Applied Soft Computing 46 (2016) 1056–1066

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Characterizing objects with SIKA features for multiclass classification

Siddharth Srivastava*, Prerana Mukherjee, Brejesh Lall

Indian Institute of Technology, Delhi, India

ARTICLE INFO

Article history:
Received 15 June 2015
Received in revised form
26 November 2015
Accepted 7 December 2015
Available online 23 December 2015

Keywords:
Object classification
SIKA
Minimal Complexity Machine
Support Vector Machine

ABSTRACT

This paper presents a novel approach for multiclass classification by fusion of KAZE and Scale Invariant Feature Transform (SIFT) features followed by Minimal Complexity Machine (MCM) as the classifier. Unlike the existing features, the paper proposes a new feature SIKA to represent characteristics of an object, as opposed to just forming a compendium of interest points in an image to represent the object characteristics. Further we append a strong and lightweight classifier, MCM to the technique. The resulting classifier easily outperforms existing techniques based on handcrafted features. Two new scores Keypoint Overlap Score (KOS) and Mean Keypoint Overlap Score (MKOS) have also been proposed as part of this work which are useful in establishing the strength of features for object representation.

1. Introduction

During the past few decades, object classification has remained an important research area for various real-world recognition applications like visual surveillance [1], object annotation [2], object tracking [3], obstacle avoidance/path planning [4] etc. In these tasks, the classifier has to handle the diverse nature of objects which makes it challenging to devise a single solution for all object classification problems. These challenges can be attributed to the following factors: number of classes, number of instances of each class, total number of images in the dataset, relative ratios of training and testing images, intra-class variance, ground truth annotations etc. Such solutions focus on two fundamental issues. First, the distinct characterization of objects of interest (features) and the second is the identification of objects (classification). Scale Invariant Feature Transform (SIFT) [5], Speeded Up Robust Features (SURF) [6], Histogram of Oriented Gradients (HOG) [7], Oriented and Rotated Brief (ORB) [8], KAZE [9] etc. have been widely used for solving the former issue. On the other hand, support vector machines (SVM) [10,11] have remained the most popular choice for the latter issue. Recently, Convolutional Neural Network (CNN) [12] has shown to outperform most of the traditional object classification and recognition benchmarks [13–15].

The techniques used for object classification can be broadly classified into three categories. The first set of techniques focus primarily on improving the input representation with the help of stronger features while using a simple classifier such as SVM. One such approach is linear Spatial Pyramid Matching based on Sparse Coding (ScSPM) [16] which uses sparse coding over vector quantization. It relaxes the cardinality constraints and introduces a regularization parameter to obtain a smaller number of non zero elements. This is then followed by max spatial pooling. It thus reduces the complexity of the classifier. In another related work [17], authors use a locality adaptor which allows to choose appropriate basis vectors corresponding to an input descriptor. Recently, authors in [15] perform experiments to demonstrate superiority of generic features extracted from CNN over handcrafted features for several recognition tasks. These features are based on the OverFeat [18] architecture. Later in this section we discuss that despite this overwhelming performance by generic features, there are gaps in the way these features are represented.

The second set of techniques focus on generating stronger training cases or using ensemble of classifiers. Stronger training cases allows the classifier to learn the peculiarities of the training set while a set of classifiers help in reducing bias by learning a more expressive representation. Authors in [19] illustrate this technique by formulating a latent SVM. It results in the problem being formulated as a convex training problem. The hard training examples are generated by using a feature similar to Histogram of Oriented Gradients (HOG). Recently, Regions with Convolutional Neural Network features (R-CNN) [14], a variant of CNN has been used to extract features from the region proposals. They perform

* Corresponding author. Tel.: +91 11 2659 1068.
E-mail addresses: eez127506@ee.iitd.ac.in (S. Srivastava), eez138300@ee.iitd.ac.in (P. Mukherjee), brejesh@ee.iitd.ac.in (B. Lall).

http://dx.doi.org/10.1016/j.asoc.2015.12.014
1568-4946/

supervised pre-training on a large dataset and then fine-tune this pretrained CNN on a relatively smaller target dataset. For this purpose they augment the strength of CNN by using category-specific SVM. These features are then classified into respective object categories.

Finally, the third set of techniques are those which try to balance the trade-off between classifier and feature strengths. In [20], authors propose a two stage sliding window approach for object localization. The main idea is to combine the classification and detection phases by considering latent properties of objects and scenes. Another technique, Selective Search [21] reduces the relative time for localizing objects by applying complementary grouping techniques for sampling. The reduction in localization is leveraged by constructing difficult negative examples to train the classifier.

These works however suffer from one or more of the following shortcomings: (a) The existing interest point based feature extraction techniques focus on characterizing content based on local information. Object features are not directly targeted by these techniques and are instead a consequence of image (and not object) interest points. The features themselves are not tuned to find or represent objects. Besides, there is very little understanding on how CNN extracts features. Moreover, recently authors in [22] showed that CNN can easily be fooled even with images that are easily identified as negatives by the human vision system. However, the authors also claim that the work provides insight into two key properties of the features from CNN. First, CNN extracts low and middle level features instead of high level features such as shape, boundary etc. Second, it learns patterns in the images which is the primary reason it was fooled. In our work we attempt to address this gap by proposing features which are representative of the characteristics of an object rather than them being a compendium of abstract representation of interest points. (b) Both CNN and SVM work well in practice, but there is no theoretical explanation of their generalization ability. This makes their use primarily based on experimentation. Moreover, CNN require huge training databases which is not usually available for many domains. We therefore use handcrafted features along with a classifier which is light-weight and guarantees generalization.

In this paper, we present a novel technique for generating a stronger feature set by using a combination of KAZE and SIFT keypoints termed as SIKA features (**SI**FT-**KA**ZE). We use these features with MCM to propose a light weight yet strong object classifier. The proposed scheme outperforms most of the existing state of the art methods. The SIKA features attempt to specifically characterize the object rather than obtaining a set of interest points. SIKA keypoints are constructed from SIFT and KAZE keypoints (described in Sections 2 and 4.1). SIFT [5] and its derivatives [23–25] show good invariance to several transformations. Since SIFT is based on Gaussian Scale Space (GSS), it inherently assigns equal importance to features on the object boundaries and to those inside it. The recently proposed KAZE [9] feature is based on non linear scale space. A useful property of KAZE is that it preserves the object boundaries as it blurs the region around edges more than the edges themselves. Therefore, it assigns more importance to features on and around the boundary. Hence SIKA features obtain a good mix of boundary (drawn from KAZE) and appearance (drawn from SIFT). The classification is performed using the recently proposed Minimal Complexity Machine (MCM) [26]. It has been shown to outperform SVM in terms of accuracy, computational complexity as well as providing sparse representation of the features. The strongest argument in favor of MCM is its provably good generalization accuracy and requirement of far lesser number of support vectors as compared to SVMs. Fewer support vectors mean faster classification of test points, and consequently due to complexity and size of the object classification datasets, MCM makes a strong case for itself. Since MCM is a recent technique for the benefit of the reader, a detailed discussion about MCM is given in Section 3.1.

