Semantic Indexing based on Focus of Attention Extended by Weakly Supervised Learning

Kimiaki Shirahama*, Tadashi Matsumura†, Marcin Grzegorzek*, and Kuniaki Uehara†

* Pattern Recognition Group, University of Siegen
† Graduate School of System Informatics, Kobe University
Email: kimiaki.shirahama@uni-siegen.de, tadashi@ai.cs.kobe-u.ac.jp, marcin.grzegorzek@uni-siegen.de, uehara@kobe-u.ac.jp

***Abstract*—Semantic INdexing (SIN) is the task to detect concepts like *Person* and *Car* in video shots. One main obstacle in SIN is the abundant information contained in a shot where not only a target concept to be detected but also many other concepts are displayed. In consequence, the detection of the target concept is adversely affected by other irrelevant concepts. To overcome this, we enhance SIN based on a human brain mechanism to effectively select important regions in the shot. Specifically, SIN is integrated with Focus of Attention (FoA) which identifies salient regions that attract user's attention. The feature of a shot is extracted by weighting regions based on their saliencies, so as to suppress effects of irrelevant regions and emphasise the region of the target concept. In this integration, it is laborious to prepare salient region annotation that assists detecting salient regions most likely to contain the target concept. Thus, we extend FoA using Weakly Supervised Learning (WSL) to generate salient region annotation only from shots annotated with the presence or absence of the target concept. Moreover, rather than the target concept, other concepts are more salient in several shots. Features of these shots falsely emphasise concepts other than the target. Hence, we develop a filtering method to eliminate shots where the target concept is unlikely to be salient. Experimental results show the effectiveness for each of our contributions, that is, SIN using FoA, FoA extended by WSL, and filtering.**

Keywords–Semantic indexing; Focus of attention; Weakly supervised learning; Filtering.

I. INTRODUCTION

For effective processing of large-scale video data, one key technology is *Semantic INdexing* (SIN) to detect human-perceivable concepts in shots [1], [2]. Concepts are textual descriptions of semantic meanings that can be perceived by humans, such as *Person*, *Car*, *Building* and *Explosion_Fire*. Below, concept names are written in italics to distinguish them from the other terms. Many sources reported that the state-of-the-art video processing can be achieved using concept detection results as an intermediate representation of a shot [3]. Regarding this, traditional features just represent visual characteristics that significantly vary depending on various changing factors like camera techniques and shooting environments. On the other hand, the intermediate representation describes the presence of semantically meaningful concepts. Thus, if we could obtain accurate results where concepts are detected irrespective of changing factors, those results would facilitate categorising/retrieving shots that are visually dissimilar, but show similar semantic meanings. Motivated by this, much research effort has been made on SIN [1], [3], [4], [5], [6], [7].

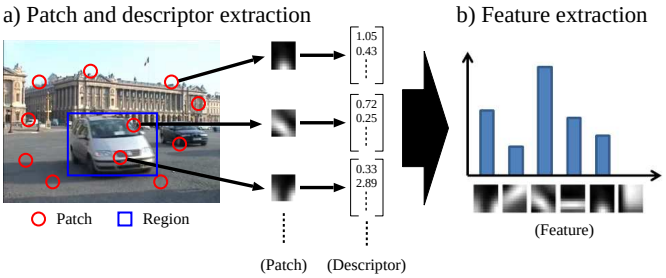

Figure 1. An illustration of feature extraction based on descriptors extracted from patches.

SIN is formulated as a binary classification problem where shots displaying a target concept are distinguished from the rest of the shots. One of the most important issues is feature extraction. Using Figure 1, we present an overview of the currently most popular approach [3], [4], [5], [6] while defining necessary terms for the following discussion. The approach consists of two main steps below:

1. Patch and descriptor extraction: This step aims to collect visual characteristics of *patches* that are small regions in a shot like red circles in Figure 1 (a). The rationale behind this is that as long as many patches are collected, some of them should keep their visual characteristics similar, irrespective of changing factors. From each patch, a *descriptor* is extracted as a vector, which numerically represents its visual characteristic. This is exemplified in Figure 1 (a), where three of patches are enlarged and their descriptors are shown on the right. It should be noted that compared to patches, we use the term 'region' to indicate a much larger region like the one of the car surrounded by the blue rectangle in Figure 1 (a).

