Structured Learning Approach to Image Descriptor Combination

Author

Zhou, J, Fu, Z, Robles-Kelly, A

Published

2011

Journal Title

IET Computer Vision

DOI

10.1049/iet-cvi.2010.0080

Rights statement

A Structured Learning Approach to Image Descriptor Combination

Jun Zhou[1,2,3] Zhouyu Fu[4] Antonio Robles-Kelly[1,2,3]

[1]NICTA,* PO BOX 8001, Canberra, ACT 2601, Australia

[2]College of Engineering and Comp. Science, ANU, Canberra, ACT 0200, Australia

[3]School of Eng. and Information Technology, UNSW@ADFA, Canberra, ACT 2600, Australia

[4]Faculty of Information Technology, Monash University, VIC 3800, Australia

Abstract

In this paper, we address the problem of combining descriptors for purposes of object categorisation and classification. We cast the problem in a structured learning setting by viewing the classifier bank and the codewords used in the categorisation and classification tasks as random fields. In this manner, we can abstract the problem into a graphical model setting, in which the fusion operation is a transformation over the field of descriptors and classifiers. Thus, the problem reduces itself to that of recovering the optimal transformation using a cost function which is convex and can be converted into either a quadratic or linear programme. This cost function is related to the target function used in discrete Markov Random Field approaches. We demonstrate the utility of our algorithm for purposes of image classification and learning class categories on two datasets.

*NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

1 Introduction

Recently, object categorisation and classification has attracted much attention in the computer vision and pattern recognition communities. In practice, object classification and image categorisation techniques [1, 2, 3] are based upon the comparison of image features, summarised in the form of a codebook, between the images in the dataset and a user-provided query image. Thus, the codebook serves as a means to recover the closest match through classification. Furthermore, descriptors are often combined so as to assign different importance to each of them in order to maximise performance.

It seems natural that the classifier output and the descriptor input should enforce consistency over the training set with respect to the object categories or classes while maximising the correlation between the training set and the classifier output. However, such an aim of computation is not straightforward due to the complexity of both the inference process and the summarisation of the dataset into a codebook of visual words. Indeed, summarisation into a codebook has been used by a number of approaches in the literature. For instance, Li and Perona [4] and Quelhas *et al* [5] have used codewords to learn scene categories. Nilsback and Zisserman [6] have combined visual vocabularies for purposes of flower image classification.

Despite the effectiveness of these approaches, the multidimensional nature of the image features makes their combination and selection a nontrivial task, which is often effected by manual tuning of algorithm parameters. This is even more important since the combination of the descriptor-set has a direct impact on the performance of the classification task. Addressing this problem, Varma and Ray [7] have used a kernel learning approach to learn the trade-off between discriminative power and invariance of image descriptors. Methods such as the one in [8] rely upon clustering algorithms to provide improved organisation of the codebook. Another alternative is to view the visual words as multidimensional data and exploit similarity information in a graph-theoretic setting using unsupervised learning. Examples include the method presented by Sengupta and Boyer [9] and that developed by Shokounfandeh et. al. [10], which employ information-theoretical criteria to hierarchically structure the dataset and pattern recognition methods to match the candidates.

In either case of recovering the optimal descriptors directly [11] or their optimal combination

[12, 13, 6, 14, 7], it is somewhat expected that the codewords should both minimise the variance across the dataset with respect to the classifier output and maximise the categorisation performance. The reason is that the classifier output is dependent on the image representation and the similarity measure employed to categorise the images with respect to the query [15]. Along these lines, the prevalent classifiers for categorisation and retrieval are Support Vector Machines (SVMs) [16]. These have often been used in conjunction with relevance feedback techniques [17], where the user provides on-line training information, i.e. positive and negative examples, on the retrieval results so as to cross-validate the parameters of the classifier used in the query operation [18]. Nonetheless, K-nearest neighbour [6], neural networks [19] and other classification methods may be used.

On the classifiers for image categorisation and recognition, fusion has been an active research topic in pattern recognition and computer vision in the past two decades. The idea is to combine multiple weak classifiers to yield a strong classifier with optimal performance. This is usually achieved by finding the optimal weights for the linear combination of classifiers. Earlier work on fusion is mostly based on heuristics in the selection of weights for the linear classifier combination. Two special cases are the majority voting rule for discrete classifier output and the sum rule for probabilistic outputs. Interested readers are referred to [20] for a comprehensive overview on weight selection for linear classifier fusion.

In the past decade, a number of principled methods have been proposed for combining classifiers [21, 22, 23, 24], among which bagging [21] and boosting [23] are the most prominent and well known in the community. Unlike heuristic-based methods, bagging and boosting have clear statistical interpretations. Bagging is based on combining classifiers trained on bootstrapping samples [21], while boosting is related to additive logistic regression [25]. The other two random classifier combination schemes, the random subspace [22] and random forest [24] also share a similar spirit to bagging-based classifiers, where, for random space, the randomness comes from splitting the feature space rather than the sample space. Note that fusion can not only be done at classifier level, but also at the feature level. The idea of feature-level fusion has been taken further in the scenario of Support Vector Machine (SVM) classifiers and kernel based learning by recovering the optimal linear combination of kernel matrices, each of which account for features obtained from a different source or modality. The kernel-target alignment [26] and Semi-definite Programming (SDP) method in [27] are two examples of kernel matrix learning.

Despite the vast literature on classifier fusion and feature combination techniques, choosing an optimal linear weighting that can be applied to all classes is still a non-trivial task. Current approaches are somewhat limited in the sense that class-specific weighting for multi-class categorisation problems [1] remains elusive. Since certain assumptions are only discriminative for some specific category, anisotropic weighting for different classes remains an attractive option.