The key contributions of this paper can be summarized as follows:

1. We propose SIKA features that characterize properties of an object. We also establish that SIFT and KAZE are complementary features. We show that a carefully chosen combination of these (as described in Section 4) boosts the classification accuracy significantly. We achieve state of the art performance on Caltech-256 dataset while close the gap to CNN based techniques on Pascal VOC 2007 dataset.
2. We show that Minimal Complexity Machine (MCM) achieves significant improvement in classification performance over the state-of-the-art work while using fewer number of training samples. We have also implemented a improvised version of MCM on GPU. To the best of our knowledge, this is the first work to demonstrate the effectiveness of MCM on images and datasets with large number of classes.

The rest of the paper is organized as follows. In Section 2, we motivate the use of KAZE and SIFT by introducing SIKA features. We explain MCM and propose its improvised version in Section 3. Section 4 describes the proposed methodology. In Section 5, we elaborate the experimental analysis and results while Section 6 concludes the paper.

2. Keypoint selection and description

Keypoint selection aims at finding a minimum set of features which helps in achieving maximum classifier performance based on certain metrics. It helps in getting rid of redundant features, resulting in simplification of the model and reduction in training time. In this work we achieve this by combining SIFT and KAZE interest points. In the following subsection, we describe how KAZE interest points can be added to complement the information represented by SIFT interest points.

2.1. Complementing SIFT

Recent studies [27,28] indicate that SIFT is the strongest feature detector available. As discussed in Section 1, SIFT focusses on appearance/ region of the entire object using high detail interest points (not necessarily boundary points) and KAZE concentrates on the boundary information. Therefore, we aim to complement the strength of SIFT with the KAZE features. The complementarity of SIFT and KAZE is due to differences in the generation mechanism of these features. The first difference is in the construction of the scale space. KAZE is based on non-linear scale space while SIFT is based on Gaussian scale space (GSS). KAZE uses non-linear diffusion filtering as given in Eq. (1) to construct the scale space.

$$\frac{\partial L}{\partial t} = div\{(c(x, y, t) \cdot \nabla(L))\} \tag{1}$$

where *div* and ∇ are divergence and gradient operators respectively, c is the conductivity function and t is scale parameter. The conductivity function c, is represented as a gradient (Eq. (2)), helping in the reduction of diffusion at edges, thus resulting in more smoothening of regions as compared to edges.

$$c(x, y, t) = g(|\nabla L_\sigma(x, y, t)|) \tag{2}$$

where ∇L_σ (luminance function) is the gradient of a Gaussian smoothed original image L where σ is the amount of blur. This property of the conductivity function makes KAZE suitable for boundary representation. There are various conductivity functions defined in

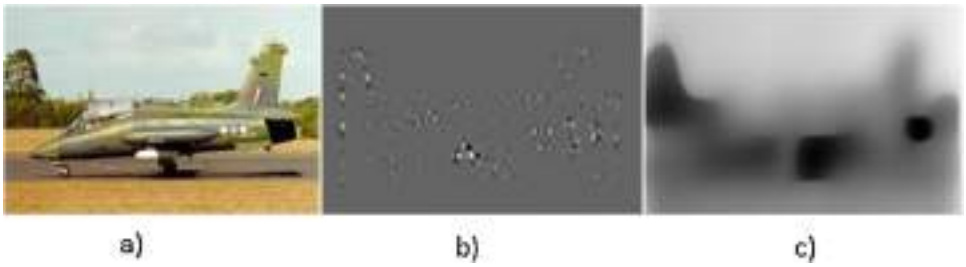

Fig. 1. (a) Original image, (b) KAZE detector response localized around the object (aeroplane) and (c) KAZE scalespace.

[29], which can be used to promote high contrast, wider regions or smoothening on both sides of the edges.

Another difference is that in SIFT, the base image for each octave is generated by downsampling the image from previous octave. On the contrary, the construction of each octave in KAZE is based on the original image. It can been observed in Fig. 1 that KAZE neither blurs out the object boundary in the detector response nor in the scale space. This helps in retaining the object boundary information at each octave and therefore contributes to the strength in obtaining boundary keypoints.

2.1.1. SIKA features

This section presents the motivation behind the proposed approach (Fig. 2) for combining SIFT and KAZE features, termed as SIKA features. The method for construction of SIKA features by selection of appropriate keypoints is provided in Section 4. SIKA features have been designed to capture the three key properties for defining an object as described in [30]: (a) representation of a defined boundary, (b) saliency and c) distinctiveness from background. The property (a) i.e. boundary representation is characterized using KAZE keypoints. KAZE features are more responsive to object boundaries as compared to other regions. The existing works do not provide a quantitative justification in support of this property. Therefore, an empirical evaluation of the effectiveness of KAZE in terms of boundary representation as compared to SIFT is provided in Appendix A. A visual depiction of KAZE and SIFT features is shown in Fig. 3. From Fig. 3(a), it can be observed that most of the KAZE keypoints are concentrated at the boundary. SIFT on the other hand looks for sharp discontinuities at all scales and hence captures keypoints in the entire region. It can also be observed (Fig. 3(b)) that SIFT gives a high number of keypoints in relatively less salient regions (like grass, clouds etc.) while KAZE keypoints are dominant around the most salient region boundaries (i.e. the object boundaries).

Property (b) i.e. saliency can be explained by the fact that salient regions are sparse as compared to other regions in the image [31]. This indicates that the keypoints in these regions are more capable of uniquely identifying an object. Since SIFT is more robust than KAZE in non boundary regions, SIFT keypoints have better response in these regions. Keypoints near the object boundaries play a crucial role in distinguishing objects against the background (Property (c) i.e. distinctiveness), therefore use of KAZE interest points helps address this requirement.

2.2. Feature representation

In order to extract a unique representation of images, the descriptors are encoded using various strategies. A comprehensive evaluation of such encoding techniques provided in [32] indicates that Fisher encoding is superior to all other strategies. The comparison also highlights that all the encoding strategies perform better than Bag of Visual Words (BoVW) approach. We now provide a brief description of these feature representation techniques.

2.2.1. Bag of Visual Words

The bag of visual words (BoVW) [33] approach is based on vector quantization of the image descriptors. The descriptors are clustered using a clustering mechanism (usually k-means) which is followed by computation of histogram of these quantized descriptors for each image in the dataset. In contrast to this hard quantization, recent methods either combine the visual words [34] or work on a differential scheme [35]. Despite the remarkable progress in feature encoding techniques, many recent works still use BoVW for evaluation due to its ease-of-use and availability of wide scientific work for comparison. A major drawback of this encoding scheme is its high computational complexity. Fisher vector is a suitable alternative and we briefly introduce it in the following subsection.

2.2.2. Fisher Vector

Fisher Vector (FV) [36] is a state-of-the-art patch encoding strategy. The main idea behind Fisher vector is to represent descriptors as a deviation from a generative model. This model is a Gaussian Mixture Model (GMM) in case of images. The deviation is defined as the gradient of the log likelihood of the local patch descriptors with respect to the GMM parameters i.e. weight, mean and covariance.

