Geometric Mining: Scaling Geometric Hashing to Large Datasets

Andrew Gilbert Richard Bowden
University of Surrey, Guildford, GU2 7XH, United Kingdom
{A.Gilbert,R.Bowden}@Surrey.ac.uk

Abstract

It is known that relative feature location is important in representing objects, but assumptions that make learning tractable often simplify how structure is encoded e.g. spatial pooling or star models. For example, techniques such as spatial pyramid matching (SPM), in-conjunction with machine learning techniques perform well [13]. However, there are limitations to such spatial encoding schemes which discard important information about the layout of features. In contrast, we propose to use the object itself to choose the basis of the features in an object centric approach. In doing so we return to the early work of geometric hashing [18] but demonstrate how such approaches can be scaled-up to modern day object detection challenges in terms of both the number of examples and their variability. We apply a two stage process; initially filtering background features to localise the objects and then hashing the remaining pairwise features in an affine invariant model. During learning, we identify class-wise key feature predictors. We validate our detection and classification of objects on the PASCAL VOC'07 and '11 [6] and CarDb [21] datasets and compare with state of the art detectors and classifiers. Importantly we demonstrate how structure in features can be efficiently identified and how its inclusion can increase performance. This feature centric learning technique allows us to localise objects even without object annotation during training and the resultant segmentation provides accurate state of the art object localization, without the need for annotations.

1. Introduction

This paper presents an approach to learning geometric configurations of feature points that represent objects in weakly supervised data i.e. where no object location or bounding box is provided. Conceptually this can be thought of as combining ideas from Geometric Hashing with learning. The origins of Geometric Hashing can be traced back to the mid 1980's [14] and is based on the central premise that an object, consisting of simple geometric features, can be efficiently matched against a database of such features using hashing schemes. However, rather than just encoding features relative to object location, as is common in object centric approaches to recognition, we attempt to find patterns of features that are geometrically consistent and invariant to affine object transformation.

Given a set of training images containing an object, the best feature set would be the one that achieves the most consistent layout of features across all examples. Exhaustively searching all possible basis sets for such features would identify the subset attributed to the object. However, even if the location of the object is known, this quickly becomes an intractable search. For example, given dense features, 250,000 features may be extracted from a single image and taking all triplets of points as possible basis vectors would generate 1.6 x 10^{16} combinations. We propose a weighted random search method to efficiently identify key feature subsets. We identify the features and configurations that are unique to each class i.e. both representative of the class, but also discriminative against competing classes.

Unlike other feature encoding schemes [16], we propose a two stage process. Firstly, we encode feature pairs and use this to suppress background features. This identifies features largely consistent with the object and reduces the complexity of the next stage. In the second stage, feature pairs are combined into candidate constellations by identifying feature triples and encoding all remaining features to this origin. Even with this two stage process, the number of combinations remains large. Traditional machine learning approaches are not suitable hence we propose an efficient learning method to identify key feature predictors.

This paper shows a number of key contributions: 1) The use of a two stage geometric encoding of low level feature collections to provide a fine grain structural awareness of the foreground description of objects. 2) The introduction of a learning method to efficiently identify discriminative encoded features. 3) The ability to localize objects in images, without explicitly labelled annotation or previous spatial information at either training or testing stages.

2. Related Work

Object recognition is a major research area within computer vision. Intuitively there are two challenges; deciding *what* objects are contained in an image and *where* these objects are. Object localization is generally deemed a harder problem and as a result, many state of the art classification approaches don't infer object locations [33, 35]. Approaches that do localize [25, 7, 19] are generally implemented via a sliding window. To remove the sliding window limitations, there has been work in identifying the foreground and background areas, to allow improved training, through negative mining [29] and object centric pooling of features [27].

Another method related to our approach is through the encoding of relationships between the feature points directly. However, in a fully connected shape model [9], as the number of combinations increases, the computation increases relative to the power of the number of features. Therefore other approximations have been proposed such as star based models [10], as the reduced dependencies of this model allows for learning in $O(N^2P)$ time instead of $O(N^P)$. More recently, pairwise features have been used in a number of ways, including encoding the fixed structure of buildings to identify connected feature patterns between images [38], in action recognition [22], or food recognition [36], as they can simplify the feature space. Mining features from data was proposed by Quack [24] but limited to a fixed spatial grid around a feature. Fernando [11] also employed mining to identify frequently reoccurring features but not higher groupings. Higher order relationships are typically ignored due to the intractability of the search. We provide a search mechanism that could be employed in any feature detection scheme to find triplets of feature points that provide the affine invariant encoding of an object. Yao [37] modified Apriori data mining for grouplet based object recognition, however Apriori association rule mining requires a full pass over the training data, which limits its use when increasing numbers of classes or instances are used.

There are approaches to use localization information [8, 29, 5, 23], that use the localization results to boost the overall classification accuracy. For example, Feng [8] resizes each image to the same size and using a grid, learn a weight based location likelihood factor for each class.