In this paper, we address the problem of descriptor combination for object categorisation and present an extension of the proposed method for classifier fusion. To do this, we pose the problem in a graph regularisation setting. Departing from a Markovian formulation, we view the finite set of descriptors as a structured field. In this manner, we can recover a transformation over the field such that the intra-class variance is minimised while enforcing accordance between image category samples and their labels. Thus, we present a means for combining descriptors and classifier-outputs in a generic fashion using techniques for inference over structured data. This is not only theoretically important, but practically useful since it permits the learning of the transformation over the codebook and over the classifier bank. The method is quite general in nature and permits the use of a number of image features and classification methods elsewhere in the literature. Moreover, it provides a unified view on classifier and descriptor combination based upon Markov random fields.

2 Structured Learning

As mentioned earlier, in this paper, we cast the problem of categorisation into a descriptor combination setting. In this section, we view descriptor combination as a transformation over the codebook aimed at minimising a cost function that both enforces consistency over the training set

[1] Since classification and categorisation are akin tasks in computer vision and pattern recognition, for the sake of conciseness, we use the word categorisation hereinafter. We do this adopting the classical view that categories, as classes, are mutually exclusive and collectively exhaustive. In this way, any image in the dataset belongs to one, and only one, of the categories or classes under consideration.

with respect to the object categories and maximises the correlation between the training set and the classifier output. This provides a general treatment based upon Markov random fields (MRF) which can be extended to classifier fusion settings later on in the paper.

In this section, we motivate the notion that descriptor combination can be cast as the recovery of the optimal transformation over an MRF. To commence, let the set of images in the dataset be $\Gamma = \bigcup_{j=1}^{N} \gamma_j$, where γ_j is the j^{th} set of images in Γ corresponding to the N categories in the dataset. Similarly, consider the codeword Φ_j corresponding to the j^{th} image in Γ.

To model the descriptor combination problem as a structured learning one, we note that each of the codewords can be further expressed in terms of the image descriptors. For the sake of simplicity, we view these codewords as a vector Y_j corresponding to the concatenation of those descriptors in the set $\Phi_j = \{\phi_1, \phi_2, \ldots, \phi_{|\Phi_j|}\}$, where ϕ_k is the k^{th} descriptor under study. Moreover, we can view these descriptors as a field subject to a transformation $\mathcal{T} : Y_j \mapsto X_j$, where X_j is the vector of combined descriptors. In practice, $\mathcal{T}$ is a family of matrices such that $X_j = \mathcal{T} Y_j$. This opens-up the possibility of casting the descriptor combination problem in a structured learning setting, in which the aim of computation is the recovery of the optimal transformation matrix $\mathcal{T}$.

2.1 Markovian Formulation

Thus, the descriptor combination processes can be viewed as a structured learning task. Viewed in this manner, the aim of computation is the recovery of the optimal transformation matrix over the field of image descriptors. As mentioned earlier, to recover this transformation matrix, we make use of MRFs by abstracting the problem into a graphical model. Let $G(\mathcal{V}, \mathcal{E})$ denote a graph with node-set $\mathcal{V} = \{V_1, \ldots, V_N\}$ and edge-set $\mathcal{E} = \{E_{i,j} | V_i, V_j \in \mathcal{V}\}$. Each $V_i \in \mathcal{V}$ is associated with a hidden variable $\mathcal{X}_i$ in the state space Λ.

With these ingredients, the joint probability distribution represented by the MRF is given by

$$P(\mathcal{X}) = \frac{1}{Z} \prod_{i,j \in \mathcal{E}} \varphi_{i,j}(\mathcal{X}_i, \mathcal{X}_j) \prod_{i \in \mathcal{V}} \zeta_i(\mathcal{X}_i) \tag{1}$$

where $\mathcal{X} = \{\mathcal{X}_i\}_{i=1,\ldots,|\mathcal{V}|}$ is the set of hidden variables and $\zeta_i(\mathcal{X}_i)$ and $\varphi_{i,j}(\mathcal{X}_i, \mathcal{X}_j)$ are unitary and binary potential functions which determine the likelihood of the graph nodes corresponding to the state $\lambda \in \Lambda$. In the equation above, $Z = \int_{\mathcal{X}} \prod_{i,j \in \mathcal{E}} \varphi_{i,j}(\mathcal{X}_i, \mathcal{X}_j) \prod_{i \in \mathcal{V}} \zeta_i(\mathcal{X}_i)$ is the normalisation factor.

Since this normalisation factor is invariant with respect to $\mathcal{X}_i$, we can remove it from further consideration. Moreover, the joint probability $P(\mathcal{X})$ over the state space Λ can be maximised by recasting the inference over the above MRF into a Maximum A Posteriori (MAP) setting. Thus, we can view the variables $\mathcal{X}_i$ in the MRF as continuous vector variables $\mathcal{X}_i = [x_{i,1}, \ldots, x_{i,|\mathcal{X}|}]$, where $x_{i,k}$ is the inner product of the rows of $\mathcal{T}$ and the codewords. This permits us to view $x_{i,k}$ as the weighted analogue of the classifier outputs.