Mathematically, the Fisher vector for images is represented as,

$$\mathcal{G}_{\lambda}^{x} = \Sigma L_{\lambda} G_{\lambda}^{x} = \Sigma L_{\lambda} \nabla \log u_{\lambda}(x_t) \tag{3}$$

where G_{λ}^{x} is the gradient of the log likelihood of the data, L_{λ} which is obtained from the Cholesky decomposition of the inverse of the Fisher Information Matrix F_{λ} as $L_{\lambda} = \sqrt{F_{\lambda}^{-1}}$, $\lambda = \{w_k, \mu_k, \Sigma_k\}$ are

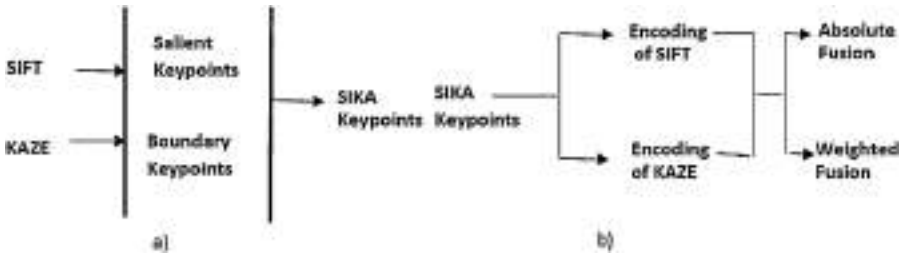

Fig. 2. (a) General idea of SIKA keypoints and (b) SIKA encoding.

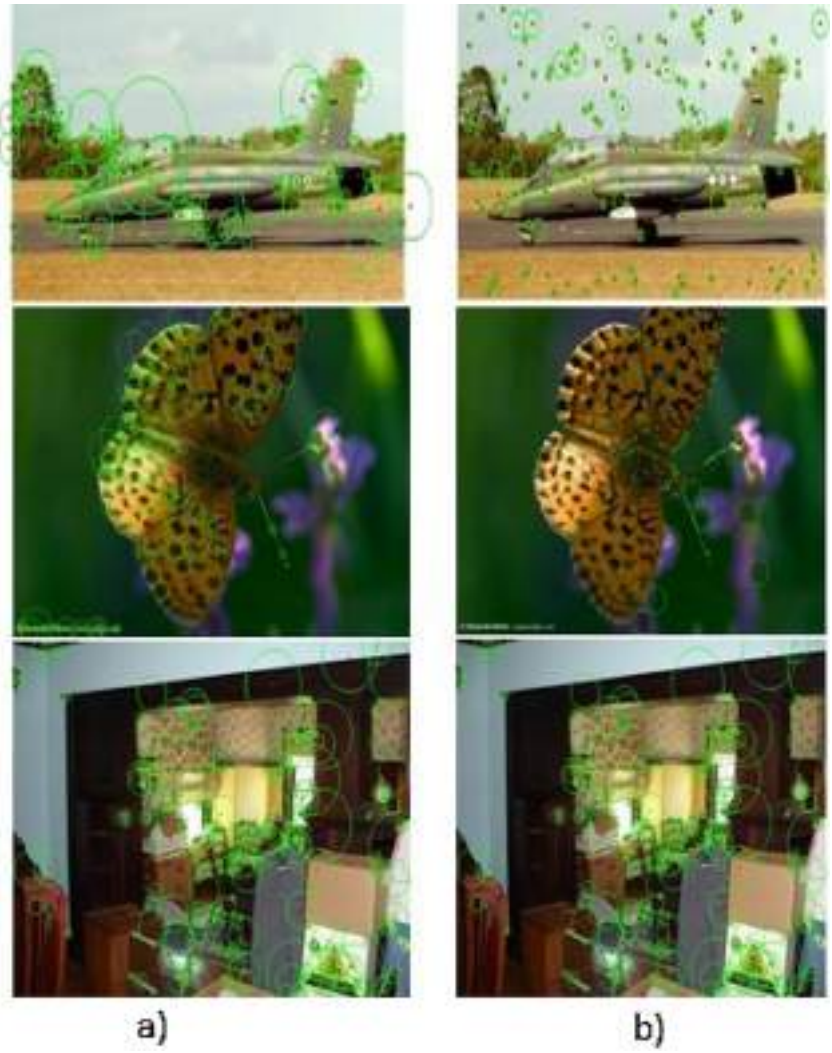

Fig. 3. (a) This shows the KAZE keypoints which are densely distributed along the object boundaries and (b) this shows the SIFT keypoints around the regions.

the parameters of the model μ_λ, u_λ is the GMM distribution and X gives the set of local patch descriptors $X = [x_1, \ldots, x_T]$, with the assumption that x_t are independent. An important feature of FV is that a linear classifier using these as feature vectors is equivalent to a non linear classifier using Fisher Kernel as the kernel. Thus, SVM and MCM being linear classifiers are a suitable choice for classification when using FV as the encoding scheme. MCM is discussed in the next section (Section 3.1) while performance of both the classifiers is evaluated in Section 5.

As compared to BoVW, FV has a comparatively lower computational cost as it requires far smaller vocabulary. FV uses GMM while BoVW uses hard quantization resulting in a less flexible model. BoVW is a particular case of FV, hence has a comparatively sparser representation. The dimensionality of FV is $(2D+1)K$ where D is the dimension of the descriptors and K is the number of clusters whereas the dimensionality of BoVW is equal to the number of cluster centers.

3. Classifier selection

In this section, we discuss and motivate the use of a recently proposed hyperplane classifier, minimal complexity machine [26] and provide a conceptual overview of it. We also draw comparisons with SVMs (Fig. 4) as they are the most popular choice of classifiers for object recognition tasks.

3.1. Minimal Complexity Machine (MCM)

MCM is a hyperplane classifier which guarantees a good generalization by minimizing the bound on Vapnik–Chervonenkis (VC) dimension [37]. VC dimension is a measure of the complexity (capacity) of a set of classification functions. VC dimension (γ) is bounded as follows,

$$\gamma \leq 1 + \min\left(\frac{R^2}{d_{\min}^2}, n\right) \tag{4}$$

where, margin $d \geq d_{min}$, R is the radius of the smallest enclosing sphere that contain all training data points. The aim is to minimize γ. One way to achieve this is to pose the optimization problem as a fractional programming problem. Since MCM extends the basic idea of SVM which is to find a maximum margin classifier, the MCM formulation introduces a variable h related to γ as follows [26],

$$\alpha h^2 \leq \gamma \leq \beta h^2 \tag{5}$$

where, h is the ratio of the maximum distance of the points from the separating hyperplane to the minimum distance of the points i.e. $\frac{R}{d_{\min}}$, and α, $\beta > 0$ are constants. Eq. (5) implies that h is an exact bound over VC dimension. Therefore, the optimization problem reduces to minimizing h^2 which further reduces to minimizing h.

Minimal Complexity Machine (MCM)	Support Vector Machine (SVM)
Given training data X of N samples with labels Y where $x_i \in R^d$, $y_i \in \{-1,+1\}$, for $i = [1,2,..,N]$.	
MCM Formulation: $\min_{w,b,q,h} h + C\sum_{i=1}^{N} q_i$ s.t. $h \geq y_i(w^T x_i + b) + q_i$ $y_i(w^T x_i + b) + q_i \geq 1$ $q_i \geq 0$	SVM Formulation: $\min_{w,b,q} \frac{1}{2}\|\|w\|\|^2 + C\sum_{i=1}^{N} q_i$ s.t. $y_i(w^T x_i + b) + q_i \geq 1$ $q_i \geq 0$
· Solves a Linear Programming Problem (LPP) · Gives fewer support vectors (lower memory requirement). · Bounded VC dimension (low complexity classifier).	· Solves a Quadratic Programming Problem (QPP) · Higher number of support vectors. · No proven bounds on VC dimension.

