Deliverable D6.2.1

Description, analysis and evaluation of confidence estimation procedures for sub-categorisation

Contract number: **FP7–216529** PinView

Personal Information Navigator Adapting Through Viewing

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under *grant agreement* n° 216529.

Identification sheet

Project ref. no.	**FP7–216529**
Project acronym	PinView
Status and version	Final, Revision: 1.0
Contractual date of delivery	31.03.2009
Actual date of delivery	09.04.2009
Deliverable number	D6.2.1
Deliverable title	Description, analysis and evaluation of confidence estimation procedures for sub-categorisation
Nature	report
Dissemination level	PU – Public
WP contributing to the deliverable	WP6 Semantic sub-categorisation features
Task contributing to the deliverable	Task 6.2 Information fusion and confidence
WP responsible	Xerox Research Centre Europe
Task responsible	Xerox Research Centre Europe
Editor	Teófilo E. de Campos, <t.de-campos@xrce.xerox.com>
Editor address	XRCE, 6, chemin de Maupertuis, 38240 Meylan, France
Authors in alphabetical order	Haider Ali, Martin Antenreiter, Peter Auer, Gabriela Csurka, Teófilo E. de Campos, Zakria Hussain, Jorma Laaksonen, Ronald Ortner, Kitsuchart Pasupa, Florent Perronnin, Craig Saunders, John Shawe-Taylor, Ville Viitaniemi
EC Project Officer	Pierre-Paul Sondag
Keywords	visual saliency, bag of patches, information fusion, confidence estimation
Abstract	This report presents contributions in two main areas: the combination of low level image features with visual saliency maps and use of confidence measures for information fusion. For the first part, we show experiments with automatic saliency estimation methods based on bottom-up and top-down approaches. We also explored maps generated by mouse-clicks. These maps were used to give weights to local image features which are then used for image categorisation in an approach based on bag-of-patches. For the second part, we explored methods to associate confidence values to predictions of each test sample. These values are used to give per-sample weights for different information sources in classifier combination.

List of annexes

None.

Contents

1 Overview

This is a Deliverable of the *Personal Information Navigator Adapting Through Viewing*, PinView, project, funded by the European Community's Seventh Framework Programme under Grant Agreement n° 216529.

The report constitutes an output of the Work Package 6 *Semantic sub-categorisation features* which aims at improving the state-of-the-art in visual categorisation performance. The main research outcomes of Task 6.2 *Information fusion and confidence* are described in this report. The prototype developed for this task, which constitutes Deliverable 6.2.2 is described in another report [8].

The research developed for this task consists in the following threads:

1. Fusion of visual saliency information and low-level features. This is a straightforward continuation of Task 6.1.

2. Investigation of confidence estimation methods for information fusion.

For the first thread, in addition to the method presented in Section 6.3 of the deliverable 6.1 [7], a number of methods for automatic estimation of visual saliency were implemented. These methods were used to weight local features in the same manner that was done when using eye gaze data, described in D6.1. Automatic methods enable the evaluation of our weighting scheme on large datasets, such as the PASCAL VOC2007 dataset, including all the 20 classes of objects. Our results with a top-down method of the automatic saliency estimation methods are promising. We also evaluated the use of saliency maps generated using the object locations pointed using mouse clicks. The categorisation results obtained with this method represent a significant improvement over the unweighted baseline. This task involved collaboration between XEROX and TKK in the generation of top-down saliency maps.

The second thread about the use of confidence estimation methods for information fusion has had a high level of collaborative work with contributions from XEROX, SOTON, UCL and MUL. The main goal of this thread is to associate a confidence score to each test sample, for each source of information. These confidence weights are associated to each classifier, potentially leading to better results than a naïve combination. We evaluated methods for three different types for confidence estimation:

- a method based on p-values of classifiers;
- a method based on distance to the margin;
- methods based on density estimation.

The work presented in this report will be continued mainly in PinView task 6.3 *Sub-categorisation with limited data*, which will use information transfer in order to further increase the robustness of an image categorisation system with limited data. Other WPs will benefit from the research outcomes from this task. For instance, the investigations on the estimation and use of saliency maps should provide insights to T2.2 *Relevance of parts of an image from the viewing pattern*. The same is true for the work on information fusion with confidence w.r.t. WP3 and WP4, since both these WPs should benefit from a combination of sources of information. The other outcome of Task 6.2, described in D6.2.2 [8] should facilitate collaborations and be helpful in the implementation of a final prototype in WP8.

2 Introduction

This document reports the work on two different fronts related to the broad concept of using confidence estimation and fusion for image categorisation. In the first part (Section 3), we focus on sub-image level confidence and explore the use of visual saliency maps as means of indicating the relevance of local features. The method described in D6.1 [7] which was evaluated with saliency maps obtained with eye gaze data is evaluated here with saliency maps obtained using automatic methods. We evaluate one bottom-up method and two top-down methods. Additionally, we also evaluate saliency maps computed using explicit feedback provided by mouse clicks.

An obvious advantage of the automatic methods is that it is much easier to evaluate them with large datasets, since they do not require gaze tracking measurements from several users. For the saliency maps based on mouse clicks, we took advantage of the laborious annotation that is provided for the PASCAL VOC challenges. The annotators have drawn bounding boxes around instances of all the 20 objects of interest in all the images of VOC 2007 and VOC 2008 (approximatelly ten thousand images per dataset).

The second part of this report (Section 4) focuses on sample-based confidence. It explores the use of different measures of confidence associated to new samples in a visual categorisation task. The goal is to use confidence scores to weight classifiers using a combination of different sources of information. In the experiments presented here, the sources of information consist of different types of low level visual features. This report concludes in Section 5.

3 Images as Sets of Weighted Features: Experiments with more Saliency Estimation Methods

This section starts with a brief review of the weighted Fisher kernel method proposed in D6.1 (Section 3.1). This is a framework that has proven useful for improving image categorisation [7] performance using saliency maps obtained from gaze data. Nevertheless, gaze data is not often available. In the absence of visual attention data obtained from gaze measurements, we can resort to some automatic saliency detection methods. Therefore, in Section 3.2 we describe some of the automatic visual saliency estimation methods that are evaluated with the weighted features framework in Section 3.3. Conclusions about these experiments are drawn in Section 3.4.

3.1 A Brief Recall of the Weighted Fisher Kernel Method

In this paragraph, we briefly recall the Weighted Fisher Kernel Method, which is described in detail in [7]. The main idea is to incorporate location relevance weights in the Fisher kernels method based image categoriser proposed in [29].

If the image is represented by a set of low level features of dimension d $\mathbf{X} = \{\mathbf{x}_t\}_{t=1..T}$ ($\mathbf{x} \in \mathbb{R}^d$), the image is characterised in the original approach by the following gradient vector:

$$\mathcal{L}(\mathbf{X}|\lambda) = \frac{1}{T}\nabla_\lambda \log p(\mathbf{X}|\lambda) = \frac{1}{T}\sum_{t=1}^{T} \nabla_\lambda \log p(\mathbf{x}_t|\lambda) \tag{1}$$

where we assume independence between the observations $\mathbf{x}_t$ and where each observation contributes equally to the global likelihood.

Here in contrast, the contribution of each individual feature vector $\mathbf{x}_t$ is weigthed based on its location relevance (see details in [7]):

$$\mathcal{L}(\mathbf{X}|\lambda) = \frac{1}{T}\sum_{t=1}^{T} \psi_t \log p(\mathbf{x}_t|\lambda) \tag{2}$$

where $\sum_t \psi_t = T$.

The location relevance ψ_t of a feature[1] can be estimated by several methods. They might come from user feedback in the form of explicit or implicit feedback or by some automatic method using the image content itself. In [7] we explored the former methods, especially the implicit feedback provided by the eye fixation captured by a gaze tracking system. The experiments showed that eye gaze data provides crucial location relevance information that can be used to improve categorisation results if incorporated as weights in the Fisher Kernel categorisation method[2]

However, gaze tracking systems are not yet ubiquitous and getting eye fixation data from a large set of images is impractical. In the absence of visual attention data obtained from gaze measurements, we can resort to automatic methods. These methods may not be able to give a very accurate location relevance measure, but they are more practical and can be much faster than getting gaze data. Therefore, in the first part of this section (Subsection 3.2), we show several possibilities to automatically obtain location relevance weights and then in Subsection 3.3 compare them based on how they influence the categorization accuracy.

3.2 Automatic Methods to Obtain Location Relevance Weights

This section describes several methods that can be used to automatically compute the location relevance weights ψ_t of a feature.

As our previous experiments with location relevance with eye fixation were successful, a first obvious choice would be to use methods that try to model the human visual attention. Most models for visual saliency detection and thumbnail extraction were inspired by the human visual system and can be grouped in bottom-up, top-down and hybrid approaches:

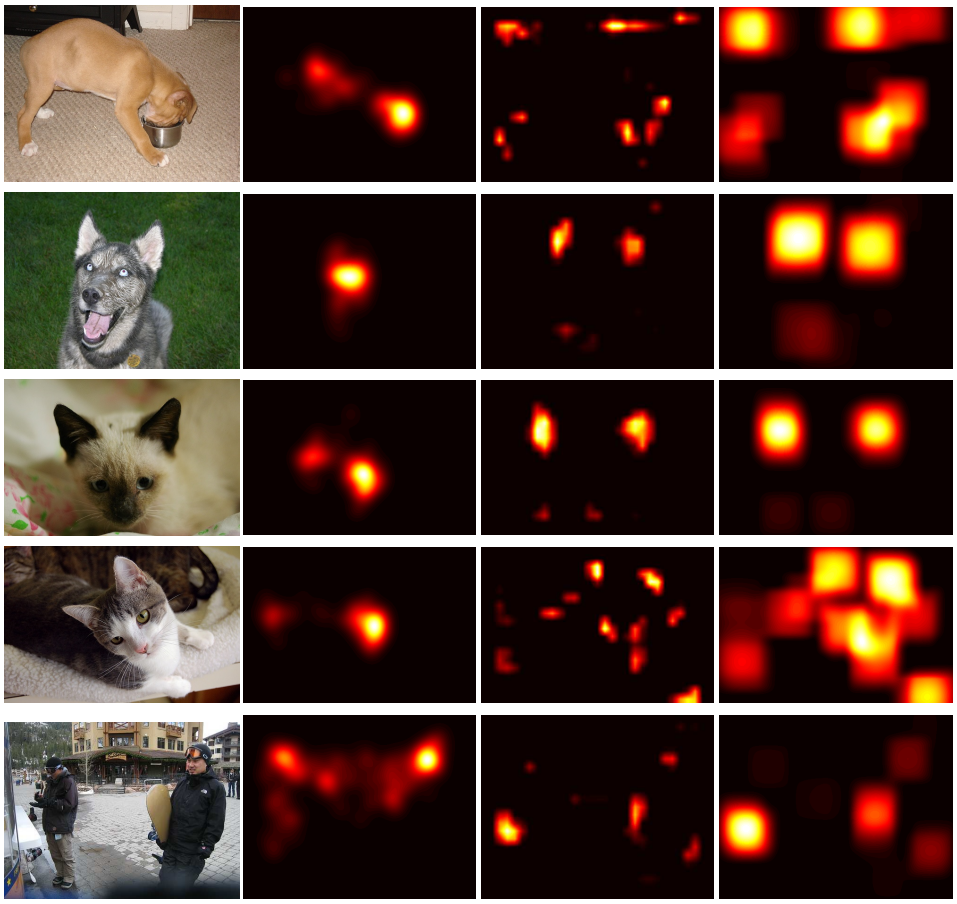
- **Bottom-Up Methods** Human attention is interpreted by some as a cognitive process that selectively concentrate on the most unusual aspects (visual surprise) of an environment while ignoring more common things. To model this behavior, various approaches were proposed based on the extraction of a set of intrinsic low level features (contrast, color, orientation) and process images without considering any high-level information of the image contents. These methods are usually based on heuristic models of biological vision systems [16, 36, 20]. Interest point detectors [25, 17] can be seen also as a particular bottom-up saliency detection model.

- **Top-Down Methods** Top-down visual attention processes are considered driven by voluntary control, and related to the observer's goal when analyzing a scene [38] and can give different responses, depending on the search task or the object of interest. These methods take into account higher order information about the image, such as context [39] , structure and often model task-specific visual search [26, 41]. They usually require a training phase, in which models for object of interest, image context or scene categories are learned. Object detection and localization can be seen as a particular case of top-down saliency detection.

- **Hybrid**. Most of the saliency detection methods are hybrid models leveraging the combinations of the bottom-up and top-down approaches [16, 3, 37, 44]. In general, they are structured in two levels, a top-down layer filters out noisy regions in saliency

[1] Note, that a feature $\mathbf{x}_t$ corresponds to a local image patch/region on which the low level feature was computed.

[2] Actually, we also showed that similar relative improvements can be obtained on classical bag-of-visual-words (BoV) [34, 4] by adding weights to the feature occurences. Nevertheless, in both cases (weighted and unweighted) we obtain lower performances than with the Fisher representation. Therefore, in this deliverable, our experiments were done only with the latter.



(a) (b) (c) (d)

Figure 1: Bottom-up saliency maps computed for sample images (a) using: (b) gaze data of 28 people ($\sigma = 2.8\%$ of the image height); (c) Itti and Koch's method (I&K) [16]; and (d) I&K's map smoothed out with the same σ used for gaze data.

maps created by the bottom-up layer. In most cases, the top-down component is actually reduced to a human face detector [16, 37] or face and text detector [3].

In what follows we describe briefly those methods which were used in our experiments.

3.2.1 Bottom-Up Saliency Maps

We have done experiments with the bottom-up saliency estimation method of Itti and Koch [16] (more specifically, the re-implementation of [43]). This model is based on the analysis of multi-scale descriptors of colour, intensity and orientations using linear filters. Center-surround structures are used to compute how much features stand out of their surroundings. The outputs of the three descriptors are combined linearly leading to a continuous map of relevance. Figure 1 shows a few examples of bottom-up saliency maps obtained with Itti and Koch's method without (c) and with Gaussian smoothing (d).

However, it was noticed that people have an overall tendency to fixate to the center-most region of the images [11]. Therefore, we also experimented with a simple method where we assumed that the image center is the only fixation point and we centered different sized non-isotropic (but with diagonal Σ) Gaussians on it. This obviously leads to a saliency map where the weights of the patches are decreasing with their distance from the center. Figure 2 shows a few such maps with varying Σ.

(a) (b) (c) (d)

Figure 2: Non-isotropic Gaussian centered saliency maps computed for sample images (a) using the following (diagonal) values for Σ: (b) $\sigma_w = 0.3w$ and $\sigma_h = 0.3h$ where w and h are the width and height of the image (c) $\sigma_w = 0.2w$ and $\sigma_h = 0.2h$ and (d) $\sigma_w = 0.1w$ and $\sigma_h = 0.1h$.

3.2.2 Top-Down Saliency Maps

We have to distinguish here two main cases. In the first case, we are learning a binary class that discriminates between salient and non-salient regions. In this case, we are not learning a specific semantic class and therefore the method leads to a single salient map. In the second case, one or several semantic visual classes are defined, and the relevance of a region is related with a given visual class. In this case we have a relevance map per visual category that we also call *class probability maps*. Nevertheless, both cases can be handled in a single example based learning framework. In the former case we have training salient and non-salient regions; in the latter case semantically labeled regions.

To create such top-down saliency maps, many object detection or semantic segmentation algorithms can be applied. We briefly describe here three methods with which we experimented:

1. ***Fisher Kernel-based Patch Class Probability Maps***

   The main idea of this method proposed in [5] is that each local patch is represented with high-level descriptors based on the Fisher Kernel. Patch level linear classifiers are then trained on labeled examples allowing us to score each local patch according to its class relevance. These posterior probabilities can further be propagated to pixels leading to class probability maps. Depending on what examplary labels are used, they can be either saliency vs non-saliency maps (a single class) or a set of semantic class maps (several classes such as people, car, cow, etc).

   Figure 3 shows the main steps of this approach. In a nutshell, given an image the proposed approach works as follows. First, patches are detected at a multi-scale grid and low-level descriptors (local RGB statistics and SIFT-like features) are computed for each patch. In both feature spaces (color and texture) a visual vocabulary is built using a Gaussian Mixture Model (GMM) where each Gaussian corresponds to a visual word. Given a low-level descriptor and the generative model (GMM), each patch is described by a high-level representation $\nabla_\lambda \log p(\mathbf{x}_t|\lambda)$ using Fisher Kernel framework (see also [29, 7] and equation (1)).

   These high-level patch descriptors are labeled based on the intersection of the corresponding patch with relevant regions in training data[3] then a linear sparse logistic

[3]Here, "relevant regions" are regions with some foreground label or labelled as "salient".

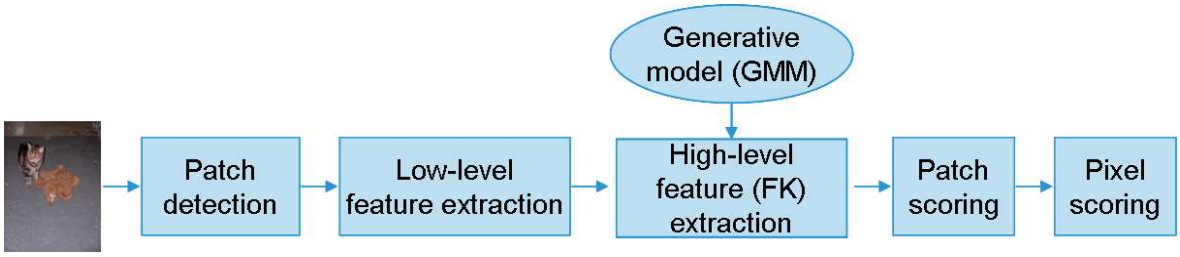


Figure 3: A scheme illustrating the Fisher Kernel based class probability maps creation.

regression method [21] is trained for each relevance class. For test images, similar high-level descriptors are computed for each patch that are scored with respect to each class. These scores are converted into probabilities and propagated from the patch center to each pixel with Gaussian smoothing, generating a smooth saliency map for each class. The saliency maps from different feature types (color statistics and SIFT-like texture) are combined (product) to obtain the class probability maps.

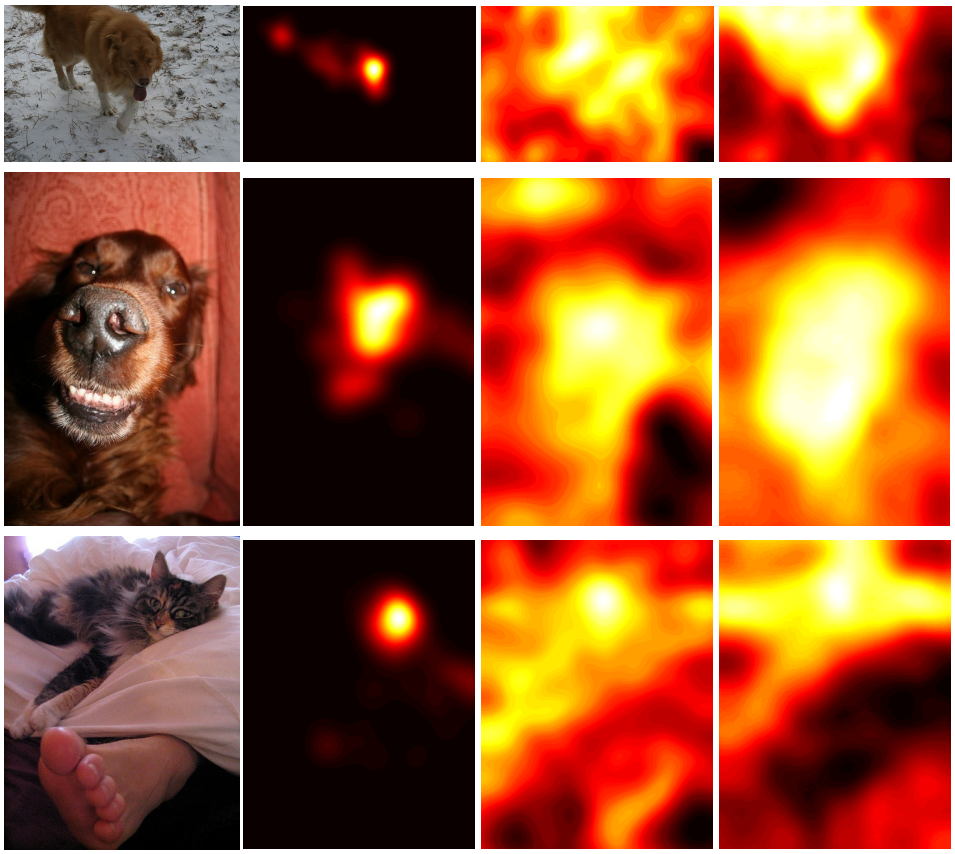
2. ***Learning saliency from labeled nearest neighbours***