Note that, by assuming non-negative codewords and, due to the fact that we can always assume a probabilistic output for the classifiers, we can constrain the vector variables to be non-negative. As a result, we have the constraint $\sum_a x_{i,a} \geq 0 \ \ \forall i$. Taking the logarithm of Equation 1, we have

$$\log P(\mathcal{X}) = \sum_{i=1}^{|\Gamma|} \sum_{a=1}^{|\mathcal{X}_i|} c_i(a) x_{i,a} + \sum_{i \sim j} \sum_{a=1}^{|\mathcal{X}_i|} \sum_{b=1}^{|\mathcal{X}_j|} w_{i,j}(a,b) x_{i,a} x_{j,b} \tag{2}$$

where $c_i(a) = \log \zeta_i(a)$ and $w_{i,j}(a,b) = \log \varphi_{i,j}(a,b)$ are determined by the potential functions, and $i \sim j$ means $\mathcal{X}_i$ and $\mathcal{X}_j$ belong to the same category.

Maximising the above cost function is equivalent to solving the original MRF inference problem, as defined in Equation 1. The cost function is in quadratic form, and, hence, it is a natural choice to apply quadratic programming techniques to solve the relaxation problem. However, the Hessian of Equation 2 is determined by the coefficients of the second order term $w_{i,j}(a,b)$ and, as a result, are not necessarily convex. A number of techniques have been proposed to relax the discrete labeling problem and convert the MRF cost functional into more tractable forms. Along these lines, some examples are SDP [28, 29, 30], SOCP [31], and spectral relaxation [32].

Here, instead of finding a continuous relaxation for the original cost function of the MRF model, we propose an alternative cost function which is closely related to Equation 1. Notice that the first and the second terms on the right-hand-side of the cost function in Equation 2 can be treated as correlation terms. The first of them measures the correlation between the label and the single node potential. The second term measures the compatibility between labels of neighbouring nodes weighed by the pairwise potential $w_{i,j}(a,b)$. By thinking of correlation as a measure of similarity which can be viewed as an inverse distance, we can transform the maximisation problem at hand into a minimisation one where the L_2 norm is a natural choice. The corresponding cost function is hence defined as follows

$$\min f(\mathcal{X}) = \sum_{k=1}^{N} \left(\eta \sum_{i \in \gamma_k}^{|\Gamma|} \sum_{a=1}^{|\mathcal{X}_i|} (c_i(a) - x_{i,a})^2 + \sum_{i \sim j;\ i,j \in \gamma_k} \sum_{a=1}^{|\mathcal{X}_i|} \sum_{b=1}^{|\mathcal{X}_j|} w_{i,j}(a,b)(x_{i,a} - x_{j,b})^2 \right) \qquad (3)$$

where η is a regularisation constant.

The constraints for the above equation are the same as that in Equation 2. Note that the reformulation of the cost function as above has several appealing properties. First, it is closely related to the MRF model in terms of its physical meaning. Like the MRF, our cost function accommodates two complementary terms, i.e. a term which measures the compatibility between the data and its transformed field variable and a smoothness term which enforces the consistency between the variables for those nodes corresponding to the images in the same category. The main difference in the cost functions with respect to Equation 2 is the replacement of the inner product with the squared distance. More importantly, the cost function defined above is convex. This leads to two desirable consequences. Firstly, we can always attain globally optimal solutions for the relaxed problem on the continuous label variables. Secondly, the problem can be reduced to that of solving a sparse linear system of equations with positive semi-definite Hessian.

Further, note that, in many applications, each coefficient $w_{i,j}(a,b)$ in Equation 3 can be decomposed into two factors $p_{i,j}$ and and $q_{a,b}$. The former depends only on the local neighbourhood for each node, while the latter depends only on the variables in $\mathcal{X}$. By assuming a constant penalty for different label pairs, i.e. $q_{a,b}$ is a constant factor, a much simplified form of the cost function can be used. This is given by

$$f(\mathcal{X}) = \sum_{k=1}^{N} \left(\eta \sum_{i \in \gamma_k}^{|\Gamma|} \sum_{a=1}^{|\mathcal{X}_i|} (c_i(a) - x_{i,a})^2 + \sum_{i \sim j;\ i,j \in \gamma_k} \sum_{a=1}^{|\mathcal{X}_i|} \sum_{b=1}^{|\mathcal{X}_j|} p_{i,j}(x_{i,a} - x_{j,b})^2 \right) \qquad (4)$$

where $p_{i,j}$ specifies the degree of penalty for disparate labeling of adjacent nodes V_i and V_j. This is akin to the discontinuity preserving MRF models with Potts prior [33] and inhomogeneous costs for different pairs of nodes. The cost function of Equation 4 is reminiscent of graph regularisation frameworks for solving semi-supervised learning problems [34, 35]. Indeed, the labeling problem in vision and the semi-supervised learning problem in machine learning can both be thought as the generic problem of inference over structured data which can be tackled using the general framework of MRFs.

2.2 L_1 vs L_2-Norm

In this section, we explore the use of the L_2 and L_1-norms for purposes of recovering the vectors $x_{i,a}$. The reason for this is that the optimal set of vectors $x_{i,a}$ can be related to the transformation matrix $\mathcal{T}$ through the relations provided in the previous section making use of an appropriate norm applied to the cost function.

2.2.1 L_2-Norm

We commence by focusing our attention on the L_2-norm. Note that, the cost function in Equation 3 is based upon the L_2-norm. To minimise it, in practice, we can treat the problem as a continuous relaxation one which leads to a convex quadratic optimisation problem akin to those used in segmentation frameworks based upon Random Walks [36].