Most recent work within the field of object recognition is exploring the use of deep convolutional neural networks [20, 17], with large improvements in recognition tasks. This includes implementing previous SIFT based localisation techniques, such as regions and spatial pyramids. R-CNN [12], extracts and weights around 2000 candidate region windows per frame via selective search to predict the location of objects. While the application of spatial pyramid pooling for CNNs has been proposed by He [13], both provide excellent performance at object detection. These indicate the power of CNNs in conjunction with spatial feature encoding; however our approach removes the rigid requirement of defined regions.

3. Motivation

As the literature has demonstrated, the use of structure in object recognition provides performance increases, over unstructured features. However, fixed grid spatial encoding [19, 24, 13] relies on training data to overcome variability in pose and appearance. To motivate and illustrate this limitation, Figure 1 shows two example images of cars from the PASCAL VOC dataset with interest points detected.

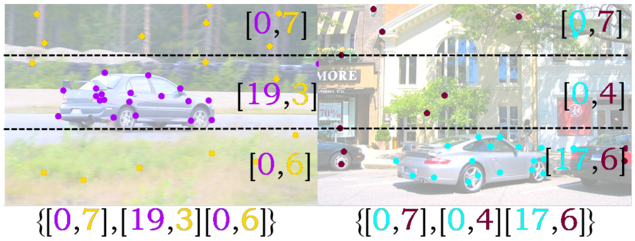

Figure 1. Encoding detected interest points in 2 images from PASCAL, encoded in a 3 way SPM, showing [FG,BG] features

A spatial pyramid could be used to encode the features (as in Figure 1), but the success of this approach is entirely dependent on the position and scale of the histogram bins. For a fixed encoding, the left image of Figure 1 learns that the car features occur in the middle bin, while the right image learns that car features occur in the lower bin. If no common features are found during learning, the classifier will fail at test time, therefore huge amounts of data are needed to generalise about the object layout.

In this work, we seek to achieve robustness to transformations of the spatial layout of features in the image by employing a spatial encoding process which identifies consistent relationships between feature points. The basic idea is shown in figure 2(a) where we see two images with constellations of features under some transformation. By choosing the correct basis vectors from the objects own set of features (the squares) and projecting all points onto these basis vectors (shown by short dotted lines in Fig 2(b)) we see which of the features are spatially consistent between the two examples (depicted by solid lines).

To do this we follow a two stage process. Initially we chose feature pairs and reject inconsistent features to roughly localise the objects. We then combine feature pairs to achieve affine invariance in the encoding. Initial pairwise encoding only provides invariance up to a similarity. However, at this stage a loose criteria is used to reject impossible configurations and reduce complexity at the second stage.

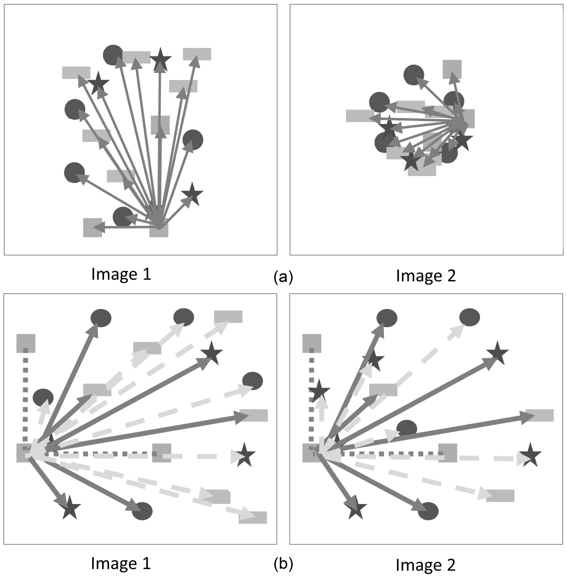

Figure 2. (a) Feature pair encoding for a single feature in two images. (b) Applying geometric hashing to both images. This transforms the feature space to a basis system, allowing similarity to be seen between the images, note the grey dashed lines indicate inconsistent features to be ignored

4. Structural Feature Descriptor

The two stage process results in a geometric hashing of features for an object class. The first stage is a pairwise hashing that intelligently filters non object features. Then in the second stage, the resulting pairs are combined into triplets to provide affine invariant encoding.

4.1. Stage 1 Encoding

In a weakly supervised setting, we do not know which features of an image are attributed to the object and which the background. The first stage therefore seeks to remove background features to reduce the features passed to the second stage. We propose to discretise the pairwise geometry of the images, to allow the geometric structure to be learnt. We define a geometric dictionary for stage 1 as Ω_1 of geometric words $\omega_1 \in \Omega_1$, where each geometric word represents a unique range in pairwise image space. The dictionary is formed over a four dimensional space, with each word defined by four pairwise geometries. The feature pair is invariant to scale, orientation and translation, based on the distance and angle between the two feature points. Given a set of features $\mathbf{K}_I$ extracted from image I and a feature pair consisting of feature κ_i and κ_j, where $\kappa_i, \kappa_j \in \mathbf{K}_I$ and $i \neq j$. Each feature has a scale κ_i^s, rotation κ_i^θ, position κ_i^{xy}, and appearance κ_i^{app} as a feature histogram. The pairwise geometry consists of the appearance of feature i, $\xi_i = \kappa_i^{app}$, the appearance of feature j, $\xi_j = \kappa_j^{app}$, the scale invariant distance $\delta_{ij} = \frac{(\kappa_i^{xy} - \kappa_j^{xy})}{|\kappa_i^s - \kappa_j^s|}$, and the absolute difference in rotation $\vartheta_{ij} = |\kappa_i^\theta - \kappa_j^\theta|$. The scale invariant distance is made invariant by being normalised by the feature scales, while the feature orientation is the absolute difference. Thus each pair of features κ_{ij}, is then encoded as