   This method proposed in [23] is based upon a simple underlying idea: images sharing visual appearance are likely to share similar salient regions. Following this principle (see also Figure 5), for each training image the K most similar images are retrieved from an indexed database and Fisher Kernel based high level descriptors were computed for each patch in the retrieved images. These patches were labeled according to manual annotations available and the color and texture high level descriptors are concatenated. Collecting independently the salient high level descriptors and the non- salient high level descriptors, we can build two Fisher Vectors using equation (1) that can be seen as salient (foreground) and non-salient (background) models. Then for each patch in the test image we again concatenate the color Fisher Vector and the texture Fisher Vector and are concatenated and based on its similarity to the foreground and background models we compute a saliency score for it. Then, similarly as above, the scores are propagated from patches (or sub-windows) to pixels generating a smooth saliency map (see for further details [23]). In Figure 6(c) we show examples of saliency maps obtained with this method on a few images.

3. ***Tree-structured Self-organizing Map-based Relevant Image Region Detection***

   This saliency map generation method takes two sets of training images as input. The generated saliency map pinpoint the image regions that are chacteristic to either one of the image sets but not both. For the experiments of this report, we always take one of the image sets to be the complement of the other set, for example set of cat and not-cat images. This saliency map generation principle has the property that background as well as foreground regions of an image can be identified as discriminative. In our example, cat images might be recognised by the bedroom environment where cats are often photographed as well as by the whiskers and nose of the cat itself. The relative importance of these two characteristics of cat images depends on the content of the non-class images, i.e. whether the non-class images also display bedroom scenes or contain other animals with whiskers.

   Technically, the saliency estimation problem is converted to a supervised classification problem by extracting a number of rectangular patches from each training image. On average, approximately 140 patches are extracted from each image. The visual properties of each patch are described with three feature vectors: a 256-dimensional colour



(a) (b) (c) (d)

Figure 4: Top-down saliency maps computed for sample images (a). While in (c) we estimated salient maps learned from eye tracking data (in (b) is the corresponding ground truth we tried to estimate), in (d) we used the labeled bounding boxes (cats,dogs) to learn class related (pet) salient maps.

histogram, a 128-dimensional edge distribution descriptor and a 128-dimensional histogram of interest point SIFT descriptors. A separate supervised soft classifier, based on the Tree-structured Self-Organizing Map (TS-SOM) clustering algorithm [19], is trained for each of the three feature spaces to separate the patches originating from the two image classes. Further details can be found in [41, 7].

When the saliency map for a novel test image is to be estimated, the image is divided into patches and each one described with the three feature vectors. For each feature, a score value is estimated with the corresponding soft TS-SOM classifier. Positive scores indicate that the patch is more likely to come from the images of the first image class rather than the second, and negative scores the opposite. Elementary classifier fusion is performed by summing the three feature-wise scores for each patch. After this, the absolute value of the score is taken, reflecting the confidence that the patch is characteristic to exactly one of the image classes. Finally, the rectified patch scores are interpolated to obtain a smooth saliency map for the whole image. Figure 7 shows a few example maps obtained with this method.

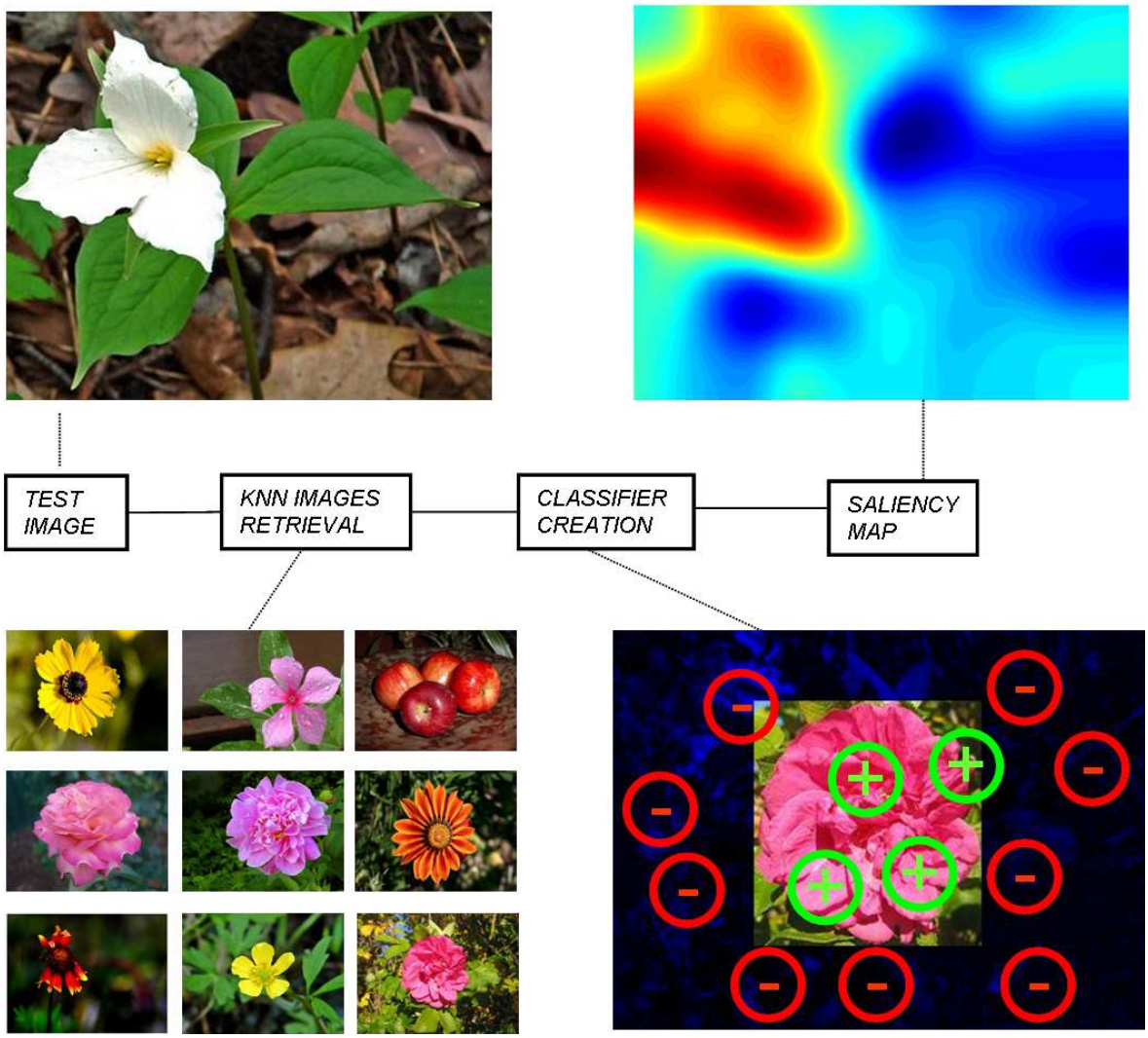


Figure 5: A schema illustrating the learning saliency from labeled nearest neighboors approach.

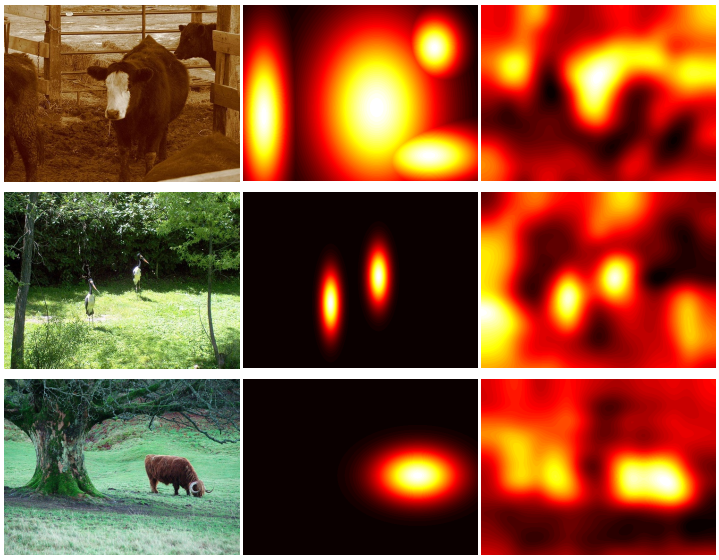
3.2.3 Location Relevance Weights from Saliency Maps

Different strategies can be used to compute the relevance weight ψ_t of a patch $\mathbf{x}_t$ from a saliency map, such as the saliency value of the patch, center or the mean or maximum value in the saliency map of all pixels that belong to the patch. These strategies were compared in the case of the eye tracking data in [7] and it was found that the mean of all the pixels gave the most stable results. Therefore, here we only use that strategy.

Furthermore, we have to distinguish two cases. First, we have a single saliency map (bottom-up or top-down trained with a single class label) and therefore a single weight per extracted feature. In the second case, we have class dependent saliency maps and therefore we generate a set of class dependent weight factors ψ_t^c for each feature $\mathbf{x}_t$. Accordingly, we obtain a different weighted Fisher representation per class when applying equation (2):

$$\mathbf{f}_w^c = \sum_{t=1}^{T} \psi_t^c \log p(\mathbf{x}_t|\lambda) \tag{3}$$

In this latter case, each one-versus-all classifier will have as input the corresponding class dependent Fisher representation.