2. Feature extraction: This step aggregates descriptors extracted from various patches to form a feature, which represents the distribution of those descriptors. For example, the histogram-type feature in Figure 1 (b) reveals that many descriptors characterise patches similar to the third one from the left, and there is no patch that is similar to the rightmost one in terms of descriptors. This kind of feature is effective for capturing detailed parts of a target concept. Especially, even if the target concept is partially invisible due to the occlusion by other concepts or the camera setting, the feature includes descriptors extracted from patches corresponding to the visible part of the target.

However, many concepts other than the target are displayed in a shot. For example, the shot in Figure 1 (a) includes the

Figure 2. Example shots where *Car* is shown in non-salient regions.

target concept *Car* and many others like *Building*, *Road* and *Sky*. Nonetheless, most of the existing SIN methods [3], [4], [5], [6] do not consider whether each patch belongs to the target concept or not. As a result, the feature is affected by patches of other concepts, and the detection performance of the target concept is degraded.

We aim to develop a SIN method that effectively spotlights a target concept while suppressing effects of the other irrelevant concepts. To this end, we incorporate *Focus of Attention* (FoA) (also called *visual attention*) into SIN. FoA implements 'selective attention' that is a brain mechanism to determine which regions in a shot (or video frame) are of most interest [8], [9], [10]. It is said that eyes are receiving visual data with the size 10^8-10^9 bits every second [9]. It is impossible for a human to completely analyse this huge size of data. Nevertheless, the human can effortlessly recognise meanings in a shot by fixating (or directing his/her gaze to) important regions. We apply this brain mechanism to SIN with the following logic: The fact that the human perceives the appearance of a target concept in the shot means that he/she fixates its region. Based on this, FoA is used to increase priorities of such regions and decrease those of the other regions, in order to construct a feature that emphasises the appearance of the target concept. In what follows, regions that attract fixations are called *salient regions*.

It should be noted that we focus only on appearances of a target concept in salient regions. In other words, we do not address its appearances in non-salient regions. For example, assuming that *Car* is the target concept, two shots in Figure 2 display it only in small background regions surrounded by red rectangles. These regions are clearly non-salient. Humans do not pay attention to or are not aware of the target concept appearing in such non-salient regions. Hence, these appearances are considered as meaningless and useless for subsequent processes like video categorisation, browsing and retrieval.

FoA consists of two main processes, *bottom-up* and *top-down*. The former implements human attention driven by stimuli acquired from the external environment. Since these stimuli can be thought as the visual information that eyes receive from a shot, they are equated with features extracted from the shot. However, salient regions detected based only on features are not so accurate because of the *semantic gap*, which is the lack of agreement between automatically extractable features and human-perceived semantics [11]. Thus, the top-down process implements attention driven by prior knowledge and expectation in the internal human mind. This biases the selection of salient regions based on human's intention, goal and situation. In our case, the top-down process utilises the knowledge about spatial relations between a target concept and surrounding ones in order to selectively localise salient regions most likely to contain the target. Finally, salient regions obtained by the bottom-up and top-down processes are combined to model their interaction.

To incorporate FoA into SIN, we address the following two issues: The first issue is the data availability of the top-down process. One typical formulation of this process is to adopt the machine learning framework, where salient regions in test shots are detected by referring to training shots in which salient regions are annotated in advance [12], [13], [14] or recorded by an eye tracker [14], [15]. However, a large number of training shots is needed to detect diverse kinds of salient regions. Due to a tremendous number of video frames in shots, it requires prohibitive cost to manually prepare many training shots. In addition, using an eye-tracker requires both labour and monetary costs. Thus, we develop an FoA method using *Weakly Supervised Learning* (WSL), where a classifier to predict precise labels is constructed only using loosely labelled training data [16]. In our case, this kind of training data are shots that are annotated only with the presence or absence of a target concept. These shots are used to build a classifier that can identify the region of the target concept in a shot, such as the blue rectangular region in Figure 1 (a) in the case where *Car* is the target. Regions identified by the classifier are used as annotated salient regions in the top-down process.

The second issue is the discrepancy that salient regions do not necessarily coincide with regions of a target concept. The reason is twofold: Firstly, it is difficult to objectively judge whether the target concept is salient or not. In other words, training shots can be annotated only with the presence of the target concept without considering its saliency. Consequently, like two shots in Figure 2, the target concept is shown in small background regions in several training shots. It is impossible or unreasonable to regard such regions as salient. The second reason for the discrepancy is possibly occurring errors in FoA. Even if the region of the target concept is salient for humans, another region may be falsely regarded as salient. A feature based on such a salient region incorrectly emphasises a non-target concept. To alleviate this, we develop a method that filters out shots where the target concept is unlikely to appear in salient regions, using regions predicted by the classifier in WSL. This enables us to appropriately capture characteristics of the target concept.