$$\kappa_{ij} = \{\xi_i, \xi_j, \delta_{ij}, \vartheta_{ij}\} \qquad (1)$$

Unlike previous work by Ta [30] who uses only the closest pairs, the feature relationships are formed between all pairs of interest points within an image. By looking for commonality across all examples, it is possible to learn which features are consistent for a class. This is used as an approximate object detector/filter to identify regions of the image where the object *may* be located and its constituent features. As such a coarse threshold rejects obvious outliers providing rough object localisation and the features are then passed to the second stage for more stringent affine encoding.

There have been many region based approaches proposed, such as objectness [2] and selective search [31]. However, our approach is able to learn an approximate object detector that has the ability to remove non object features, resulting in $\mathbf{V_I}$ filtered features, instead of a coarse bounding box based region of interest.

4.2. Stage 2: Building Constellations of Features

Given the initial image feature filtering through the pairwise features of section 4.1, a second stage of encoding is performed to provide affine invariance. This is achieved through the relative encoding of feature points, with respect to a coordinate system identified from a triplet of features.

To form a triplet of features, all feature pairs which share a common node in $\mathbf{K}_I$ are found, with respect to the filtered geometric pairs from the first stage $\mathbf{V}_I$

$$\mathbf{V}_I = \{i : (i,j) \in \mathbf{K}_I\} \cup \{j : (i,j) \in \mathbf{K}_I\} \qquad (2)$$

which gives all possible triplets as shown in equation 3.

$$\mathbf{Y}_I = \{\kappa(a,i,j) : a,i,j \in \mathbf{V}_I, \kappa(a,i), \kappa(a,j) \in \mathbf{K}_I\} \qquad (3)$$

For each triplet $\kappa(a,i,j)$ in $\mathbf{Y}$, all remaining features $r \in \mathbf{V}_I$ are encoded to the same basis set using κ_{ar}. The 2nd stage encoding is visualised in Figure 3. In this example the interest points a, i, j, r_1, r_2, r_3 are in the set $\mathbf{K}_I$, and the interest point pairs $\kappa(a,i)$ and $\kappa(a,j)$ are within $\mathbf{V}_I$ sharing the common interest point a. All remaining points r_1, r_2, r_3 are spatially encoded to the basis vectors $\vec{ai}$ and $\vec{aj}$. This process is repeated for every feature triplet within $\mathbf{V}_I$ and for every image within a class. Once the links between all features have been encoded, a class wise model is learnt.

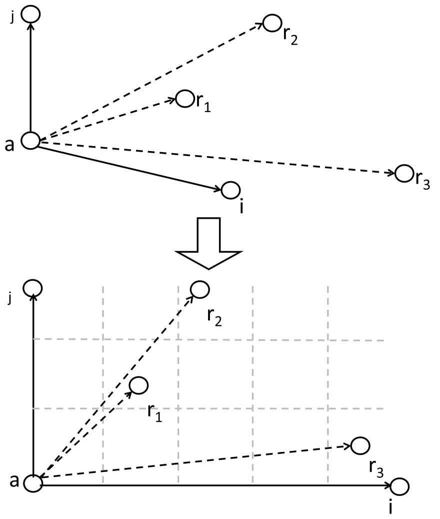

Figure 3. The visualisation of the 2nd stage feature encoding

5. Learning

The size of the feature sets at each stage of encoding can be prohibitively large; we therefore require an efficient method to identify the subsets that are both common across a class and discriminate against other classes.

Instead of modelling all the data, our learner identifies features that can provide the most impact or change on the dataset and this generally results in a simple feature set that is both distinctive and descriptive with respect to a class. In order to learn this set of features we employ mining. Two metrics are applied to the candidate features, *lift* and *support*.