(a) (b) (c)

Figure 6: Row (c) shows top-down saliency maps computed for sample images (a) of VOC dataset using the method based on labelled nearest neighbours. To train the foreground and background model, the bounding boxes of all classes were considered as relevant region. In (b) we show a smoothed version of these bounding boxes that we used to simulate an implicit feedback experiment on PASCAL VOC data (see details in section 3.3).

3.3 Experiments with Automatic Weighting Methods

In this section, we present categorisation experiments using weighted representations obtained automatically. The methods were evaluated on two datasets:

- The *Cats&Dogs Dataset* was built to collect eye gaze data. It is a subset of images of the PASCAL VOC 2007 dataset [9]. Details on the eye gaze data collection are detailed in [7], here we just briefly recall that it contains three classes: cats (105 images), dogs (105 images) and a third class which consists of other objects, which we labeled as the 'neither' class (52 images). It was important to do experiments with this data in order to compare the automatic weighting results with the method where the weights were obtained from eye scanpaths.

- The *PASCAL VOC* dataset is a benchmark dataset [9] that contains images of 20 classes of objects. The categorisation challenge consists of detecting classes of objects in the images and the results are evaluated with the mean average precision for all the classes. All the objects were manually labeled with rectangular bounding boxes. To train the classifier we used the PASCAL VOC 2008 [10] training dataset (both train and validation containing 2113 and 2227 images, respectively). The graund trutht labels of the VOC 2008 test set are not available yet, so we evaluated our method using the VOC 2007 test set containing 4952 images.

In our experiments we compared the weighted Fisher Kernel method using the following saliency maps:

- **BSL** – The baseline method obtained with unweighted Fisher representations. This also corresponds to a uniform saliency map.

- **IF-Gaze** – Saliency method using implicit feedback. In case of the Cats&Dogs we used the available eye tracking data. Here we show the best results obtained by weighting

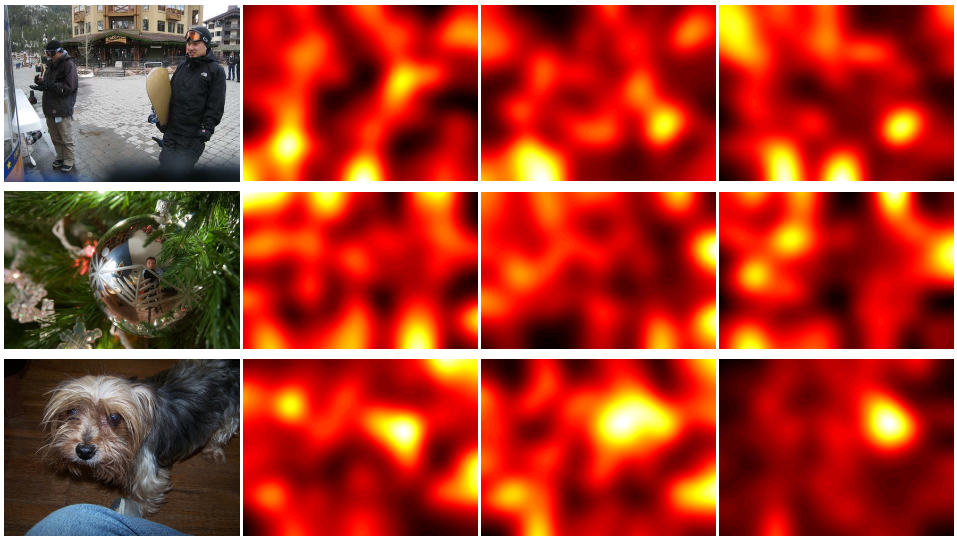


Figure 7: Top-down saliency maps computed using the TS-SOM method for sample images for classes cat, dog and neither.

the Fisher representations using smoothed eye fixation maps (see also [7]).

- **IF-SBB** – As we do not have eye tracking data for the whole PASCAL VOC dataset, we used an oracle experiment, which uses saliency maps computed from bounding boxes for both training and testing images (can be seen as a user drawing bounding boxes as explicit feedback). In order to preserve background information in our representation, instead of using the bounding boxes as binary maps, we centered non-isotropic Gaussians on the bounding boxes. Their standard deviation for each dimension is proportional to the width and height of the boxes. For this reason they are referred to as the Soft Bounding Boxes (**SBB**), see Figure 4(d) for examples.

- **I&K** – The bottom-up saliency maps of Itti and Koch [16]. It has the advantage of not requiring any training step or data. On the other hand, it leads to disappointing categorisation results, possibly because they are too sharply peaked and noisy, as illustrated in Figure 1(f).

- **S-I&K** – Similarly to what was done in the case of eye gaze data (see [7]), we experimented with smoothing the saliency maps obtained by Itti and Koch's method [16]. This was done by convolution with Gaussian using $\sigma = 8\%h$ where h is the image height[4].

- **CGM** – Centered Gaussian Maps. We tested different diagonal Σ of the Gaussian, with varying s, where $\sigma_w = s \cdot w$ and $\sigma_h = s \cdot h$ where w and h corresponding to the the width and height of the image. Here we report the best results which were obtained with $s = 0.2$ for Cats&Dogs Dataset (more focused to the center) and $s = 0.3$ for PASCAL VOC (where we require more knowledge about the background).

- **KNN** Learning saliency from labeled nearest neighbours where the 30 nearest neighbour were used to train the foreground and background model with all available bounding boxes (all classe in case of PASCAL VOC dataset) considered as salient regions in the retrieved images.

[4]We chose the sigma value that performed the best in the case of eye tracking data on.

- **FK-PCPM** – Fisher Kernel-based Patch Class Probability Maps were trained on gaze data when tested on the Cats&Dogs Dataset. Here we trained a salient versus non salient patch classifier. The positive labels were obtained by thresholding gaze-based saliency maps, so the features from areas of high visual attention were used as foreground (salient) samples. Since there are two classes of patches (foreground/background), a single classifier is trained and one probability map is created from each image[5] In the case of the PASCAL VOC dataset, we considered all classes as relevant and after computing the class probability maps we combined them in a single map[6] (considering the max probability).
- **TS-SOM** - The class maps obtained by the Tree-Structured Self-Organizing Map-based relevant image region detection method was tested only on Cats&Dogs Dataset (as we do not have yet the maps for the PASCAL VOC dataset).

Table 1 shows different results on the Cats&Dogs Dataset obtained with different maps. As this is a small dataset a visual vocabulary of 32 Gaussians gave already good performances (the improvement obtained using larger vocabulary lead to a small or no improvement on the classification accuracy in most cases). As classifier, we used the linear sparse logistic regression of [21] and the combination of color with texture was done by late fusion (score averaging).

CR	Color	Texture	Combined
BSL	45.80	55.34	54.96
IF-Gaze	51.90	65.27	66.41
I&K	44.27	54.58	50.38
S-I&K	44.27	55.34	51.91
CGM	51.53	59.16	56.87
FK-PCPM	48.47	54.20	56.49
KNN	47.71	54.96	57.25
TS-SOM	42.37	56.87	51.91

Table 1: Correct classification rate (in % of categorisation accuracy) results obtained with diverse automatic saliency estimation methods on the Cats&Dogs dataset.

Analysing the table first we can note that for this dataset the color features lead to relatively poor results and combined with the texture, they even decrease the classification accuracy compared to the pure texture based classification in most cases. Furthermore, none of the proposed automatic saliency maps reach (even close) to the performances that can be achieved with saliency obtained by an eye tracker, which shows the usefulness of integrating an eye tracker in the classification process if available.

Concerning the automatic saliency maps, suprisingly, the best improvement over the baseline was obtained with the simplest method, i.e. the Centered Gaussian Maps (CGM). The classical Itti and Koch's bottom-up method does not work at all, it even decreases the baseline performance. We tried to improve the maps by smoothing them, but the results were not satisfactory neither.

Finally, we tested a few more complex top-down saliency maps. The Fisher Kernel-based Patch Class Probability Maps (FK-PCPM) was trained on the eye tracking data and used a single class (salient) leading to a single class independent map. They allowed to improve the color scores and the combined (colour+texture) score, but they decreased the accuracy

[5]Here actually we try to learn to predict the eye fixation data.

[6]We still need to perform the experiments where for each classifier we use a class dependent weighting without combining the maps into a single saliency map.

	Color	Texture	Combined
BSL	39.28	47.62	49.06
IF-SBB	43.8	54.78	56.42
I&K	28.98	33.37	36.69
S-I&K	34.26	40.67	43.13
CGM	39.28	48.68	50.01
FK-PCPM	39.64	49.18	50.41
KNN	39.41	46.72	48.79

Table 2: Average precision for different saliency maps when training on PASCAL VOC 2008 and testing on PASCAL VOC 2007.

when texture alone was used. The results of the KNN method were similar to the FK-PCPM, which is not suprising as the maps are similar due to the fact that we used in both cases the Fisher Vectors of the patches as high level representations and learned saliency from them. In contrast, the tree-structured self-organizing map-based (TS-SOM) class dependent maps worked better on the texture (improved the baseline results), but they decreased the color and the combined classification accuracy.

Table 2 show the results obtained on the PASCAL VOC dataset. As the database was larger (almost 10000), here we used a larger visual vocabulary of 128 Gaussians. As classifier, we used the kernel[7] sparse logistic regression [21].

Again, we can notice that using an implicit feedback map (simulated in an oracle type experiment with ground truth smoothed bounding boxes), we can obtain significant classification performance increases by weighted features with location relevance. On this database, even if color perfors worse than texture their combination generally leads to a significant improvement of the classification accuracy, showing teh importance of the color for this database.

Concerning the bottom-up saliency maps, again we obtain some improvement with the Centered Gaussian Maps (CGM), but Itti and Koch's maps lead to decreased classification performances. The FK-PCPM top-down methods allowed for similar improvement as CGM, but the KNN method was less performant in this case than the CGM.

3.4 Concluding Remarks

In D6.1 [7], we introduced a novel image representation in which images are modeled as order-less sets of weighted low-level local features. We showed how this framework could be integrated in the Fisher Kernel framework (FK). The main idea was to weight each extracted feature based on a saliency map. We described experiments with saliency maps built from gaze data.