Thus, the cost function in Equation 3 can be re-written in the following matrix form

$$\min f(\mathbf{X}) = tr(\mathbf{X}^T \mathbf{L} \mathbf{X}) + \eta ||\mathbf{X} - \mathbf{C}||^2 \tag{5}$$

where $\mathbf{X}$ is a matrix of transformed vectors for all images in the dataset, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian of the graph, $\mathbf{W}$ is the adjacency matrix of the graph with entries $p_{i,j}$, and $\mathbf{D}$ is a diagonal matrix of node degrees whose diagonal elements are given by the row-sums of $\mathbf{W}$. We have introduced the matrix $\mathbf{C}$, which corresponds to the variables $c_i(a)$ in Equation 3.

The optimisation problem in Equation 5 is a convex one whose global minimum can be found by solving the following systems of linear equations.

$$(\mathbf{L} + \eta \mathbf{E}^T \mathbf{E}) \mathbf{X}_{.,j} = \eta \mathbf{E}^T \mathbf{E} \mathbf{C}_{.,j} (j = 1, \ldots, |\Gamma|) \tag{6}$$

where $\mathbf{X}_{.,j}$ denotes the j^{th} column of the matrix of transformed vectors $\mathbf{X}$ and $\mathbf{C}_{.,j}$ denotes the j^{th} column of the matrix $\mathbf{C}$ comprised by the expectation vectors. Note that, in the equation above, we have introduced the matrix $\mathbf{E}$ which corresponds to the category-label vectors for the images in Γ, i.e. $e(i,j) = 1$ if and only if the ith image in the dataset corresponds to the jth category γ_j.

This system of equations can be solved with Gaussian-Siedel or conjugate gradient methods. Further, due to the sparse nature of $\mathbf{L}$ and $\mathbf{E}$, we can solve them efficiently by making use of sparse Cholesky factorisation [37]. First $\mathbf{L} + \eta \mathbf{E}^T \mathbf{E}$ is factorised into the product of an upper-triangular matrix $\mathbf{Q}_h$ and its transpose $\mathbf{Q}_h^T$, which is a lower-triangular matrix. Then two trivial linear equations, $\mathbf{Q}_h \mathbf{H} = \mathbf{E}^T \mathbf{E} \mathbf{C}_{\cdot,j}$ and $\mathbf{Q}_h^T \mathbf{X}_{\cdot,j} = \mathbf{H}$ are solved via forward and backward substitution to recover $\mathbf{X}_{\cdot,j}$. As a result, the sparse Cholesky factorisation only needs to be applied once for recovering multiple columns of $\mathbf{X}$.

2.2.2 L_1-Norm

We now turn our attention to L_1-norm as an alternative to the L_2-norm. By substituting the L_1-norm for the L_2-norm, the cost function becomes

$$f(X) = \sum_{k=1}^{N} \left(\eta \sum_{i \in \gamma_k}^{|\Gamma|} \sum_{a=1}^{|X_i|} |c_i(a) - x_{i,a}| + \sum_{i \sim j;\ i,j \in \gamma_k} \sum_{a=1}^{|X_i|} \sum_{b=1}^{|X_j|} w_{i,j} |x_{i,a} - x_{j,b}| \right) \tag{7}$$

where, for convenience, we have written $w_{i,j}$ as an alternative to $p_{i,j}$ in Equation 4.

By introducing auxiliary variables $r_{u,j}$ and $z_{u,v,j}$ for the j^{th} category for the images abstracted as nodes u and v, we can rewrite Equation 7 in terms of a linear programme (LP) over the variables X_i, $r_{u,j}$ and $z_{u,v,j}$ in the following manner

$$\min \quad f(\mathcal{X}, \mathcal{Z}, \mathcal{R}) = \eta \sum_{u,j} r_{u,j} + \sum_{u,v,j} w_{u,v} z_{u,v,j} \tag{8}$$

$$s.t. \quad \begin{cases} |x_{u,j} - c_u(j)| \leq r_{u,j} \ \forall \ u, j \\ |x_{u,j} - x_{v,j}| \leq z_{u,v,j} \ \forall \ u, v, j \\ r_{u,j} \geq 0 \ \forall \ u, j \text{ and } z_{u,v,j} \geq 0 \ \forall \ u, v, j \end{cases}$$

The above LP formulation has the following dual form

$$\max \quad f(\mathcal{P}, \mathcal{Q}) = \sum_{u,j} q_{u,j} \tag{9}$$
$$s.t. \quad \begin{cases} \sum_{u \sim v} p_{u,v,j} - \sum_{u \sim v} p_{v,u,j} = 0 \; \forall \; u, j \\ 0 \leq p_{u,v,j} \leq w_{u,v} \\ 0 \leq q_{u,j} \leq \eta \end{cases}$$

It is straight forward to see that the primal and dual forms in Equations 8 and 9 correspond to the mincut and maxflow problems in network flow theory with multiple terminal nodes respectively [38]. This is as the first and second constraints in Equation 9 correspond to the mass balance constraint and the capacity constraint of the network flow. Similar results on L_1-norm regularisation were noted in [39] for binary labelling problems. For the binary case, the duality between minimum cut and maximum flow can also be explained by the Ford-Fulkerson theory [38]. Thus, the above case is a more general result which applies to multiple classes, i.e. multiple categories in the data set.

3 Discussions and Implementation Issues

From an alternative viewpoint, the optimisation task presented in the previous section can also be viewed as a problem of regression, where the norms defined in both the data and smoothness terms in Equation 3 reflect the fitting error corresponding to the field transformation. The ideal case is given when all the images in γ_i should have exactly the same classifier output. In the Least Squares regression, i.e. the case of L_2-norm, the cost increases quadratically with the error. This is somehow quite sensitive to outliers and noise corruption. The L_1-norm based regression is more robust, but the cost still increases linearly with respect to the error. In M-estimator based methods [40], a non-linear error measure or influence function is employed with suppressed cost for larger error. The framework presented here is quite general and can accommodate M-estimators in a straightforward manner. This is as the two terms in the cost function can be defined in terms of distance functions corresponding to robust statistics rather than L_1 or L_2-norms. Moreover, some specific choices of influence functions, such as the Huber one are combinations of L_1 and L_2-norms. The Tukey's biweight function, on the other hand, bounds the cost at its supremum for increasing degree of error.