The objective function or *lift* of a given feature or compound of features is a measure of change that the given features will make to the outcome class distribution compared to the baseline class distribution. With a set of training classes, $C = \{c_1, c_2, ..., c_\alpha\}$, and a large set of training examples E (in this case images), each example, $I \in E$ will contain a set of encoded features, $\mathbf{K}_I$. Each image is labelled with respect to a specific class c from the training data for example $\{\mathbf{K}_1, \mathbf{K}_{20}, \mathbf{K}_{24}...\} \longrightarrow c_1$ etc. Within the training examples, the frequency of the images labelled to a specific class is given by $\{F_1, F_2, ..., F_\alpha\}$ and is used as a normalisation factor. The overall aim of the learner is to identify the short feature subsets $\mathbf{V}_I$ that provide the greatest improvement in the overall class distribution. To achieve this, given the set of training classes, the frequency of the subset $\mathbf{V}$ occurring within each class is computed as $f_c^{\mathbf{V}}$. Ideally the subset will have a high frequency of occurrence in the positive class and low occurrence elsewhere and this will provide maximal lift. We define lift as

$$lift_c(\mathbf{V}) = \sum_{c=0}^{\alpha} \frac{U_c f_c^T}{F_c} \tag{4}$$

where U_c is a class weight, where

$$U_\alpha = \begin{cases} 1 & c = Positive \\ -1 & otherwise \end{cases} \tag{5}$$

If the lift is greater than 0, the subset of features is improving the input or baseline distribution of the class, i.e. the subset is more frequent in the positive class and less so in the negative classes. Ideally it will have a large lift, and this can achieved by making it more specific. This can cause over fitting, causing the unwelcome property of the subset of features only occurring in a very small selection of the positive class.

In order to avoid over fitting, the learner uses a threshold on its *minimum support*. This is the ratio of the frequency of occurrence of the subset of features within the positive class, with respect to the frequency of the occurrence of the subset in the rest of the dataset as shown in equation 6

$$support(\mathbf{V}) = \frac{f_{cp}^{\mathbf{V}}}{\sum_{c=0}^{C} f_c^{\mathbf{V}}} where\ c \neq cp \tag{6}$$

where *cp* is the positive class label. To avoid over fitting, a minimum support threshold of 0.2 is used, this ensures the proposed subset is representative of the input data.

The lift and support assesses the effectiveness of a subset, but the possible candidate subsets need to be generated. A naive approach would test all possible feature combinations. However, even for a single example image, this would be infeasible. Therefore we use a weighted random sampling to ensure we select the best single features to combine together and to reduce training time.

5.1. Generation of Subsets

To form the feature subsets, we propose an intelligent method that is weighted towards identifying and using subsets made up of single features that already represent the training class well. This ensures that it is more likely that when these representative features are joined to form compound subsets of features, that they will represent the training class well.

Initially, a random subset of individual features is selected, and these are scored by their individual lift. These lift scores are then converted into a cumulative distribution function. Λ subsets of the weighted random features are selected, with each containing up to M features (typically 200). These are scored and sorted with respect to their lift and support. Then a further Λ random subsets of individual features, formed of up to M features are then selected and weighted. This process is repeated for ρ iterations and the top A selected for the class. The learner is run independently for each class c using the training data, to produce a set of subsets for each class $M(c) = \{m(c)_1, m(c)_2, ...m(c)_A\}$.

6. Classification

Given the focus of the approach to produce concise representative feature subsets, the classification of images is implemented in a class specific hash table. To classify, the possible candidate features of a given image are encoded, and given the learnt model for each stage, each candidate feature in turn is compared to the entries in the hash table. The response score of the classifier R to a particular class label c is given by

$$R(T_i, c) = \frac{1}{K} \sum_{j=0}^{M(c)_K} \frac{1}{A} m(T_i, M(c)_j) \qquad (7)$$

where

$$m(T_i, M(c)_j) = \begin{cases} 1 & T_i \in M(c)_j \\ 0 & otherwise \end{cases} \qquad (8)$$

After the 1st stage, the feature pairs that match to the learnt subsets for a specific class are identified and used to compute the geometric hash of the 2nd stage. In the case of the classification of test images, the response score is repeatedly computed over all class labels, and the maximum response classifies the unseen image with that class label. As the learner identifies a small subset of features efficiently encoded in a hash table, this is very fast to compute for each image.

7. Detection of Objects

Given the exhaustive nature of the pairwise and geometric encoding and its inherent spatial awareness, it is possible to localise the classified object. Figure 4a shows an example classified feature pairs after the 1st stage of our approach. It can be seen in Figure 4b and c, that the relationships of the interest points, encode the structural information of the object. We use these spatial locations of the resulting interest points, from the maximal class label, to provide the initial seed location of the object within the image. The image is initially segmented using a water shed algorithm [32], and the nodes for the graphs are the resultant segmented regions. The foreground seed locations are given by the centre of the feature interest points from the classified pairwise relationships. These pixel locations are represented by small watershed regions and the image segmented via the GrabCut algorithm [26]. Figure 4d shows the resulting segmentation of the object. By maximising a rectangular bounding box to cover the segmented foreground mask, localization of the object is possible, without the introduction of any explicit spatial annotation during training.

8. Experiments

We validate our approach on three different sets of images, the challenging PASCAL07 dataset, PASCAL VOC2011 [6] and fine grained CarDb [21]. The CarDatabase (CarDb) consists of 13,473 photos of cars labelled with the model year from 1920 -1999. The dataset is defined using a 70/30 train and test split as used by [21]. The cars have a low inter class variation with only small areas of contrasting features, examples of the images through the years can be seen in Figure 5.

