extract local co-occurrences by combining information from the convolutional layers of spatial and temporal DCNNs. The adapted bilinear DCNN is then fused with the two-stream approach (early fusion) to combine spatial and temporal information at the local and global level.

The study is performed on a new and challenging video dataset of birds, consisting of 1,416 video clips of 100 species birds taken by expert bird watchers. The dataset contains several compounded challenges, such as clutter, large variations in scale, camera movement and considerable pose variations. Experiments show that classification performance is improved from 23.1% (using single images) to 41.1% when using the spatio-temporal bilinear DCNN approach, which outperforms 3D ConvNets as well as both forms of the two-stream approach.

We highlight the importance of performing early fusion, either at the input layer (3D ConvNets) or feature layer (adapted bilinear DCNN), as this consistently outperforms late fusion (ie. the original two-stream approach). Incorporating automatically detected bounding box location further improves the classification accuracy of the spatio-temporal bilinear DCNN approach to 53.6%.

We continue the paper as follows. Section 2 describes the studied methods and our adaptations, while Section 3 describes the new video-based bird dataset. Section 4 is devoted to comparative evaluations. The main findings are summarised in Section 5.

2 Combining Spatial and Temporal Information

In this section we first describe two baseline networks that make use of either image or temporal information. We then outline the deep 3-dimensional convolutional network [23], extend the two-stream approach [21] and adapt the bilinear DCNN approach [17] to encode local spatial and temporal co-occurrences.

2.1 Underlying Spatial and Temporal Networks

Our baseline systems are DCNNs that use as input either optical flow (temporal) or image-based features. The temporal network $\mathcal{T}$ uses as input the horizontal flow $\mathbf{O}_x$, vertical flow $\mathbf{O}_y$, and magnitude of the optical flow $\mathbf{O}_{mag}$ combined to form a single optical feature map $\mathbf{O} \in \mathrm{R}^{h \times w \times 3}$, where $h \times w$ is the size of the feature map (image). The spatial network $\mathcal{S}$ uses RGB frames (images) as input. Both $\mathcal{S}$ and $\mathcal{T}$ use the DCNN architecture of Krizhevsky et al. [15] which consists of 5 convolutional layers, $\mathbf{S}^{c1}, \mathbf{S}^{c2}, \ldots, \mathbf{S}^{c5}$, followed by 2 fully connected layers, $\mathbf{S}^{fc6}$ and $\mathbf{S}^{fc7}$, prior to the softmax classification layer, $\mathbf{S}^{o}$. The networks are trained by considering each input frame from a video (either image or optical flow) to be a separate instance, and are fine-tuned to the specific task (and modality) by using a pre-trained network. Fine-tuning [24] is necessary as we have insufficient classes and observations to train the networks from scratch (preliminary experiments indicated that training the networks from scratch resulted in considerably lower performance).

When performing classification, each image (or frame of optical flow) is initially treated as an independent observation. For a video of N_f frames this leads to N_f classification decisions. To combine the decisions, the max vote of these decisions is taken.

2.2 Deep 3D Convolutional Network

The deep 3-dimensional convolutional network (3D ConvNet) approach [23], originally proposed for action recognition, utilises 3-dimensional convolutional kernels to model L frames of information simultaneously. In contrast to optical flow features where temporal information is explicitly modelled, the approach implicitly models the information within the deep neural network structure.

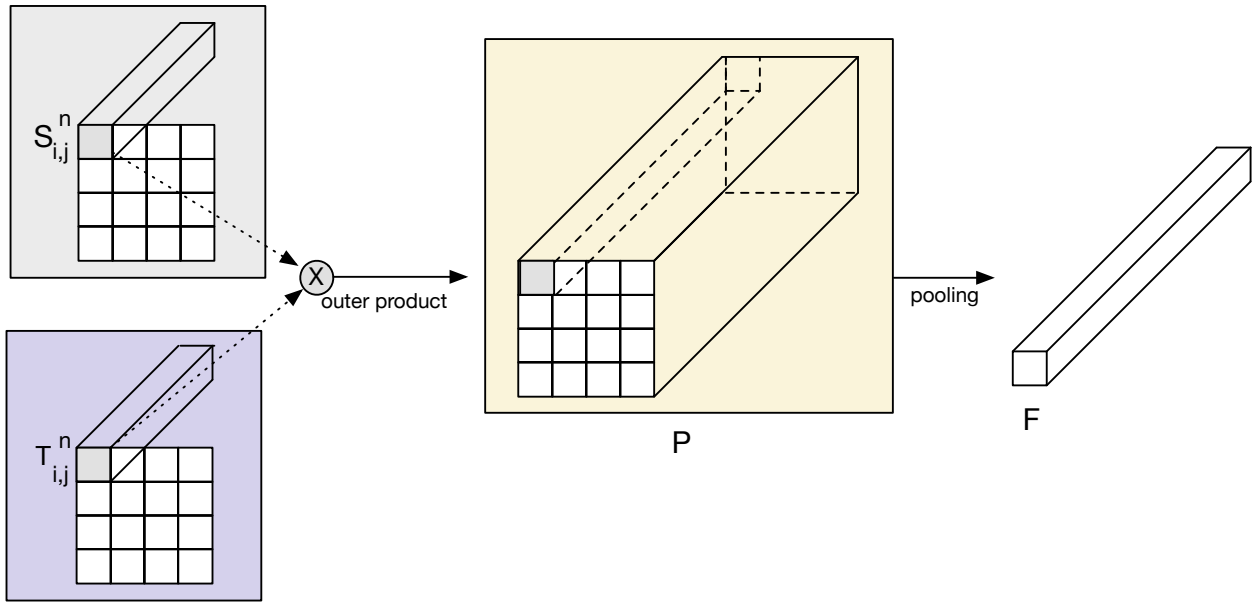


Fig. 1. Conceptual illustration of the spatio-temporal co-occurrence based approach.