This paper is an extended version of our previous paper that only briefly illustrates our SIN method based on FoA due to the space limitation [1]. Specifically, the survey of related methods was quite insufficient in [1]. In contrast, the next section of this paper gives a comprehensive comparison of our method to various methods in four research fields, namely FoA, salient object detection, discriminative saliency detection, and SIN. In addition, while only a brief and conceptual explanation of our method was introduced in [1], its details and mathematical formulations are presented in Section III of this paper. Furthermore, although the experimental results in Section IV is the same to those in [1], Section V offers new ideas of how our method can be applied to different state-of-the-art features. Finally, for the sake of clarity, the Appendix provides a list of many abbreviations used in this paper.

II. Related Work

FoA has been studied in the fields of computer vision, psychology and neurobiology for a long time. In particular, the development of a principled top-down process is one of the most important research topics [9]. Below, some types of

knowledge used in the top-down process of existing methods are presented. First, *contextual cueing* means that a user can easily search a particular object, if he/she saw the same or similar spatial layout of objects in the past [12], [13], [14], [15]. Salient regions in a test shot are adaptively extracted based on salient regions in training shots with similar spatial layouts. The *symmetry* indicates that while viewing a symmetric object, eye fixations are concentrated on the centre of symmetry [17]. Based on this, salient regions are preferentially located around centres of regions, which individually have a symmetric pattern of intensity or colour values. In addition, the *focusness prior* represents that a camera is often focused on the most salient object [18]. According to this, regions with low degrees of blur are more likely to be regarded as salient. Furthermore, the *centre prior* expresses that the main content is displayed near the centre, and is used to emphasise regions around the centre as salient [19]. Please refer to [9], [10] for other types of knowledge in the top-down process.

Among the knowledge described above, we use contextual cueing because it can be generally applied to any kind of videos. In particular, we target 'unconstrained' web videos that can be taken by arbitrary camera techniques and in arbitrary shooting environments [20]. Apart from contextual cueing, the symmetry highly depends on directions of a concept. Although the frontal appearance of the concept is symmetric, its side appearance may not be so. In addition, the focusness and centre priors are considered as valid only for professional videos, which follow shooting and editing rules to clearly convey the content to viewers. On the other hand, web videos are usually created by amateurs without taking such rules in account. As a result, the main content in a shot is often captured unfocused, and is not necessarily displayed near the centre of a video frame. In contrast, the generality of contextual cueing can be enhanced using a large amount of training shots, so that a variety of salient regions in web videos can be covered. Also, while existing methods based on contextual cueing require training shots that are annotated with salient regions [12], [13], [14] or eye fixations [14], [15], we use WSL to generate such annotation from shots labelled only with the presence or absence of a concept.

Our method is now compared to two extensions of FoA. The first is *Salient Object Detection* (SOD) that extracts the region of an object attracting the most user attention [10], [21], [22]. Since the principle of FoA is to detect regions where people look as salient, it is not guaranteed that salient regions correspond to semantically meaningful objects. It often happens that salient regions only characterise parts of an object, where these parts are visually distinctive or contrastive compared to the surrounding ones. Thus, SOD detects regions that not only are salient but also characterise meaningful objects. Also, there is an experimental evidence indicating that salient regions are strongly correlated with attractive objects [23]. Our method differs from SOD in the following two points: First, although most SOD methods need training shots where regions of salient objects are annotated [21], [22], our method using WSL only needs training shots annotated with the presence or absence of a target concept. Second, SOD just detects the region of a salient object without identifying its category. In contrast, the category of a target concept is considered in our method based on WSL. Here, regions of the target concept are identified as the ones that are commonly contained in training shots annotated with its presence, but are not contained in training shots annotated with its absence. Using these identified regions, depending on the target concept, we adaptively find regions that are not only salient but also likely to contain it.