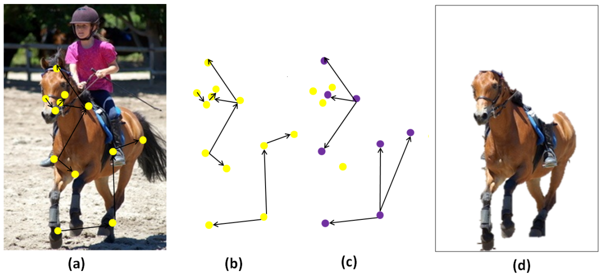

Figure 4. a) The detected pairwise relationships, for the object class horse, b) The structure of the object is visible based on the 1st stage pairwise relationships, c) After the 2nd stage, the features are able to encode more of the structure of the horse d) Interest points, are used as foreground seeds, to solve a graph based segmentation of the desired object

8.1. Implementation

To represent the images, two feature types are applied and compared, a colour invariant SIFT descriptor, C-SIFT [1] and a densely detected region based CNN. This is to indicate the feature agnostic ability of the pairwise encoding. The densely detected colour invariant SIFT descriptor, C-SIFT [1], is used in a softly assigned codebook of size 1024, and for each image around 250,000 pairwise features are encoded at the 1st stage and 60,000 are generated at the second 2nd stage. The CNN features are extracted from a deep CNN model pre-trained with the ImageNet dataset. We extract features from the sixth layer of the network which has the same architecture as that proposed by [17], and won ILSVRC2012. Because deep CNN-based features are extracted from the network, which is trained for recognition tasks, we can regard it as a feature that expresses discriminative information of an image. In our tests, we use the Caffe implementation [15]. We use this 4096 feature response rather than a codebook. The CNN regions are 25x25 pixels and are densely sampled with a 3 pixel overlap on the image, with around 220,000 feature pairs per image at the 1st stage and 50,000 pairs at the 2nd.

In order to provide discrete symbols for the encoded features, each feature histogram element from both the CNN response and C-SIFT codebook response is used to form the same number of new but unique symbols as the frequency. Therefore, given the sample frequency histograms, $H_1 = \{3; 0; 1\}$ and $H_2 = \{1; 3; 2\}$, the resulting symbolisation will be $H_1 = \{A1; A2; A3; C1\}$ and $H_2 =$

$\{A1; B1; B2; B3; C1; C2\}$. While computing the encoded stage 1 pairwise features in equation 1, the scale invariant distance δ_{ij} value is rounded to an integer value and in the case of C-SIFT features, the feature's rotation used in computing the absolute difference in rotation ϑ_{ij}, it is quantized into 1 of 12 equal sized radial polar bins. In the case of the CNN feature's, the absolute difference in rotation is set to 1, as it is unused.

8.2. CarDatabase Year Classification

To indicate the performance of our two stage pairwise feature encoder, we test on the CarDb dataset with an aim to indicate the year of a car present. Training data provides 10 year categories for cars, examples of these classes are shown in Figure 5. The mean error of the classified year for both our C-SIFT and CNN Feature Relationship data Mining (FRM) after stage 2 is compared to a linear SVM (BOW) with and without SPM spatial pooling (SPM), discriminatively mined sub patches [28] and visual changes over time [21].

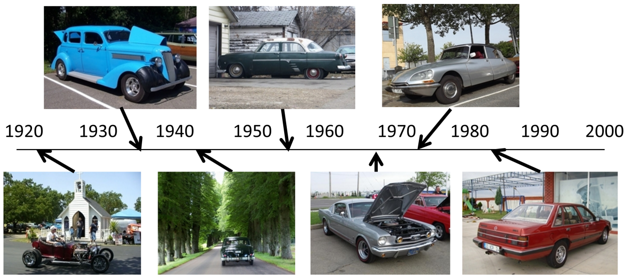

Figure 5. Examples of the CarDb dataset with class label

Approach	Mean Error (years)
BoW	15.39
SPM	11.81
Singh [28]	9.72
lee [21]	8.56
C-SIFT-FRM Stg 2	8.12
CNN	11.1
CNN-FRM-Stg2	6.5

Table 1. Mean error of classified year on the CarDb dataset.

The table shows the excellent performance of our approach especially with the CNN features. C-SIFT results can be compared to other approaches using SIFT features, with a 2.5 year reduction in the predicted car age error over SPM and 7 year reduction in error for a BoW approach using the same C-SIFT features. The CNN features reduce this error metric a further 2 years to a mean error of just 6.5 years. Most of the other approaches (BoW, SPM, Singh)have no mechanism to explicitly model the stylistic feature differences, instead trying to model the overall image for the class label and thus resulting in a loss of the fine detail necessary to learn the age of the cars. The work by lee [21], does model appearance based changes, but an interesting point is that they don't use a spatial pyramid on this dataset, despite using one on other datasets within their paper. This is likely to be due to the larger variation in the positioning of the cars in the photos as shown in Figure 5. This indicates the importance of feature hashing through pairwise relationships.