In this report we described other methods to build saliency maps and compared them using two databases, the Cats&Dogs dataset of D6.1 and the PASCAL VOC 2008/2007 datasets. These saliency detection methods consist on bottom-up unsupervised models and supervised top-down methods. The top-down methods and the simple CGM (a weighting scheme based on a Gaussian placed on the centre of the images) lead to a minor improvement in visual categorisation performance. On the other hand, the popular bottom-up method of Itti & Koch gave results worse than the baseline method, which does not use weights from saliency maps.

[7]Similar conclusions can be deduced when using a linear classifier, but the performances are in general lower than in the case of the kernel classifier

4 Confidence Estimation and Information Fusion

Our aim is to use some type of confidence estimation method in order to give weights to different sources when fusing information. Visual categorisation frameworks like ours allow information fusion to happen in any level: early (e.g. using concatenation of feature vectors, as done in [6]), at kernel level (as done in [40, 24] and in D3.1 [15]) or with late fusion by combining the output classifiers. In all of the above, a weight factor can be associated to each source of information. The weights can be pre-set or learnt from training data, but for most of the methods in the literature, once the weights are determined, they remain fixed for new samples. We are interested in associating a confidence score to each prediction obtained from each information source, i.e., a per-sample confidence. For this case, it is more practical to work with late fusion.

Several methods have been proposed for confidence estimation, most of them to prune classification results. A number of techniques have been proposed for fusion in multimodal biometrics [1] and in multimedia applications [30, 18]. In the subsections below, we review the methods for confidence estimation which we evaluated in this report.

4.1 Conformal Predictors

In this section we briefly review a method for producing confidence estimates for predictions based on the idea of typicalness and conformal predictors (see e.g. [31] and [42]). In [31], the authors propose a transductive method for providing confidence values for SVM predictions. Essentially, the framework is based around the the idea that for an exchangeable distribution, all possible orderings of a sequence are equally probable. Therefore, one can estimate the randomness of a particular sequence by bounding the probability of a particular event occurring. This leads to a method where for each test example, every possible label is postulated for it and it is added to the training set (forming a new 'sequence'). A randomness test is then performed on the sequence and if the sequence is deemed non-random, then the test example is 'untypical', and the label postulated is unlikely to be the correct one. See [42] for more details on the motivation for this approach. Essentially however this method can be seen as approximating a universal test for randomness and has strong connections with the Kolmogorov complexity of a sequence.

In order to describe the approach, we first follow the transductive method for SVMs presented in [31] and then describe the more computationally efficient Inductive Confidence Machine (ICM) which was introduced by [27].

The transductive method follows the procedure outlined above. We assume we have some strangeness function $A(\cdot)$ which takes in a sequence of examples and labels $\{(x_1, y_1), \cdots, (x_l, y_l)\}$ and for each example (x_i, y_i) in the sequence it outputs an associated real-valued *strangeness* value α_i. The important requirement for the function A is that it should be *consistent*; in the sense that if the order of input examples are permuted, then so are the associated α values. Irrespective of the order of the inputs, the function must always assign the same strangeness value to the same input pair for a specific sequence. In [31] the authors trained a support vector machine (SVM) on the sequence and used the α values of the Lagrange multipliers directly as strangeness values.

Therefore, when a test example x_{new} is presented, it is first assigned a label of $+1$ and added to the training set. An SVM is then trained on the new set and a strangeness value for each example α_i is simply the Lagrange multipliers at the SVM solution. Given that under the exchangeability assumption all possible sequences of strangeness values are equally likely, one can ask what the probability of the strangeness value associated with the test example, α_{new} being among the largest n strangeness values is. Let $\alpha_1, \ldots, \alpha_l, \alpha_{new}$ be the strangeness values obtained, let $\mathbf{z}$ represent the training set plus the test example with postulated label,

then we can define the function A as

$$A(\mathbf{z}) = \frac{\#\{i = 1, \cdots, l : \alpha_i \geq \alpha_{new}\}}{l+1}, \tag{4}$$

note that this in turn is a p-value as we have [42]:

$$P\{A(\mathbf{z}) \leq n\} \leq n \ , \tag{5}$$

where n is a significance level. For a postulated label, we often refer to the output of $A(\mathbf{z})$ simply as the p-value:

$$p = \frac{\#\{i : \alpha_i \geq \alpha_{new}\}}{l+1}, \tag{6}$$

Intuitively, if the test example has been given the correct label, then it is likely to be a 'typical' example and the associated α_{new} is likely to be zero (it is not a support vector). Therefore the p-value above will be one. If however the incorrect label is assigned, the example is likely to be a Support Vector with a high alpha-value (and incorrectly labelled example in a cloud of points with another label); and the associated p-value will be small.

In practice, a test sample is added to the training set once with label 1 and once with label -1, leading to two p-values: p^+ and p^-. One then predicts the classification associated to the highest p-value (p_1) and outputs as confidence in the prediction one minus the second-highest p-value ($1-p_2$). The appealing feature of this approach is that it produces a *conformal predictor*[8]. That is, if one chooses to reject classifications at a particular confidence level, say 5%, then using the above method by only predicting those labels for which the confidence is 95% or higher, will only result in an error equal to or less than 5%. This is due to the p-value above being a valid test for randomness [42]. The second-largest p-value gives a value at which we can *reject* the associated classification, and by Eq. (6), the probability of this occurring is less than p_2. Note that the use of Support Vector Machines here was just an example. Any algorithm which leads to a consistent function $A(\cdot)$ could be used (for example, k-Nearest Neighbour distance measures on the $l+1$ sequence could be taken as strangeness values, as they respect the permutation invariance required for consistency).

4.2 Inductive Confidence Machines

The transductive approach outlined above is often inefficient: a new classifier has to be trained and evaluated for every label on every test point. Inductive Confidence Machines (ICM) [27] introduced a method in which the confidence can be estimated almost as efficiently as the underlying algorithm. Note that in the paper they focused on a regression method for producing confidence, here however we continue to consider the classification setting.

Given a training set of attributes and labels $\{(x_1, y_1), \cdots, (x_l, y_l)\}$, the training set is separated into a training set for the algorithm $\{(x_1, y_1), \cdots, (x_m, y_m)\}$ with $m < l$, and a calibration set $\{(x_{m+1}, y_{m+1}), \cdots, (x_l, y_l)\}$. One then trains the algorithm being used (e.g. SVMs) on the smaller training set only. Strangeness values are then obtained using the calibration set and the test example. In the case of SVMs, one could take for example the distance $\alpha_i = y_i f(x_i)$ for each point. That is, the distance from the hyperplane multiplied by the example's label. The procedure outlined above can then be followed: a new example is obtained and both labels are postulated, each time (6) is used to compute the p-value. Note here however, that no retraining is required: strangeness values are calculated from the output of the algorithm using the calibration set and test point. Therefore this process is much more efficient.

[8] Strictly speaking, here we consider non-conformal predictors, that is predictors where the label disagrees with our base notion. This is the approach we use as it is consistent with the statistical notion of p-values in hypothesis testing.

For the multi-class multi-label setting considered here (where one example can belong to many classes), we use the following approach. First we postulate all possible labels and obtain p-values using the method described above, then p-values are ranked in descending order $p_1, p_2, p_3, \ldots$. For a given rejection threshold n, we reject all labels for which $p_i < n$ and predict the set of labels for which $p_i \geq n$. This gives us a conformal predictor under the definition of Vovk [42]. If we need to reject examples rather than labels, we use the following as a proxy. We order all test examples according to their associate p_1 value (associated to the most likely first label) and reject the lowest $n\%$ examples with the lowest p_1 values. Note that in this case, the calculation of (6) is done only over the examples with positive labels (including the postulated one); that is, (6) becomes:

$$p = \frac{\#\{i : \alpha_i \geq \alpha_{new} \wedge y_i = 1\}}{l_+ + 1}, \tag{7}$$

where l_+ is the number of positive samples y_+ in the calibration set.

For the experiments presented in Section 4.6, we use the Inductive Confidence Machine approach with two different underlying classification algorithms: a sparse Fisher discriminant method and the sparse logistic regression method [21]. In both cases we used $\alpha_i = -y_i$, which intuitively produces a large output if the example is unusual (i.e. a positive value if it is 'missclassified') and a small output if the example is typical (i.e. the example in the validation set which lies furthest on the correct side of the decision boundary will produce the lowest value negative number). Sparse logistic regression is also used in our baseline classifier in the experiments presented here and in D6.1 [7].

4.3 Model selection via nonconformity

Following the approach outlined above and presented in [42], we apply a recent algorithm for model selection using nonconformity [13] to the VOC challenge data set. Given 20 classes of objects we concentrate on the classification task, where we would like to predict with certain confidence levels that our predictions exist within particular classes. Each example, however, can exist in several classes simultaneously.

We use the shorthand notation $z = (x, y)$ to denote an input-output pair. Let $S = \{z_1, \ldots, z_\ell\}$ be the training sample and let $S^v = \{z_1^v, \ldots, z_n^v\}$ denote the validation set. As mentioned in Section 4.2, we have a nonconformity measure for the SVM as follows:

$$A(S, z) = yf(x)$$

where $f(\cdot)$ is the function output by the SVM. By using (6), we now can obtain a p-value using this for postulated labels of $y \in \{\pm 1\}$:

$$\epsilon_z = \epsilon_{(x,y)}(f, S^v) = \frac{|\{j = 1, \ldots, n+1 : A(S, z^v) \leq A(S, z)\}|}{n+1}. \tag{8}$$

As mentioned above, we are seeking to reject classifications which are below a certain threshold of strangeness (for which p_2 is low), and therefore we can predict the opposite classification with high confidence. For a specific p-value threshold however, say 0.05, it may be the case that we cannot reject either prediction at this level (both p-values are less than this value) and therefore we have to predict the set containing both labels in order to be certain we have included the correct one with more than 95% confidence. Obviously this is undesirable, and we wish to make point predictions of only one label. Observe that to produce a high confidence we are seeking to minimise:

$$\bar{\epsilon}_x = \min_{y \in \{-1, +1\}} \epsilon_{(x,y)}(f, S^v). \tag{9}$$

and predict the opposite label. In the case of ties for this value (both labels give the same p-value), then we predict randomly.