The second extension of FoA is *Discriminant Saliency Detection* (DSD) that extracts salient regions based on the discrimination power of descriptors for recognising a target concept (object) [24], [25]. Roughly speaking, DSD first regards the extraction of descriptors from patches as the bottom-up process, because they can be directly derived from images/videos (i.e., stimuli from the external environment). Then, the top-down process is performed as the selection of 'salient' descriptors, which best discriminate between the target concept and the others. Salient regions are computed based on locations of salient descriptors. However, as seen from the above-mentioned overview, DSD is significantly biased towards the recognition task, and does not care whether the region of the target concept is perceptually salient or not. In other words, DSD regards the target concept as salient even if it is shown in a small background region. Compared to this, we develop a filtering method that eliminates shots where the target concept is unlikely to appear in salient regions, by checking the coincidence between salient regions detected by FoA and regions identified by WSL. Appearances of the target concept in non-salient regions are considered as useless, because they do not attract user attention.

Finally, SIN is established in TRECVID that is a NIST-sponsored annual worldwide competition on video analysis and retrieval [2]. New SIN methods are being developed every year. The most popular approach is to extract a feature of a shot by encoding the distribution of descriptors using a histogram [4], using Gaussian Mixture Model (GMM) representing both means and variances of the descriptor distribution [5], [6], and using Fisher vector considering the first and second order differences between the distribution and the reference one [7]. Recently, researchers have started to adopt deep learning where a multi-layer convolutional neural network is used to extract a feature hierarchy with higher-level features formed by the composition of lower-level ones [7]. Despite this advancement of features, to our best knowledge, no method utilises FoA to enhance the quality of features. In this paper, we demonstrate the effectiveness of FoA to improve the most standard histogram-type feature.

III. SIN BASED ON FOA EXTENDED BY WSL

Figure 3 presents an overview of our SIN method. We call training shots annotated with the presence and absence of a target concept *positive shots* and *negative shots*, respectively. Since the target concept is *Car* in Figure 3, it is displayed and not displayed in positive and negative shots, respectively. For each of these training shots, FoA is performed to obtain its *saliency map* as shown in the middle of Figure 3. This map is an image representing the degree of saliency at each pixel. The higher saliencies of pixels are, the brighter they are depicted. Figure 3 shows saliency maps obtained for the positive and negative shots presented at the left. In the positive shot, the region of the moving car is regarded as salient. In the negative shot where a person is making a hand gesture, the region of his moving hand is regarded as salient.

After FoA, the feature of a training shot is extracted by weighting each descriptor based on the saliency of the patch from which the descriptor is derived. More concretely, red dot-

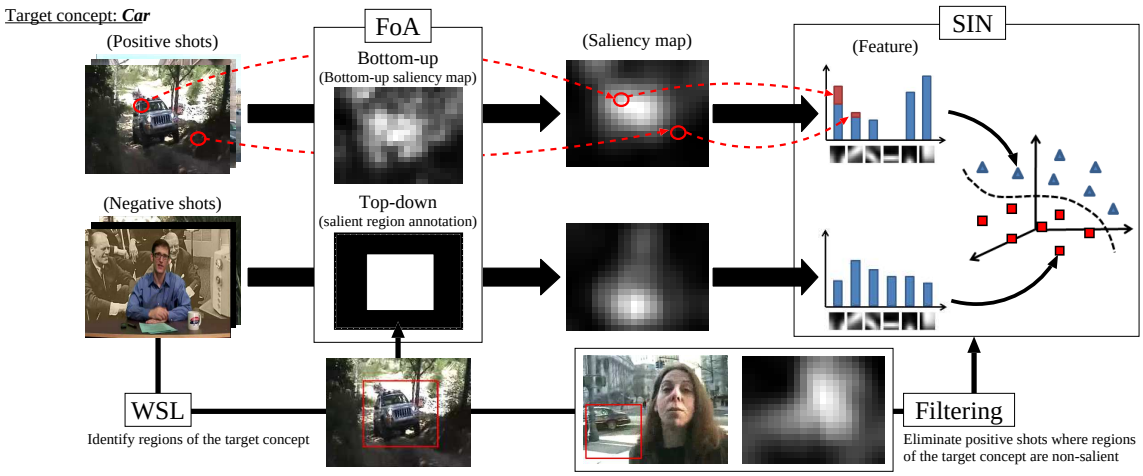

Figure 3. An overview of our SIN method using FoA extended by WSL

ted arrows starting from the positive shot in Figure 3 illustrate that the descriptor extracted from a patch in a salient region and the one extracted from a patch in a non-salient region have large and small influences on the feature, respectively. Finally, SIN is carried out by regarding training shots represented with such features as points in the multi-dimensional space. For the sake of visualisation, Figure 3 only depicts a three-dimensional space where triangles and rectangles indicate positive and negative shots, respectively. As shown in the dotted curve in this space, a *detector* is constructed to discriminate between positive and negative shots, and used to examine the presence of the target concept in test shots.