To provide qualitative results, Figure 6 shows both the resultant 1st and 2nd stage C-SIFT features that correctly classified the year of the car. It can be seen that after the 1st stage of the process, there is already far fewer features. Initially there are around 250,000 densely sampled interest points in each image, this is reduced to around 50 - 150 after the 1st stage, and reduces further after the 2nd stage to on average around 10. Furthermore the features that remain after the 2nd stage are in the areas that have the greatest contrast with other ages of cars, for example the wheels and headlights. This makes sense as visual features on other areas of the car or the background will have little discriminatory information and therefore are rejected by learning due to a low lift score.

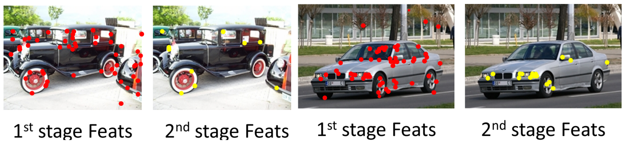

Figure 6. Resultant 1st and 2nd stage classified C-SIFT features for example images from the CarDB dataset

8.3. PASCAL Image Classification

The PASCAL07 [6] dataset's 20 object categories still provide a challenge for recognition and localisation due to the large amounts of clutter, scale change, viewpoint and environment variation. Results for a number of CNN and SIFT based BoW approaches are shown, together with our performance for both stages and both feature types in Table 2.

The table shows the use of our proposed approach Feature Relationship data Mining (FRM), with both CNN and SIFT features against baseline approaches (SPM), the fisher kernel feature encoding within an SPM and SVM framework, FK [3], and the recent CNN approaches of Chatfield [4], Wei [34] and CNN within a SPM framework [13]. It can be seen that our C-SIFT approaches compares favourably to other non CNN approaches, however when the densely sampled CNN features are used, performance improves significantly. Our approach, **CNN-FRM-Stg2** improves the classification on 13 out of the 20 object classes. This is impressive when you consider we do not use the labelled training annotations. Compared to other approaches, **CNN-FRM-Stg2** achieves improvement on the classes; boat, car, chair, and sofa, indeed the use of the feature relationships,

	type	aero	bike	bird	boat	bot	bus	car	cat	chr	cow	tble
SPM	sift	71.4	56.8	50.3	63.2	22.4	60.1	76.4	57.5	51.9	42.6	48.2
FK[3]	sift	79.0	67.4	51.9	70.9	30.9	72.2	80.0	61.4	56.0	49.6	58.4
SPP [13]	cnn	-	-	-	-	-	-				-	-
Wei [34]	cnn	95.1	90.1	92.8	89.9	51.5	80.0	91.7	91.6	57.7	77.8	**70.9**
Chatfield [4]	cnn	95.3	**90.4**	92.5	89.6	**54.4**	81.9	91.5	91.9	64.1	76.3	53.8
C-SIFT-FRM Stg 2	sift	77.1	65.3	43.1	67.3	28.8	67.2	80.2	58.3	53.4	48.8	53.8
CNN-FRM-Stg2	cnn	**95.6**	90.0	**92.9**	**91.3**	51.9	**82.8**	**92.9**	**92.5**	**64.7**	**78.7**	55.3

	type	dog	hrse	mbke	pers	plnt	shp	sofa	trn	TV	mAP
SPM	sift	36.9	75.3	62.8	82.9	18.2	37.1	43.3	69.4	50.9	53.9
FK[3]	sift	44.8	78.8	70.8	85.0	31.7	51.0	56.4	80.2	57.5	61.7
Wei [34]	cnn	89.3	89.3	85.2	93.0	**64.0**	85.7	62.7	94.4	**78.3**	81.5
SPP [13]	cnn	-	-	-	-	-	-	-	-	-	80.1
Chatfield [4]	cnn	**89.7**	**92.2**	86.9	**95.2**	60.7	82.9	68.0	95.5	74.4	82.4
C-SIFT-FRM Stg 2	sift	39.8	79.5	69.5	84.6	18.4	46.0	52.3	71.4	51.9	57.9
CNN-FRM-Stg2	cnn	**89.7**	91.6	**87.4**	95.1	60.7	**86.6**	**70.8**	**96.1**	76.4	82.3

Table 2. Classification of average precision of the 2 stage our approach (CNN-FRM-Stg2) compared to other approaches on the PASCAL VOC 2007 dataset

	aero	bike	bird	boat	bot	bus	car	cat	chr	cow	tble
DPM [7]	33.2	46.5	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7
R-CNN [12]	68.1	72.8	56.8	43.0	36.8	66.3	74.2	67.6	34.4	63.5	54.5
SPP-net [13]	68.6	69.7	57.1	41.2	40.5	66.3	71.3	72.5	34.4	34.8	61.7
C-SIFT-FRM-Stg2	34.3	41.8	14.2	17.4	28.8	59.2	62.4	25.5	24.2	19.7	30
CNN-FRM-NonTrained	75.9	75.3	54.0	44.3	40.7	67.1	72.4	68.1	29.4	40.0	54.8
CNN-FRM-Stg2	75.5	77.4	58.2	44.8	41.0	67.8	74.9	76.1	38.4	45.7	60.5