This approach obtains state-of-the-art performance on various action recognition datasets such as UCF-101 [22] and ASLAN [14]. The network is fine-tuned for our classification task by taking a sliding window of $L = 15$ frames and moving the sliding window one frame at a time; each sliding window is considered to be a separate instance. This results in $N_f - 14$ classification decisions which are combined using the max vote.

2.3 Spatio-Temporal Two-Stream Network: Early and Late Fusion

The two-stream network proposed for action recognition by Simonyan and Zisserman [21] uses the two independent spatial and temporal networks $\mathcal{S}$ and $\mathcal{T}$. The softmax output of these two networks is then concatenated and used as a feature vector that is classified by a multi-class support vector machine (SVM). We refer to this network as *Two-Stream (late fusion)*; it is conceptually illustrated in Fig. 2(a).

A potential downside of this approach is that fusion of spatial and temporal information is done at the very end. This limits the amount of complementary information captured as scores (or decisions) from the softmax classification layer are combined. To address this issue, we propose to combine the two streams of information much earlier (early fusion) by combining the $fc6$ outputs, $\mathbf{S}^{fc6}$ and $\mathbf{T}^{fc6}$; $fc6$ is the first fully connected layer and is often used to extract a single feature from DCNNs [5]. We refer to this modified network as *Two-Stream (early fusion)*. See Fig. 2(b).

2.4 Joint Spatial and Temporal Features via Co-occurrences

We adapt the recently proposed bilinear DCNN approach by Lin et al. [17] via combining the convolutional layers of the baseline spatial and temporal networks

by calculating co-occurrences. The rationale behind is that different species of birds may have different appearance and motion patterns and their combination. Specifically, let the feature maps of the n-th layer of the spatial and temporal networks be $\mathbf{S}^n \in \mathbb{R}^{h \times w \times d_n}$ and $\mathbf{T}^n \in \mathbb{R}^{h \times w \times d_n}$, where d_n is the number of dimensions for the feature map (number of kernels). The two feature maps are combined by calculating an outer product:

$$\mathbf{P}_{i,j} = \text{vec}\left(\mathbf{S}^n_{i,j} \mathbf{T}^n_{i,j}{}^{\intercal}\right) \tag{1}$$

where $\mathbf{S}^n_{i,j} \in \mathbb{R}^{d_n}$ and $\mathbf{T}^n_{i,j} \in \mathbb{R}^{d_n}$ are the local feature vectors of the spatial and temporal streams at location (i, j), $\text{vec}(\cdot)$ is the vectorization operation, and $\mathbf{P} \in \mathbb{R}^{h \times w \times d_n^2}$, with $\mathbf{P}_{i,j} \in \mathbb{R}^{d_n^2}$ being the co-occurrence feature at location (i, j). As such, the outer product operation captures the co-occurrence of the visual and motion patterns at each spatial location. Max pooling is applied to all the local encoding vectors $\mathbf{P}_{i,j}$ to create the final feature representation $\mathbf{F} \in \mathbb{R}^{d_n^2}$. Finally, L_2 normalisation is applied to the encoding vector [17]. The overall process is conceptually illustrated in Fig. 1.

The spatio-temporal bilinear DCNN feature is combined with the $fc6$ spatial and temporal features used for *Two-Stream (early fusion)*. This allows us to combine the spatial and temporal information at both the local and global level. The resultant features are fed to an SVM classifier. See Fig. 2(c) for a conceptual illustration. We refer this system as *Spatio-Temporal Co-occurrence*.

3 Dataset: Videos of Birds 100 (VB100)

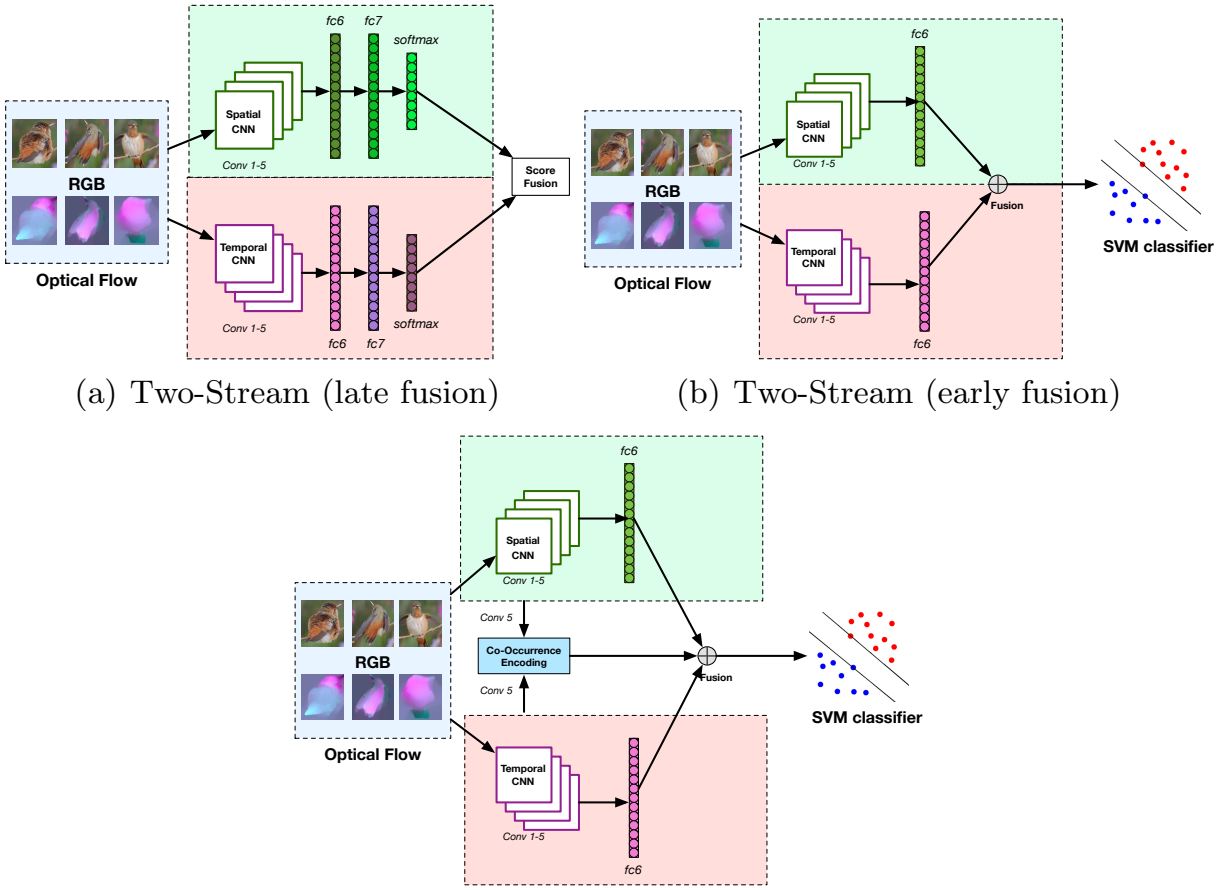
To investigate video-based fine-grained object classification we propose a new and challenging dataset consisting of 1,416 video clips of 100 bird species taken by expert bird watchers. The birds were often recorded at a distance, introducing several challenges such as large variations in scale, camera movement and considerable pose variations; a link to the dataset will be provided upon publication. See Fig. 3 for examples.