By using the nonconformity measure in Equation (9) we have $1-\bar{\epsilon}_x$ confidence of rejecting the class minimised for example x, or equivalently $1-\bar{\epsilon}_x$ confidence of accepting the opposite class.

Now let us move to the situation where we run the SVM over a range of parameter values. Let $T = \{1, \ldots, |T|\}$ be an index set of different parameters generating $|T|$ decision functions $f_1(\cdot), \ldots, f_{|T|}(\cdot)$ from $|M|$ different models. By minimising over these different models we would still be finding the p-value that was most confident in rejecting the class. Therefore we have the following p-value $\bar{\epsilon}_x(m)$ for example x found over $|M|$ models:

$$\bar{\epsilon}_x(m^*) = \min_{m \in M} \min_{y \in \{-1,+1\}} \epsilon_{(x,y)}(f_m, S^v), \tag{10}$$

where m^* corresponds to the function $f_t(\cdot)$ realised by the minimum. We would like to apply this model selection procedure [13] to the VOC challenge data set containing 20 (object) classes. Therefore this minimisation would not be sufficient. Hence, denote $\bar{\epsilon}^1_x(m^*), \bar{\epsilon}^2_x(m^*), \ldots, \bar{\epsilon}^{20}_x(m^*)$ as the values found using Equation (10) for each class when training the SVM using a 1-vs-all methodology, where the true classes are considered positive ($y = +1$) and the remaining classes negative ($y = -1$). Therefore our confidence p^c_x for example x over the $|M|$ models for class $c \in \{1, 2, \ldots\}$ would be

$$p^c_x = \begin{cases} 1 - \bar{\epsilon}^c_x(m^*) & \text{if } \epsilon^c_{(x,y=-1)}(f_{m^*}, S^v) < \epsilon^c_{(x,y=+1)}(f_{m^*}, S^v) \\ \bar{\epsilon}^c_x(m^*) & \text{otherwise.} \end{cases}$$

Given d test examples $(x_1, \ldots, x_d)$ and $|C| = |\{1, 2, \ldots\}|$ classes we can generate the following matrix of p-values,

$$P = \begin{pmatrix} p^1_{x_1} & \cdots & p^{|C|}_{x_1} \\ \vdots & \ddots & \vdots \\ p^1_{x_d} & \cdots & p^{|C|}_{x_d} \end{pmatrix}$$

giving us a confidence for each example existing in a particular class. The pseudocode for the procedure described above is given below in Algorithm 1.

Algorithm 1 Multi-class nonconformity model selection.

Input: Training samples S, validation samples S^v and $|T|$ SVM parameters
Output: Matrix P of p-values

1: **for** $c = 1, \ldots, |C|$ **do**
2: Set class c as positive and $C \setminus \{c\}$ as negative.
3: Train $|T|$ models (*i.e.*, SVM) on training data S to find $f_1(\cdot), \ldots, f_{|T|}(\cdot)$.
4: **for** each test example **do**
5: For example x find

$$\begin{aligned} \epsilon^+ &= \min_{t \in T} \epsilon_{(x,y=+1)}(f_m, S^v) \\ \epsilon^- &= \min_{t \in T} \epsilon_{(x,y=-1)}(f_m, S^v) \end{aligned}$$

6: Compute p-value:

$$p^c_x = \begin{cases} 1 - \epsilon^- & \text{if } \epsilon^- < \epsilon^+ \\ \epsilon^+ & \text{otherwise.} \end{cases}$$

7: **end for**
8: **end for**

4.3.1 Fusion of p-values

We generate the following p-values for several different models:

1. SVM using Algorithm 1 for COL (svm-col-ms-pval)
2. SVM using Algorithm 1 for ORH (svm-orh-ms-pval)
3. SVM using standard conformal predictor for COL (svm-col-cp-pval)
4. SVM using standard conformal predictor for ORH (svm-orh-cp-pval)
5. SMLR using Algorithm 1 for COL (smlr-col-ms-pval)
6. SMLR using Algorithm 1 for ORH (smlr-orh-ms-pval)
7. SMLR using standard conformal predictor for COL (smlr-col-cp-pval)
8. SMLR using standard conformal predictor for ORH (smlr-orh-cp-pval)

We can fuse together any combination of these p-values. We show some of these combinations in Section 4.6. The p-values for SMLR were generated without using several different models but just a single one by ignoring the two minimisations in Algorithm 1.

4.4 Methods based on Density Estimation

The confidence estimation methods of the above sections were designed to evaluate the confidence based on predictors. This section reviews some methods which analyse the samples without taking classification predictions into account. Instead, they rely on measures related to sample density or, more accurately, to distances between points and sets. The discriminative classifiers which we use give classification scores (or posterior probability scores) for each class, but they do not take into account how strange a new sample is w.r.t. the known samples. In other words, a classifier outputs a high score for a sample because it falls far from a decision boundary in the binary classification case. But this might happen because this sample is very different from any seen sample, rather than because the classification is trustworthy. In this case the analysis of the training sample density can give the cue needed to estimate the confidence of the prediction in a feature space, for a given sample. So we hypothesise that a class-independent measure of strangeness complements classification scores.

This approach relates to novelty or outlier detection [33], in which the goal is to identify if sample data is not related to the training set. But we are interested in "smooth" novelty scores. Another related problem is that of density-based clustering [35, 2], in which a density measure of sets is used to assign elements to clusters or create new clusters. Several density measures have been proposed for this application.

Below are the measures of this type evaluated in this report:

- **KNN**: distance to the k^{th} nearest neighbour. This distance can be normalised using the maximum distance between pairs of training samples.
- **Sphere**: given a parameter r and a new sample x, count how many n neighbours of x are within the distance r from x. The value of r can be computed, for instance, using the median of distances between samples in a validation set. The confidence of sample x is proportional to this count n.

- **SphereS**: the 'Sphere' method can be applied for a range of different values of r and the counts can be combined by their average. In our experiments, we tried $r = \{m, m/2, m/4, m8, m/16\}$, where m is the median of distances between training samples.

- **Rank**: first, compute $d(x, x_k)$, the distance between sample x and its k^{th} nearest neighbour (x_k). At training time, evaluate the distances between all the pairs of training samples and rank them. The rank of the value $d(x, x_k)$ among the distance values of the training set should be used to estimate the confidence of x.

- **Parzen**: kernel density estimation (KDE) [12] is a non-parametric way of estimating the PDF:

$$f_h(x) = \frac{1}{T \cdot h} \sum_{t=1}^{T} K\left(\frac{x - x_t}{h}\right) , \tag{11}$$

  where K is a kernel and h is the bandwidth (smoothing parameter). The kernel is often taken to be a Gaussian with zero mean with unit variance. This method is often referred to as the Parzen window density estimation method. Previous experiments with Fisher have shown that the similarity measure below leads to better classification accuracy than all of the commonly used kernels:

$$\cos L1(x, x_t) = -\left|\left|\frac{x}{||x||_1} - \frac{x_t}{||x_t||_1}\right|\right|_1 , \tag{12}$$

  where $||x||_1$ is the L1 norm of x. Although this is obviously not a Mercer kernel, we refer to this measure as "$\cos L1$ kernel" because of its analogy with $2\cos(x, x_t) - 2$ in the case of the L2 norm. This measure gives values in the range $[-2, 0]$. We adapt equation (11) with $\cos L1$ by

$$f(x) = \frac{1}{T} \sum_{t=1}^{T} 1 - \frac{\cos L1(x, x_t)}{2}. \tag{13}$$

- **ExpParzen**: Alternatively, one can give the version below, which enhances the difference in confidence level of samples which are closer to denser areas:

$$f(x) = \frac{1}{T} \sum_{t=1}^{T} \exp\left(\beta \cos L1(x, x_t)\right) , \tag{14}$$

  where $\beta > 1$ is an input parameter.

Several other methods can also be used. One of the promising possibilities is the one-class-SVM [32]. The goal is to estimate a $C(\alpha)$ minimum volume that contains at least a fraction α of the probability mass. It searches for a function f that gives positive values (+1) for a small $C(\alpha)$ region capturing most of the data points and -1 elsewhere. The data is mapped into the feature space using a kernel K (e.g. RBF) that separates it from the origin with maximum margin. The goal is to optimise:

$$\min_{\alpha} \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \text{ subject to } 0 \leq \alpha_i \leq \frac{1}{\nu l}, \ \sum_i \alpha_i = 1 \ . \tag{15}$$

Here $\nu \in (0, 1]$ and l is the number of data points. The distance of a test sample to the margin detects how novel this sample is. One problem is that this method requires Mercer kernels to enable a convex optimisation, which is not the case for the $\cos L1$ kernel. For this reason we did not perform experiments with this method for this report, but exploring this possibility can be an interesting direction for future work.

4.5 Direct Fusion Approaches

In this section we propose two simple and direct methods to combine the output of different classifiers $\mathcal{C}_j$ into a final prediction. Here we consider only binary classification problems, but the methods can be extended also to multi-class problems. We assume that for any input x each classifier predicts a score $\mathcal{C}_j(x)$. The sign of the score gives the binary prediction of the classifier, while the absolute value $|\mathcal{C}_j(x)|$ indicates the confidence in this prediction (high values meaning high confidence).

Assuming that the scores $\mathcal{C}_j(x)$ describe the performance of the individual classifiers reasonably well, we need to learn a fusion function $F(s_1, \ldots, s_k) = s_{\text{fus}}$, $s_{\text{fus}} \in \mathbb{R}$, which takes the scores of the individual classifiers s_i as input and produces a fusion score s_{fus}.

4.5.1 Learning a Fusion Function by SVMs

Since we are considering classification problems, support vector machines are an obvious choice for learning the fusion function. This is done by transforming each training example $(x_i; y_i)$ with $y_i \in \{+1, -1\}$, into a training example $(s_{i1}, \ldots, s_{ik}; y_i)$ for the fusion SVM, where $s_{ij} = \mathcal{C}_j(x_i)$ is the score of classifier $\mathcal{C}_j$ for input x_i. The fusion SVM can then be trained on the transformed examples by any SVM training method.

In initial cross-validation experiments we found that for a good fusion function a linear SVM was preferable. This linear SVM was rather insensitive to choices of training parameters, unless the distribution of positive and negative training examples was very uneven. In this case the positive examples (which are fewer) need to receive a higher weight in the SVM objective function.

