Chapter I

Introduction

Image classification has always been an attractive research direction in computer vision, since it is closely related to many interesting applications such as identifying an image on the web. The image content could be a certain type of human action, a kind of object, or a scene. The computer understands the query images in the desired way and classifies them into different categories automatically. Following the development of this field, we have introduced a novel image classification scheme that takes the advantages of visual saliency.

1.1. Nearest-Neighbour Classifiers

Every year quite a few approaches are invented for image classification. Generally, these classifiers are either parametric or non-parametric. Common parametric methods include the support vector machine (SVM), decision trees, boosting, and neural networks. They learn the model parameters from annotated training data. Non-parametric methods process information without the procedure of learning. Therefore, normally they are simpler than the learning-based classifiers with slightly degraded performance. But the value of non-parametric approaches has always been underrated.

Nearest-Neighbours is among the non-parametric classifiers. Typically, the classification procedure consists of four steps: feature detection, feature extraction or image representation, image distance calculation, and classification based on the distance (similarity). Boiman *et al*.

[1] apply image-to-class (I2C) distances instead of image-to-image (I2I) distances since they claim image descriptor quantisation and I2I distance computation can affect the performance of Nearest-Neighbour based classifiers. Hence the Naive-Bayes Nearest-Neighbour (NBNN) was proposed. They show that the NBNN method can estimate the optimal classification based on the naive Bayes hypothesis. Annotated images are only employed as references and no prior learning or training is required, which is similar to the original Nearest-Neighbour classifiers. The I2C distances specify the similarities between an input image and the classes formed by the images with a same label. Although very simple in concept, NBNN ranks among the leading methods in term of its performance.

Following [1], Tuytelaars *et al*. [2] have improved the original NBNN by incorporating a kernel that concatenates the I2C distances from all the classes. The kernel, which is a vector, can be employed to train a SVM classifier. Because the kernels preserve more discriminative feature-level information, when used with SVM they produce better results than the original NBNN does. Besides Tuytelaars *et al*. [2], Bechmo *et al*. [3] and Wang *et al*. [4] have their own works towards the optimal NBNN. Bechmo *et al*. [3] commence their investigation from the hypothesis of NBNN. NBNN simplifies the class estimation problem by assuming that the probability of each class-dependent feature can be approximated by the Parzen kernel, which is mostly a Gaussian distribution and class-independent. Bechmo *et al*. [3] set the parameters such as the bandwidth and the normalisation factor of the kernel different for the features in different classes. The parameters are learned using hinge-loss optimisation from the training data. Wang *et al*. [4] combine a learned Mahalanobis metric with the I2C distance. The class-specific metric defines a large margin, which is optimised by the gradient descent method, to separate the corresponding I2C distance of the expected class from the participation of other classes discriminatively. Although [2-4] inherit the merits from NBNN, they are essentially

learning-based parametric classifiers. Because prior training is necessary, their frameworks are also more complex.

Inspired by [5, 6], McCann and Lowe [7] have proposed another non-parametric Nearest-Neighbour based classification method, named local NBNN. Without calculating I2C distance to every class, local NBNN finds the most relevant classes inside the whole image set for the features and computes the I2C distance in the local neighbourhood. This algorithm narrows the searching space and as a consequence the classification procedure is speeded up. McCann and Lowe claim that local NBNN outperforms NBNN and NBNN kernel with a fine-tuned area of searching, given the fact that only the categories inside a local neighbourhood make the most significant and reliable contribution to the posterior probability under the Bayes assumption. Limiting feature comparison to local neighbourhoods for a query descriptor ignores the distant categories which are less meaningful.

1.2. Motivations and Contributions

Though NBNN and local NBNN have displayed their potentials in image classification, they have their weaknesses. For example, NBNN and local NBNN use all the local features identically. But apparently, some of these features carry more valuable information. In some cases, irrelevant features can disrupt the I2C distance and errors are brought in. For instance, the background features such as patches representing grass or sky from a cricket-playing image and the horse-riding images can be similar. As a result, the I2C are not sufficiently separated enough to make them distinguishable from each other, especially when the reference images in the same category have a large intra-class variability. On the other hand, the background is not useless. After all, images from one category usually share similar context. For example,

croquet is always played on grass. This common character can make croquet-playing images identical.

Based on the above-mentioned reasons, we start to investigate the feasibility of treating different regions of an image in I2C computation separately. Naturally, humans often focus on somewhere that attracts them most in an image. They spend more time observing that part than anywhere else. The things in a scene that are capable of drawing the attention of people are defined as the salient part. Hence, we choose the visually salient areas as the foreground while the remaining regions are considered to be the background with contextual content.

Saliency arises from the contrasts between the object and its neighbourhood. Inspired by the various saliency detection methods, we manage to divide the images into foreground and background. With the identified regions of object and context, we have built unique efficient context-aware (or saliency-aware) Nearest-Neighbour classifiers that calculate I2C distances for different isolated regions respectively. Our contributions can be concluded as: firstly, we use a saliency detector to recognise the features from the object and the context; secondly, we calculate I2C distances for the object and the context instead of treating all the features as a whole; thirdly, we have developed a voting scheme for the outcomes indicated by the multiple I2C distances, which is able to correct the misleading results and thus brings an enhancement in accuracy; finally, we accelerate the classifier by setting anchor points, which are generated through clustering within a class, to replace the massive features involved in the I2C distance computation in original NBNN and local NBNN. With the benefit from the above solutions, our approach costs significantly less time but is superior to the original NBNN and local NBNN in image classification.

1.3. Image Datasets

There are many public datasets in the computer vision field. In order to demonstrate the applicability, we test our method on three datasets that contain images with different attributes. In this section, we will describe some basic information of these datasets, such as scales, image resolutions, and colour depth.

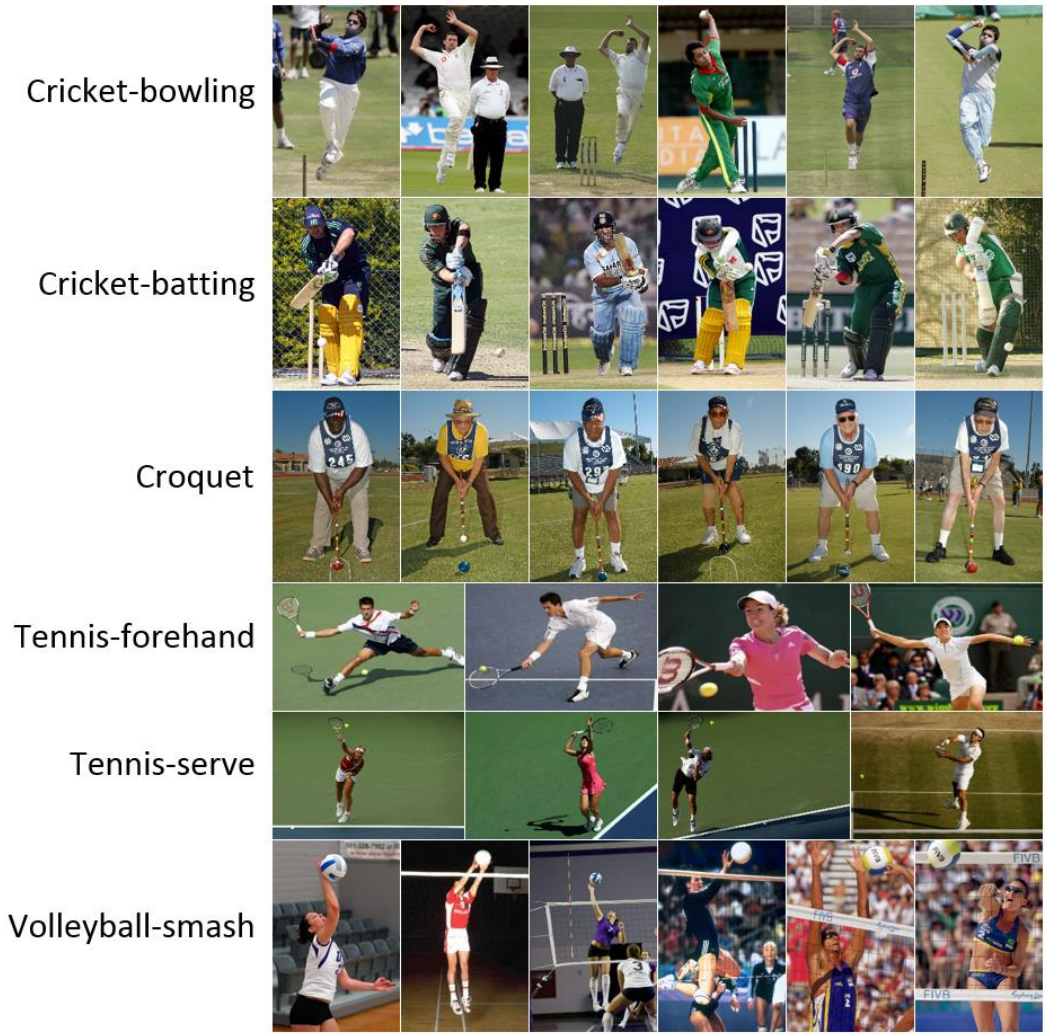

Fig. 1.1: Pami-09 datasets

1.3.1. Pami-09

The datasets (Fig. 1.1) include six sports classes and are originally published by Gupta *et al*. [8] in their research of human-object interactions. The six categories are cricket-bowling, cricket-batting, croquet, tennis-forehand, tennis-serve, and volleyball-smash, with 50 images

each. As indicated by Gupta *et al*. [8], the classification task can be very challenging because the actions have limited inter-class variations. The similar poses and the scenes in the images can bring significant confusion. The images are in 24-bit colour depth PNG format. Resolutions vary from 250×150 to 2560×1920.

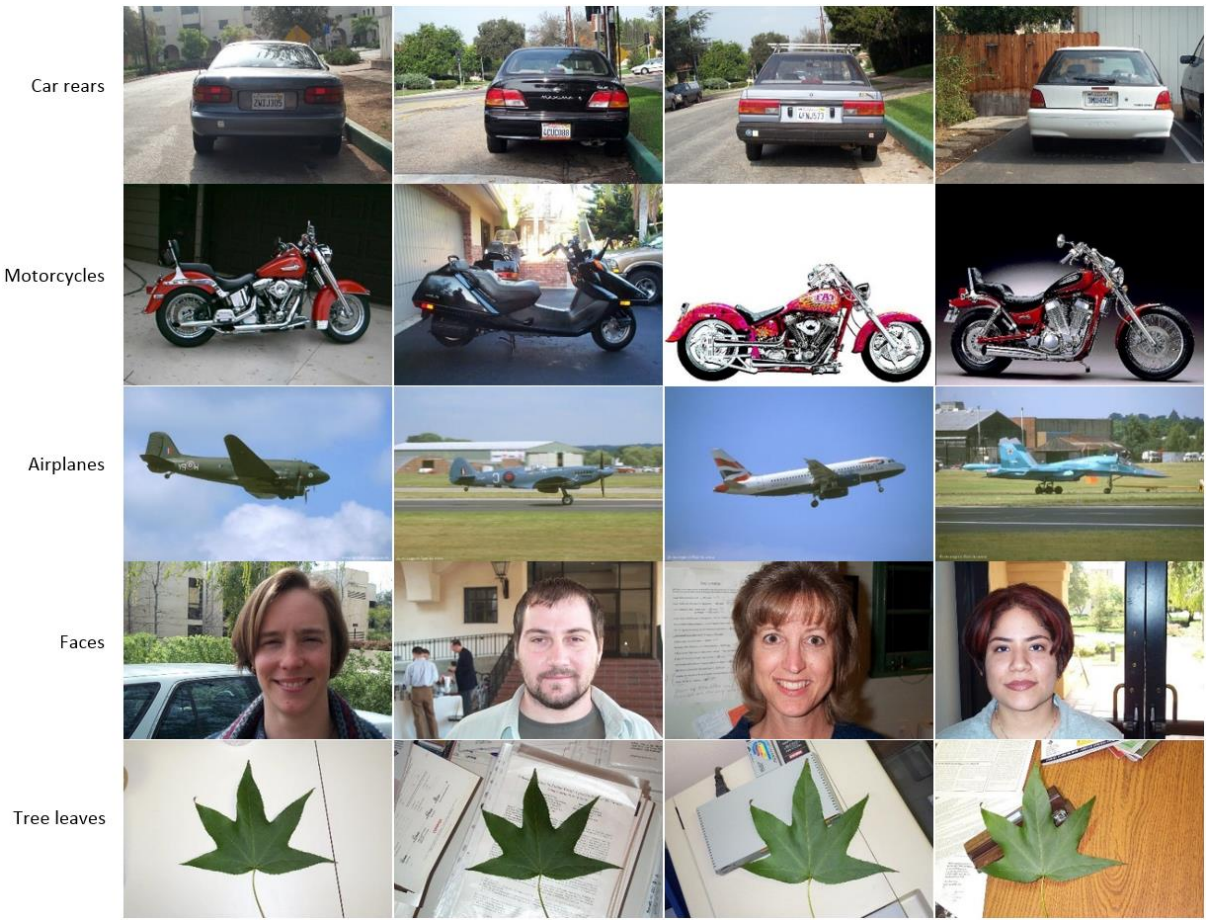

Fig. 1.2: Caltech-5 datasets

1.3.2. Caltech-5

Here we use the initial edition of the Caltech Vision Lab object categorisation datasets (Fig. 1.2), which were built and expanded to Caltech-101 by Fei-Fei *et al*. [9, 10]. We combine the car rears 2001 with the car rears 1999 as they are considered to be the same object. Thus, the datasets contain 5 categories: car rears, motorcycles, airplanes, faces, and tree leaves. The number of images belonging to each category varies from 186 to 1074. All images are in 24-

bit colour JPG format. The sizes of leaf and face images are 896×592. The car rear images have two sizes: 360×240 and 896×592. The airplane and motorbike images vary from 200×113 to 1000×699.