In Figure 3, the FoA module first conducts the bottom-up process on each training shot to compute its 'intermediate' saliency map, called *bottom-up saliency map*. This map is based only on features because the bottom-up process implements how eyes react to the visual information (see Section I). Specifically, regions that are visually different from surrounding ones are regarded as salient. However, it is difficult to accurately detect salient regions only using features. For example, Figure 3 shows the bottom-up saliency map for the positive shot, where in addition to the region of the car many background regions are also regarded as salient. Hence, the top-down process is needed to refine bottom-up saliency maps. To this end, WSL is firstly applied to training shots in order to prepare salient region annotation necessary for the top-down process. As a result, a classifier that identifies regions of the target concept is built. In Figure 3, the image under the FoA module and the black-and-white image over it indicate that, regions identified in positive shots (i.e., red rectangle) are used as annotated salient regions. Based on this, the top-down process is performed to refine a bottom-up saliency map into the final one.

The SIN module in Figure 3 involves filtering. Let us consider the positive shot and its saliency map on the left of the "Filtering" box in Figure 3. The positive shot shows *Car* only in the small background region depicted by the red rectangle. Correspondingly, this region is not so salient while the region of the woman in the foreground is regarded as the most salient. The feature extracted from this positive shot falsely emphasises the non-target concept *Person*, and misleads a detector to detect it. Thus, filtering is performed to eliminate positive shots where the target concept is unlikely to appear in salient regions. Below, we describe the bottom-up and top-down processes, WSL method, and SIN method with filtering.

A. Bottom-up Process

Figure 4 illustrates an overview of the bottom-up process where the positive shot on the left of Figure 3 is used as an example. We use a retina model to design how the bottom-up saliency map of a shot is created based on the visual information received by human eyes [14]. As shown in the upper part of Figure 4, it is known that the visual information is sequentially processed by horizontal, bipolar and Amacrine cells in the retina. The first cells perform smoothing to emphasise contrasts in the visual information, the second cells detect edges (or contours), and the last ones conduct the second smoothing to emphasise detected edges. Finally, according the feature integration theory [26], the above cells process different types of visual information in parallel, and the brain integrates processing results to focus the attention on certain regions. In what follows, we explain how to implement each cell's mechanism and how to integrate processing results.

First of all, as the encoding of the visual information that arrives at eyes, the following six features related to cell responses in the retina are extracted [14]:

$$\text{Intensity:} \quad I = \frac{r+g+b}{3}, \tag{1}$$

$$\text{Red-Green (RG) contrast:} \quad RG = \frac{r-g}{\max(r,g,b)}, \tag{2}$$

$$\text{Blue-Yellow (BY) contrast:} \quad BY = \frac{b-\min(r,g)}{\max(r,g,b)}, \tag{3}$$

$$\text{Flicker:} \quad F = I - I', \tag{4}$$

$$\text{Motion direction:} \quad \Theta = \tan^{-1}(\frac{v}{u}), \tag{5}$$

$$\text{Motion strength:} \quad \Gamma = \sqrt{u^2+v^2}, \tag{6}$$

where r, g and b represent the red-, green- and blue-channel values of a pixel in a video frame, respectively. In Equation (4), I' is the intensity in the previous video frame. In Equations (5) and (6), u and v are the horizontal and vertical displacements of the optical flow starting at a pixel. It should be noted that the above-mentioned features are extracted from each pixel in the video frame. Thus, each feature is represented as an