4.5.2 Optimising the average precision directly

Since retrieval performance is often measured by average precision, we also tried to optimise this value directly. We concluded from the SVM experiments that a linear fusion function is sufficient. Thus we optimised the weights which combine the scores from the individual classifiers, such that the average precision on the training set is maximized. This is not completely straight forward as the average precision as a function of the weights can be quite erratic (e.g. Figure 8). As expected, we find in our experiments that optimising average precision directly gives the best results for this quantity.

4.6 Fusion Experiments

Our experiments are based on the PASCAL VOC 2007 challenge [9], which contains 2501 images in the training set, 2510 in the validation set and 4952 in the test set. Each image contains at least one object of 20 different classes. In our experiments, the method of [29] was used as the main visual categorisation framework. This is the same method that was used as baseline in D6.1 [7] and in Section 3 of this report. We performed classifier fusion experiments and evaluated the results using the mean of the average precision across all the 20 categories of this dataset, which is the evaluation method used to rank the results submitted to the PASCAL VOC challenges 2007 and 2008.

We used two different types of image features: ORH (edge ORientation Histograms, based on [22]) and COL (Gaussian weighted local COLour histograms). Both were extracted using dense grids at five different scales with an overlap of 50% in the area of neighbouring patches of the same scale. The Fisher kernels framework is applied for each feature type separately and SLR [21] is trained. The combination between the two SLR results is done by:

$$S(x) = S_{ORH}(x) + S_{COL}(x) \ , \tag{16}$$

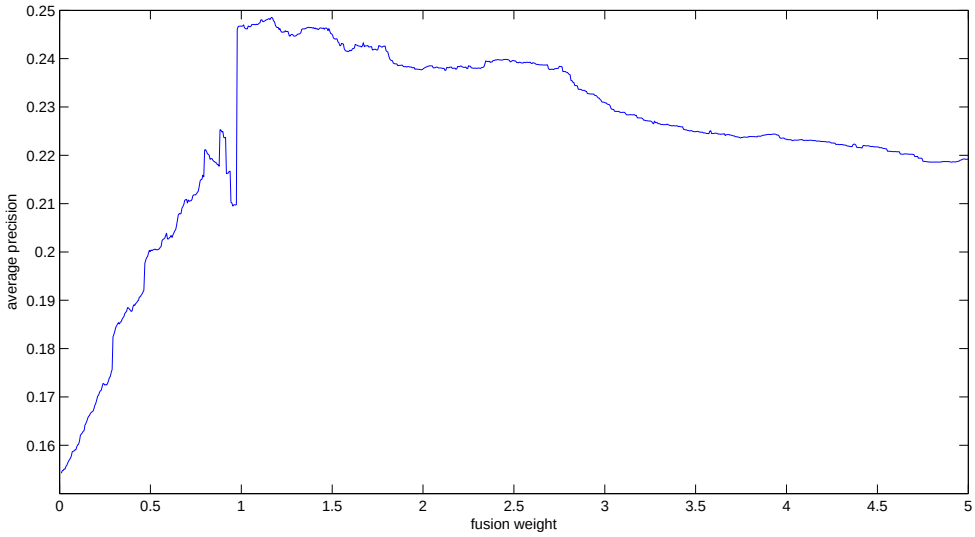


Figure 8: Average precision on the training data of class "potted plant" as a function of the fusion weight.

where $S_{feature}(x) = wx + b$ is the classification score obtained for sample x with w and b learned from the training set by SLR. As before, we used one classifier per category. The above combination method is our baseline and gives a mean average precision of 54.48% for VOC 2007.

The direct fusion approaches of Section 4.5 output a combination score directly. The other methods output confidence measures which can be associated to prediction scores in a number of different ways. One possibility is to associate confidence weights to the classification scores in (16) leading to:

$$S(x) = \alpha_{ORH}(x)S_{ORH}(x) + \alpha_{COL}(x)S_{COL}(x) \ , \tag{17}$$

where $\alpha_{feature}(x)$ is the confidence associated to sample x. This α can be class-dependent (as with p-values) or class-independent (for density-based measures of confidence). For weights which do not sum to one, it may be more appropriate to use the following:

$$S(x) = \frac{\alpha_{ORH}(x)S_{ORH}(x) + \alpha_{COL}(x)S_{COL}(x)}{\alpha_{ORH}(x) + \alpha_{COL}(x)} \ . \tag{18}$$

Another possibility is to combine the inputs using the probabilities obtained by fitting a sigmoid to the classification scores:

$$p(x) = p_{ORH}(x) \cdot p_{COL}(x) \ , \tag{19}$$

where

$$p_{feature}(x) = \frac{1}{1 + \exp(-S_{feature}(x))} \ . \tag{20}$$

Confidence weights can be integrated by

$$p(x) = p_{ORH}(x)^{\alpha_{ORH}(x)} \cdot p_{COL}(x)^{\alpha_{COL}(x)} \ , \tag{21}$$

It is straightforward to compute α for the class-independent density estimation methods. For the conformal predictors, we explored three possibilities:

COL alone	45.12
ORH alone	52.91
Linear combination (Eq. 16)	<u>54.84</u>
Product combination (Eq. 19)	54.21
Direct fusion (Sec. 4.5.1)	**55.11**
Direct fusion optimising AP (Sec. 4.5.2)	**55.68**

Table 3: Baseline results results and results with the direct fusion methods of Section 4.5, for PASCAL VOC 2007, in % of the mean average precision for all categories. The underlined value is used as our main baseline in this report, the bold values highlight results that are better than this baseline.

- Simply associating α to the confidence weight and giving an individual weight for each class c

$$\alpha^c_{feature}(x) = p^c_{feature}(x) \ . \tag{22}$$

- Using the maximum confidence value across all the classes. This generates a single value per sample, which is used across all classes to compute $S(x)$:

$$\alpha_{feature}(x) = \max_{\forall c} p^c_{feature}(x) \ . \tag{23}$$

  This means that we assume high confidence for a sample x when at least one of the classes c gives a high confidence value $p^c(x)$.

- We can also assume that a predictor is highly confident when the predictors of all the other classes are not very confident:

$$\alpha^c_{feature}(x) = 1 - \max_{\forall i \neq c} p^i_{feature}(x) \ . \tag{24}$$

  This is a sensible combination method for problems in which a single label is associated to each sample, which is not the case for many of the images in the PASCAL dataset.

Table 3 presents the baseline results obtained with each information source separately and with the straightforward combinations. The same table also presents results with the direct fusion methods. Notice that the result obtained by fusion as a linear combination is better than with the product-based fusion, so we will adopt the linear combination method for the remaining of this report.

Table 4 shows the results obtained by using the linear combination method of Eq. (17) to combine sources. Here the α values are determined by the density-based confidence estimation methods. Here, the parameters for KNN, Sphere, ExpParzen and Ranking methods were determined by experiments in a validation set. Similarly, table 5 shows results obtained by these methods with the normalisation of (18).

Notice that the different in results between normalised and non-normalised confidence weights is very small and the best overall result was obtained without normalisation. Since the evaluation, which is based on mean average precision, depends on a ranking of the scores (and therefore on their magnitude), some information is lost when normalisation is applied. For the next experiments, we do not use the normalisation of Eq. (18).

For the ICM method (Sec. 4.2), we evaluated two underlying predictors, the SLR classifier, which is used for our baseline experiments, and two variants of the kernel-based extension of Fisher discriminant analysis (KFDA) of [14]:

- **KFDA**$_1$ uses the connection between Fisher discriminant analysis and least-squares problem. The complexity is controlled by L_q penalty function where $0 < q \leq 1$. This

KNN	**55.26**
Sphere	49.23
SphereS	51.93
Rank	54.84
Parzen	54.70
ExpParzen	**55.46**

Table 4: Results (% of mean AP in VOC 2007) obtained with the density-based confidence estimation methods of Sec. 4.4 by applying them for linear combination (Eq. 17). The highlighted results are better than the baseline of Table 3.

KNN	**55.02**
Sphere	53.02
SphereS	54.59
Rank	**54.86**
Parzen	54.79
ExpParzen	**55.36**

Table 5: Similar to Table 4, but with the confidence weights normalised using Eq. (18).

penalty function is well-known to have sparsity-inducing property and leads to non-smooth formulation. The problem is solved by the majorise-minimize principle, this gives a very simple iterative algorithm. Here, we use L_1 penalty function to induce parsimonious solutions.

- **KFDA**$_{Jef}$: in this variant, the penalty function requires a choice of regularisation parameter which controls the degree of parsimony. This involves an extra parameter apart from kernel parameter in the optimisation which must be found via, e.g. cross-validation. A Jefrey's noninformative hyperprior is adopted through a hierarchical-Bayes interpretation of the Laplacian prior distribution. Hence, this leads to a non-requirement of the regularisation parameter. More details can be found at [28].

Previous experiments have shown that KFDA$_{Jef}$ outperforms KFDA$_1$, so we do not show results with KFDA$_1$. Table 6 shows results of KFDA$_{Jef}$ as a classification method for each information source separately and for the linear combination of colour (COL) and texture (ORH) without confidence weights (16).

Table 7 shows the results of the conformal predictors of sections 4.2 and 4.3 applied using the linear combination scheme of (17) with class-dependent confidence values (22). For the ICM method, we evaluated the use of both SLR and KFDA$_{Jef}$ as the underlying predictor, denoted by ICM+SLR and ICM+KFDA$_{Jef}$, respectively. The same table also shows results obtained by the methods of 4.3.

Tables 8 and 9 shows results of the same confidence estimation methods applied using the maximum across all classes (23) and $1 -$ max all other classes (24).

COL alone	45.42
ORH alone	52.82
Linear combination	**55.55**

Table 6: Baseline results in the same format as Table 3 but using KFDA$_{Jef}$ for classification, instead of SLR.

ICM+SLR	48.67
ICM+KFDA$_{Jef}$	22.97
svm-ms	46.71
svm-cp	50.42
smlr-ms	47.50
smlr-cp	50.59

Table 7: Results (mean AP in %) with methods based on confidence estimation with conformal predictors applied to (17) with **class-dependent** confidence values (22).