1.3.3. 15-Scene

15 natural scenes, including places such as bedroom, living room, kitchen, office, store, industry and so on (Fig. 1.3) [11-13]. Each scene category has at least 200 images and there are 4485 images in total. The images are in 8-bit greyscale JPG format. The resolutions of the pictures taken from MIT are 256×256 while other sets vary from 240×200 to 509×220.

Fig. 1.3: 15-Scene datasets

1.4. Thesis Outline

We first introduce the motivations, the datasets, and the related background knowledge of image classifications in general in this chapter. In the following chapter, we will illustrate the algorithms of NBNN and local NBNN, which pave the way to our framework. In Chapter

III, we have discussed the mainstream saliency models proposed in different periods in terms of their principles. Moreover, we have described how we select the model and make it applicable to our method. In Chapter IV, the details of our framework are explained. We have also presented some discoveries that have enhanced our method to another level. In Chapter V, we show the performance of our method against NBNN and local NBNN. In the final chapter, we give our conclusions and possible future research directions.

Chapter II

Nearest Neighbour Classification Based on Naive Bayes Assumption

2.1. Naive Bayes Nearest Neighbour

Boiman *et al*. [1] have introduced the NBNN classifier, based on the claim that feature quantisation can degrade the performance. Many learning based classifiers use dimensionality reduction or codebooks [14, 15] to generate compact image representation. This avoids huge computational load and possible overfitting but also sacrifices the most discriminative features. Usually, simple features such as edges and corners that can be largely found in the datasets are preserved better while infrequent features can have big errors under the designed quantisation framework.

According to [1], I2I is efficient due to intra-class variability under some cases. However, features from an image can find their counterpart more easily when the features from a category are put together. As a result, they compute I2C. Only a few labelled images are required and no prior learning is needed. Despite being conceptually simple, NBNN can compete with the state-of-the-art classifiers.

Assume $d_1, \dots, d_n$ are the extracted local image descriptors (features) from a test image, NBNN finds a class C that minimises

$$\sum_{i=1}^{n} \|d_i - NN_C(d_i)\|^2 \tag{2.1}$$

where $NN_C(d_i)$ denotes the Nearest Neighbour descriptor that has a minimum distance to d_i in class C. Given a query image Q, using the maximum a posteriori (MAP) model which minimises the error, the estimation can be decided by

$$\hat{C} = argmax_c P(C|Q) = argmax_c P(Q|C) \tag{2.2}$$

$\hat{C}$ is the estimated label. When the prior $P(C)$ is uniform, based on Bayes theory, this has become a maximum-likelihood (ML) problem. In a naive Bayes case, each local descriptor d_i is independent, $P(Q|C)$ can be formulated as the product of $P(d_i|C)$:

$$P(Q|C) = \prod_{i=1}^{n} P(d_i|C) \tag{2.3}$$

Introducing the log probability, it is modified to:

$$\hat{C} = argmax_c \log \prod_{i=1}^{n} P(d_i|C) = argmax_c \sum_{i=1}^{n} \log P(d_i|C) \tag{2.4}$$

$P(d_i|C)$ can be expressed by the Parzen kernel, which is typically a Gaussian function, and for NBNN only the nearest neighbour is considered:

$$\hat{P}(d_i|C) = \frac{1}{L}\sum_{j=1}^{L} K(d_i - {d_j}^c) = K\big(d_i - NN_C(d_i)\big) = \exp\left(-\frac{\|d_i - NN_C(d_i)\|^2}{2\sigma^2}\right) \tag{2.5}$$

K represents the kernel function and L is the number of descriptors in a class. Thus, the ultimate estimation can be written as:

$$\hat{C} = argmax_c \sum_{i=1}^{n} \log e^{-\frac{\|d_i - NN_C(d_i)\|^2}{2\sigma^2}} = argmin_c\left(\sum_{i=1}^{n} \|d_i - NN_C(d_i)\|^2\right) \tag{2.6}$$

Above all, the NBNN image classifier can be summarised in Algorithm 1.

Algorithm 1 NBNN

Require: descriptors of reference images with class label c
Input: local image descriptors $d_1, \ldots, d_n$ of a test image I

for all descriptors $d_i \in I$ **do**

for all classes C **do**
find the nearest neighbour of d_i in C: $NN_C(d_i)$
do $\sum_{i=1}^{n} \|d_i - NN_C(d_i)\|^2$
end for
end for

Output: $\hat{C} = argmin_c \sum_{i=1}^{n} \|d_i - NN_C(d_i)\|^2$

2.2. Local Naive Bayes Nearest Neighbour

McCann and Lowe [7] have developed NBNN by restricting the feature searching space to a much smaller local neighbourhood that determines the posterior probability estimation. The neighbourhood only consists of a part of all categories. Their theory has been justified by proving the deduction of log-odds update.

Let C stand for some classes and $\bar{C}$ for all others. Q is a query image. Assuming all the local features are independent from each other, based on Bayes rule the odds (O) of class C can be expressed as

$$O_C = \frac{P(C|Q)}{P(\bar{C}|Q)} = \frac{P(Q|C)P(C)}{P(Q|\bar{C})P(\bar{C})} = \prod_{i=1}^{n} \frac{P(d_i|C)P(C)}{P(d_i|\bar{C})P(\bar{C})} \tag{2.7}$$

Taking the log probability equation (2.7) becomes

$$\log(O_C) = \sum_{i=1}^{N} \log \frac{P(d_i|C)}{P(d_i|\bar{C})} + \log \frac{P(C)}{P(\bar{C})} \tag{2.8}$$

By applying Bayes rule again, equation (2.8) can be written as

$$\log(O_C) = \sum_{i=1}^{N} \log \frac{P(C|d_i)P(\bar{C})}{P(\bar{C}|d_i)P(C)} + \log \frac{P(C)}{P(\bar{C})} \tag{2.9}$$

The prior odds are $\frac{P(C)}{P(\bar{C})}$, the update is determined by the posterior odds $\frac{P(C|d_i)}{P(\bar{C}|\overline{d_i})}$. When the posterior odds are greater than the prior odds, the increment is positive. If the posterior odds are smaller, the increment is negative. Based on the assumption that the class priors are equal, the classification procedure can be simplified as

$$\hat{C} = argmax_c(\sum_{i=1}^{N} \log \frac{P(C|d_i)P(\bar{C})}{P(\bar{C}|d_i)P(C)}) \quad (2.10)$$

The above formulation clarifies the role the increment is playing. It proves that only the remarkable update affects. The steps of local NBNN is given below.

Algorithm 2 Local NBNN

Require: descriptors of reference images with class label c
Input: local descriptors $d_1, \dots, d_n$ of a test image I, number of nearest neighbours k

for all descriptors $d_i \in I$ **do**
    Find $p_1, \dots, p_{k+1}$ nearest neighbours of d_i: $NN_C(d_i)$
    for all the k classes C having one of $p_1, \dots, p_k$ do
        $\sum_{i=1}^{n} \|d_i - NN_C(d_i)\|^2$
    end for
end for

Output: $\hat{C} = argmin_c \sum_{i=1}^{n} \|d_i - NN_C(d_i)\|^2$

Besides k nearest neighbours, one more search continues for the background, which can be considered as an upper bound. These distances will not affect label estimation. Hence the classification results are independent of this additional searching.

2.3. Summary

In this chapter, we have reviewed two simple but effective unsupervised (non-parametric) nearest neighbour classifier called NBNN and local NBNN. The images are represented locally using Bag-of-Words (BoW) model, without the procedure of putting them into codebooks. In other words, the model is loaded with a collection of local features. The sequences or the spatial relationships between those features will not be considered. In the next chapter, we will discuss the feasibility of dividing the bag into "smaller bags" using saliency detectors.

Chapter III

Saliency Detection

3.1. Classification of Saliency Detectors

Visual Saliency has been actively explored during the last 30 years. According to Borji and Itti [16], the available detection approaches can be divided into two modes: the bottom-up and the top-down models. Bottom-up models directly make use of the information encoded in scene characteristics. As addressed by Borji and Itti [16], the bottom-up methods are usually faster and more straightforward than the top-down methods, while top-down models are driven by the cognitive information, including targets and expectations. Therefore, their performances rely heavily on prior knowledge (even require training). However, in our framework, we want the approach to be non-parametric, which requires us to concentrate on the effectiveness of the bottom-up detectors only.

3.2. Bottom-up Saliency Detectors

Itti *et al*. [17] have proposed one of the earliest visual attention models. They filter the input image to nine spatial scales using the dyadic Gaussian pyramids [18] and apply a series of "centre-surround" analyses to three feature channels, the orientation, the intensity, and the colour channel, separately. Though this method has established a standard for the follow-ups, its performance relies heavily on the types of its feature maps.

Harel *et al*. [19] have proposed another early invention, the Graph-Based Visual Saliency (GBVS) model. Similar to [17], this method extracts maps from several feature channels at different image scales. Based on the feature vectors at different locations, the method uses a Markov approach to form an activation map, whose nodes are fully connected with a graph. As demonstrated in [19], GBVS provides more accurate predictions on human fixations than the previous methods as it is biologically plausible. Moreover, it is able to be reformed to a multi-resolution counterpart and thus more promising outcome can be potentially achieved.

Sometimes the computation of visual saliency can be rather simple. Hou and Zhang [20] have developed a model based on Spectral Residual (SRS). The method has no reliance on prior knowledge such as features and category labels, owing to the fact that a number of natural images share a similar part in frequency domain (spectrum) statistically. From the point of view of information theory, in the frequency domain, the common part, which is redundant, can be subtracted. The remaining part, which carries the discriminative information of each individual image, can be employed to draw the saliency map followed by a Gaussian filtering process for the purpose of visualisation. This method requires limited computational resource so it runs very efficiently.

In recent years, this field has been consistently developed. Tavakoli *et al*. [21] have proposed another centre-surround method named Fast & Efficient Saliency (FES), which uses sparse sampling and kernel density estimation to obtain the saliency map under the Bayesian rule. Hou *et al*. [22] introduce a sparse foreground detector by defining a simple but powerful image descriptor called Image Signature (IS). Murray *et al*. [23] base their method on colour appearance and centre-surround windows, whose sizes are determined by a Gaussian Mixture Model with training data. This Saliency by Induction Mechanisms (SIM) method decomposes the images into multiple scales and integrates the scaled images by wavelet and inverse wavelet transforms respectively.

Vikram *et al*. [24] have proposed another centre-surround model (RCSS). This model computes saliency in terms of intensity differences of the pixels in a number of sub-windows, whose sizes and positions are decided by a discrete uniform probability distribution function in three channels at the original scale of the Gaussian filtered image, followed with a saliency map fusion. A usage of window-sliding technique can be also found in the Conditional Random Filed model (CRF) [25].

Different from the above-mentioned models, the techniques Saliency Detection by Self-Resemblance (SDSR) [26] and Region Covariance-based Visual Saliency (CovSal) [27] use non-linear features, instead of the ordinary linear features such as a Gabor filter. They claim non-linear features and their integration can preserve local structures better.

Riche *et al*. [28] have proposed another bottom-up detector, which observes a mechanism defined as rarity (RARE-2012) in various channels with multiple images scales. This method is developed from their previous designs in 2007 (RARE-2007) [29] and 2011 (RARE-2011) [30]. RARE-2007 only considers colour information. The orientation is ignored. RARE-2011 uses Gabor filter. RARE-2012 improves RARE-2011 by introducing parallel and serial features extraction.

3.3. Bottom-up Methods with Top-down Prior

Apart from the above-mentioned pure bottom-up methods, there are many detectors that combine the bottom-up concept with top-down prior. For instance, Bruce and Tsotsos [31] have introduced a visual attention detector based on information maximisation (AIM). Their method estimates the saliency probability distribution through observing the correspondences between a number of small local image patches and a set of basis coefficients representing the patches from the natural images database, determined by independent component analysis (ICA) [32].

Thus, it requires a large sampling from available natural scenes, which results in a reliance on the database.

Zhang *et al*. [33] have proposed another saliency model using natural statistics (SUN), which is similar to [31]. On the other hand, besides the local image information, the top-down knowledge, which is an object location prior independent from the features, is incorporated in their approach based on a Bayesian framework. The difference of Gaussian (DoG) and ICA-derived descriptors have been employed. They claim SUN can outperform or at least compete with the most influential techniques at that time and its features can be developed to a higher level to further release the potential of SUN.