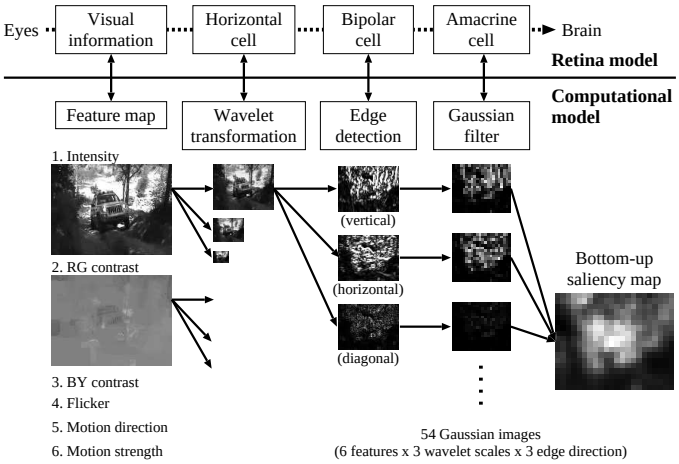

Figure 4. An overview of the bottom-up process in our FoA method

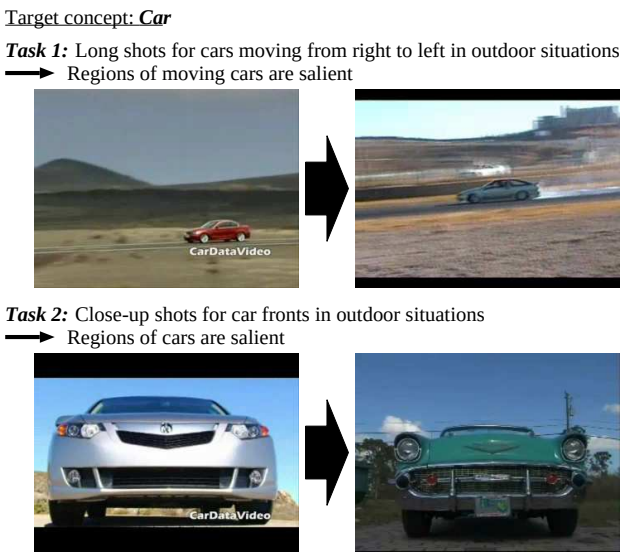

Figure 5. Two conceptual examples of tasks

image, called 'feature map', which has the same size to the video frame, as shown at the left of Figure 4.

Then, smoothing by horizontal cells is implemented as wavelet transform on each feature map. As shown in Figure 4, the feature map is scaled down into $1/2$, $1/4$ and $1/8$ sizes of images, termed as 'wavelet images'. This down-scaling is useful for emphasising contrasts in the feature map while removing noises. In addition, wavelet images with three scales facilitate flexibly detecting salient regions with different sizes. Subsequently, edge detection by bipolar cells is simulated by applying high-pass filtering to each wavelet image. Three Sobel filters are used to extract edges (high-frequency components) in the vertical, horizontal and diagonal directions, as seen from Figure 4. This converts the wavelet image into three 'edge images' where an edge represents the saliency of the corresponding pixel, because this edge indicates the difference between the pixel and surrounding ones. Afterwards, to highlight such edges and suppress noises, the second smoothing by Amacrine cells is conducted using Gaussian filter for each edge image. We name the resulting image as a 'Gaussian image'.

As a result of the aforementioned steps, 54 Gaussian images are obtained for each video frame (i.e., 6 feature maps $\times$ 3 wavelet scales $\times$ 3 edge directions). These Gaussian images are now integrated into a bottom-up saliency map. For computational efficiency, each Gaussian image is firstly scaled to the size 22×18 pixels. Note that each pixel in this scaled image corresponds to a region in the original video frame, based on the relative positional relation between the pixel and the region. In this sense, pixels in the scaled Gaussian image are called *macro-blocks*. The subsequent bottom-up saliency map extraction and top-down process use macro-blocks as the unit. Also, keep in mind that in Figures 3, 4 and 7, each saliency map with the size 22×18 pixels is resized to the original video frame size. As is clear from red dotted arrows in Figure 3, this resizing allows us to determine the saliency of each patch in the original video frame. The bottom-up saliency map of the video frame is created by taking the average of 54 Gaussian images for each macro-block. In addition, this map is normalised so that the most and least salient macro-blocks have 0 and 1, respectively.

B. Top-down Process

The top-down process implements attention related to *tasks*. According to the contextual cueing described in Section II, we define a task as the expectation that, for shots with a certain type of spatial layouts, a human supposes to locate salient regions of a target concept at similar places. In Figure 5, where the target concept is *Car*, let us consider the situation where the human already saw the top-left shot and confirmed that the region of the moving car is salient. Based on this experience, the human expects that the region of the moving car in the top-right shot is also salient, because it has the similar spatial layout to the top-left shot. Similarly, when the human knows that the region of the car front in the bottom-left shot is salient, he/she should apply the same logic to the bottom-right shot. Like this, a task is the human's expectation for salient regions of the target concept based on the similarity in camera techniques and shooting environments. However, only using such tasks lacks the examination of whether detected regions are visually (perceptually) salient or not. To resolve this, it is important to integrate the top-down and bottom-up processes. Hence, the top-down process in our method works to refine the bottom-up saliency map, so that salient regions detected by the bottom-up process are biased based on task-related attention described above.