ICM+SLR	54.82
ICM+KFDA$_{Jef}$	50.30
svm-ms	54.84
svm-cp	54.78
smlr-ms	53.78
smlr-cp	54.77

Table 8: Similar to Table 7, but using (23) to combine the confidence values across all the classes using the **maximum**.

ICM+SLR	**54.93**
ICM+KFDA$_{Jef}$	50.60
svm-ms	47.00
svm-cp	48.13
smlr-ms	46.52
smlr-cp	50.09

Table 9: Similar to Table 7, but using (24) to set the confidence for each class as 1-max of the confidence of other classes.

4.7 Discussion

For most of the conformal predictors of Sections 4.2 and 4.3, which give confidence values per classifier, the best results were obtained using combination of the confidence values using the maximum across all classes (23). The only exception is for ICM using SLR as the underlying estimator, which in fact gave the best result among all the methods of Sections 4.2 and 4.3, using the 1-maximum of all other classes strategy (24). Still, this result is marginally superior to the baseline of fusion without confidence weights, indicating that the conformal predictors were disappointing for this problem.

The KFDA classifier gave results better than the SLR for the individual features and for the weight-less fusion. However, when KFDA was used as the underlying classifier for ICM and this confidence score was associated to the classification score, the result was disappointingly worse than that of SLR. This is probably explained by the fact that the obtained ICM+KFDA p-values were often zero, which is probably due to overfitting to the validation set.

Among the methods evaluated, the direct fusion methods of Section 4.5.1 gave the best results. This is somewhat expected, since these methods directly aim at optimising the fusion weights.

The density-based methods ExpParzen and KNN (Section 4.4) seconded the direct fusion methods in terms of performance improvement. These are simple methods which do not require training or pre-processing and they recycle distance computations done for the kernel computation for classification. They complement classifiers by adding a measure of data strangeness as a weight factor.

The obtained improvement in mean average precision was of less than 1% in our best result (direct fusion). This is probably due to the fact that local colour and texture features are both rich and highly complementary, making it difficult to improve over the baseline. We expect that a more significant performance improvement will be obtained with the fusion of more sources of information.

5 Conclusions

This report presented the progress of research done in Task 6.2 of PinView. It comprised two branches of work.

The first one is a direct continuation of the work done in Task 6.1 about exploring saliency maps as a measure of local feature confidence. The method developped in Task 6.1 was evaluated with a number of alternative to eye gaze measurements, from explicit relevance feedback using mouse-clicks to automatic saliency estimation methods. The results show that our framework takes good profit of saliency maps if they are able to highlight the objects of interest, which is the case for gaze data and for maps built from explicit feedback. But the automatic methods only gave a marginal performance over the baseline without our local feature weighting scheme. This shows that there is still room for improvement in the area of estimating visual saliency automatically.

In the second part, we dealt with the higher level problem of fusing different sources of information in order to improve visual categorisation. We evaluated a number of confidence estimation methods as means to combine the output of classifiers. The direct fusion methods gave the best result, but this was still a mild performance improvement compared to the baseline of fusion by simply averaging out classification scores. However, we believe that the experiments performed here will be very useful for the upcoming work if a higher number of impormation sources is used.

Acknowledgements

We wish to thank Dr. Zakria Hussain of the University College London for his valuable comments on drafts of this report.

References

[1] S. Bengio, C. Marcel, S. Marcel, and J. Mariéthoz. Confidence measures for multimodal identity verification. *Information Fusion*, 3(4):267–276, 2002.

[2] Stefan Brecheisen, Hans-Peter Kriegel, and Martin Pfeifle. Efficient density-based clustering of complex objects. In *ICDM*, 2004.

[3] L.-Q. Chen, X. Xie, X. Fan, W.-Y. Ma, H.-J. Zhang, and H.-Q. Zhou. A visual attention model for adapting images on small displays. *ACM Multimedia Systems Journal*, 9(4), 2003.

[4] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C Bray. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, Prague, Czech Republic, May 11-14 2004.

[5] Gabriela Csurka and Florent Perronnin. A simple high performance approach to semantic segmentation. In *Proc 19th British Machine Vision Conf, Leeds*, 2008.

[6] Teófilo de Campos and David Murray. Regression-based hand pose estimation from multiple cameras. In *Proc IEEE Conf on Computer Vision and Pattern Recognition, New York NY, June 17-22*, 2006.

[7] Teofilo de Campos, Florent Perronnin, Ville Viitaniemi, Jorma Laaksonen, and Marco Bressan. Description and evaluation of novel local features with usable sub-categorisation performance. Technical report, PinView, October, 1st 2008. Deliverable 6.1, available from www.pinview.eu.

[8] Teófilo E. de Campos, Herve Poirier, Bernhard Lackner, and Michael kumar. Description, analysis and evaluation of confidence estimation procedures for sub-categorisation. Technical report, PinView, April 2009. Deliverable 6.2.1, available from www.pinview.eu.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html.

[11] Tom Foulsham and Geoffrey Underwood. What can saliency models predict about eye movements? spatial and sequential aspects of fixations during encoding and recognition. *Journal of Vision*, 8(2):1–17, February 2008.

[12] Mark Girolami and Chao He. Probability density estimation from optimally condensed data samples. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25:1253–1264, 2003.

[13] David R. Hardoon, Zakria Hussain, and John Shawe-Taylor. A nonconformity approach to model selection for svms. Technical report, University College London, 2009.

[14] Robert F. Harrison and Kitsuchart Pasupa. A simple iterative algorithm for parsimonious binary kernel fisher discrimination. *Pattern Analysis & Applications*, 2009. In press.

[15] Zakria Hussain, John Shawe-Taylor, Craig Saunders, and Kitsuchart Pasupa. Basic metric learning. Technical report, PinView, December 2008. Deliverable 3.1, available from www.pinview.eu.

[16] Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40:1489–1506, 2000.

[17] Timor Kadir and Michael Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, November 2001.

[18] Katrin Kirchhoff and Jeff A. Bilmes. Dynamic classifier combination in hybrid speech recognition systems using utterance-leval confidence values. In *Proc. IEEE International Conference on Acouostics, Speech and Signal Processing (ICASSP)*, 1999.

[19] Pasi Koikkalainen. Progress with the tree-structured self-organizing map. In *11th European Conference on Artificial Intelligence*. European Committee for Artificial Intelligence (ECCAI), August 1994.

[20] Gert Kootstra, Arco Nederveen, and Bart de Boer. Paying attention to symmetry. In *Proc 19th British Machine Vision Conf, Leeds*, 2008.

[21] B. Krishnapuram, L Carin, M. Figueiredo, and A. Hastemink. Sparse multimodal logistic regression: Fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):957–968, June 2005.

[22] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2), 2004.

[23] L. Marchesotti, C. Cifarelli, and G. Csurka. A framework for visual saliency detection with applications to image thumbnailing. In *ICCV*, 2009. submitted.

[24] Marcin Marszałek, Cordelia Schmid, Hedi Harzallah, and Joost van de Weijer. Learning object representations for visual object class recognition, oct 2007. Visual Recognition Challange workshop, in conjunction with ICCV.

[25] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.

[26] Frank Moosmann, Diane Larlus, and Frederic Jurie. Learning saliency maps for object categorization. In *ECCV International Workshop on The Representation and Use of Prior Knowledge in Vision*. Springer, 2006.

[27] Harris Papadopoulos, Kosta Poedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In 12^{th} *European Conference on Machine Learning (ECML)*, 2001.

[28] Kitsuchart Pasupa. *Data Mining and Decision Support in Pharmaceutical Databases*. PhD thesis, Department of Automatic Control & Systems Engineering, University of Sheffield, November 2007.

[29] F. Perronnin and C. Dance. Fisher kernel on visual vocabularies for image categorization. In *Proc of the IEEE Conf on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA., June 2007.

[30] Gerasimos Potamianos and Chalapathy Neti. Stream confidence estimation for autio-visual speech recognition. In *Proceedings of the International Conference on Spoken Language Processing*, volume III, pages 746–749, Beijing, 2000.

[31] Craig Saunders, Alexander Gammerman, and Volodya Vovk. Transduction with confidence and credibility. In 16^{th} *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 722–726, 1999.

[32] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Article communicated by vladimir vapnik estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.

[33] Sameer Singh and Markos Markou. A black hole novelty detector for video analysis. *Pattern Analysis Applications*, 8(1-2):102–114, 2005.

[34] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477, October 2003.

[35] Benno Stein and Muchael Busch. Density-based cluster algorithms in low-dimensional and high-dimensional applications. In *International Workshop on Text-Based Information Retrieval*, 2005.

[36] Fred W. M. Stentiford. Attention based similarity. *Pattern Recognition*, 40(3):771–783, March 2007.

[37] B. Suh, H. Ling, B.B. Bederson, and D.W. Jacobs. Thumbnail cropping and its effectiveness. In *ACM User Interface Software and Technology*, 2003.

[38] Veronica Sundstedt, Alan Chalmers, Kirsten Cater, and Kurt Debattista. Top-down visual attention for efficient rendering of task related scenes. In *In Vision, Modeling and Visualization*, pages 209–216, 2004.

[39] Antonio Torralba, Aude Oliva, Monica Castelhano, and John Henderson. Contextual guidance of eye movements and attention in real-world scenes: The role of global features on object search. *Psychological Review*, 113(4):766–786, October 2006.

[40] Manik Varma and Deb Ray. Learning the discriminative power-invariance trade-off. In *Proc 11th Int Conf on Computer Vision, Rio de Janeiro, Brazil, Oct 14-20*, 2007.

[41] Ville Viitaniemi and Jorma Laaksonen. Use of image regions in context-adaptive image classification. In *International Conference on Semantic and Digital Media Technologies (SAMT 2006)*, LNCS, pages 169–183, Athens, Greece, December 2006. Springer.

[42] Vladimir Vovk, Alexander Gammerman, and Flenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

[43] D. Walther and C. Koch. Modeling attention to salient proto-objects. *Neural Networks*, 19:1395–1407, 2006.

[44] Z. Wang and B. Li. A two-stage approach to saliency detection in images. In *ICASSP*, 2008.