Other models that incorporate top-down knowledge include [34-35]. Torralba *et al*. [34] have proposed a contextual guidance model that makes use of both local and global features. The local features identify the spatial locations while the global features, which could be used for the scene recognition, indicate the expected positions holistically. The two pathways work independently in parallel. Besides the low-level features that have been applied in the existing models [17, 34-37], Judd *et al*. [38] proposed a more advanced technique based on machine learning using mid-level gist features [14], high-level face [39] and human detectors [40], and a centre prior, based on the assumptions about what kinds of objects and how they appear in natural images, respectively. Since SVMs are involved, this detector needs a number of training data. Another example that combines different levels of features and the visually psychological rules has been presented in [41].

3.4. Saliency Detector Selection

We have briefly described a number of mainstream saliency detectors proposed in the recent years. Essentially, we are looking for an unsupervised image classification framework.

Saliency detectors incorporating any learned priors or top-down assumptions, such as [16, 19, 20-21, 30], will be excluded since their performance can be dependent on the training dataset. Generally, existing bottom-up models are simpler, and they can offer enough discrimination in finding salient regions.

We have compared seven saliency detectors, including SRS [20], FES [21], IS with LAB and RGB [22] channels, RCSS [24], CRF [25], SDSR [26], and RARE-2012 [28]. The saliency maps of an image from *Pami-09* [8] produced by different detectors are shown in Fig. 3.1. We decide to choose SDSR [26] as our detector because it brings decent local image structures due to its non-linear feature combination property, instead of focusing on a few points or going into details. Though the detectors have multiple parameters to define, we assume the default settings are proper as claimed by most authors. Furthermore, SDSR is robust to data uncertainty [26]. We want our method to be applicable to various types of image datasets.

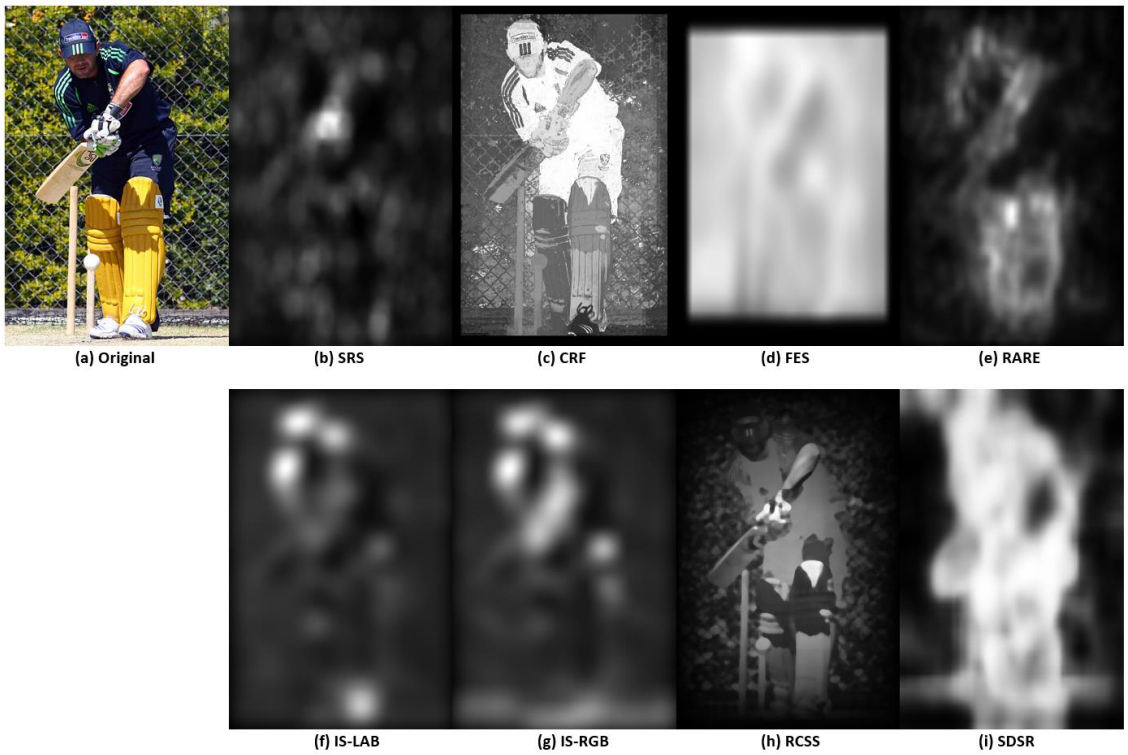

Fig. 3.1: Saliency maps generated via different models: by setting a threshold we can separate the foreground and background easily using SDSR, without breaking the original structures.

There are many criteria for comparing these saliency detectors such as Kullback-Leibler (KL) divergence, normalised scanpath saliency (NSS), string editing distance, area under curve (AUC), linear correlation coefficient (CC), and visually subjective scores [16]. However, these measurements only evaluate saliency detectors in one aspect, such as probability distribution, signal detection metric, or statistical relationship.

3.5. Self-resemblance Saliency Detection

As mentioned above, this model shows great performance in preserving local structures. For instance, it indicates the salient region without breaking the object into isolated pieces or greater pixels. In other words, if the foreground and the background cannot be separated from each other effectively, the I2C distances between the object features or the contextual features will not have significant variations, since these features are still in a mixture of foreground and background.

Different from correlation methods, SDSR finds dissimilarities between a pixel and its neighbourhood, based on the non-linear local regression kernels. The kernels that encode the dissimilarity are estimated in a non-parametric way.

Similar to other models, for each pixel $\boldsymbol{x}_i$ in an image, if it is salient is formulated by

$$t_i = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \text{ is salient} \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

where t_i denotes the saliency of $\boldsymbol{x}_i = (x_i, y_i)$, $i = 1, \dots, M$, M is the number of pixels. According to Seo and Milanfar [26], the saliency of SDSR at pixel $\boldsymbol{x}_i = (x_i, y_i)$ is a posterior probability:

$$s_i = P(y_i = 1 | \boldsymbol{F}_i) \tag{3.2}$$

where $\boldsymbol{F}_i = [\boldsymbol{f}_i^1, \dots, \boldsymbol{f}_i^L]$ is the feature matrix that includes a number of feature vectors at pixel $\boldsymbol{x}_i$, L is the number of vectors inside a specified window. Generally, employing multiple features performs better than using a single vector. Let $\boldsymbol{F} = [\boldsymbol{F}_1, \dots, \boldsymbol{F}_N]$ denote the collection of the centre feature matrices surrounding $\boldsymbol{x}_i$, N is the number of pixels in a neighbourhood. Based on Bayes rule, equation (3.2) can be expressed as

$$s_i = P(t_i = 1|\boldsymbol{F}) = \frac{p(\boldsymbol{F}|t_i=1)P(t_i=1)}{p(\boldsymbol{F})} \tag{3.3}$$

$P(t_i = 1)$ is assumed to be equal for all the pixels and $p(\boldsymbol{F})$ is uniform, finding s_i is to estimate the conditional probability density $p(\boldsymbol{F}|t_i = 1)$.

3.5.1. Local Regression Kernel

In order to better capture the local data structure, local steering kernels (LSKs) [42] are used as image features. The kernel is modelled as

$$K(\boldsymbol{x}_l - \boldsymbol{x}_i) = \frac{\sqrt{\det(\boldsymbol{C}_l)}}{h^2} \exp\left(\frac{(\boldsymbol{x}_l-\boldsymbol{x}_i)^T \boldsymbol{C}_l (\boldsymbol{x}_l-\boldsymbol{x}_i)}{-2h^2}\right) \tag{3.4}$$

where $l = 1, \dots, P$ shows the size of the kernel sampling region, h is a global smoothing parameter. For 2D LSKs, the covariance matrix $\boldsymbol{C}_l$ can be derived using the matrix $\boldsymbol{J}_l$:

$$\boldsymbol{J}_l = \begin{bmatrix} z_x(\boldsymbol{x}_1) & z_y(\boldsymbol{x}_1) \\ \vdots & \vdots \\ z_x(\boldsymbol{x}_P) & z_y(\boldsymbol{x}_P) \end{bmatrix} \tag{3.5}$$

where z_x and z_y are the first derivatives along x and y axes. Let (q_1, q_2) and $(\boldsymbol{v}_1, \boldsymbol{v}_2)$ stand for the singular values and singular vectors given by the singular value decomposition (SVD) [42] of

$$\boldsymbol{J}_l = \boldsymbol{U}_l \boldsymbol{S}_l {\boldsymbol{V}_l}^T = \boldsymbol{U}_l diag[q_1, q_2]_l {[\boldsymbol{v}_1, \boldsymbol{v}_2]_l}^T \tag{3.6}$$

Then a robust estimate of $\boldsymbol{C}_l$ can be written as

$$\boldsymbol{C}_l = \gamma \sum_{i=1}^{2} a_i^2 \boldsymbol{v_i} \boldsymbol{v_i}^{\boldsymbol{T}} \tag{3.7}$$

with

$$a_1 = \frac{q_1+\lambda'}{q_2+\lambda'}, a_2 = \frac{q_2+\lambda'}{q_1+\lambda'}, \gamma = (\frac{q_1 q_2+\lambda''}{P})^{\alpha} \tag{3.8}$$

λ'=1 and λ''=10^{-7} are the parameters set to depress noise and prevent the denominators from being 0, and α is set to 0.008 to control γ.

3.5.2. Self-resemblance Saliency

Before constructing the feature matrix $\boldsymbol{F_i}$, linear regression kernels are normalised as

$$W_i = \frac{K(\boldsymbol{x}_l - \boldsymbol{x}_i)}{\sum_{l=1}^{L} K(\boldsymbol{x}_l - \boldsymbol{x}_i)} \tag{3.9}$$

$i = 1, \dots, M$ are the pixel numbers. As mentioned above, L is feature window size, which also stands for the number of selected features. For example, if L is 3×3, at the pixel $\boldsymbol{x}_i$, $\boldsymbol{F}_i = [\boldsymbol{f}_i^1, \dots, \boldsymbol{f}_i^9]$. If the larger neighbourhood has 7×7 pixels centred at $\boldsymbol{x}_i$, $\boldsymbol{F} = [\boldsymbol{F}_1, \dots, \boldsymbol{F}_{49}]$.

Using the constructed feature matrices, Seo and Milanfar [26] estimate the saliency in a surrounding neighbourhood as:

$$s_i = \hat{p}(\boldsymbol{F}|t_i = 1) = \frac{G_i(\overline{\boldsymbol{F}}_i - \overline{\boldsymbol{F}}_i)}{\sum_{j=1}^{N} G_i(\overline{\boldsymbol{F}}_i - \overline{\boldsymbol{F}}_j)} \tag{3.10}$$

with

$$\overline{\boldsymbol{F}}_i = [\frac{\boldsymbol{f}_i^1}{\|\boldsymbol{F}_i\|_F}, \dots, \frac{\boldsymbol{f}_i^L}{\|\boldsymbol{F}_i\|_F}] \tag{3.11}$$

$j = 1, \ldots, N$, $\|\cdot\|_F$ is the Frobenius norm, $G_i(\cdot)$ is the kernel function. By introducing the concept of [43],

$$G_i\left(\overline{\boldsymbol{F}}_i - \overline{\boldsymbol{F}}_j\right) = \exp\left(\frac{-\left\|\overline{\boldsymbol{F}}_i - \overline{\boldsymbol{F}}_j\right\|_F^2}{2\sigma^2}\right) = \exp\left(\frac{-1+\rho(\boldsymbol{F}_i, \boldsymbol{F}_j)}{\sigma^2}\right) \tag{3.12}$$

$\rho(\boldsymbol{F}_i, \boldsymbol{F}_j)$ is the matrix cosine similarity [44-46] and can be defined as Frobenius inner product, σ controls the fall-off weight:

$$\rho\left(\boldsymbol{F}_i, \boldsymbol{F}_j\right) = trace\left(\frac{\boldsymbol{F}_i{}^T \boldsymbol{F}_j}{||\boldsymbol{F}_i||_F ||\boldsymbol{F}_j||_F}\right) \tag{3.13}$$

When dealing with colour images, the detector decomposes the image into 3 channels c_1, c_2, c_3 (CIE L/a/b or RGB), $\mathbb{F}_i = [\boldsymbol{F}_i^{c_1}, \boldsymbol{F}_i^{c_2}, \boldsymbol{F}_i^{c_3}]$. As a result, the saliency map becomes

$$s_i = \hat{p}(\mathbb{F}|t_i = 1) = \frac{1}{\sum_{j=1}^{N} \exp\left(\frac{-1+\rho(\mathbb{F}_i, \mathbb{F}_j)}{\sigma^2}\right)} \tag{3.14}$$

3.5.3. From Saliency Map to Context Map

The above saliency detection approach can be concluded as Algorithm 3.