Fig. 4. Comparison between properties of MCM and SVM.

For a hyperplane represented by set of linear equations $u^T x + v = 0$, the optimization problem becomes,

$$Min_{u,v} h = \frac{Max_{i=1,2,\ldots,M} y_i(u^T x^i + v)}{Min_{i=1,2,\ldots,M} y_i(u^T x^i + v)} \tag{6}$$

This is a linear fractional programming problem, it can be further reduced by the Charnes–Cooper transformation [38] to,

$$Min_{w,b,h} h \tag{7}$$

$$h \geq y_i \cdot \left[w^T x^i + b\right], \quad i = 1, 2, \ldots, M \tag{8}$$

$$y_i \cdot \left[w^T x^i + b\right] \geq 1, \quad i = 1, 2, \ldots, M \tag{9}$$

where $w \in \mathbb{R}^n$, b, $h \in \mathbb{R}$. When the input samples are mapped using a kernel function $\Phi(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $m > n$ the optimization problem can be formulated as,

$$Min_{w,b,h,q} h + C \cdot \sum_{i=1}^{M} q_i \tag{10}$$

$$h \geq y_i \cdot \left[w^T \Phi(x^i) + b\right] + q_i, \quad i = 1, 2, \ldots, M \tag{11}$$

$$y_i \cdot \left[w^T \Phi(x^i) + b\right] + q_i \geq 1, \quad i = 1, 2, \ldots, M \tag{12}$$

$$q_i \geq 0, i = 1, 2, \ldots, M \tag{13}$$

These equations imply that MCM guarantees good generalization i.e. low error rates. In the original paper [26], the authors claim that MCM produces far less number of support vectors as compared to SVM while also achieving significant improvements in execution time. A point to note is that those experiments have been performed on datasets from the UCI machine learning repository [39], which contains a maximum of 6 classes, 12,626 features and 1567 samples in different datasets. Review on recent literature shows that MCM has not been evaluated on the datasets that represent the complexity associated with typical image datasets (large number of classes, high dimensional features and high number of keypoints). To the best of our knowledge, this is the first work to evaluate MCM on image datasets. We have evaluated both the linear and kernel versions of MCM (Section 5). We observe that the kernel MCM requires extremely high training time for these datasets. In order to reduce the training time, we have developed an improvised implementation of MCM, details of which follows next.

3.1.1. Improvised implementation of MCM

We observe that as the MCM linear programming formulation in Eqs. (6)–(13) requires two constraints for each data point, the training time for MCM involving large datasets is high. Moreover, in experiments involving Fisher vector, the high dimensionality of Fisher vector was found to be a major computational bottleneck. To circumvent these problems, we implemented an improvised version of MCM.

Our implementation addresses the following specific computational bottlenecks:

- Computation of the constraint matrix which in turn requires computation of kernel matrix (linear or non linear).
- Solving the linear programming problem (LPP).

A key observation for experiments involving Fisher vector was that they were consistently sparse for SIKA features. So instead of storing the complete matrix of Fisher vectors, we store the sparse representation in the memory. As the computation of the constraint matrix required multiplication of high dimensional vectors, we delegate this operation to GPU.

Secondly, complexity of the MCM LPP becomes high since the number of constraints is directly proportional to the number of data points. To address this, we use Mosek Optimization Toolbox [40] for solving the LPP. Mosek achieves this by using the following steps: Presolve, Dualizer, Scaling and Optimize. We disabled the eliminator of the Presolve step in our implementation since as per the formulation discussed earlier, there are no free variables. We also disabled the Dualizer step as MCM solves the primal problem. Also, we used interior-point algorithm to solve the linear programming problem. With the above modifications to the implementation, we achieved approximately 58% computational speedup as compared to the original implementation of MCM for training the classifier.

4. Proposed methodology

The basic workflow of our technique is shown in Fig. 5. The various steps are described in the following subsections.

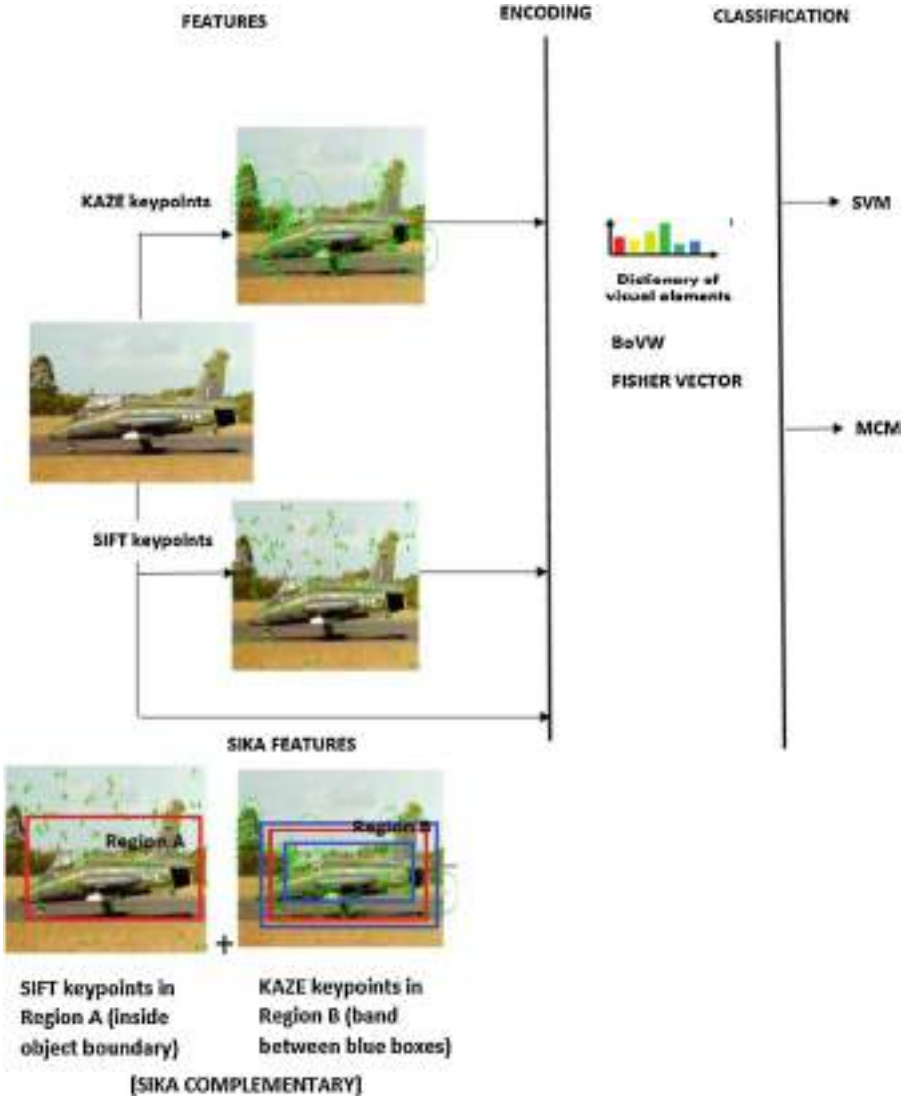

Fig. 5. Workflow of the proposed methodology.