Note that, in general, the L_1 and L_2-norm based regression can be viewed as particular cases of M-estimators with certain influence functions. In practice, M-estimator based regression is tackled in an iterated re-weighted Least Squares (IRWS) framework. Thus, the cost function can be optimised by starting with some initial estimate of transformation matrix and solving the weighted version of the L_2-norm labeling problem in Equation 5 with an updated weight matrix. The (u, v)th entry of the weight matrix is then given by $w'_{u,v} = w_{i,j}\rho(d(u, v))$, where $\rho(d(u, v))$ is the weight function associated with the robust M-estimator employed, $d(u, v)$ is the L_2 or squared Euclidean distance. In this case, the data terms are also modified accordingly.

At this point, it is also worth noting that throughout the previous sections we have not made any assumption on the potential $c_i(a)$ rather than the fact that it should be non-negative. For purposes of descriptor combination, here we use the expectation over the probabilistic output Q_{ξ,γ_i} in Equation 14. That is, for each image in the category γ_a, the potential $c_i(a)$ is given by

$$c_i(a) = E[\gamma_a \mid \mathbf{Q}] = \frac{\sum_{\gamma_a} Q_{\Phi_i,\gamma_a}\Phi_i}{\sum_{\gamma_a} Q_{\Phi_i,\gamma_a}} \tag{10}$$

where $E[\cdot]$ is the expectation operator. In this manner, in our implementation, the potential is the expected output of the classifiers trained over a set of codewords. This implies that the cost function will enforce smoothness over the image category while favoring transformations that bring the codewords "closer" to the expected classifier outputs.

For the classifier fusion we have used a different approach and set $c_i(a)$ to unity if and only if the i^{th} image in the data set belongs to γ_a and zero otherwise. As a result, the optimisation will aim at recovering a transformation matrix that brings the classifier-bank output closer to its hard limits while, again, enforcing smoothness over the classification results for the images in the category under study.

3.1 Soft vs Hard Constraints

The minimisation schemes above can be viewed as an optimisation in which the category consistency for the images in Γ is posed as soft constraint. This can be slightly modified in terms of its formulation as follows

$$\min f(\mathcal{X}) = \sum_{k=1}^{N} \Bigg(\sum_{i \sim j;\ i,j \in \gamma_k} \sum_{a=1}^{|X_i|} \sum_{b=1}^{|X_j|} w_{i,j}(x_{i,a} - x_{j,b})^2 \Bigg) \qquad (11)$$

$$\text{s.t.} \quad c_i(a) = x_{i,a}\ \forall i, a$$

This corresponds to the case of infinite η for the formulation in Equation 3. As a result, the category indexes for images in the set Γ can be fixed to their ground truth values. This is in contrast with the previous section formulation the category indexes are allowed to change.

Note that the hard constraint case does not apply to the categorisation problem as posed here since the aim of computation is to recover a transformation matrix and not the image labels. Nonetheless, this link provides a means to showing that the L_1 and L_2-norm methods above coincide, in its hard limit formulations, with Graph Cuts [33] and the Random Walker segmentation algorithm [36]. Specifically, for the random walks on the L_1-Norm, the solution is given by the following equation,

$$\mathbf{L}_{u,u}\mathbf{X}_{u(.,j)} = -\mathbf{L}_{u,a}\mathbf{C}_{.,j} \text{ for } j = 1, \ldots, |\Gamma| \qquad (12)$$

where $\mathbf{X}_u$ refers to label vectors of unlabeled items only, and $\mathbf{C}_{.,j}$ has been defined in section 2.2.1. This can be solved efficiently via the same sparse Cholesky Factorisation technique mentioned previously. Thus, employing hard constraints is equivalent to setting the t-link weights to infinity instead of η except to the jth t-link for any labeled item u that belongs to j^{th} category. This clamps the image u to label j and prevents it from connecting to other t-nodes.

3.2 Extensions to Classifier Fusion

Further, note that the idea of recovering the optimal transformation over a structured field is a quite general one that can also be applied for purposes of classifier fusion in image categorisation. This is because, given several descriptors, we can always combine them into codewords comprised by all their permutations. That is, for each image we can have a family of codewords corresponding to transformations of permuted descriptors. As a result, we can adopt an approach akin to pairwise frameworks elsewhere in the literature [41] aimed at building a classifier for each of the categories γ_i, where the final classification result is obtained by the majority voting results over all the permutations of the descriptors under study. Thus, the classifier bank $\Psi = \{\psi_1, \psi_2, \ldots, \psi_{|\Psi|}\}$ delivers at output the probability of the j^{th} image belonging to the category γ_i given the transformations over the set of permutations of the descriptors in Φ_j.

By using the probabilistic output of each classifier ψ_i [42], we can obtain a version of multiclass classification which can be viewed as a generalisation to majority voting. Hard decisions can then be obtained by thresholding the posterior probabilities into binary values. A natural way of combining the classification results over the set Ξ of transformed descriptor permutations is maximising the total posterior probability given by

$$\varrho = \arg\max_{\Gamma} \sum_{\xi \in \Xi} Q_{\xi,\gamma_i} \tag{13}$$

where Q_{ξ,γ_i} is the probabilistic output of the classifier for the permutation $\xi \in \Xi$ of image descriptors and the i^{th} category γ_i.