Algorithm 3 Self-resemblance Saliency Detection

Input: image I, size of LSK P, number of LSKs in the feature matrix for each sampling point L, size of the neighbourhood to compute self-resemblance N, fall-off weight σ, and smoothing parameter h

Step 1: Extract Features
Compute normalised LSK W_i and vectorise it to $\boldsymbol{f}_i$

Step 2: Compute Self-Resemblance Saliency
for $i = 1, \ldots, M$ **do**
  if I is a grey-scale image **then**
    identify feature matrices $\boldsymbol{F}_i, \boldsymbol{F}_j$

$$s_i = \frac{1}{\sum_{j=1}^{N} \exp\left(\frac{-1 + \rho(\boldsymbol{F}_i, \boldsymbol{F}_j)}{\sigma^2}\right)}$$

  else identify feature matrices

$$s_i = \frac{1}{\sum_{j=1}^{N} \exp(\frac{-1 + \rho(\mathbb{F}_i, \mathbb{F}_j)}{\sigma^2})}$$

end if

end for

Output: saliency maps s_i, $i = 1, \dots, M$

The saliency at $\boldsymbol{x}_i$ can be treated as a weight of the local feature for image classification. However, context information is not futile. Therefore, we separate foreground and background to make use of them to a better extent. Similar to [20, 22, 23, 25-28, 35], we simply threshold saliency maps to derive object maps:

$$o_i = \begin{cases} 1 & s_i \geq thr \\ 0 & s_i < thr \end{cases} \tag{3.15}$$

o_i is the object map value at $\boldsymbol{x}_i$. Conversely, the contextual part or the background can be highlighted as

$$b_i = \begin{cases} 1 & s_i < thr \\ 0 & s_i \geq thr \end{cases} \tag{3.16}$$

The threshold can be set as a fixed value, however, in some cases, the object map scale can be either too large or too small, depending on the map intensities. As a result, the object and the context cannot be separately effectively, which can lead to a degradation of the image classifier.

3.6. Summary

In this chapter, we have briefly reviewed the mainstream bottom-up saliency detection methods, and illustrated the reasons choosing SDSR in our framework. In the following chapter, we will present the detail of our classifier, including the role the saliency detector plays.

Chapter IV

Context-aware I2C Distances

Although the original NBNN and local NBNN have achieved impressive accuracies, selecting salient features can further improve their performance. For example, the local features from backgrounds of a horse-riding and a cricket-playing image be identical. As a result, the I2C distances are not discriminative. Thus, we group the local features into object and context. By calculating I2C distances for different groups and category label voting, we have successfully enhanced the performance of nearest neighbour based classifiers.

4.1. An Overview

Using the related and extended works illustrated in the above sections, a unique image classification method that incorporates naive Bayes nearest neighbour classifiers with saliency detection has been proposed. An overview of the framework is presented as Fig. 4.1.

Given a few query images and the reference images with class labels, in the beginning we generate their saliency maps by detecting self-resemblance. Afterwards, we threshold the saliency maps to obtain the desired object and context maps, followed by a multiplication with the original images. When the foreground and the background have been specified, we then extract local features from each part and compute their I2C distances.

The images are represented using BoW model. There are nine I2C distance pairs in total between the foreground, background and original image, as shown in Fig. 4.1. It is worth noting

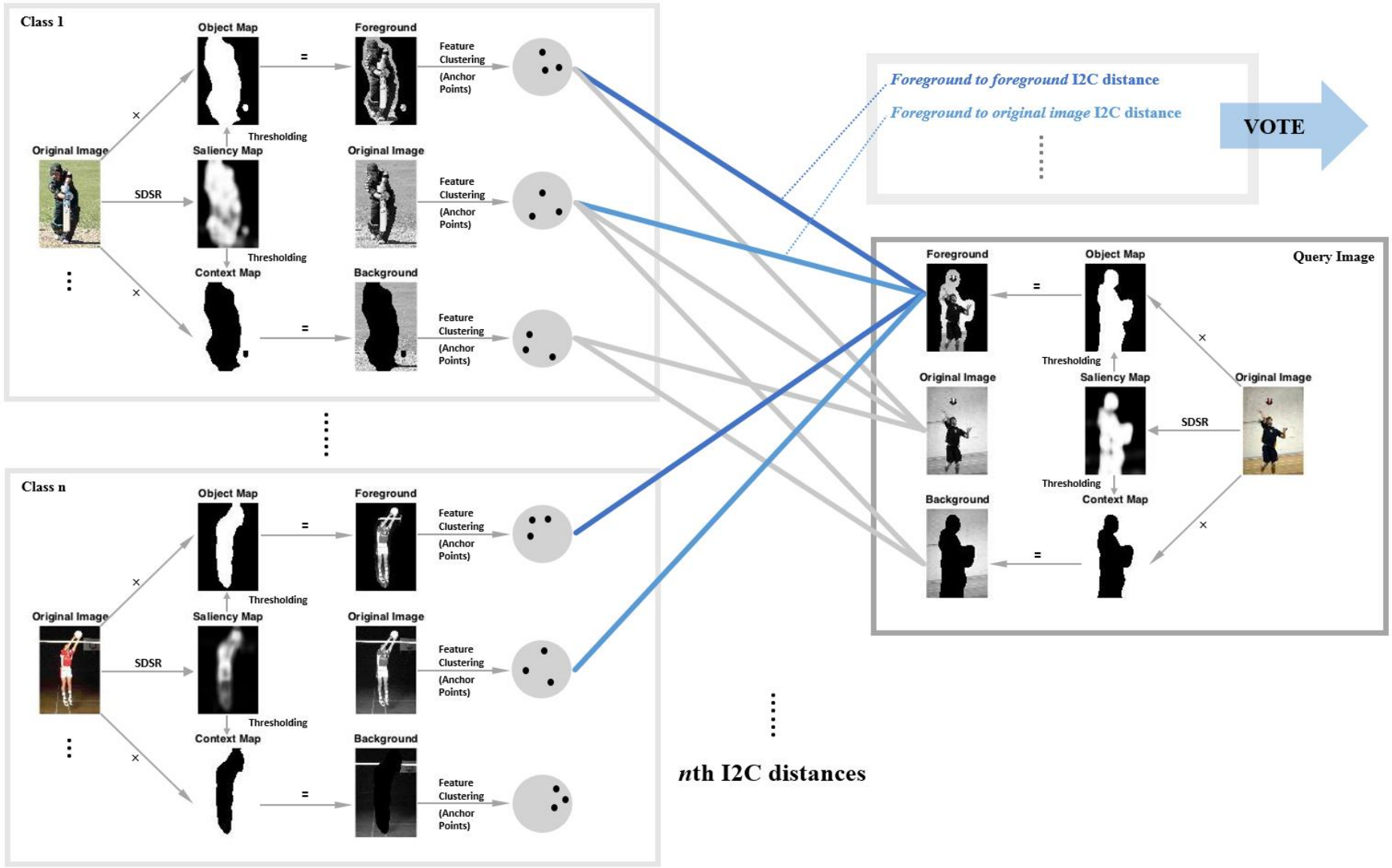

Fig. 4.1: An overview: the framework of the proposed method

that a few pairs deteriorate the last decision. For instance, it is pointless to search object features in a background bag. As a result, the foreground/background and background/foreground have been removed from voting.

We hope only that the I2C distances which are powerful enough to distinguish relevant classes contribute. For the sake of simplifying the system further, we sort the distances based on a test using a small number of images, which are selected from the database on a random basis. When each I2C gives a different label, we trust the one that shows best performance in validation. In all, our framework consists of six critical steps:

- Draw saliency maps for both reference and query images
- Identify the object from the original image with produced saliency maps, and the remaining sections are the background
- Represent images using BoW model, by extracting local features from the object and the background (context)
- Compute the I2C distances between segmented regions and the original images
- Rank the I2C distances and choose validated I2C distances to classify query images by NBNN and local NBNN
- Implement majority voting for the final category label.

Though the framework seems to be complex and time-consuming, we speed up the whole process without degrading its performance. During the following sections, we will describe the technical details that have made our approach a success.

4.2. Image Scales for Saliency Detection

Though most saliency detectors including [26] can draw full-resolution saliency maps, it is still necessary to resize the images to an appropriate scale, not only for the computational

efficiency, but also for the object segmentation. When the input image has a relatively large scale, the saliency detector focuses on edges and corners, while if the image scale is too small, the detector has limited power for identifying different regions (see Fig. 4.2). The resizing only happens for feature detection and the saliency maps are up-sampled to the original resolution for the generation of object and context maps.

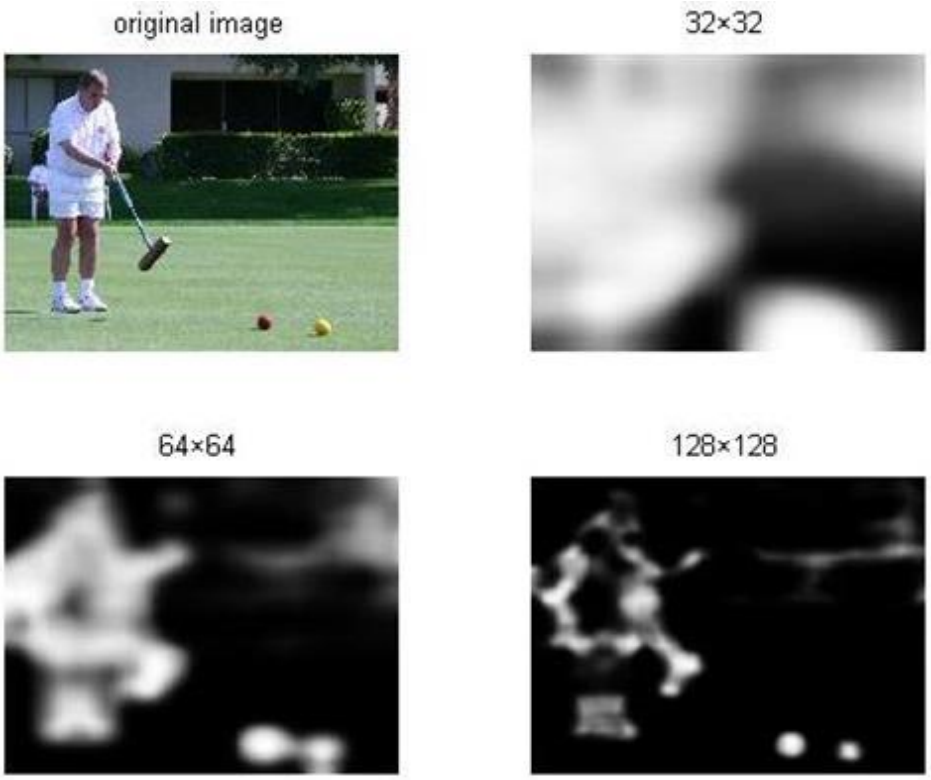

Fig. 4.2: The impact of image scale: saliency maps when input image is rescaled to different sizes.

4.3. Feature Extraction

A single Scale-Invariant Feature Transform (SIFT) [47] is employed. Similar to [48], we extract SIFT descriptors in 16×16 patches. The patches are densely sampled from the original images on a grid. The patch location is defined as its centre point position $\boldsymbol{x}_i$. To categorise the features into foreground or background, we multiply them with a weight factor, which is either 1 or 0, assigned by the value of corresponding object map o_i or context map b_i.

4.4. Feature Clustering for I2C Distances

Despite competitive performance, the computation of I2C distances can be quite time-consuming. The time complexity of NBNN is $O(cN_D N_C \log(N_D N_T))$ [1]. For local NBNN, the complexity becomes $O(cN_D \log(N_C N_D N_T))$ [7]. N_T is the number of reference images inside each category, N_C is the number of categories, N_D is the mean number of features per image, and c denotes the times of comparisons of I2C distances. Normally, N_D can be hundreds or thousands. As a result, the total quantity of features from the reference categories can easily increase to millions. In our framework, local NBNN and NBNN will be repeated for object, context, and original image. With the purpose of reducing such heavy computational load, we commence to investigate the feasibility of representing each class in a more compact but still discriminative way for NBNN and local NBNN.

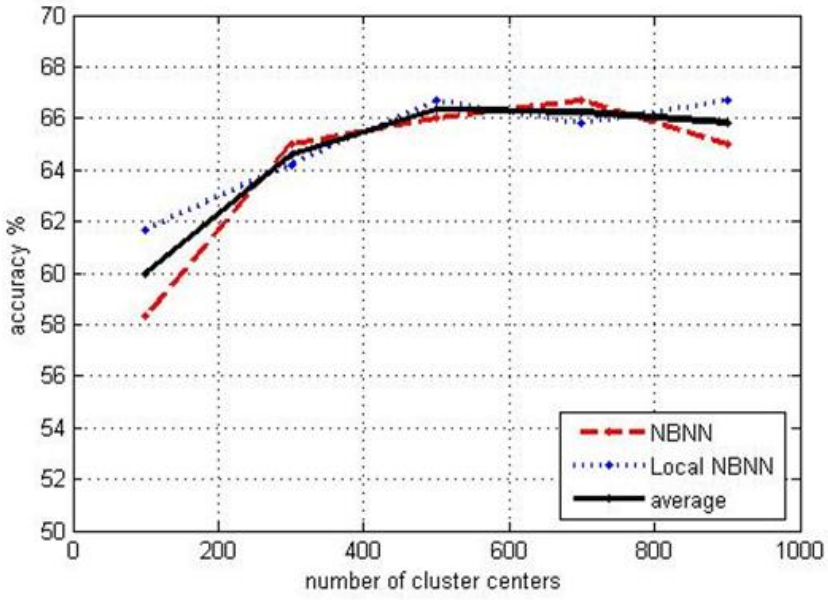

Fig. 4.3: The relationship between the number of anchor points and the classifier performance: the accuracy goes up quickly at the beginning, then slackens its pace and stays around the peak when the quantity of anchor points remains increasing.