	dog	hrse	mbke	pers	plnt	shp	sofa	trn	TV	mAP
DPM [7]	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
R-CNN [12]	61.2	69.1	68.6	57.6	33.4	92.9	51.1	62.5	64.8	58.5
SPP-net [13]	62.3	71.0	69.8	57.6	29.7	59.0	50.2	65.2	68.0	59.2
C-SIFT-FRM-Stg2	12.0	65.7	58.6	41.4	15.6	29.9	31.0	46.0	47.2	35.2
CNN-FRM-NonTrained	58.7	73.4	59.9	58.4	29.0	56.1	50.8	60.8	69.9	56.9
CNN-FRM-Stg2	63.2	73.0	71.6	59.2	31.5	56.7	57.5	63.7	72.1	63.3

Table 3. Comparison of detection performance on PASCAL07

increases performance consistently over most classes, especially where the objects are smaller and often not centred or in fixed locations. The classes, sofa and chair can be difficult to classify as they are often heavily occluded and the improvements for these classes shows good invariance to occlusions due to the use of the encoded feature relationships, allowing the foreground structural relationships to be learnt and recognized. There are three class categories that proved especially difficult for **CNN-FRM-Stg2** to improve; bottle, plant, and table. These are complex classes in PASCAL07, with a high degree of occlusion and class variation, and these classes are challenging when even the labelled annotations are used [7, 13].

A key component of the learning is the weighted combination of encoded features. This allows the underlying geometric structure of the objects to be efficiently described in far greater detail, allowing the overall accuracy on the PASCAL07 dataset to increase from 45.3% (1st stage single pair relationships) to a maximum of 82.3% (2nd stage with feature groups containing up to 4 feature relationships) as shown in Table 5. For the 1st stage, this increasing accuracy plateaus with a maximum number of feature relationship of 73.5%. This plateau occurs due to the minimum support threshold in the learning, ensuring that it does not over fit to the training set. In the 2nd stage, a single feature relationship will encode the relationship between 4 separate interest points providing the higher initial accuracy of 56.7% and also the lower plateau in performance using only up to 4 feature relationships.

CNN-	Classification mAP (%)						
FRM	1	2	3	4	5	7	9
Stg1	45.3	62.4	65.5	71.3	73.5	73.5	73.5
Stg2	56.7	75.7	80.4	82.3	82.3	82.3	82.3

Table 5. The effect of increasing the maximum treatment size (M) on the interest point pairwise relationships CNN-FRM

	aero	bike	bird	boat	bot	bus	car	cat	chr	cow	tble	dog	hrse	mbke	pers	plnt	shp	sofa	trn	TV	mAP
R-CNN [12]	68.1	63.8	46.1	29.4	27.9	56.6	57.0	65.9	26.5	48.7	39.5	66.2	57.3	65.4	53.2	26.2	54.5	38.1	50.6	51.6	49.6
CNN-FRM-Stg2	85.4	72.4	23.4	38.5	60.4	75.4	58.1	58.4	9.1	47.9	35.4	48.2	39.7	60.4	55.8	33.1	60.1	28.7	55.8	44.9	49.6

Table 4. Performance on PASCAL VOC 2011

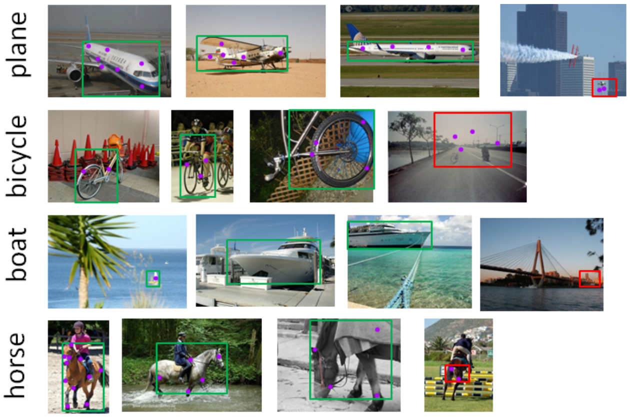

Figure 7. Object detections, learnt without any labelled annotation. Green boxes indicate positive detection, with false positive detections in red. Seed interest points are shown in purple

8.4. PASCAL Detection Results

The PASCAL07 dataset is challenging for localization, and our approach is able to both work with and without training annotation, we use the same localization accuracy criterion as [23], the window intersection over the union $\geq 50\%$. For an unseen image, the image is classified, and the features used as the seeds for segmentation. The maximum of the segmentation forms the outer limits of the bounding box, average classwise detection results for PASCAL VOC07 test set are shown in Table 3.

Our 2 stage approach using either SIFT and CNN features achieves 35.2% and 63.3% detection performance respectively. These are able to outperform the other related approaches, such as SPP [13] and R-CNN [12] as our approach uses relative pairwise hashes to encode the relative geometric structure, which is more invariant to changes in the object structure than a fixed region or window approach. Furthermore, we are able to localise the objects without using the training annotation boxes and this is shown in Table 3, as **CNN-FRM-NonTrained**, 56.9 this is an excellent result, considering that no annotation detail was provided in training. It is simply using the learning to identify unique feature combinations applicable to a particular object class. This is important for future work, as providing annotation for training images is expensive. Figure 7 shows examples of positive and negative detections.