4.1. Selection of SIKA keypoints

In this subsection, we describe two approaches for selecting SIKA keypoints. First approach for generation of SIKA keypoints is to select all keypoints from SIFT and KAZE detectors. We term these keypoints as *SIKA ALL* (Eq. (14)).

$$SIKA \quad ALL_{keypoints} = SIFT_{keypoints} \cup KAZE_{keypoints} \tag{14}$$

The second approach for selection of SIKA keypoints is to use the complementarity of SIFT and KAZE keypoints (Eq. (15)) to specifically represent an object. We term these keypoints as *SIKA Complementary* and given as,

$$SIKA \quad Comp_{keypoints} = SIFT_{keypoints(object)} \cup KAZE_{keypoints(boundary)} \tag{15}$$

The *SIKA Complementary* keypoints consist of (a) the KAZE keypoints along the boundary of the object. The keypoints are selected in the band of region around the object's bounding box. The detail of how the bounding box is defined is given in Appendix A. (b) SIFT keypoints which lie strictly inside the bounding box of the objects.

4.2. Encoding

The SIKA keypoints are described using the respective descriptors and encoded using Fisher vector or BoVW. Let the corresponding encoding of SIKA, SIFT and KAZE keypoints be represented as E_{SIKA}, E_{SIFT} and E_{KAZE}, respectively. Suppose, K is the number of words for Fisher vector or BoVW encoding.The final encoding can be generated using either the simple fusion (Eq. (16)) or weighted fusion (Eq. (17)).

$$E_{SIKA} = (\boldsymbol{E_{SIFT}}, \boldsymbol{E_{KAZE}}) \tag{16}$$

$$E_{SIKA} = w_{SIFT} \cdot \boldsymbol{E_{SIFT}} + w_{KAZE} \cdot \boldsymbol{E_{KAZE}} \tag{17}$$

where the weights are defined as $w_{SIFT} = \frac{\sigma^2_{E_{SIFT}}}{\sigma^2_{E_{SIFT}} + \sigma^2_{E_{KAZE}}}$ and $w_{KAZE} = \frac{\sigma^2_{E_{KAZE}}}{\sigma^2_{E_{SIFT}} + \sigma^2_{E_{KAZE}}}$ and σ^2 is the variance.

The classification is then performed using MCM and we compare our results with those achievable by SVM. The results are discussed in the next section.

5. Experiments and results

5.1. Experimental Setup

The experiments were performed on a machine with 32GB RAM, Xeon 1650 processor and 1GB NVIDIA Graphics Card. Matlab 2014a was used as the programming platform. We used the libSVM [41] implementation of SVM. For calculating Fisher vector we used VLFeat Toolbox [42]. The datasets used for evaluation of the proposed methodology are Caltech-256 [43] and Pascal VOC 2007 [44]. In the following subsections, we discuss the results and compare them with other state of the art methods.

5.2. Caltech-256

In the experiments we evaluated SIFT, KAZE and the proposed SIKA features using SVM and MCM. On the Caltech-256 dataset, we represent the features as both BoVW and FV. The vocabulary size for BoVW is 512 while that for FV is chosen as 256. These are then provided to SVM and MCM for classification. As already mentioned in Section 2.2, BoVW is comparatively weaker and computationally expensive as compared to FV. Moreover, most prior art which uses Caltech-256 is based on Bag of Visual Words. Therefore, for a fair comparison with prior works and to better establish the superiority of the proposed SIKA features, we have provided results on BoVW also. We have used the one-vs-one approach for multiclass classification. Moreover, as the patterns represented by SIFT features are linearly separable [16], we have chosen a linear kernel for classification. We have found that the patterns represented by KAZE features are also linearly separable, since our experiments with a non linear (RBF) kernel consistently performed worse than those with a linear kernel. The results are shown in Table 1 for SIFT, KAZE and *SIKA ALL* features. We have provided evaluation with *SIKA ALL* features as Caltech-256 dataset does not contain bounding box annotations for computing *SIKA Complementary* features. In addition to classification accuracy, we also compare the number of unique support vectors required by SVM and MCM. We use unique support vectors because in one-vs-one approach, a training sample could belong to multiple binary classifiers as a support vector. Fig. 6 compare the number of unique support vectors found with MCM and SVM, demonstrating that MCM consistently finds 30-60% fewer support vectors than SVM. Table 2 shows the performance of the state of the art technique on Caltech-256 dataset. As can be seen that our proposed method beats the current state of the art method of CNN using ImageNet pretrained [45] by 4.64% for 15 training images/class while also outperforming it for 60 training images/class.

Table 1
Classification accuracy for MCM and SVM for SIFT, KAZE and SIKA ALL features on Caltech-256 dataset.

Training samples	SIFT			
	MCM		SVM	
	BoVW	FV	BoVW	FV
15	52.79	65.12	19.82	30.03
30	55.08	67.02	26.82	31.89
45	56.45	68.67	28.98	32.33
60	57.20	69.45	30.91	34.56
Training samples	KAZE			
	MCM		SVM	
	BoVW	FV	BoVW	FV
15	51.83	57.65	18.24	27.45
30	52.00	58.70	21.08	28.09
45	52.70	59.23	22.86	29.48
60	52.90	61.52	24.23	30.73
Training samples	SIKA ALL			
	MCM		SVM	
	BoVW	FV	BoVW	FV
15	56.93	70.34	26.86	29.05
30	57.13	72.08	34.92	38.55
45	58.68	74.69	38.95	39.05
60	59.66	75.81	42.60	45.55

5.3. Pascal VOC 2007

Pascal VOC 2007 dataset is more challenging than Caltech-256 as it consists of multiple objects in an image with varying degrees of complexities.

We extract *SIKA ALL* and *SIKA Complementary* features for the training images in the dataset. The *SIKA Complementary* features use the ground truth bounding box annotations of the Pascal VOC 2007 dataset. We extract 128 dimensional descriptors for SIFT and KAZE keypoints. These descriptors are then reduced to 80 dimensions using PCA as suggested in [32]. The descriptors are then encoded using the Fisher vector with 256 words. Since the dimensionality of the Fisher vector is high, we use the weighted fusion of the SIFT and KAZE encodings to generate the SIKA encoding. The results in terms of Mean Average Precision(mAP) are shown in Table 3. The number of support vectors required by MCM and SVM are shown in Table 4. It can be seen that *SIKA All* with MCM significantly outperforms other combinations of features and classifiers. It can also be noted that MCM requires fewer support vectors than SVM; and with *SIKA All* resulting in minimum number of support vectors.

Table 5 compares results from various contemporary works on Pascal VOC 2007 dataset. The *SIKA ALL* with MCM significantly outperforms the works using SVM and SIFT [47,48,17,35] and is very close to the CNN based approach of Zeiler & Fergus [45] while having much lower complexity.