In order to keep this classifier unsupervised, we naturally come up with data clustering. Although clustering can be regarded as feature quantisation method and has the potential in

reducing the power of current classifiers [1], however, it has never been applied to NBNN and local NBNN. With the curiosity in discovering how the clustering may affect the performance of NBNN and local NBNN, we use cluster centroids retaining the properties of a category as anchor points. We only cluster reference data. The query features stay unquantized. The anchor points in the I2C distance calculations will replace the large number of image features. The number of anchor points N_A for each class has to be carefully chosen. Insufficient anchor points may affect the precision of I2C distances. On the other hand, if too many anchor points are put into use, the optimisation becomes intractable.

Based on the above reasons, we complete our verification on dataset *Pami-09* [8]. For each class, the first 20 images are used as reference and the next 20 are used for test. We do *k*-means clustering [49] for all the features inside each class and the number of anchor points N_A is set to 100 initially, and goes up to 900 with a step of 200. For local NBNN, we choose to search four nearest neighbourhoods. As can be seen from Fig. 4.3, the accuracies of NBNN and local NBNN grow quickly when N_A is increasing from 100 to 500. Then their performances stay around 66% despite the continuous increment of the number of anchor points. The trend has proved that there is no need to employ a large number of centroids inside each category. What is more, NBNN and local NBNN can only reach 62.5% and 67.5% without clustering, it is demonstrated that by introducing anchor points the performance of the I2C distances will not be degraded significantly. For local NBNN, its accuracy drops a little, while NBNN even shows better results, with a slight increase of 2%-3%.

4.5. Validation and Label Voting

A group of images were randomly chosen for the ranking of different I2C distances. For each dataset, this process has only to be done for local NBNN and NBNN respectively once.

If too many data are involved, the computational time can increase significantly. When the best I2C distances (set to 3) have been confirmed, this step does not need to be repeated. Assume that we have three responses at hand, if more than two of them give the same class label, then the final decision follows. If three labels appear, the one that receives the highest score in the validation stage wins. This label voting corrects the mistakes when the features from different regions of the images possess similar characteristics in the computation of I2C distances.

4.6. The Algorithm

Based upon the above illustrations, we can now summarise the proposed context-aware nearest neighbour image classification method as Algorithm 4.

Algorithm 4 Context-aware Image Classification

Require: reference images $\boldsymbol{I}_R$ with labels $\boldsymbol{c}_R$ and validation images $\boldsymbol{I}_V$ with labels $\boldsymbol{c}_V$
Input: query image $\boldsymbol{I}_Q$, number of nearest searching neighbours k for local NBNN

for all classes $c \in \boldsymbol{c}_R$ **do**
  for all images $\boldsymbol{I}_R \in c$ **do**
    extract local features $\boldsymbol{d}$
    draw saliency map s
    draw object map o and context map b
    classify $\boldsymbol{d} \rightarrow \boldsymbol{d}_o, \boldsymbol{d}_b, \boldsymbol{d}_I$
  end for
  clustering $\boldsymbol{d}_o, \boldsymbol{d}_b, \boldsymbol{d}_i \rightarrow anchor\ points\ \boldsymbol{a}_o, \boldsymbol{a}_b, \boldsymbol{a}_i$
end for
for all classes $c \in \boldsymbol{c}_V$ **do**
  for all images $\boldsymbol{I}_V \in c$ **do**
    extract local features $\boldsymbol{d}'$
    draw saliency map s
    draw object map o and context map b
    classify $\boldsymbol{d}' \rightarrow \boldsymbol{d}_o', \boldsymbol{d}_b', \boldsymbol{d}_i'$
    $\hat{c}_1$=**nbnn** or **local nbnn** $[\boldsymbol{d}_o', \boldsymbol{a}_o]$
    $\hat{c}_2$=**nbnn** or **local nbnn** $[\boldsymbol{d}_o', \boldsymbol{a}_i]$
    $\hat{c}_3$=**nbnn** or **local nbnn** $[\boldsymbol{d}_b', \boldsymbol{a}_b]$
    $\hat{c}_4$=**nbnn** or **local nbnn** $[\boldsymbol{d}_b', \boldsymbol{a}_i]$
    $\hat{c}_5$=**nbnn** or **local nbnn** $[\boldsymbol{d}_i', \boldsymbol{a}_o]$
    $\hat{c}_6$=**nbnn** or **local nbnn** $[\boldsymbol{d}_i', \boldsymbol{a}_b]$
    $\hat{c}_7$=**nbnn** or **local nbnn** $[\boldsymbol{d}_i', \boldsymbol{a}_i]$

for j=1:7
$score_j += \hat{c}_j \&\& c$
end
end for
end for
sort($score_j$)→$\{l, m, n\}$
for all images $\boldsymbol{I}_Q$ **do**
extract local features $\boldsymbol{d}'$
draw saliency map s
draw object map o and context map b
classify $\boldsymbol{d}' \rightarrow \boldsymbol{d}_o', \boldsymbol{d}_b', \boldsymbol{d}_i'$
compute $\hat{c}_l, \hat{c}_m, \hat{c}_n$
$\hat{c} = \text{mode}(\hat{c}_l, \hat{c}_m, \hat{c}_n)$
end for

Output: estimated label $\hat{c}$

$\boldsymbol{d}_i$ and $\boldsymbol{d}_i'$ stand for the collections of features from the original images.

4.7. Summary

In this chapter, we have presented our classification scheme with detailed reasoning and some techniques that have successfully improved the system efficiency. Under the concept of I2C distance, SDSR categorises the words, which are essentially local image features, in the bag into different groups including foreground and background. The foreground carries object information while the background is the context. After that, we calculate I2C distances between those groups separately. Given the class labels estimated from different I2C distances, we vote to receive a final decision. In the next chapter, we will show the experimental results to prove the superiority of our classifier over the original NBNN and local NBNN.

Chapter V

Experimental Results

We have evaluated our method on 3 datasets, which have been presented in Chapter I. Not only the image classification accuracy but also the runtime will be examined. For each database, we repeat our method four times. Each time the reference images and the test images are randomly selected, which means each repetition gives a different partition of data. For each repetition, all methods (NBNN, local NBNN, NBNN and local NBNN based on saliency detection) use same reference and test images. Hence, they compete with equal opportunities. SIFT descriptor is used throughout the experiment. The PC is equipped with an i5-3470 (3.2 GHz) CPU and 8GB RAM, with 64-bit Windows 7 OS installed. Time consumption considers the procedure of classification only. The runtimes of pre-processing and feature extraction are not counted.

According to [5], to fully release the potential of local NBNN, the quantity of the nearest neighbours in searching must be carefully tuned. The details of the influence of tuning can be found in [5]. However, this is not what we want to address so there is no guarantee that local NBNN outperforms NBNN each time. In our experiment, *k*, the number of nearest neighbours, is simply set to $\left\lfloor \frac{1}{2} N_c \right\rfloor$, where N_c is the number of categories belonging to each dataset.

5.1. Pami-09

The datasets are introduced by Gupta *et al*. [8]. Six sports actions include tennis-forehand, tennis-serve, volleyball smash, cricket-defensive shot, cricket-bowling and croquet-shot. All

the actions images are downloaded from internet except the class croquet-shot. These datasets are originally used for the evaluation of image interpretation due to the possession of significant confusion.

Since each class has 50 images, we use 20 of them for reference, 10 for validation, and 20 for test. The results of all the observations are shown in Table 4.1 and Table 4.2. The average confusion matrices of NBNN and local NBNN with or without context awareness are given in Table 4.3 to Table 4.6.

Observation (Sample)	***NBNN + saliency***		***NBNN***		***Local NBNN + saliency***		***Local NBNN***	
	Accuracy	Runtime (s)	Accuracy	Runtime (s)	Accuracy	Runtime (s)	Accuracy	Runtime (s)
# 1	0.692	14.8	0.642	64.9	0.717	18.4	0.70	45.8
# 2	0.667	10.2	0.658	49.1	0.708	14.0	0.642	40.5
# 3	0.792	10.4	0.725	43.6	0.80	14.1	0.70	40.4
# 4	0.642	11.1	0.608	50.1	0.675	14.5	0.650	43.9
Average	**0.698**	**11.6**	**0.658**	**51.9**	**0.725**	**15.3**	**0.673**	**42.7**

Table 4.1: Results on Pami-09 (1).

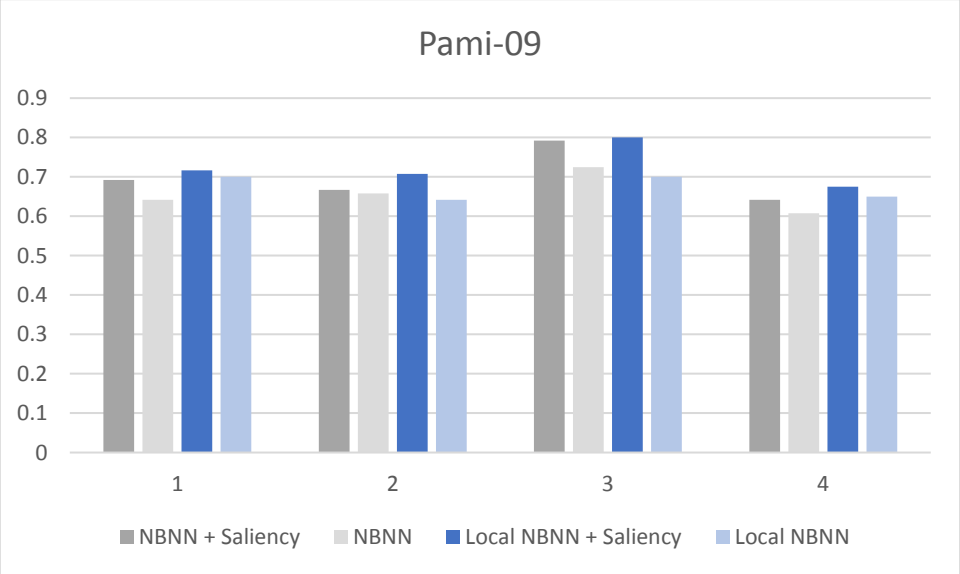

Table 4.2: Results on Pami-09 (2).

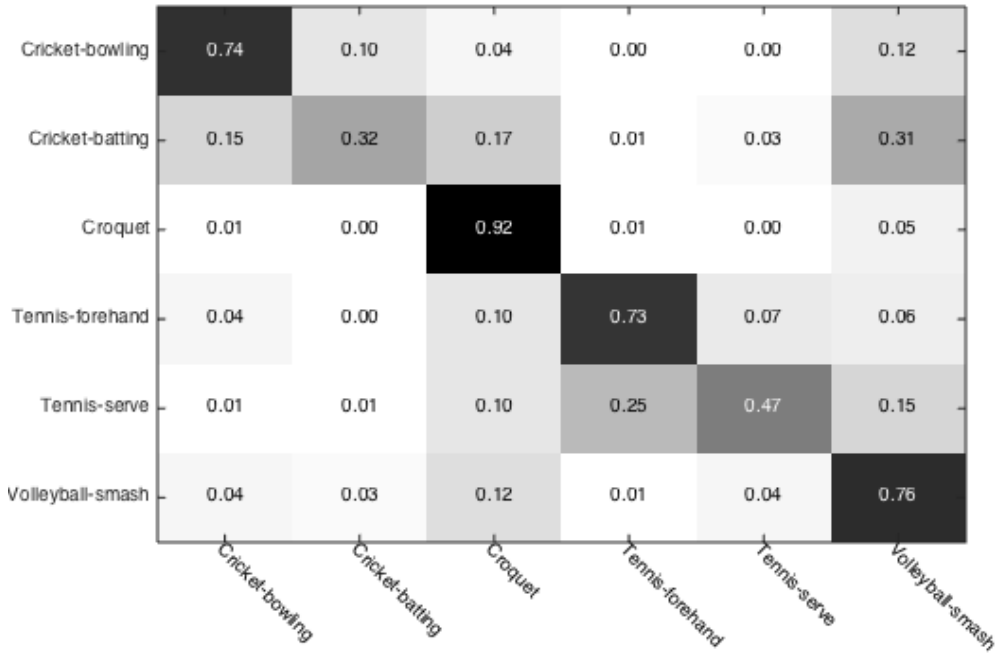

Table 4.3: Confusion matrix of NBNN on Pami-09.