Table 4 provides a breakdown of performance on the validation data of the more recent VOC2011 dataset compared against the R-CNN approach which uses similar features. The important point to note is that although the per class performance varies, the overall mAP score is almost identical. However, our approach uses no bounding box information during training to achieve this comparable result. It is also interesting to note that our approach tends to perform better for rigid objects where an affine assumption is more valid but less well for deformable objects that obviously break this assumption.

9. Conclusion

Our geometric feature mining approach is a flexible solution for the challenge of image detection. It employs a two stage learning and pairwise encoding of feature relationships invariant to the affine transformation of feature points and is agnostic to the features used. The use of a two stage process allows all interest points to initially be examined and to filter out background and non-descriptive features. While the 2nd stage encodes more complex affine invariant relationships between the remaining features, providing excellent performance on several challenging datasets. Importantly, without the use of any training annotation.

10. Acknowledgements

This work was supported by the EPSRC project, Learning to Recognise Dynamic Visual Content from Broadcast Footage (EP/I011811/1)

References

[1] A. Abdel-Hakim and A. Farag. Csift: A sift descriptor with color invariant characteristics. *In Proc of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2:1978–1983, 2006.

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. In *PAMI*, volume 34, pages 2189–2202, 2012.

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

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[5] T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. In *ECCV 2010*, pages 452–466, 2010.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes VOC Challenge . *In Proc. of IJCV*, 88:303–338, 2010.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. In *PAMI*, volume 32, pages 1627–1645, 2010.

[8] J. Feng, B. Ni, Q. Tian, and S. Yan. Geometric p-norm feature pooling for image classification. In *In Proc CVPR*, pages 2609–2704. IEEE, 2011.

[9] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *In CVPR 2003*, 2:II–264, 2003.

[10] R. Fergus, P. Perona, and A. Zisserman. A sparse object category model for efficient learning and exhaustive recognition. *In CVPR 2005*, 1:380–387, 2005.

[11] B. Fernando, E. Fromont, and T. Tuytelaars. Effective use of frequent itemset mining for image classification. In A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, editors, *Computer Vision ECCV 2012*, volume 7572 of *Lecture Notes in Computer Science*, pages 214–227. Springer Berlin Heidelberg, 2012.

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

[13] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *In ECCV 2014*, pages 346–361, 2014.

[14] W. H.J. and I. Rigoutsos. Geometric hashing: An overview. *IEEE Computational Science and Engineering*, 4(4):10–21, 1997.

[15] 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.

[16] L. Karlinsky, M. Dinerstein, and S. Ullman. Unsupervised feature optimization (ufo): simultaneous selection of multiple features with their detection parameters. In *CVPR 2009*, pages 1263–1270, 2009.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, pages 1097–1105, 2012.

[18] Y. Lamdan and H. J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *ICCV 2013*, pages 238–249, 1988.

[19] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR 2006*, pages 2169–2178, 2006.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[21] Y. J. Lee, A. A. Efros, and M. Hebert. Style-aware mid-level representation for discovering visual connections in space and time. In *ICCV 2013*, 2013.

[22] P. Matikainen, M. Hebert, and R. Sukthankar. Representing pairwise spatial and temporal relations for action recognition. *in ECCV 2010*, pages 508–521, 2010.

[23] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 1307–1314, 2011.

[24] T. Quack, V. Ferrari, B. Leibe, and L. Gool. "Efficient Mining of Frequent and Distinctive Feature Configurations". *In Proc. of IEEE International Conference on Computer Vision (ICCV'07)*, 2007.

[25] X. Ren and D. Ramanan. Histograms of sparse codes for object detection. In *CVPR 2013*, 2013.

[26] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (SIGGRAPH)*, volume 23, pages 309–314, 2004.

[27] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei. Object-centric spatial pooling for image classification. In *In Proc ECCV 2012*, pages 1–15. Springer, 2012.

[28] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV 2012*, pages 73–86, 2012.

[29] P. Siva, C. Russell, and T. Xiang. In defence of negative mining for annotating weakly labelled data. *In Proc ECCV*, pages 594–608, 2012.

[30] A.-P. Ta, C. Wolf, G. Lavoue, A. Baskurt, and J. Jolion. Pairwise features for human action recognition. In *ICPR*, pages 3224–3227, 2010.

[31] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013.

[32] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE transactions on pattern analysis and machine intelligence*, 13(6):583–598, 1991.

[33] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR 2010*, 2010.

[34] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan. Cnn: Single-label to multi-label. *arXiv preprint arXiv:1406.5726*, 2014.

[35] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR 2009*, pages 1794–1801, 2009.

[36] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar. Food recognition using statistics of pairwise local features. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2249–2256. IEEE, 2010.

[37] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 9–16. IEEE, 2010.

[38] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR 2011*, pages 809–816. IEEE, 2011.