For each class (species of bird), the following data is provided: video clips with activity annotations, sound clips, automated bounding box detection, as well as taxonomy and distribution location. See Fig. 4 for an example.

The median length of a video is 32 seconds with the the shortest being 8 seconds and longest being 118 seconds. Each class has on average 15 clips, with the lowest being 6 and the highest being 23. Most videos (977) were captured at 30 frames per second (fps), while 422 were captured at 25 fps, 10 at 60fps, and 1 at 100 fps. Often the camera will need to move in order to track the bird, keeping it in view. This form of camera movement is present in 798 videos, with the remaining 618 videos obtained using static cameras.

4 Experiments

Two sets of experiments are presented in this section. In the first set (Section 4.1), we evaluate the performance without taking into account whether each video clip



(a) Two-Stream (late fusion) (b) Two-Stream (early fusion)

(c) Spatio-Temporal Co-Occurrence

Fig. 2. Overview of the Two-Stream and Spatio-Temporal Co-Occurrence approaches for fine-grained video classification. In (a) the Two-Stream approach uses *late fusion*, where features are combined from the softmax layer. In (b) the Two-Stream approach uses *early fusion*, where features are combined from the $fc6$ layer. The Spatio-Temporal Co-Occurrence approach (c) combines the co-occurrence (bilinear DCNN) features with the features from $fc6$.

was recorded by a static or moving camera. In the second set (Section 4.2), we study the effect of camera movement on performance. In all cases, to obtain a per video classification decision we use the max voting from the classified frames. For the Spatio-Temporal Co-occurrence approach, initial experiments found that using the last convolutional layer $n = c5$ provided the best performance; this leads to $d = 65,536$ for the spatio-temporal bilinear features. The input frame size for all networks is 224×224. Training and testing is performed using Caffe [12].

The dataset is divided into 730 training videos (train set) and 686 testing videos (test set). Results are presented in terms of mean classification accuracy. Classification accuracy is calculated on a per video basis and per class basis, with accuracy $= N_p^c/N^c$, where N_p^c is the number of correctly classified videos for the c-th class and N^c is the number of videos for the c-th class. The mean classification accuracy is then calculated across all of the classes.

4.1 Comparative Evaluation

We first investigate the performance of two independent networks for spatial and temporal information: Spatial-DCNN and Temporal-DCNN. We then compare the performance of 3D ConvNets [23] fine-tuned for our bird classification task (referred to as 3D ConvNets-FT), the two-stream approach [21] (which combines the Spatial-DCNN and Temporal-DCNN networks), and the spatio-temporal co-occurrence approach. Finally we evaluate the performance of the co-occurrence approach in conjunction with an off-the-shelf bird detector/locator. For this we use the recent Faster Region CNN [20] approach with default parameters learned for the PASCAL VOC challenge [6]; only bird localisations are used, with all other objects ignored. Examples of localisation are shown in Fig. 5.

Network Setup. The Spatial-DCNN uses the AlexNet structure pre-trained on the ImageNet dataset [15] before being fine-tuned for our bird classification task. It is trained by considering each frame from a video to be a separate instance (image). Two variants of Spatial-DCNN are used: (i) randomly selecting one frame per video clip, and (ii) using 5 frames per second (fps) from each video

Fig. 3. Example frames from video clips in the VB100 dataset. Each row shows four sample frames for a unique class. The first frame in each row (left to right) shows an easy situation, followed by three images showing variations in pose, scale and background.

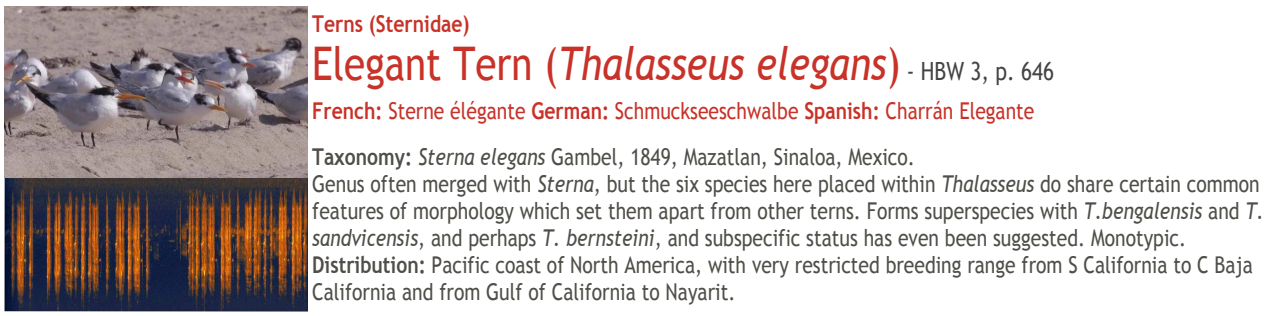


Fig. 4. An example for the class *Elegant Tern* in the new video-based bird dataset. Top-left: a still shot from one of the video clips. Bottom-left: spectrogram created from the corresponding audio file. Right: taxonomy information about the class.