5.4. Discussions

The results demonstrate that *SIKA ALL* along with MCM outperforms state-of-the-art techniques involving SVM for both the considered datasets. *SIKA ALL* with Fisher vector and MCM outperforms the CNN based state of the art technique on Caltech-256 dataset. On Pascal VOC 2007, *SIKA Complementary* features with SVM and MCM perform marginally better than SIFT with SVM and MCM respectively. This observation is important for two reasons. Firstly, this indicates that *SIKA Complementary* features are at least as discriminative as SIFT features for object representation. Secondly, it achieves this despite the fact that the number of *SIKA Complementary* keypoints is significantly lesser than the number of SIFT keypoints for an image. This results in faster computations as compared to using the complete set of keypoints from SIFT detector. As can be observed, *SIKA All* with MCM give highest mAP amongst all combinations of SVM, SIFT and KAZE (Table 3). In case of Caltech-256, the weighted mixture of SIFT and KAZE (Table 1) outperforms the other two (SIFT and KAZE individually) approximately by 6-14% for MCM and around 10%-15% for SVM. This further strengthens the claim that SIFT and KAZE are complementary features and can effectively define an object within an image. This can also be understood by observing the fact that while KAZE effectively incorporates the boundary characteristics, the chosen SIFT keypoints capture the region properties. It is also important to reiterate that the contemporary works achieving state of the art performance (Tables 2 and 5) using SVM, performed strong pre-processing on the features or were trained with specifically constructed hard negatives from the training examples whereas in this work, we have used the

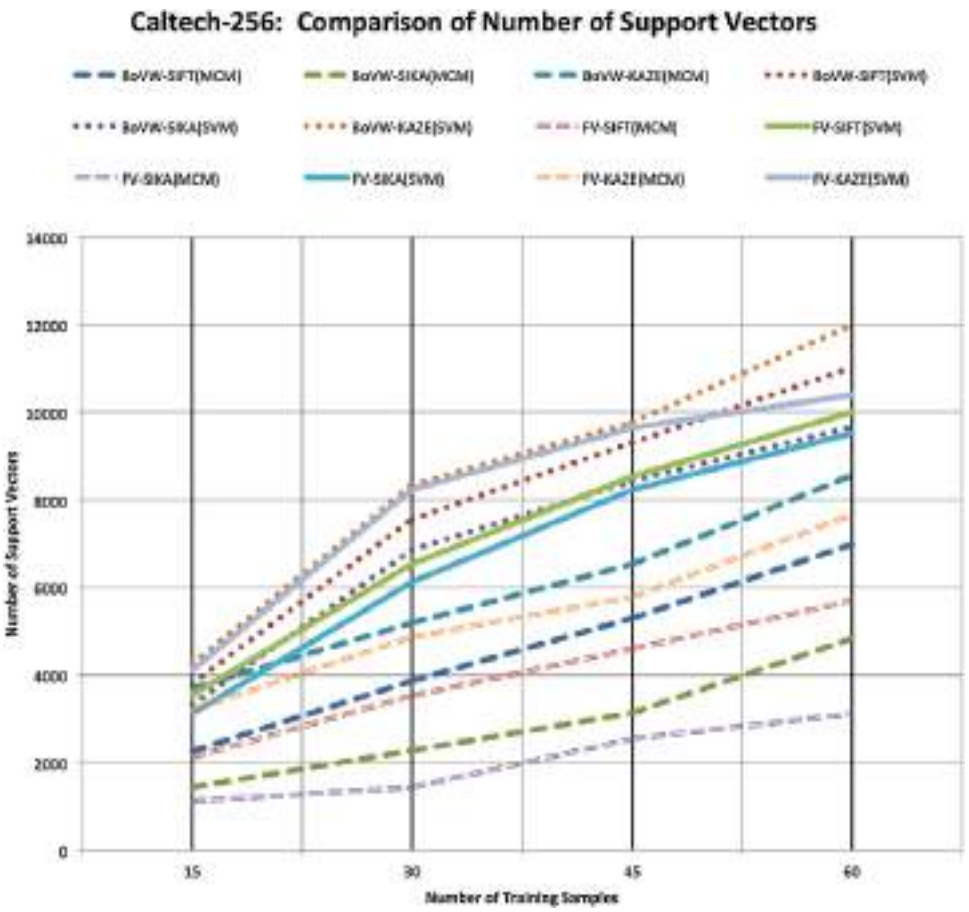

Fig. 6. Comparison of number of support vectors found by MCM and SVM with bag of visual words and Fisher vector encoding.

simplest representation of features and classifiers. This demonstrates the robustness of the SIKA features for object representation as compared to other feature extraction techniques.

In addition to superior classification accuracy, MCM also gives fewer number of support vectors for both the datasets (Fig. 6 and Table 4). Fewer support vectors mean faster classification of test samples. An interesting trend that can be observed from Fig. 6 is that the number of support vectors are minimum for *SIKA All* among SIFT, KAZE and *SIKA All* features with both MCM and SVM while having the highest classification accuracy. This indicates that the SIKA features are less ambiguous as compared to SIFT and KAZE features irrespective of the classification technique used.

It can also be observed that we beat state of the art results on Caltech-256 while reducing the gap with CNN based techniques in terms of classification accuracy on PASCAL VOC 2007 dataset. The first and intuitive reason for this is that the proposed SIKA features are inherently designed to characterize an object as per its theoretical definition [30]. This makes them less susceptible to attacks such as those in [22] as compared to CNN where the primary reason given for such failure is that patterns partially represent the salient regions within an image and not the object exactly. Hence patterns cannot be a definitive criterion for representing objects unlike the proposed SIKA features. Another strength of the proposed technique is that MCM gives a theoretical bound on VC dimension to reduce the generalization error of the classifier. This presents an alternative to classification using CNN where the claims of generalization are based on trial-and-error that too primarily on natural images. Lastly, a key advantage of our technique is that it is simple, requires lesser training time and gives good results with far less training samples as compared to CNN which are known to be data-hungry and requiring huge training time. For example, as shown in Table 2, our technique was able to outperform state of the art techniques even with as low as 15 training samples. This can help in extending the technique to other domains. Moreover, the state of the art results using CNN on Pascal VOC 2007 are not based on raw features provided by CNN instead they perform additional post-processing [54] or mid-level processing between outputs of various layers. This makes the already complex CNN architecture even more complicated. These observations indicate that the hand crafted SIKA features are efficient and precise for representing object characteristic. This property is not achieved even by the computationally heavy CNN features. CNN despite its high computational complexity and additional post-processing beats our light-weight approach only for some cases.

Table 2
State of the art classification accuracy on Caltech-256.

Technique	15	30	45	60
ScSPM[a] [2009] [16]	27.73	34.02	37.46	40.14
LLC[b] [2010] [17]	34.36	41.19	45.31	47.68
Multipath Sparse Coding [2012] [46]	40.5	48.0	51.9	55.20
SIFT+Fisher vector [2013] [36]	38.5	47.4	52.1	54.8
SIFT+LCS[c]+Fisher vector [2013] [36]	41.0	49.4	54.3	57.3
CNN using ImageNet pretrained [2014] [45]	65.7	70.6	72.7	74.2
Ours (*SIKA ALL*+FV+MCM)	**70.34**	**72.08**	**74.69**	**75.81**

[a] Spatial pyramid matching based on sparse coding.
[b] Locality-constrained linear coding.
[c] Local color statistics.

Table 3
Evaluation on PASCAL VOC 2007 using SVM, MCM and SIKA features.