By viewing the classifier bank as a field, we can introduce the transformation $\mathcal{S}$, where again, the transformation $\mathcal{S}$ can be viewed as a family of matrices such that the posterior becomes

$$\varrho = \arg\max_{\Gamma} \mathcal{S}\mathbf{Q} \tag{14}$$

where $\mathbf{Q}$ is a matrix whose entry indexed j, i is given by Q_{ξ,γ_i}.

To recover the transformation matrix $\mathcal{S}$, we can use a formulation akin to that in Equation 3 when the pairwise potential $w_{i,j}$ is null. To see this more clearly, note that, if the true labels for the input images are known, the equation above can be simplified to the following expression

$$\min f(\mathcal{X}) = \sum_{i=1}^{N} \sum_{\xi \in \Xi}^{|\Xi|} (c_{\xi,i} - Q_{\xi,\gamma_i})^2 \tag{15}$$

where $c_{\xi,i}$ is a vector with the same dimension as Q_{ξ,γ_i} whose i-th entry corresponding to the class γ_i is set to 1 and 0 otherwise.

4 Experiments

In this section we demonstrate the utility of our structured learning algorithm for purposes of object classification and image categorisation. To this end, we use the the Oxford flower dataset [6] for image classification and the ETH-80 dataset [43] for categorisation.

Following the feature extraction method in [6], we have recovered the SIFT descriptors [44], colour and texture features making use of the MR-filter band [2] for each of the images in our datasets. For purposes of recovering a codebook, these features have been clustered using a K-means algorithm [45]. Each image is then represented as a visual word corresponding to the frequency histogram of the features in the codebook. In our experiments, we have used the code-words corresponding to each individual feature and the vector yielded by their concatenation. In the training stage, a Support Vector Machine (SVM) classifier-bank is generated from different features of the training set. Then feature optimisation and classifier fusion is performed so as to evaluate the corresponding performance on the testing set.

The descriptor optimisation process commences by computing the correct classification rate on the training set using the trained SVM classifier-bank. An expected feature vector for each class is then computed using the features for the correctly classified training images. These are used to generate the transformation matrix for each of the codewords so as to recover the optimised descriptor vectors. This treatment allow for both classification on a per-codeword basis and upon the concatenated feature vectors.

Using the same method, classifier fusion is performed by recovering the transformation matrix for the SVM classifier-bank trained upon the optimised features. Here, for the SVM training and testing we use LIBSVM [46] set so as to generate probabilistic estimates for each class at output. In the fusion step, we recover the transformation matrix for these probabilites. The transformed probabilities are then used as input to a further application of a SVM classifier, whose output is taken as the final classification result in our experiments.

It is worth noting that, in our experiments we have explored the use of both the L_2-norm in Equation 3 and the L_1-norm in Equation 7 for learning the optimal transformation matrix. In practice, we found L_1-norm regulariser does not differ much from L_2-norm regulariser. With

Figure 1: Sample images for the Oxford flowers database.

regard to the computational complexity, the L_2-norm regulariser is much more computationally efficient than the L_1 counterpart.

4.1 Object Classification

The first of our experimental vehicles is the Oxford flower dataset [6], which contains flower images of 17 species against different backgrounds. For our experiments, we divide the dataset into a training set of 680 images, and testing and validation sets with 340 images each. Sample images from the database are shown in Figure 1.

We commence by extracting the regions of interest (ROI) in the image. To do so, we follow the saliency extraction model by Itti *et al.* [47]. This model adopts a bottom-up strategy that decomposes visual input into component feature maps. For each feature map, saliency is extracted separately and then combined into a global map. Once the saliency maps are at hand, we compute an optimal cut-off τ for each image. We build on the method proposed in [48] and view τ as the optimal split arising from a binomially distributed set of univariate random variables which correspond to the saliency values for every pixel in the image. For classification, we use the regions of interest (ROIs) given by the binarisation of the saliency maps. Examples of saliency maps and their corresponding ROIs are displayed in Figure 2.

With these ROIs at hand, we proceed to recover from them the color, SIFT and texture outputs for purposes of training and testing. In Table 1, we show the testing results when descriptor combination is performed. In the table, we show classification performance with and without op-

Figure 2: Saliency map and ROI.

timisation for the three features, i.e. SIFT (S), colour (C) and texture (T). From the table, we can conclude that the feature optimisation process on single features can improve the performance of the system. The classification rate is clearly improved when the three descriptors are combined. Table 1 also suggests that from the performance viewpoint, the election of norm for the recovery of the transformation matrices is not a critical choice. This is somewhat deceiving since the optimisation task in the quadratic form is less computationally demanding. Note that the results in Table 1 also capture the application of the classifier fusion algorithm in Section 3.2. When optimisation on the classifier bank is applied, the L_1-norm optimisation can further boost the performance to $83.4 \pm 1.4\%$.