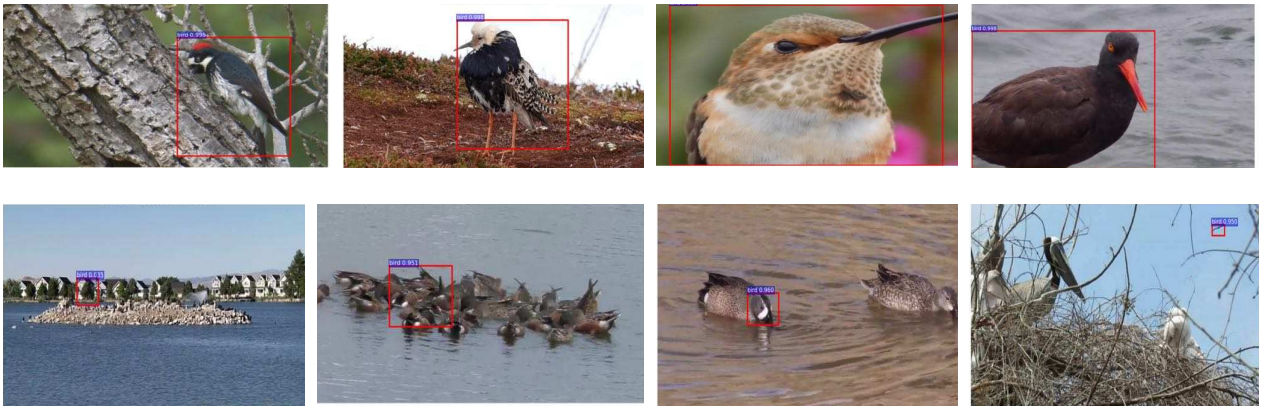


Fig. 5. Examples of bird localisation (red bounding box) using the default settings of Faster R-CNN [20]. Top row: good localisations. Bottom row: bad localisations due to confounding textures, clutter, small objects, and occlusions.

clip[1]. The Temporal-DCNN uses dense optical flow features computed from the Matlab implementation of Brox et al. [3]. For the sake of computational efficiency, we have calculated the optical flow every 5 frames.

It is generally beneficial to perform zero-centering of the network input, as it allows the model to better exploit the rectification non-linearities and for optical flow features provides robustness to camera movement [21]. Therefore, for both Spatial-DCNN and Temporal-DCNN we perform mean normalisation of the input data. For Spatial-DCNN we subtract the mean value for each RGB channel, while for Temporal-DCNN mean flow subtraction is performed for the temporal input.

For the two-stream approach we use two forms (as described in Section 2.3): (i) early fusion, where the first fully connected features (fc6) from the Spatial-DCNN (with 5 fps) and Temporal-DCNN networks are concatenated, and (ii) late fusion, where the softmax output of the two networks is concatenated. For the two-stream and the spatio-temporal co-occurrence approaches, the resultant feature vectors are fed to a multi-class linear SVM for classification.

Quantitative Results. The results presented in Table 1 show that using more frames from each video (ie. more spatial data) leads to a notable increase in accuracy. This supports the use of videos for fine-grained classification. The results also show that spatial data provides considerably more discriminatory information than temporal data. In all cases, combining spatial and temporal information results in higher accuracy than using either type of information alone, confirming that the two streams of data carry some complementary information.

In contrast to the using late fusion in the standard two-stream approach, performing early fusion yields a minor increase in accuracy (37.5% vs 38.9%) and slightly exceeds the accuracy obtained by 3D ConvNets-FT (38.6%). Using the co-occurrence approach leads to the highest fusion accuracy of 41.6%.

[1] The video clips were normalised to 5 fps, as this was computationally more efficient. Preliminary experiments indicated that using 5 fps leads to similar performance as normalising at 25 fps.

Table 1. Fine-grained video classification results on the VB100 video dataset.

Method	**Mean Accuracy**
Spatial-DCNN (random frame)	23.1%
Spatial-DCNN (5 fps)	37.0%
Temporal-DCNN ($\Delta = 5$)	22.9%
Two-Stream (early fusion)	38.9%
Two-Stream (late fusion)	37.5%
3D ConvNets-FT	38.6%
Spatio-Temporal Co-occurrence	41.1%
Spatio-Temporal Co-occurrence + bounding box	53.6%

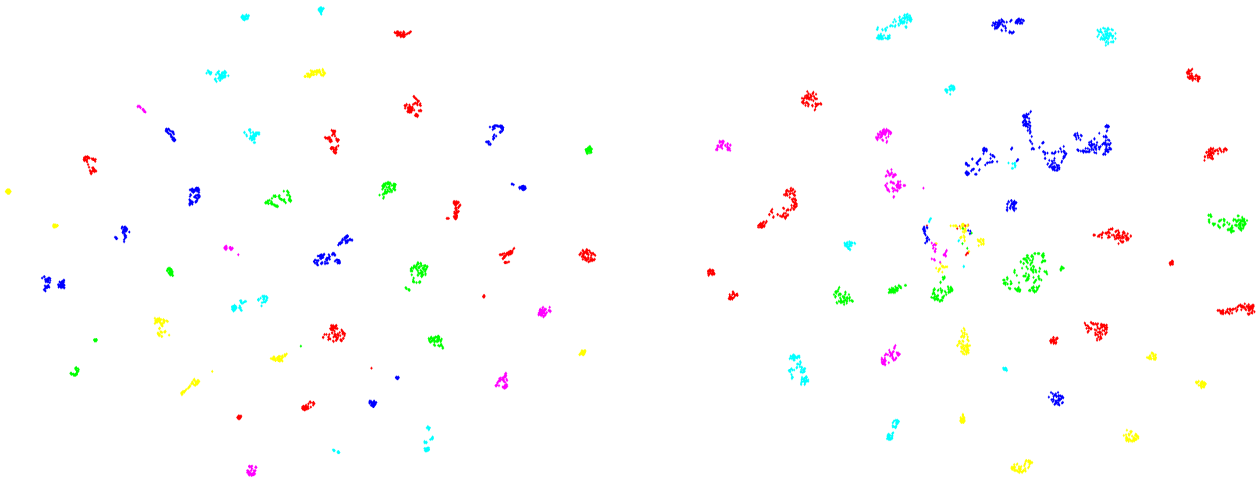


Fig. 6. Qualitative evaluation using t-SNE [19] to visualise the data for 10 classes (indicated by unique colours). Left: using Spatial-DCNN features. Right: using Spatio-Temporal Co-occurrence features. For both approaches several distinct clusters are formed for each class. By using the co-occurrence approach fewer separated clusters are formed, and the separated clusters tend to be closer together.