Class/method	SIFT		KAZE		SIKA(ALL)		SIKA(COMP.)	
	MCM	SVM	MCM	SVM	MCM	SVM	MCM	SVM
Aeroplane	85.67	83.51	65.14	67.41	89.65	86.13	86.5	84.62
Bicycle	69.54	67.12	41.28	44.63	79.8	72.48	67.23	66.3
Bird	60.08	58.13	24.86	30.75	70.5	64.43	61.48	59.1
Boat	71.77	73.14	22.7	23.58	79.4	75.19	73.2	72.9
Bottle	27.32	27.8	8.6	9.82	38.27	29.97	27.65	28.5
Bus	65.91	67.13	29.76	28.15	74.56	68.26	66.76	66.82
Car	81.45	82.61	55.6	54.73	85.76	83.15	82.6	81.44
Cat	59.58	58.5	26.56	25.52	72.5	60.5	59.27	59.23
Chair	51.99	51.52	30.2	29.23	59.2	52.65	51.27	53.2
Cow	47.41	43.54	13.6	14.5	59.87	48.15	48.8	43.2
Diningtable	62.63	58.61	18.56	16.5	71.15	63.15	63.23	59.23
Dog	48.56	43.5	23.7	23.8	57.84	49.65	49.7	44.5
Horse	87.83	82.64	48.78	49.8	91.23	89.91	88.67	81.4
Motorbike	67.11	66.13	40.56	41.5	75.17	69.76	69.9	66.07
Person	83.1	85.4	66.45	67.8	89.8	88.45	81.7	85.8
Pottedplant	31.2	30.2	13.4	14.7	45.7	37.54	30.4	31.6
Sheep	53.76	48.5	14.5	14.6	60.53	55.24	53.2	49.6
Sofa	61.92	57.32	21.86	19.9	67.1	59.53	60.5	58.65
Train	88.26	82.5	44.35	40.2	88.2	87.6	88.3	83.2
Tvmonitor	59.12	53.4	32.22	26.5	66.4	58.7	60.4	53.2
mAP	63.21	61.06	32.13	32.18	**71.13**	65.02	63.54	61.43

The value in bold signifies the highest mAP value in this table.

Table 4
Number of support vectors (SV) on Pascal VOC 2007 dataset using SIFT, KAZE, SIKA All and KAZE with SVM and MCM.

	SIFT		KAZE		SIKA(ALL)		SIKA(COMP.)	
	MCM	SVM	MCM	SVM	MCM	SVM	MCM	SVM
#SV	11055	18479	14340	22450	7453	9330	9741	11303

Table 5
Classification results on Pascal VOC 2007.

Technique	mAP(%)
Vector quantization [2006] [47]	56.07
SuperVector encoding (sv-soft) [2010] [48]	61.1
Locality-constrained linear coding (LLC) [2010] [17]	59.3
Fisher Kernel [2010] [35]	61.69
Context-SVM [2011] [49]	70.5
GHM[a] [2012] [50]	64.70
AGS[b] [2013] [51]	71.1
Zeiler & Fergus [45]	75.90
Oquab et al. [52]	77.7
Chatfield et al. [53]	82.42
Spatial pooling in deep CNN [2014] [54]	82.44
Ours (*SIKA ALL* + MCM)	**71.13**
Ours (*SIKA Complementary* + MCM)	63.54

[a] Generalized hierarchical matching.
[b] Ambiguity guided subcategory mining.

6. Conclusion

In this paper, we have proposed a novel feature SIKA, which has been constructed by exploiting the complementary nature of SIFT and KAZE. We establish the property that SIFT and KAZE represent complementary information of an object. We go on to demonstrate the efficacy of this proposed features by combining them with MCM and running experiments on standard image databases. The results clearly establish the superiority of the proposed object classification technique over the other object classifiers proposed in the literature. The set of techniques proposed in this paper are simple yet powerful and we trust that they have the potential to improve the classification further if used in conjunction with CNN based techniques.

Appendix A. Effectiveness of KAZE in boundary representation

In order to empirically evaluate the effectiveness of KAZE in representing object boundaries, we define two measures: Keypoint Overlap Score (KOS) and Mean Keypoint Overlap Score (MKOS).

The Keypoint Overlap Score is defined as the percentage of the number of keypoints within a region in an image and is formulated below,

$$KOS = \frac{1}{K}\left[\sum_{o=1}^{O}\sum_{k=1}^{K}\chi(A, KP_k)\right] \tag{A.1}$$

where O is the number of objects in the image, K is the total number of keypoints detected in the image, A is the region of interest, KP_k is the k^{th} keypoint and $\chi(A, KP_k)$ specifies if a keypoint KP_k lies within the region A, which is given as,

$$\chi(A, KP_k) = \begin{cases} 1 & \text{if } KP_k \text{ within } A \\ 0 & \text{otherwise} \end{cases} \tag{A.2}$$

We consider two type of regions within an image. First is the bounding box of the objects, BB_o where o is the object while the second, A_{band} is a region defined by a band around the boundary of the object and is discussed later.

Since KOS is image specific, we define a generic goodness measure MKOS as the average over all the images considered for evaluation as follows,

$$MKOS = \frac{1}{I}\sum_{i=1}^{I} KOS_i \tag{A.3}$$

where KOS_i is the Keypoint Overlap Score for an image and I is the total number of images. The KOS and MKOS are calculated on PASCAL VOC 2007 [44] dataset using the ground truth annotations. To characterize the boundaries from the ground truth annotations, we consider a region around the boundary of the ground truth bounding box BB_o by extending and reducing it by a factor of β as shown

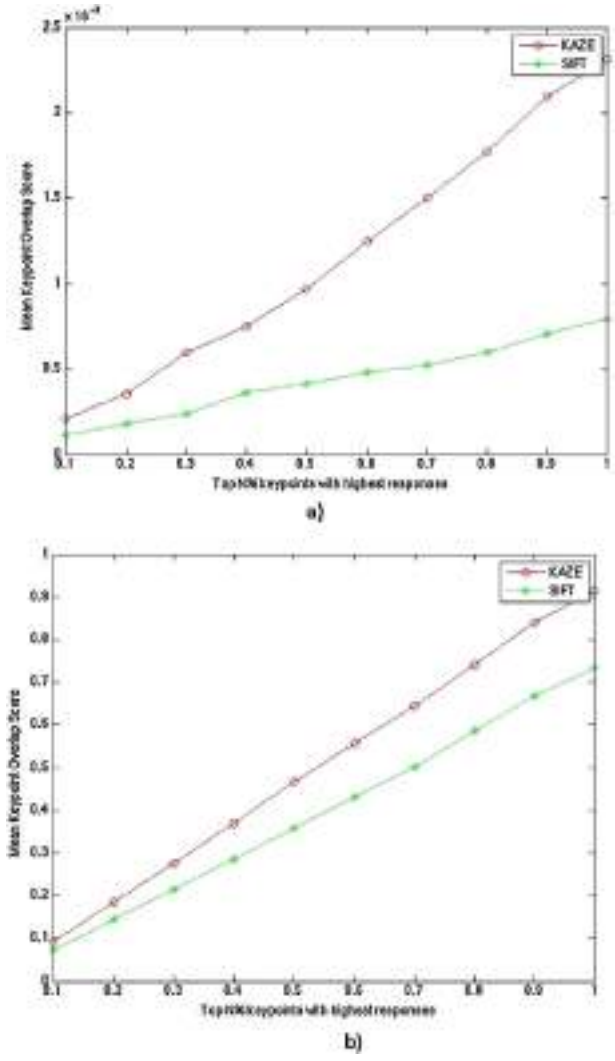

Fig. A7. (a) Mean Keypoint Overlap Score vs top N% keypoints with highest responses (for all keypoints within A_{band} with β = 0.1) (b) Mean Keypoint Overlap Score vs top N% keypoints with highest responses (for all keypoints within the bounding box BB_o).