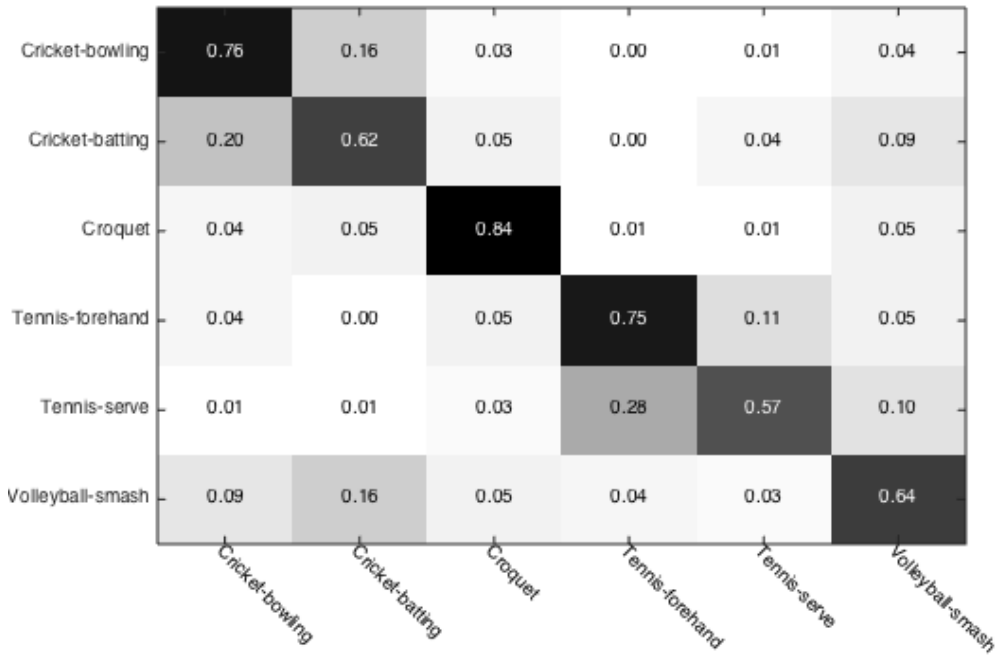

Table 4.4: Confusion matrix of context-aware NBNN on Pami-09.

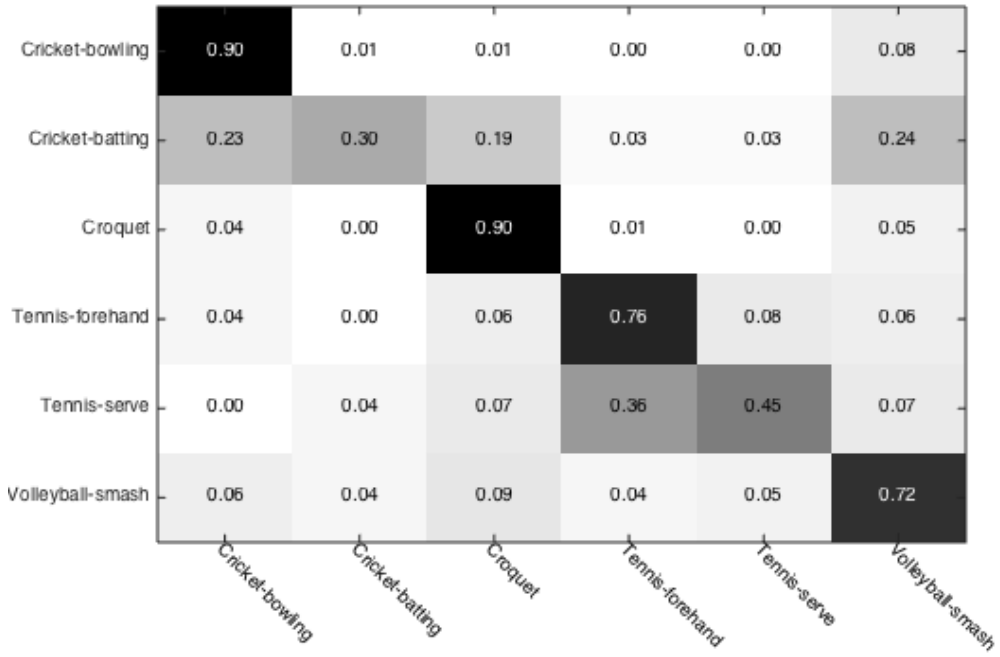

Table 4.5: Confusion matrix of local NBNN on Pami-09.

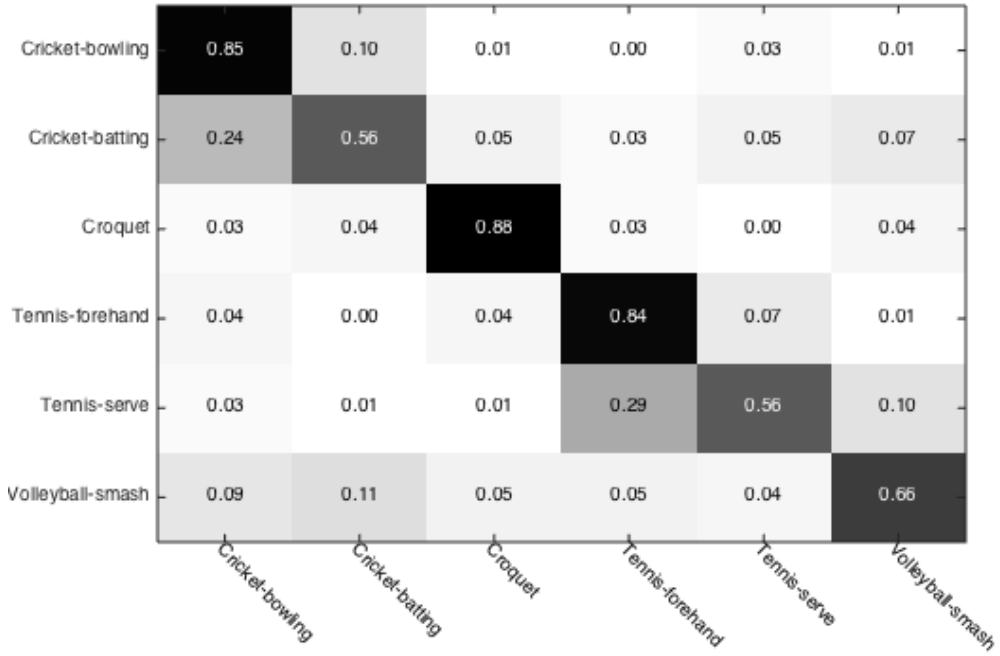

Table 4.6: Confusion matrix of context-aware local NBNN on Pami-09.

As can be seen from Table 4.1, based on saliency detection, the performances of NBNN and local NBNN have increased 3% and 5.2% respectively. Meanwhile, the time consumptions of NBNN and local NBNN have decreased by 77.6% and 64.2%. Furthermore, based on the information from the confusion matrices, the effectiveness on some unimpressive categories such as tennis-serve and cricket-batting has been greatly improved (at least 10%).

5.2. Caltech-5

The datasets used in our experiment are the initial versions that consist of five classes of objects: motorcycles, aeroplanes, human faces, cars and tree leaves [9, 10]. Each category has at least 186 images. Therefore, we take 50 images per class as reference, 20 for validation, and 50 for test.

The results are presented in Table 4.7 and Table 4.8. The average confusion matrices of NBNN and local NBNN on Caltech-5 are given in Table 4.9 to Table 4.12.

Observation (Sample)	***NBNN + saliency***		***NBNN***		***Local NBNN + saliency***		***Local NBNN***	
	Accuracy	Runtime (s)	Accuracy	Runtime (s)	Accuracy	Runtime (s)	Accuracy	Runtime (s)
# 1	0.964	21.8	0.956	217.2	0.972	28.1	0.972	173.6
# 2	0.952	26.2	0.968	222.3	0.964	31.8	0.960	171.9
# 3	0.984	20.7	0.972	217.6	0.980	27.1	0.976	174.5
# 4	0.956	17.5	0.952	195.1	0.964	25.6	0.964	153.8
Average	**0.964**	**21.6**	**0.962**	**213.1**	**0.970**	**28.2**	**0.968**	**168.5**

Table 4.7: Results on Caltech-5 (1).

Context-aware NBNN and local NBNN outperform original NBNN and local NBNN in average, with a minor lead of 0.2%. In other words, one wrongly labelled image by the original NBNN or local NBNN has been corrected out of every 500 images. Since the original NBNN and local NBNN have already achieved decent performance on this object category database,

the improvement is insignificant. However, context-aware classifiers only use 10.1% to 16.7% processing time of their counterparts.

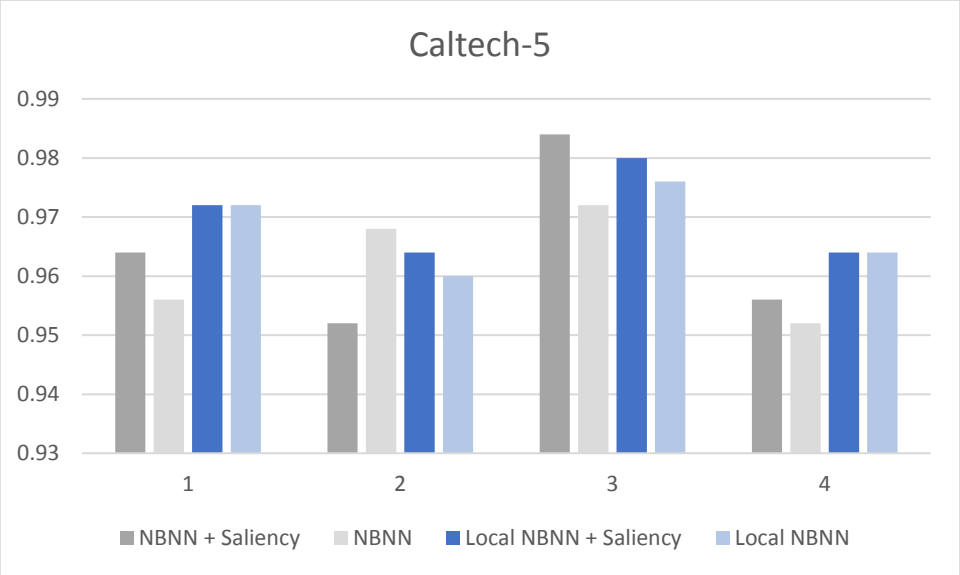

Table 4.8: Results on Caltech-5 (2).

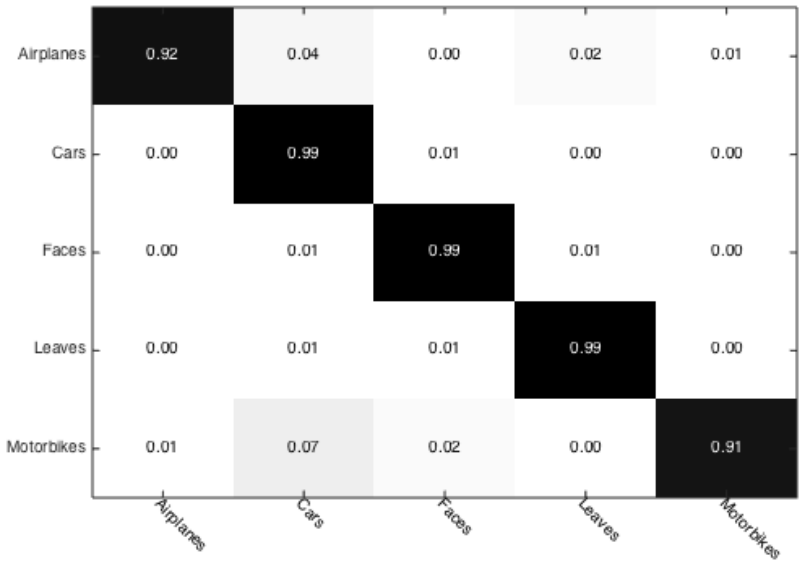

Table 4.9: Confusion matrix of NBNN on Caltech-5.

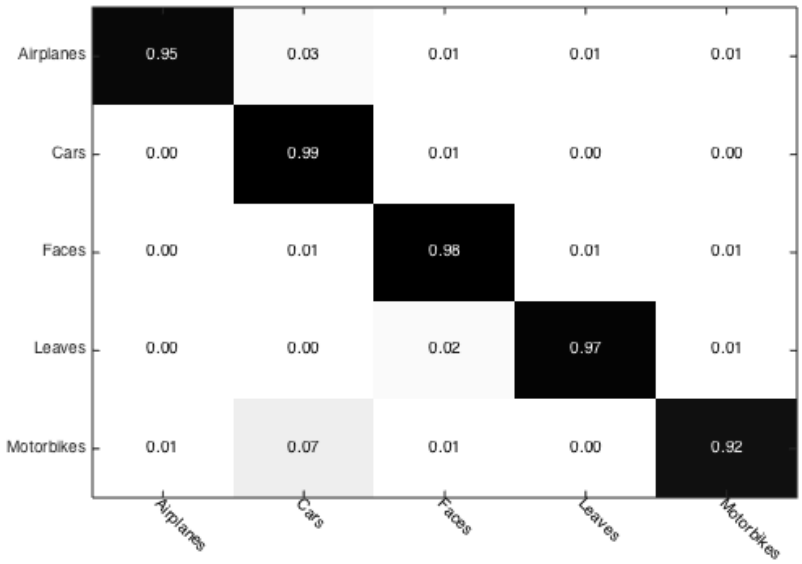

Table 4.10: Confusion matrix of context-aware NBNN on Caltech-5.