This highlights the importance of making use of the extra information from the video domain for object classification. Finally, using the Spatio-Temporal Co-occurrence system in conjunction with an automatic bird locator increases the accuracy from 41.6% to 53.6%. This in turn highlights the usefulness of focusing attention on the object of interest and reducing the effect of nuisance variations.

Qualitative Results. To further examine the impact of incorporating temporal information via the co-occurrence approach, we visualise 10 classes with features taken from the Spatial-DCNN and Spatio-Temporal Co-occurrence approaches. To that end we use the t-Distributed Stochastic Neighbour Embedding (t-SNE) data visualisation technique based on dimensionality reduction [19]. In Fig. 6 it can be seen that both sets of features yields several distinct clusters for each class. However, by using the co-occurrence approach fewer separated clusters are formed, and the separated clusters tend to be closer together. This further indicates that benefit can be obtained from exploiting temporal information in addition to spatial information.

Table 2. Effect of static and moving cameras on performance, using a 21 class subset of the VB100 dataset without bounding box detections. Temporal-DCNN (no zero-norm) is trained without applying mean subtraction to the input features.

Network	Camera Type	Mean Accuracy
Spatial-DCNN	Static	57.6%
Spatial-DCNN	Moving	47.8%
Temporal-DCNN (no zero-norm)	Static	28.9%
Temporal-DCNN (no zero-norm)	Moving	23.7%
Temporal-DCNN	Static	32.2%
Temporal-DCNN	Moving	33.3%
Spatio-Temporal Co-occurrence	Static	**61.1%**
Spatio-Temporal Co-occurrence	Moving	**53.7%**

4.2 Effect of Camera Type: Static vs Moving

In this section we explore how camera motion affects performance. Camera motion is a dominant variation within the VB100 dataset as it contains 618 video clips recorded with a static camera and 798 video clips recorded with a moving camera, which follow bird movement (eg., flight). Fig. 7 shows examples from two videos of Elegant Tern recorded by static and moving cameras.

Previous work in action recognition [11, 16], rather than fine-grained object classification, has presented conflicting results regarding the impact of camera motion. Jain et al. [11] showed that features which compensated for camera motion improved performance, while Kuehne et al. [16] showed that the presence of camera motion either had little effect or improved performance.

We manually select 21 classes with videos recorded with and without camera movement, and examine the performance of the Spatial-DCNN, Temporal-DCNN and the Spatio-Temporal Co-occurrence approach. The setup of the networks is the same as per Section 4.1. The results in Table 2 show that Spatial-DCNN is adversely affected by camera movement with the accuracy dropping from 57.6% to 47.8%. This leads to a similar degradation in performance for the Spatio-Temporal Co-occurrence approach: from 61.1% to 53.7%. We attribute the degradation in performance of the spatial networks to the highly challenging conditions, such as the difference between stationary and flying bird presented in Fig. 7. By contrast, performance of Temporal-DCNN is largely unaffected.

We hypothesise that the Temporal-DCNN is robust to camera movement due to the mean subtraction operation that can reduce the impact of global motion between frames. To test the above hypothesis we re-trained the Temporal-DCNN without mean subtraction (no zero-norm). This results in the performance for the Static case reducing from 32.2% to 28.9%, while for the Moving case the performance reduced considerably further: from 33.3% to 23.7%. This supports our hypothesis and highlights the importance of the mean subtraction pre-processing stage for temporal features in the presence of camera motion.

Fig. 7. Top row: examples of video frames recorded by a static camera. Bottom row: examples of video frames recorded by a moving camera, manually tracking the bird.

5 Main Findings

In this work, we introduced the problem of video-based fine-grained object classification along with a challenging new dataset and explored methods to exploit the temporal information. A systematic comparison of state-of-the-art DCNN based approaches adapted to the task was performed which highlighted that incorporating temporal information is useful for improving performance and robustness. We presented a system that encodes local spatial and temporal co-occurrence information, based on the bilinear CNN, that outperforms 3D ConvNets and the Two-Stream approach. This system improves the mean classification accuracy from 23.1% for still image classification to 41.1%. Incorporating bounding box information, automatically estimated using the Faster Region CNN, further improves performance to 53.6%.

In conducting this work we have developed and released the novel video bird dataset VB100 which consists of 1,416 video clips of 100 bird species. This dataset is the first for video-based fine-grained classification and presents challenges such as how best to combine the spatial and temporal information for classification. We have also highlighted the importance of normalising the temporal features, using zero-centering, for fine-grained video classification.

Future work will exploit other modalities by incorporating the audio (sound), taxonomy information, and the textual description of the video clips.