in Eqs. (A.4) and (A.5). The scores are then calculated for the region represented by A_{band}.

$$A_{extended} = BB_o * (1 + \beta) \tag{A.4}$$

$$A_{reduced} = BB_o * (1 - \beta) \tag{A.5}$$

$$A_{band} = A_{extended} - A_{reduced} \tag{A.6}$$

The KAZE and SIFT keypoints were calculated for each image in the dataset. The keypoints were then sorted according to the keypoint strength provided by respective detectors. The MKOS was then calculated for top N% of the keypoints. Fig. A7 (a) and (b) show the MKOS for this dataset. As can be seen, the density of KAZE keypoints is consistently higher around the object boundaries as compared to SIFT keypoints.

References

[1] M. Tsuchiya, H. Fujiyoshi, Evaluating feature importance for object classification in visual surveillance, in: 18th International Conference on Pattern Recognition, ICPR 2006. vol. 2, IEEE, 2006, pp. 978–981.

[2] C. Wang, D. Blei, F.-F. Li, Simultaneous image classification and annotation, in: CVPR 2009. IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 1903–1910.

[3] O. Javed, M. Shah, Tracking and object classification for automated surveillance, in: Computer Vision – ECCV 2002, Springer, 2002, pp. 343–357.

[4] U. Regensburger, V. Graefe, Object classification for obstacle avoidance, in: Fibers' 91, Boston, MA, International Society for Optics and Photonics, 1991, pp. 112–119.

[5] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2004) 91–110.

[6] H. Bay, T. Tuytelaars, L. Van Gool, Surf: speeded up robust features, in: Computer Vision – ECCV 2006, Springer, 2006, pp. 404–417.

[7] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: CVPR 2005. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, IEEE, 2005, pp. 886–893.

[8] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, Orb: an efficient alternative to sift or surf, in: IEEE International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 2564–2571.

[9] P.F. Alcantarilla, A. Bartoli, A.J. Davison, Kaze features, in: Computer Vision – ECCV 2012, Springer, 2012, pp. 214–227.

[10] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.

[11] C.J. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, vol. 2, Springer, 1998, pp. 121–167.

[12] Y. LeCun, Y. Bengio, Convolutional Networks for Images, Speech, and Time Series 3361, 1995.

[13] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. (2012) 1097–1105.

[14] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 580–587.

[15] A.S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN features off-the-shelf: an astounding baseline for recognition, in: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 2014, pp. 512–519.

[16] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009. IEEE, 2009, pp. 1794–1801.

[17] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 3360–3367.

[18] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks, ICLR, 2014.

[19] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1627–1645.

[20] H. Harzallah, F. Jurie, C. Schmid, Combining efficient object localization and image classification, in: IEEE 12th International Conference on Computer Vision, IEEE, 2009, pp. 237–244.

[21] J.R. Uijlings, K.E. van de Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, Int. J. Comput. Vis. 104 (2013) 154–171.

[22] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: high confidence predictions for unrecognizable images, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[23] K.E. Van De Sande, T. Gevers, C.G. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2010) 1582–1596.

[24] Y. Ke, R. Sukthankar, PCA-SIFT: a more distinctive representation for local image descriptors, in: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004. vol. 2, IEEE, 2004, pp. II–506.

[25] E.N. Mortensen, H. Deng, L. Shapiro, A sift descriptor with global context, in: CVPR 2005. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, IEEE, 2005, pp. 184–190.

[26] Jayadeva, Learning a hyperplane classifier by minimizing an exact bound on the {VC} dimension, Neurocomputing 149 (Part B) (2015) 683–689.

[27] N. Khan, B. McCane, S. Mills, Better than sift? Mach. Vis. Appl. (2015) 1–18.

[28] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 1615–1630.

[29] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anal. Mach. Intell. 12 (1990) 629–639.

[30] B. Alexe, T. Deselaers, V. Ferrari, What is an object? in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 73–80.

[31] Z. Liu, W. Zou, O. Le Meur, Saliency tree: a novel saliency detection framework IEEE Trans. Image Process. 23 (2014) 1937–1952.

[32] K. Chatfield, V.S. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods., in: BMVC, volume 2, 2011, p. 8, No. 4.

[33] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Workshop on Statistical Learning in Computer Vision, ECCV, Prague, 2004, pp. 1–2, volume 1.

[34] J.C. van Gemert, J.-M. Geusebroek, C.J. Veenman, A.W. Smeulders, Kernel codebooks for scene categorization, in: Computer Vision – ECCV 2008, Springer, 2008, pp. 696–709.

[35] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: Computer Vision – ECCV 2010, Springer, 2010, pp. 143–156.

[36] J. Sánchez, F. Perronnin, T. Mensink, J. Verbeek, Image classification with the Fisher vector: theory and practice, Int. J. Comput. Vis. 105 (2013) 222–245.

[37] V.N. Vapnik, V. Vapnik, Statistical Learning Theory, vol. 1, Wiley, New York, 1998.
[38] S. Chandra, M.A. Jayadeva, Numerical Optimization with Applications, Alpha Science International, Oxford, UK, 2009.
[39] K. Bache, M. Lichman, UCI Machine Learning Repository, 2013.
[40] A. MOSEK, The MOSEK Optimization Toolbox for MATLAB Manual, Version 7.1 (Revision 31), MOSEK ApS, Denmark, 2015.
[41] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011), 27:1-27:27. Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm.
[42] A. Vedaldi, B. Fulkerson, VLFeat: An Open and Portable Library of Computer Vision Algorithms, 2008 http://www.vlfeat.org/.
[43] G. Griffin, A. Holub, P. Perona, Caltech-256 Object Category Dataset, 2007.
[44] M. Everingham, L. Van Gool, C. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2008.
[45] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: Computer Vision – ECCV 2014, Springer, 2014, pp. 818–833.
[46] L. Bo, X. Ren, D. Fox, Multipath sparse coding using hierarchical matching pursuit, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 660–667.
[47] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE, 2006, pp. 2169–2178.
[48] X. Zhou, K. Yu, T. Zhang, T.S. Huang, Image classification using super-vector coding of local image descriptors, in: Computer Vision – ECCV, vol. 6315, Springer, 2010, pp. 141–154.
[49] Z. Song, Q. Chen, Z. Huang, Y. Hua, S. Yan, Contextualizing object detection and classification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 1585–1592.
[50] Q. Chen, Z. Song, Y. Hua, Z. Huang, S. Yan, Hierarchical matching with side information for image classification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 3426–3433.
[51] J. Dong, W. Xia, Q. Chen, J. Feng, Z. Huang, S. Yan, Subcategory-aware object classification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2013, pp. 827–834.
[52] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2014, pp. 1717–1724.
[53] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: delving deep into convolutional nets, in: British Machine Vision Conference, 2014.
[54] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: Computer Vision – ECCV 2014, Springer, 2014, pp. 346–361.