For the sake of comparing our saliency approach to the recovery of the region of interest with other detection methods, we have also implemented the algorithm in [6]. In contrast with the use of saliency as a means to recover ROIs, the alternative requires hand-labeling of foreground and background pixels so as to provide a suitable prior for the segmentation step, which is based upon graph cuts. For consistency with results reported elsewhere, we have computed the graph cut prior using the image segmentation ground truth provided by the authors [2]. With the priors at hand, we

[2]The segmented imagery can be found at http://www.robots.ox.ac.uk/ vgg/data/flowers/17/index.html

Our approach (Saliency)					
	C	S	T	C-S-T	Classifier Fusion
Raw features	63.9 ± 1.9	69.2 ± 2.3	55.0 ± 3.5	81.5 ± 1.9	82.7 ± 1.7
L_1 Combined Feature	64.3 ± 1.7	69.7 ± 2.1	56.6 ± 3.2	82.9 ± 1.7	83.4 ± 1.4
L_2 Combined Feature	64.2 ± 1.5	69.3 ± 2.1	56.9 ± 3.1	83.1 ± 1.5	83.4 ± 1.4
Method in [6] (Graph Cuts)					
	C	S	T	C-S-T	Classifier Fusion
Raw features	63.0 ± 2.3	75.9 ± 1.7	57.0 ± 1.8	81.7 ± 2.3	82.2 ± 1.0
L_1 Combined Feature	63.7 ± 2.8	76.0 ± 1.6	58.3 ± 2.2	83.2 ± 2.3	83.4 ± 1.6
L_2 Combined Feature	63.9 ± 2.2	76.0 ± 1.7	58.1 ± 1.6	83.3 ± 2.2	83.3 ± 1.3

Table 1: Classification results for individual features, their combination and the application of classifier fusion, when saliency detection and graph cuts are used for the region of interest detection. Results show the percentage and standard deviation of correct classification using ten-fold cross validation.

apply the graph cut algorithm so as to recover suitable ROIs, which can then be used to perform feature optimisation and classifier fusion. The results for the alternative is shown in Table 1. It can be observed that feature optimisation step has improved the classification performance. After classifier fusion, the performance of our saliency-based approach is comparable to the alternative, despite the fact that our approach does not require the computation of a prior for the segmentation step as needed by the method in [6].

Now, we turn our attention to the performance of the proposed method (feature combination and classifier fusion) as compared to that yielded by a number of alternatives which employ the same set of image descriptors as our method and, thus, provide a fair ground for comparison. In [6], the feature combination is cast as finding the optimal weights for individual features. These weights are used to combine descriptors and features in the codebook and are uniform across all classes. In [6], the weights are recovered through an intensive search on the validation set. Moreover, this method requires manual tuning of the weights for the descriptor combination. This contrasts with

Classification Method	Classification Rate
proposed method (L_1)	83.4 ± 1.4
proposed method (L_2)	83.4 ± 1.4
Bach [49]	77.8 ± 2.1
Nilsback [6]	81.3
Varma [7]	82.6 ± 0.3
Nilsback [50]	83.3 ± 1.4

Table 2: Performance comparison on Oxford flower dataset. Results show the correct classification percentage and standard deviation for ten-fold cross validation.

our method, where the weight for each component of the descriptor is automatically optimised. Note that, despite the manual tuning of the weights, the classification performance delivered by our method, as shown in Table 1, is better than that yielded by the algorithm in [6].

The method of Varma and Ray [7] also treats the feature transformation as a feature combination problem. Their method recovers domain specific descriptors by learning the trade-off between invariance and discriminative power of the classification system. Additionally, we also provide comparison with the performance yielded by the Block 1-norm method (MKL-Block) [49, 7], which employs a multiple kernel learning approach based upon regularisation. The comparison results are shown in Table 2. Note that our method achieved correct classification rates of up to 83.4%, which provides a margin of improvement over the alternatives. Thus, we can conclude that, nonetheless our method is devoid of intensive search schemes, our method can achieve classification rates comparable to other methods elsewhere in the literature. This can greatly reduce the burden of manual parameter tuning by computing the optimal image descriptor combination.

4.2 Image Categorisation

Now we illustrate the utility of our method for purposes of image categorisation. To this end, we use the ETH-80 dataset, which contains images from 8 classes. Each class contains 10 objects, for which there are 41 images of each from different views against the same background. Examples

Figure 3: Sample images for the ETH-80 dataset.

of the images in the dataset are shown in Figure 3. To test the effectiveness of our method, we have randomly divided the dataset into a training set and testing set, both of equal sizes. We then perform ten-fold cross validation on the datasets for parameter selection. In the testing stage, for purposes of evaluation, so long as a testing image is assigned a label with correct category, it is considered a correct categorisation result.

For the sake of consistency, we have recovered the features and visual code words as per our object classification experiments. For our categorisation experiments, we have tuned the SVM parameters via cross validation for optimum performance. Here, the feature transformation process commences by computing the classification results on the training set using the SVM classifiers. The results yield an expected feature vector from correctly classified images for each image class, i.e. a K-dimensional vector specifying the class conditional mean for the images in the training data. These expected feature vectors and the input image descriptors are then used to recover the transformation matrix. The transformation matrix is then used to transform the input descriptors so as to combine them for the testing phase. In this experiment, and since the background of the objects is not varying greatly across different images, we have not extracted ROIs but rather used the descriptors over the whole image.

	C	S	T	C-S-T	Classifier Fusion
Raw Feature	95.4 ± 0.4	57.6 ± 1.0	84.9 ± 0.8	95.7 ± 0.6	96.8 ± 0.5
Feature Combination (L_1)	95.7 ± 0.4	57.7 ± 1.1	84.9 ± 0.8	95.9 ± 0.6	97.1 ± 0.5
Feature Combination (L_2)	95.8 ± 0.4	57.7 ± 1.2	84.9 ± 0.8	95.9 ± 0.6	97.0 ± 0.4
Majority Voting	—	—	—	—	89.9 ± 0.9
Boosted SVM	—	—	—	—	95.8 ± 0.4

Table 3: Categorisation results for the ETH-80 dataset. The results show the correct categorisation percentage and standard deviation using ten-fold cross validation.