References

1. Berg, T., Belhumeur, P.N.: How do you tell a blackbird from a crow? In: ICCV (2013)
2. Berg, T., Belhumeur, P.N.: POOF: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In: CVPR (2013)
3. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: ECCV (2004)
4. Chai, Y., Lempitsky, V., Zisserman, A.: Symbiotic segmentation and part localization for fine-grained categorization. In: ICCV (2013)
5. Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T.: DeCAF: A deep convolutional activation feature for generic visual recognition. ICML (2014)
6. Everingham, M., Gool, L.V., Williams, C.K., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. International Journal of Computer Vision 88(2), 303–338 (2010)
7. Farrell, R., Oza, O., Zhang, N., Morariu, V.I., Darrell, T., Davis, L.S.: Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In: ICCV (2011)
8. Gavves, E., Fernando, B., Snoek, C.G., Smeulders, A.W., Tuytelaars, T.: Fine-grained categorization by alignments. In: ICCV (2013)
9. Ge, Z., Bewley, A., McCool, C., Corke, P., Upcroft, B., Sanderson, C.: Fine-grained classification via mixture of deep convolutional neural networks. WACV (2016)
10. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: NIPS (2015)
11. Jain, M., Jergou, H., Bouthemy, P.: Better exploiting motion for better action recognition. In: CVPR (2013)
12. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: ACM International Conference on Multimedia. pp. 675–678 (2014)
13. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: CVPR (2014)
14. Kliper-Gross, O., Hassner, T., Wolf, L.: The action similarity labeling challenge. IEEE Trans. Pattern Analysis and Machine Intelligence 34(3), 615–621 (2012)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1097–1105 (2012)
16. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: A large video database for human motion recognition. In: ICCV (2014)
17. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear CNN models for fine-grained visual recognition. In: ICCV (2015)
18. Liu, J., Kanazawa, A., Jacobs, D., Belhumeur, P.: Dog breed classification using part localization. In: ECCV (2012)
19. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. Journal of Machine Learning Research 9, 2579–2605 (2008)
20. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS (2015)
21. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS (2014)
22. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)

Chapter 7

Conclusion

The objective of this thesis has been to investigate a general and robust fine-grained classification system to answer the research questions "how can images and videos of sub-categories in challenging scenarios be robustly classified?" To achieve these objectives we have proposed methods and modelling techniques for fine-grained classification that can be applied to multiple fine-grained tasks such as food, fish, plant and bird classification. This chapter summarise the contributions made in this thesis. We then discuss potential usages and future research directions for this area.

7.1 Summary of Contributions

The four contributions made in this thesis are:

(i) Proposed the novel local inter-session variability modelling (Local ISV) for fine-grained classification. The first major contribution of this thesis is to answer the question of modelling different instances of the same class under various environments. We implemented inter-session variability modelling (ISV) and extended of this to model local regions for fine-grained (fish and food) image classification. The proposed Local ISV approach is able to capture the crucial local identity information and also model and suppress noise locally. From the experiment result of applying Local ISV to fine-grained fish classification, the proposed method provides a relative improvement of 38% over standard ISV on the QUT fish dataset. We then explored how advances in deep convolutional neural networks (DCNNs) could be used to

improve the robustness of the local features used in the ISV framework. We proposed a layer-restricted tuning method to reduce the dimensionality of the DCNN and used this to extract local features. To do this we proposed a two-step retraining method to perform dimensionality reduction on the original pre-trained DCNN model. Combining the local DCNN feature with Local ISV, comparative experiments show that considerable performance improvements can be achieved on the challenging Fish and UEC FOOD-100 datasets.

(ii) Novel hierarchical learning framework. The second contribution is to proposed a novel hierarchical learning framework which first groups visually similar classes into the same subset and then train an expert classifier for each subset. This hierarchical-based approach leverages the weights of both local and global information to generate more discriminative and robust classifiers for fine-grained bird classification. Evaluations on the challenging CUB-200 bird dataset, with parts detection algorithms such as DPM and DPD on top of our proposed approach, shows that classification accuracy can be increased from 64.5% to 72.7%, a relative improvement of 12.7%. However, by using the ground-truth subset labels the best performance can be achieved through this approach is 78.6% which indicates that performing more accurate assignment of a sample to its subset can yield considerable performance improvements. To fill in this gap, we later improved this system by introducing subset feature learning into this framework so that subset-specific features could be learnt and extracted. A combined representation which uses both the subset-specific and globally learned features was then used to achieve state-of-the-art performance of 77.5% for fully automatic fine-grained bird image classification.

(iii) Novel Mixture of DCNNs. The third major contribution of this thesis is to propose a novel mixture of DCNNs. This mixture of DCNNs extends the hierarchical learning framework by probabilistically assigning a sample to a network, during both training and testing. This allows us to jointly train the subset networks in an end-to-end manner. The final decision of each sample is weighted by the occupation probability of each DCNN component. The occupation probability obviates the need for a separate subset selector and highlights the importance of being able to adaptively weight samples based on their relevance to a DCNN component. Empirical evaluations showed that this approach outperforms previous subset feature learning methods with an average relative performance improvement of 12.7% and achieves consistently improved performance over several related methods such as an ensemble of classifiers, Gated-DCNN and subset feature learning.

(iv) Video-based fine-grained classification. The fourth contribution is to demonstrate the potential of exploiting temporal information to improve the robustness of fine-grained classification. We explore a new direction for fine-grained classification, fine-grained video classification. In our proposed method, temporal information is captured by optical flow and these motion features from various videos are used to train a temporal DCNN while raw video frame pixels are fed into a spatial DCNN to learn spatial information. We propose a novel adaptation of bilinear pooling to extract local co-occurrences by combining information from the convolutional layers of spatial and temporal DCNNs. Furthermore, we also introduced a bird video dataset VB100 which consists of 1,416 video clips of 100 bird species. A systematic comparison of state-of-the-art DCNN based approaches is performed on the VB100 bird dataset. These experiments demonstrate the effectiveness of our proposed novel spatial and temporal co-occurrence features which outperform other previous state-of-the art algorithms including 3D ConvNets and the Two-Stream approach.

7.2 Future Work

Although multiple aspects are covered in this thesis, there are still many to be explored in the future work.

1. For fine-grained bird classification, numerous bird pictures are available on the internet. Semi-supervised or unsupervised labelling could automatically annotate large numbers of birds images and provide numerous training images. Such an approach would likely to lead to considerable performance improvements as it would provide orders of magnitude more data to train deep networks which are known to require an enormous amount of labelled data.