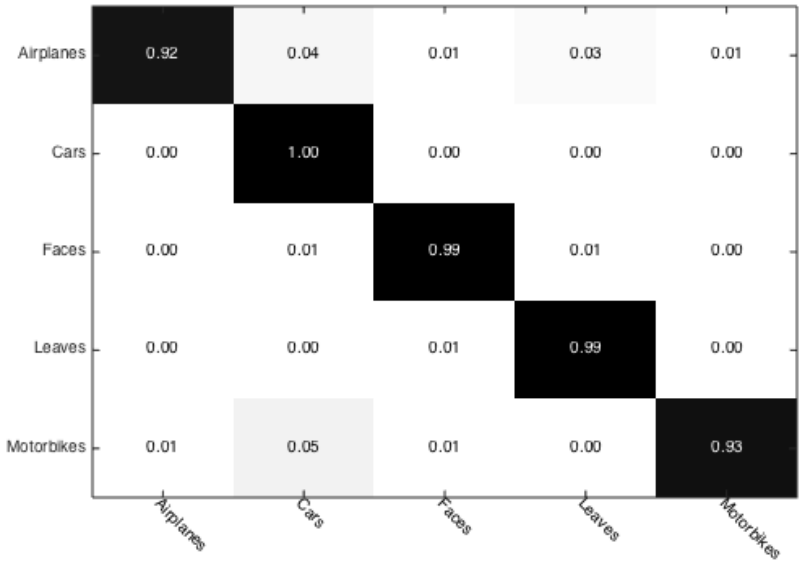

Table 4.11: Confusion matrix of local NBNN on Caltech-5.

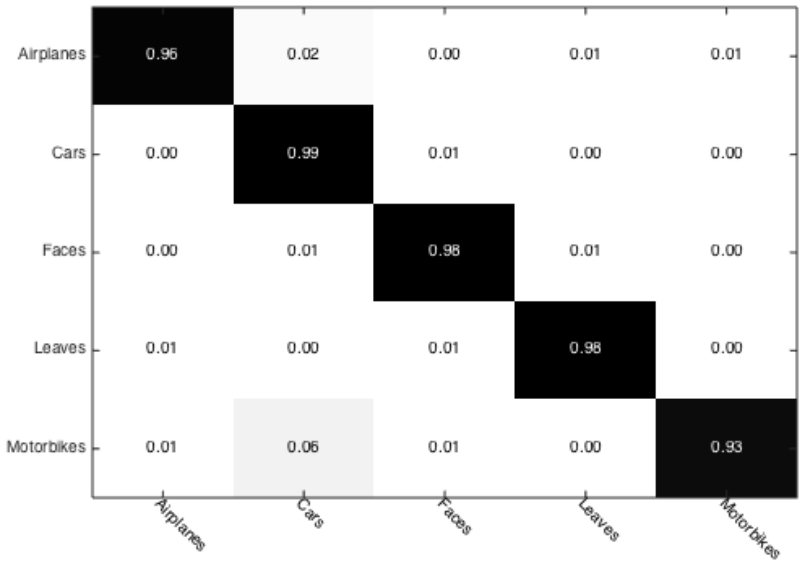

Table 4.12: Confusion matrix of context-aware local NBNN on Caltech-5.

5.3. 15-Scene

Thirteen of the fifteen classes are provided by [11, 12]. Lazebnik *et al*. [13] collect two other of them. The datasets contain 4485 images in total, with 200 to 400 per class. The images are from personal photographs and Google and they are all natural scenes with no artificialities.

Observation (Sample)	***NBNN + saliency***		***NBNN***		***Local NBNN + saliency***		***Local NBNN***	
	Accuracy	Runtime (s)	Accuracy	Runtime (s)	Accuracy	Runtime (s)	Accuracy	Runtime (s)
# 1	0.666	368.5	0.563	7099.6	0.653	425.4	0.531	6768.8
# 2	0.674	354.4	0.549	7127.2	0.649	425.5	0.543	6805.4
# 3	0.675	378.7	0.565	7280.0	0.623	420.4	0.527	6796.8
# 4	0.643	365.1	0.582	7111.4	0.628	431.6	0.564	6620.0
Average	**0.665**	**366.7**	**0.565**	**7136.5**	**0.638**	**425.7**	**0.541**	**6747.8**

Table 4.13: Results on 15-Scene (1).

We use 80 items per category as reference images, 40 for validation, and 80 for test. The performance of our framework compared to the original methods has been given in Table 4.13 and Table 4.14. The average confusion matrices of NBNN and local NBNN are listed in Table 4.15 to Table 4.18.

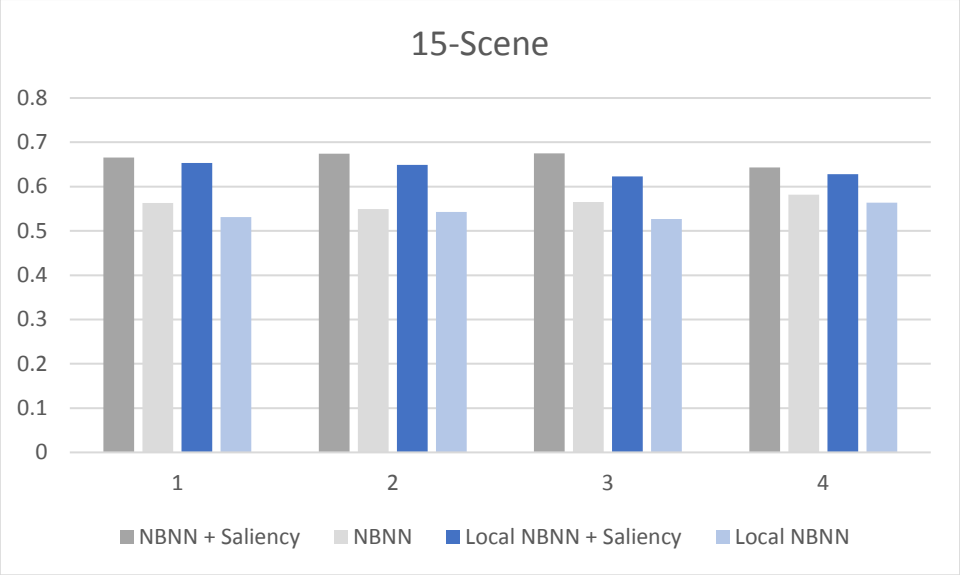

Table 4.14: Results on 15-Scene (2).

	CALsuburb	MITcoast	MITforest	MIThighway	MITinsidecity	MITmountain	MITopencountry	MITstreet	MITtallbuilding	PARoffice	bedroom	industrial	kitchen	livingroom	store
CALsuburb	0.41	0.02	0.03	0.00	0.24	0.01	0.23	0.04	0.01	0.00	0.00	0.00	0.00	0.00	0.00
MITcoast	0.00	0.86	0.00	0.03	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MITforest	0.00	0.00	0.89	0.00	0.00	0.02	0.07	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
MIThighway	0.00	0.08	0.00	0.80	0.03	0.01	0.02	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MITinsidecity	0.00	0.00	0.00	0.00	0.90	0.00	0.01	0.03	0.06	0.00	0.00	0.00	0.00	0.00	0.00
MITmountain	0.00	0.02	0.04	0.01	0.00	0.74	0.15	0.03	0.02	0.00	0.00	0.00	0.00	0.00	0.00
MITopencountry	0.00	0.21	0.00	0.01	0.00	0.03	0.73	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00
MITstreet	0.00	0.01	0.00	0.01	0.06	0.00	0.03	0.79	0.11	0.00	0.00	0.00	0.00	0.00	0.00
MITtallbuilding	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.01	0.95	0.00	0.01	0.00	0.00	0.00	0.00
PARoffice	0.00	0.01	0.00	0.00	0.46	0.00	0.00	0.01	0.07	0.23	0.11	0.00	0.07	0.03	0.00
bedroom	0.00	0.00	0.00	0.00	0.20	0.00	0.01	0.03	0.07	0.00	0.55	0.00	0.08	0.04	0.00
industrial	0.00	0.04	0.02	0.02	0.32	0.02	0.10	0.09	0.25	0.00	0.02	0.11	0.00	0.00	0.00
kitchen	0.00	0.00	0.00	0.00	0.43	0.00	0.00	0.00	0.09	0.01	0.14	0.00	0.31	0.02	0.00
livingroom	0.00	0.01	0.00	0.00	0.38	0.01	0.00	0.06	0.12	0.02	0.18	0.00	0.05	0.18	0.00
store	0.00	0.01	0.18	0.00	0.35	0.04	0.18	0.08	0.12	0.00	0.00	0.00	0.00	0.01	0.02

Table 4.15: Confusion matrix of NBNN on 15-Scene.

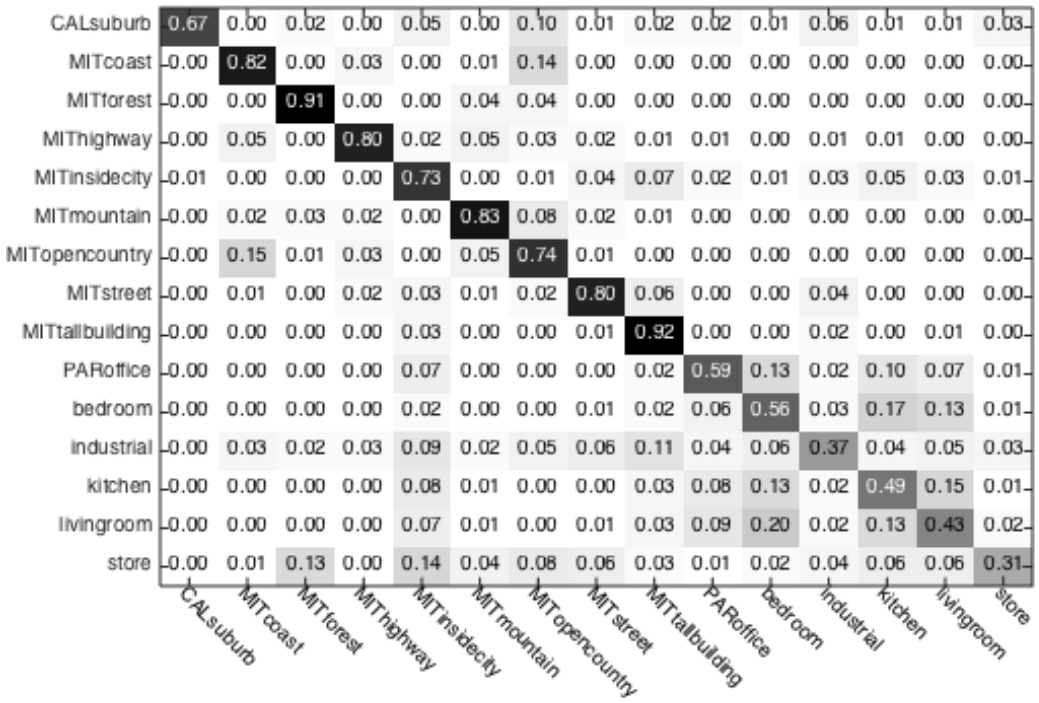

Table 4.16: Confusion matrix of context-aware NBNN on 15-Scene.

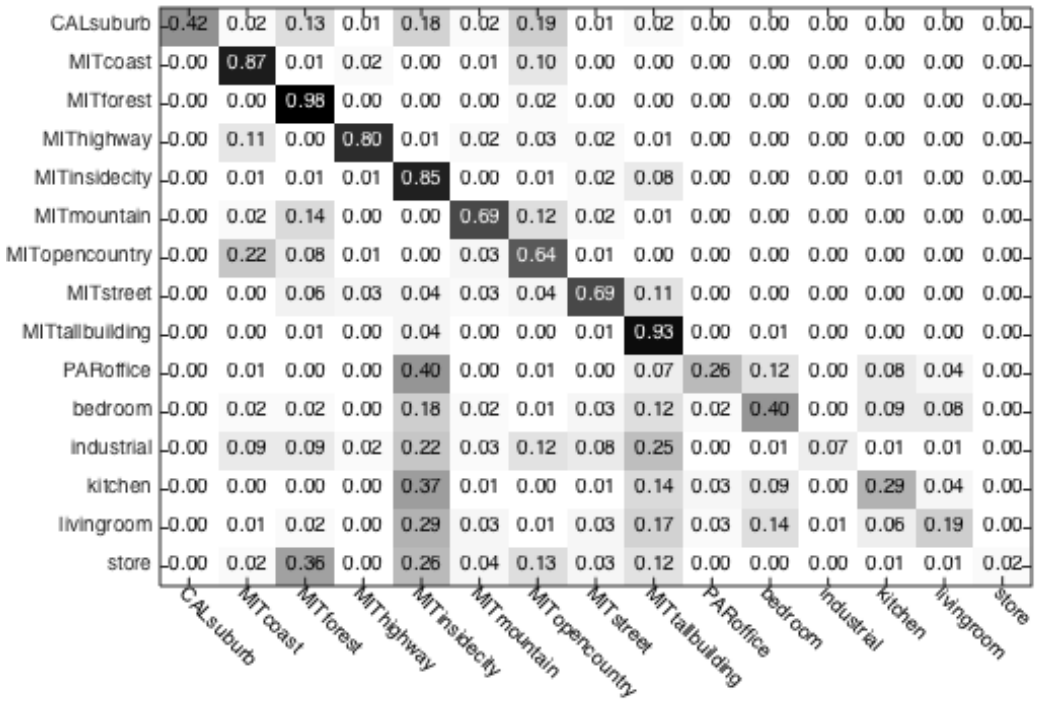

Table 4.17: Confusion matrix of local NBNN on 15-Scene.