The results on feature optimisation are shown in table 3. Note that, for our categorisation experiments, the feature combination does not overly improve the categorisation performance. This suggests that, at the feature level, the codebook generated by clustering may be almost optimal. However, when classifier fusion is applied, the performance increases to 97.1 ± 0.5 for the L_1-norm and 97.0 ± 0.4 for the L_2-norm. Here, we also compare our results against competitive fusion methods. To this end, we have performed experiments using majority voting and AdaBoost [23]. For the majority voting, we have trained an SVM classifier per feature. For the AdaBoost, each of the weak learners operates on a different feature. As seen in table 3, majority voting delivers lower categorisation rates than our method or AdaBoost. Note that, despite delivering better results than majority voting, AdaBoost cannot outperform the performance of our optimised features of our classifier fusion approach. This may be due to overfitting in the AdaBoost training step.

5 Conclusions

In this paper, we have proposed a method to perform both descriptor combination and classifier fusion for object classification and image categorisation. The method presented here is quite general in nature and hinges on the recovery of the optimal transformation over a structured field. In this manner, we can cast the problem into a learning setting, in which the aim of computation is the recovery of the optimal transformation up to a cost function. Here, this cost function has clear links to MRF and is convex in nature. We have shown the the utility of our method for classification and categorisation making use of the Oxford Flower database and the ETH-80 dataset.

References

[1] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.

[2] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 370 – 377, 2005.

[3] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[4] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 524–531, 2005.

[5] P. Quelhas, F. Monay, J. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool. Modelling scenes with local descriptors and latent aspects. In *Proceedings of the IEEE International Conference on Computer Vision*, volume I, pages 883–890, 2005.

[6] M. E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1447–1454, 2006.

[7] M. Varma and D. Ray. Learning the discriminative powerinvariance trade-off. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2007.

[8] Y. Chen, J. Z. Wang, and R. Krovetz. Clue: Cluster-based retrieval of images by unsupervised learning. *IEEE Trans. on Image Processing*, 14(8):1187–1201, 2005.

[9] K. Sengupta and K. L. Boyer. Using geometric hashing with information theoretic clustering for fast recognition from a large cad modelbase. In *IEEE International Symposium on Computer Vision*, pages 151–156, 1995.

[10] A. Shokoufandeh, S. J. Dickinson, K. Siddiqi, and S. W. Zucker. Indexing using a spectral encoding of topological structure. In *Proceedings of the Computer Vision and Pattern Recognition*, pages 491–497, 1998.

[11] S. Winder and M. Brown. Learning local image descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[12] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *ACM International Conf. on Image and Video Retrieval*, pages 401–408, 2007.

[13] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, 2005.

[14] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. Journal of Computer Vision*, 73(2):213–238, 2007.

[15] N. Vasconcelos. On the efficient evaluation of probabilistic similarity functions for image retrieval. *IEEE Trans. on Informaiton Theory*, 50(7):1482–1496, 2004.

[16] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[17] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[18] Jun Zhou, Li Cheng, and Walter F. Bischof. Spatial-temporal modeling of interactive image interpretation. *Spatial Vision*, 22(5):455–472, 2009.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.

[20] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

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

[22] T.K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[23] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[24] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[25] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting, 1998.

[26] N. Cristianini, J. Shawe-Taylor, J. Kandola, and A. Elisseeff. On kernel-target alignment. In *Advances in Neural Information Processing Systems*, pages 367–373, 2002.

[27] G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

[28] J. Keuchel, C. Schnorr, C. Schellewald, and D. Cremers. Binary partitioning, perceptual grouping, and restoration with semidefinite programming. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(11):1364–1379, 2003.

[29] J. Keuchel. Multiclass image labeling with semidefinite programming. In *European Conference on Computer Vision*, pages 454–467, 2006.

[30] P.H.S. Torr. Solving markov random fields using semi definite programming. In *Intl Workshop on Artificial Intelligence and Statistics*, 2003.

[31] M.P. Kumar, P.H.S. Torr, and A. Zisserman. Solving markov random fields using second order cone programming relaxations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1045–1052, 2006.

[32] T. Cour and J. Shi. Solving markov random fields with spectral relaxation. In *International Conference on Artificial Intelligence and Statistics*, 2007.

[33] Y. Boykov and M-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In *Intl. Conf. on Computer Vision*, pages 105–112, 2001.

[34] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Neural Information Processing Systems*, 2003.

[35] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *20th Intl. Conf. on Machine Learning*, 2003.

[36] L. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[37] T. Davis. *Direct Methods for Sparse Linear Systems*. SIAM, 2006.

[38] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[39] A. K. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and ramdom walker which yields a new algorithm. In *Proc. of ICCV*, 2007.

[40] P. Huber. *Robust Statistics*. Wiley, 1981.

[41] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2002.

[42] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74, 2000.

[43] Bastian Leibe and Bernt Schiele. Analyzing appearance and contour based methods for object categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 409–415, 2003.

[44] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[45] R. O. Duda and P. E. Hart. *Pattern Classification*. Wiley, 2000.

[46] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. $http://www.csie.ntu.edu.tw/ cjlin/libsvm$.

[47] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI*, 20(11):1254–1259, 1998.

[48] N. Otsu. A thresholding selection method from gray-level histobrams. *IEEE Trans. on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[49] Francis R. Bach, Romain Thibaux, and Michael I. Jordan. Computing regularization paths for learning multiple kernels. In *Neural Information Processing Systems COnference (NIPS)*, 2004.

[50] M. E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008.