2. Furthermore, multiple information source fusion is a interesting direction to explore. We have shown the potential of combining temporal information for fine-grained bird classification and much more work could be conducted in this area. Also, for plants or fish classification tasks, prior knowledge about the geographical location of the image being taken is extremely important to filter out any irrelevant results.

3. For the algorithm perspective, our proposed subset-based learning system was able to group bird classes in terms of visual appearance. It is interesting to explore alternative

features such as pose and background information to initialise those clusters and observe the impact for the final classification results.

4. Close the gap between the fine-grained classification and general classification is a trend in the near future. Many proposed algorithms for fine-grained image classification have been proven to be useful for other classification tasks such as texture and scene classification. Furthermore, recent proposed methods for fine-grained bird classification can be trained without explicit parts annotations, which make the training objective same as the general image classification (with a single image class label).

Literature Cited

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

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). SURF: Speeded up robust features. In *ECCV*.

Belhumeur, P. N., Chen, D., Feiner, S., Jacobs, D. W., Kress, W. J., Ling, H., Lopez, I., Ramamoorthi, R., Sheorey, S., White, S., et al. (2008). Searching the worlds herbaria: A system for visual identification of plant species. In *ECCV*.

Belhumeur, P. N., Jacobs, D. W., Kriegman, D. J., and Kumar, N. (2011). Localizing parts of faces using a consensus of exemplars. In *CVPR*, pages 545–552. IEEE.

Bendersky, M., Garcia-Pueyo, L., Harmsen, J., Josifovski, V., and Lepikhin, D. (2014). Up next: Retrieval methods for large scale related video suggestion. In *SIGKDD*, pages 1769–1778. ACM.

Berg, T. and Belhumeur, P. N. (2013). POOF: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *ICCV*.

Bottou, L. and Bousquet, O. (2011). The tradeoffs of large-scale learning. *Optimization for Machine Learning*, page 351.

Bourdev, L. and Malik, J. (2009). Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*.

Brown, M., Hua, G., and Winder, S. (2011). Discriminative learning of local image descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):43–57.

Chai, Y., Lempitsky, V., and Zisserman, A. (2013a). Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*.

Chai, Y., Lempitsky, V., and Zisserman, A. (2013b). Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*.

Chai, Y., Rahtu, E., Lempitsky, V., Van Gool, L., and Zisserman, A. (2012). Tricos: A tri-level class-discriminative co-segmentation method for image classification. In *ECCV*.

Chatfield, K., Lempitsky, V. S., Vedaldi, A., and Zisserman, A. (2011). The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*.

Chaudhry, R., Ravichandran, A., Hager, G., and Vidal, R. (2009). Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *CVPR*. IEEE.

Chen, M., Zheng, A., and Weinberger, K. (2013). Fast image tagging. In *ICML*.

Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*.

Dalal, N. and Triggs, B. (2005a). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*.

Dalal, N. and Triggs, B. (2005b). Histograms of oriented gradients for human detection. In *CVPR 2005*.

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

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

Deng, J., Krause, J., and Fei-Fei, L. (2013). Fine-grained crowdsourcing for fine-grained recognition. In *CVPR*.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. (2014). DeCAF: A deep convolutional activation feature for generic visual recognition. *ICML*.

Duan, K., Parikh, D., Crandall, D., and Grauman, K. (2012). Discovering localized attributes for fine-grained recognition. In *CVPR*.

Everingham, M., Gool, L. V., Williams, C. K., Winn, J., and Zisserman, A. (2010). The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338.

Farrell, R., Oza, O., Zhang, N., Morariu, V. I., Darrell, T., and Davis, L. S. (2011). Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*.

Fei-Fei, L., Fergus, R., and Perona, P. (2007). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645.

Fergus, R., Perona, P., and Zisserman, A. (2003). Object class recognition by unsupervised scale-invariant learning. In *CVPR*.

Gavves, E., Fernando, B., Snoek, C. G., Smeulders, A. W., and Tuytelaars, T. (2013). Fine-grained categorization by alignments. In *ICCV*.

Gehler, P. and Nowozin, S. (2009). On feature combination for multiclass object classification. In *CVPR*.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*.

Glorot, X., Bordes, A., and Bengio, Y. (2011a). Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics*.

Glorot, X., Bordes, A., and Bengio, Y. (2011b). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*.

Goring, C., Rodner, E., Freytag, A., and Denzler, J. (2013). Nonparametric part transfer for fine-grained recognition. In *CVPR*.

Gosselin, P.-H., Murray, N., Jégou, H., and Perronnin, F. (2013). Boosting the fisher vector for fine-grained classification. *INRIA Technical Report*.

Grgic, M., Delac, K., and Grgic, S. (2011). Scface–surveillance cameras face database. *Multimedia tools and applications*, 51(3):863–879.

Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 object category dataset. *Technical Reports*.

Hariharan, B., Malik, J., and Ramanan, D. (2012). Discriminative decorrelation for clustering and classification. In *ECCV*.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.

Hinton, G. E., Vinyals, O., and Dean, J. (2014). Distilling the knowledge in a neural network. In *NIPS 2014 Deep Learning Workshop*.

Jaakkola, T., Haussler, D., et al. (1999). Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, pages 487–493.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial transformer networks. In *NIPS*.

Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *CVPR*, pages 3128–3137.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *CVPR*.

Klaser, A., Marszałek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3d-gradients. In *BMVC*.

Krause, J., Gebru, T., Deng, J., Li, L.-J., and Fei-Fei, L. (2014). Learning features and parts for fine-grained recognition. In *ICPR*.

Krause, J., Jin, H., Yang, J., and Fei-Fei, L. (2015a). Fine-grained recognition without part annotations. In *CVPR*.

Krause, J., Sapp, B., Howard, A., Zhou, H., Toshev, A., Duerig, T., Philbin, J., and Fei-Fei, L. (2015b). The unreasonable effectiveness of noisy data for fine-grained recognition. *arXiv preprint arXiv:1511.06789*.

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114.

Kumar, N., Belhumeur, P. N., Biswas, A., Jacobs, D. W., Kress, W. J., Lopez, I. C., and Soares, J. V. (2012). Leafsnap: A computer vision system for automatic plant species identification. In *ECCV*.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*.