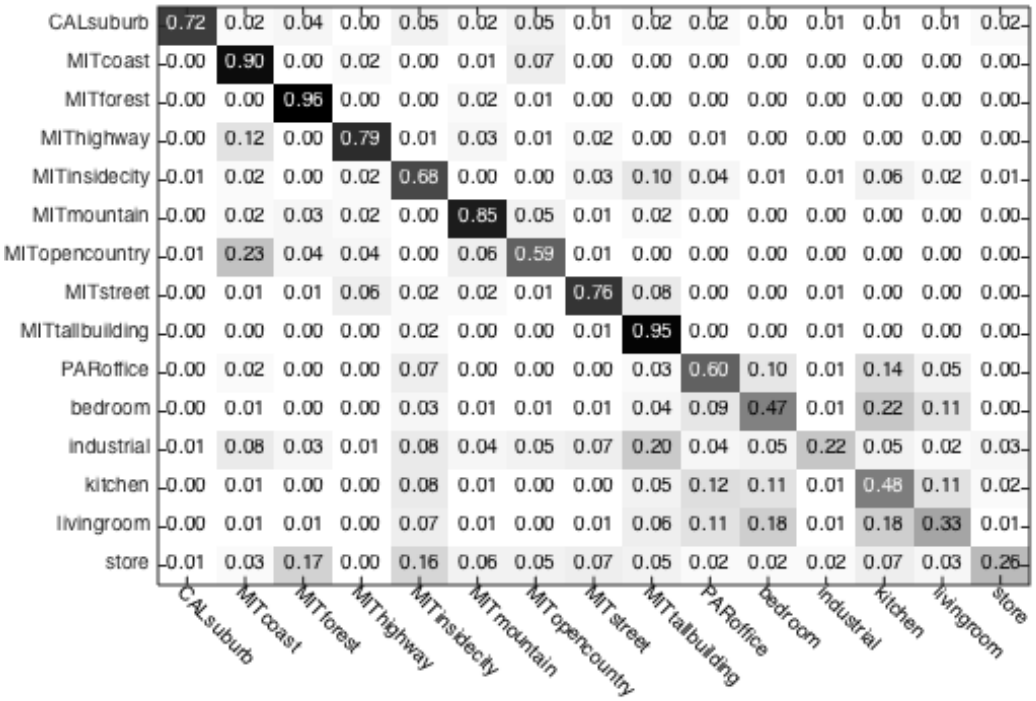

Table 4.18: Confusion matrix of context-aware local NBNN on 15-Scene.

Context-aware approaches show better performance in terms of all samples. On average, saliency based NBNN has an advantage of 10% over the original NBNN, while saliency based local NBNN is 9.7% ahead of local NBNN. At the same time, context-aware approaches save 94.9% to 93.7% running time, and are more efficient. More specifically, as can be seen from Table 4.15 to Table 4.18, context-aware approaches have enhanced the performance on a few classes that may bring confusion to the original NBNN and local NBNN (especially between MIT-inside-city and the bottom classes, such as industrial, kitchen, living room, and store).

5.4. Discussion & Summary

In this chapter, we have presented the performance of our novel approach on three public databases: Pami-09, Caltech-5, and 15-Scene. Overall, context-aware NBNN and local NBNN

have shown good results. Not only the effectiveness of classification but also the time cost has been improved. Specifically, context-aware classifiers lead the original ones on all samples, except one of Caltech-5 the context-aware NBNN lost (context-aware local NBNN still wins). Meanwhile, the time consumption has been reduced significantly in all cases. By grouping and calculating I2C distances between features from object and context separately, we limit feature searching in a more likely neighbourhood. This prevents the mistake of matching a foreground feature to a similar background feature in another category. Furthermore, there is a chance that different objects or human actions can have similar backgrounds when we compute the total distances of all the features. I2C may not be distant enough. If both object and context match, this gives us the confidence in assigning the two images a same label. If not, the voting scheme plays its role.

Generally, this method brings huge difference in classifying images from a database that has more complex context, such as 15-Scene. On other simple database that the original NBNN and local NBNN have already proved their efficiency, the enhancement is modest. Also, we cannot guarantee that the context-aware local NBNN uses less time than the context-aware NBNN when the original local NBNN uses less time than NBNN, even with a proper number of nearest searching neighbours selected [7]. This is due to the replacement of features using clustering. Further detailed reasons need to be investigated.

In the final chapter, we will conclude our works and propose possible future research directions to make this framework more powerful and more widely applicable.

Chapter VI

Conclusions & Future Work

We have proposed a unique NBNN classifier based on image contextual awareness. Based on the BoW model, the original NBNN and local NBNN put all reference features into a bag for each category. Then they compute the distances from the features of a test image to those bags, which are called I2C distances. In our framework, we further separate those reference features within one bag into smaller packs containing object and context respectively, based on saliency detection. The tuning and the role that the saliency detection plays have been comprehensively discussed. We have also demonstrated that by clustering the data inside each reference class the classification procedure can be accelerated. Using the produced anchor points carrying the discrimination of a class, the computation of the I2C distances of every single image feature has been bypassed. Therefore, the time consumption of NBNN and local NBNN have been reduced remarkably. More importantly, owing to the separation of salient regions and image context generated by the saliency maps, we have enhanced the performance of both NBNN and local NBNN. The class label estimation given by the voting of different regions of an image is more robust. This improvement has been verified on three databases: Pami-09, Caltech-5, and 15-Scene. Generally, the improvement is more significant when the datasets contain complex contextual information, such as on 15-Scene.

Possible future research directions include improving context-aware naive Bayes nearest neighbour classifiers by using multiple features, instead of a single SIFT, or finding a linear or

non-linear combination of the salient and context image features, which is capable of raising the discrimination of I2C distances.

Bibliography

[1] Boiman, O., Shechtman, E., and Irani, M. (2008). In defense of nearest-neighbor based image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1-8.

[2] Tuytelaars, T., Fritz, M., Saenko, K., and Darrell, T. (2011). The NBNN kernel. In *IEEE International Conference on Computer Vision*, pages 1824-1831.

[3] Behmo, R., Marcombes, P., Dalalyan, A., and Prinet, V. (2010). Towards optimal naive Bayes nearest neighbor. In *European Conference on Computer Vision*, pages 171-184. Springer Berlin Heidelberg.

[4] Wang, Z., Hu, Y., and Chia, L. T. (2010). Image-to-class distance metric learning for image classification. In *European Conference on Computer Vision*, pages 706-719. Springer Berlin Heidelberg.

[5] Liu, L., Wang, L., and Liu, X. (2011). In defense of soft-assignment coding. In *IEEE International Conference on Computer Vision*, pages 2486-2493.

[6] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. (2010) Locality-constrained linear coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3360-3367.

[7] McCann, S., and Lowe, D. G. (2012). Local naive Bayes nearest neighbor for image classification. In *IEEE Conference on Computer Vision and Pattern* Recognition, pages 3650-3656.

[8] Gupta, A., Kembhavi, A., and Davis, L. S. (2009). Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(10), 1775-1789.

[9] Fei-Fei, L., Fergus, R., and Perona, P. (2003). A Bayesian approach to unsupervised one-shot learning of object categories. In *IEEE International Conference on Computer Vision*, pages 1134-1141.

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

[11] Oliva, A., and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, *42*(3), 145-175.

[12] Fei-Fei, L., and Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 524-531.

[13] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2169-2178.

[14] Bart, E., and Ullman, S. (2004). Class-based matching of object parts. In *IEEE Workshop on Computer Vision and Pattern Recognition*, page 173.

[15] Tuytelaars, T., and Schmid, C. (2007). Vector quantizing feature space with a regular lattice. In *IEEE International Conference on Computer Vision*, pages 1-8.

[16] Borji, A., and Itti, L. (2013). State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 185-207.

[17] Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(11), 1254-1259.

[18] Greenspan, H., Belongie, S., Goodman, R., Perona, P., Rakshit, S., and Anderson, C. H. (1994). Over-complete steerable pyramid filters and rotation invariance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 222-228.

[19] Harel, J., Koch, C., and Perona, P. (2006). Graph-based visual saliency. In *Advances in Neural Information Processing Systems*, pages 545-552.

[20] Hou, X., and Zhang, L. (2007). Saliency detection: A spectral residual approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1-8.

[21] Tavakoli, H. R., Rahtu, E., and Heikkilä, J. (2011). Fast and efficient saliency detection using sparse sampling and kernel density estimation. In *Scandinavian Conference on Image Analysis*, pages 666-675. Springer Berlin Heidelberg.

[22] Hou, X., Harel, J., and Koch, C. (2012). Image signature: Highlighting sparse salient regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(1), 194-201.

[23] Murray, N., Vanrell, M., Otazu, X., and Parraga, C. A. (2011). Saliency estimation using a non-parametric low-level vision model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 433-440.

[24] Vikram, T. N., Tscherepanow, M., and Wrede, B. (2012). A saliency map based on sampling an image into random rectangular regions of interest. *Pattern Recognition*, *45*(9), 3114-3124.

[25] Rahtu, E., Kannala, J., Salo, M., and Heikkilä, J. (2010). Segmenting salient objects from images and videos. In *European Conference on Computer Vision*, pages 366-379. Springer Berlin Heidelberg.

[26] Seo, H. J., and Milanfar, P. (2009). Static and space-time visual saliency detection by self-resemblance. *Journal of Vision*, *9*(12), 15-15.

[27] Erdem, E., and Erdem, A. (2013). Visual saliency estimation by nonlinearly integrating features using region covariances. *Journal of Vision*, *13*(4), 11-11.

[28] Riche, N., Mancas, M., Duvinage, M., Mibulumukini, M., Gosselin, B., and Dutoit, T. (2013). Rare2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis. *Signal Processing: Image Communication*, *28*(6), 642-658.

[29] Mancas, M. (2008). Relative influence of bottom-up and top-down attention. In *International Workshop on Attention in Cognitive Systems*, pages 212-226. Springer Berlin Heidelberg.

[30] Riche, N., Mancas, M., Gosselin, B., and Dutoit, T. (2012). Rare: A new bottom-up saliency model. In *IEEE International Conference on Image Processing*, pages 641-644.

[31] Bruce, N., and Tsotsos, J. (2005). Saliency based on information maximization. In *Advances in Neural Information Processing Systems*, pages 155-162.

[32] Hyvärinen, A., Karhunen, J., and Oja, E. (2004). *Independent Component Analysis* (Vol. 46). John Wiley & Sons.

[33] Zhang, L., Tong, M. H., Marks, T. K., Shan, H., and Cottrell, G. W. (2008). SUN: A Bayesian framework for saliency using natural statistics. *Journal of Vision*, *8*(7), 32-32.

[34] Torralba, A., Oliva, A., Castelhano, M. S., and Henderson, J. M. (2006). Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological Review*, *113*(4), 766.

[36] Rosenholtz, R. (1999). A simple saliency model predicts a number of motion popout phenomena. *Vision Research*, *39*(19), 3157-3163.

[37] Simoncelli, E. P., and Freeman, W. T. (1995). The steerable pyramid: a flexible architecture for multi-scale derivative computation. In *IEEE International Conference on Image Processing*, pages 444-447.

[35] Judd, T., Ehinger, K., Durand, F., and Torralba, A. (2009). Learning to predict where humans look. In *IEEE International Conference on Computer Vision*, pages 2106-2113.

[38] Itti, L., and Koch, C. (2000). A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, *40*(10), 1489-1506.

[39] Viola, P., and Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, *57*(2), 137-154.

[40] Felzenszwalb, P., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1-8.

[41] Goferman, S., Zelnik-Manor, L., and Tal, A. (2012). Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(10), 1915-1926.

[42] Takeda, H., Farsiu, S., and Milanfar, P. (2007). Kernel regression for image processing and reconstruction. *IEEE Transactions on Image Processing*, *16*(2), 349-366.

[43] Seo, H. J., and Milanfar, P. (2010). Training-free, generic object detection using locally adaptive regression kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(9), 1688-1704.

[44] Fu, Y., and Huang, T. S. (2008). Image classification using correlation tensor analysis. *IEEE Transactions on Image Processing*, *17*(2), 226-234.

[45] Fu, Y., Yan, S., and Huang, T. S. (2008). Correlation metric for generalized feature extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(12), 2229-2235.

[46] Ma, Y., Lao, S., Takikawa, E., and Kawade, M. (2007). Discriminant analysis in correlation similarity measure space. In *International Conference on Machine Learning*, 577-584.

[47] Lowe, D. G. (2004). Distinctive image features from scale-invariant key-points. *International Journal of Computer Vision*, *60*(2), 91-110.

[48] Yang, J., Yu, K., Gong, Y., and Huang, T. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1794-1801.

[49] Hartigan, J. A., and Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *28*(1), 100-108.