Le, Q. V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., and Ng, A. Y. (2011). Building high-level features using large scale unsupervised learning. *arXiv preprint arXiv:1112.6209*.

Le Cun, B. B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer.

Lin, T.-Y., RoyChowdhury, A., and Maji, S. (2015). Bilinear CNN models for fine-grained visual recognition. In *ICCV*.

Liu, J., Kanazawa, A., Jacobs, D., and Belhumeur, P. (2012). Dog breed classification using part localization. In *ECCV*.

Liu, J., Luo, J., and Shah, M. (2009). Recognizing realistic actions from videos in the wild. In *CVPR*.

Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440.

Lowe, D. G. (2004a). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110.

Lowe, D. G. (2004b). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

McCool, C., Wallace, R., McLaren, M., El Shafey, L., and Marcel, S. (2013). Session variability modelling for face authentication. *IET biometrics*, 2(3):117–129.

Mutch, J. and Lowe, D. G. (2008). Object class recognition and localization using sparse features with limited receptive fields. *International Journal of Computer Vision*, 80(1):45–57.

Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987.

Parikh, D. and Grauman, K. (2011). Interactive discovery of task-specific nameable attributes. In *CVPR*.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. (2012). Cats and dogs. In *CVPR*.

Perronnin, F., Sánchez, J., and Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. In *ECCV*.

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2008). Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*.

Philippe, G. and Naila, M. (2012). Boost the fisher vector for fine-grained classification. *INRIA Technical Report*.

Ponce, J., Berg, T. L., Everingham, M., Forsyth, D. A., Hebert, M., Lazebnik, S., Marszalek, M., Schmid, C., Russell, B. C., Torralba, A., et al. (2006). Dataset issues in object recognition. In *Toward category-level object recognition*, pages 29–48. Springer.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*.

Rother, C., Kolmogorov, V., and Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. In *TOG*.

Saenko, K., Kulis, B., Fritz, M., and Darrell, T. (2010). Adapting visual category models to new domains. In *ECCV*, pages 213–226.

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):222–245.

Schüldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: a local svm approach. In *ICPR*.

Scovanner, P., Ali, S., and Shah, M. (2007). A 3-dimensional sift descriptor and its application to action recognition. In *International Conference on Multimedia*, pages 357–360. ACM.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., and LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv:1312.6229*.

Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *NIPS*.

Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *ICCV*.

Soomro, K., Zamir, A. R., and Shah, M. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*.

Sun, L., Jia, K., Yeung, D.-Y., and Shi, B. E. (2015). Human action recognition using factorized spatio-temporal convolutional networks. In *ICCV*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions. *CVPR*.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. *ICCV*.

Van De Sande, K. E., Gevers, T., and Snoek, C. G. (2010). Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596.

Van de Sande, K. E., Uijlings, J. R., Gevers, T., and Smeulders, A. W. (2011). Segmentation as selective search for object recognition. In *ICCV*.

Van De Weijer, J., Schmid, C., Verbeek, J., and Larlus, D. (2009). Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523.

Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., and Belongie, S. (2015a). Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *CVPR*.

Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., and Belongie, S. (2015b). Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *CVPR*.

Vogt, R. and Sridharan, S. (2008). Explicit modelling of session variability for speaker verification. *Computer Speech & Language*, 22(1):17–38.

Wah, C., Branson, S., Perona, P., and Belongie, S. (2011a). Multiclass recognition and part localization with humans in the loop. In *ICCV*.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. (2011b). The caltech-ucsd birds-200-2011 dataset. *Technical Reports*.

Wallace, R., McLaren, M., McCool, C., and Marcel, S. (2011). Inter-session variability modelling and joint factor analysis for face authentication. In *International Joint Conference on Biometrics*.

Wand, M. and Schultz, T. (2011). Session-independent emg-based speech recognition. In *Biosignals*. Citeseer.

Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. (2013). Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1):60–79.

Wang, H. and Schmid, C. (2013). Action recognition with improved trajectories. In *ICCV*.

Wang, H., Ullah, M. M., Klaser, A., Laptev, I., and Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *BMVC*.

Willems, G., Tuytelaars, T., and Van Gool, L. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*.

Xie, L., Tian, Q., Yan, S., and Zhang, B. (2013). Hierarchical part matching for fine-grained visual categorization. Technical report, Technical Report, Department of Computer Science and Technology, Tsinghua Univerity.

Xie, S., Yang, T., Wang, X., and Lin, Y. (2015). Hyper-class augmented and regularized deep learning for fine-grained image classification. In *CVPR*.

Xu, Z., Huang, S., Zhang, Y., and Tao, D. (2015). Augmenting strong supervision using web data for fine-grained categorization. In *ICCV*.

Yang, S., Bo, L., Wang, J., and Shapiro, L. (2012). Unsupervised template learning for fine-grained object recognition. In *Advances in Neural Information Processing Systems 25*, pages 3131–3139.

Yao, B., Bradski, G., and Fei-Fei, L. (2012). A codebook-free and annotation-free approach for fine-grained image categorization. In *CVPR*.

Yao, B., Khosla, A., and Fei-Fei, L. (2011). Combining randomization and discrimination for fine-grained image categorization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1577–1584. IEEE.

Zeiler, M. D. and Fergus, R. (2013). Visualizing and understanding convolutional neural networks. *arXiv:1311.2901*.

Zhang, N., Donahue, J., Girshick, R., and Darrell, T. (2014). Part-based R-CNNs for fine-grained category detection. In *ECCV*.

Zhang, N., Farrell, R., and Darrell, T. (2012). Pose pooling kernels for sub-category recognition. In *CVPR*.

Zhang, N., Farrell, R., and Darrell, T. (2013a). Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*.

Zhang, N., Farrell, R., Iandola, F., and Darrell, T. (2013b). Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*.

Zhang, N., Shelhamer, E., Gao, Y., and Darrell, T. (2015). Fine-grained pose prediction, normalization, and recognition. *arXiv:1511.07063*.

Zitnick, C. L. and Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In *ECCV*.