First, we explain how to model task-related attention, which generally occurs by adjusting responses of cells in the retina to a specific type of stimuli [13], [14], [15]. Based on this, we re-use the retina model in Figure 4 and model task-related attention as the adjustment of 54 Gaussian images to a target concept [14]. Let us assume P positive shots, where each of them is associated with L $(= 54)$ Gaussian images that are individually represented with N $(= 22 \times 18)$ macro-blocks. For the ith positive shot $(1 \leq i \leq P)$, we create an L-dimensional vector $\boldsymbol{x}_{in} = (x_{in}^1, \cdots, x_{in}^L)$ by aggregating values at the nth macro-block $(1 \leq n \leq N)$ in L Gaussian images. For example, assuming that the positive shot in Figure 4 is the ith one, $\boldsymbol{x}_{i1}$ is the collection of values at the top-left macro-block in 54 Gaussian images. Note that the exact definition is $\boldsymbol{x}_{ii'n}$ corresponding to the nth macro-block for the i'th video frame in the ith positive shot. But, this makes

the discussion unnecessarily complex. Thus, we use $\boldsymbol{x}_{in}$ for simplicity. Extending $\boldsymbol{x}_{in}$ to $\boldsymbol{x}_{ii'n}$ is straightforward, and our experiments are conducted using video frames sampled at every second.

A task is modelled as a function f_t to adjust $\boldsymbol{x}_{in}$. Here, f_t is a linear function $f_t(\boldsymbol{x}_{in}) = \sum_{l=1}^{L} w_t^l x_{in}^l$ where $\{w_t^l\}_{l=1}^{L}$ is a parameter set estimated using salient region annotation obtained by WSL in the next section. However, it is difficult to deterministically decide which task is used for a positive shot. In other words, it is impossible to objectively find to what extent each task is applicable for the positive shot, in terms of differences in appearances of the target concept, camera techniques and shooting environments. For example, in comparison to the top-left shot in Figure 5, let us consider a shot where a car is moving from left to right, the camera is placed closer to the car, and the situation is urban. It is unknown whether "Task 1" in Figure 5 can be used for this shot. Hence, we adopt a 'soft assignment' approach where functions $\{f_t\}_{t=1}^{T}$ for T tasks are probabilistically related to each positive shot. That is, $\boldsymbol{x}_{in}$ is adjusted by $\sum_{t=1}^{T} c_{it} f_t(\boldsymbol{x}_{in})$ where c_{it} represents the weight of f_t for the ith positive shot.

Using task-related attention based on f_ts, we explain how to refine a bottom-up saliency map. Let b_{in} be the value at the nth macro-block in the bottom-up saliency map for the ith positive shot. We carry out the refinement of b_{in} as the weighted combination of b_{in} and the adjustment of $\boldsymbol{x}_{in}$, that is, $\sum_{t=1}^{T} c_{it} f_t(\boldsymbol{x}_{in}) + \alpha_{ib} b_{in}$. Here, α_{ib} is the weight representing the importance of the bottom-up saliency map. The top-down process estimates the following two components: The one is a set of parameter sets for T functions $F = \{\{w_t^l\}_{l=1}^{L}\}_{t=1}^{T}$, and the other is a set of weight vectors $C = \{\boldsymbol{c}_i = (c_{i1}, \cdots, c_{iT}, \alpha_{ib})\}_{i=1}^{P}$ where $\boldsymbol{c}_i$ represents weights of functions and the bottom-up saliency map for the ith positive shot. These F and C are estimated so as to accurately approximate salient region annotation $y_{in} \in \{0, 1\}$, where $y_{in} = 1$ means that the nth macro-block in the ith positive shot is salient, otherwise non-salient. Note that by regarding the binary value y_{in} as continuous, F and C are estimated as the regression problem of such continuous values using $\sum_{t=1}^{T} c_{it} f_t(\boldsymbol{x}_{in}) + \alpha_{ib} b_{in}$.

In particular, for effective estimation of F, we employ *multi-task learning* that simultaneously estimates the parameter set $\{w_t^l\}_{l=1}^{L}$ for each of T functions by considering their correlation [14]. Compared to estimating such sets independently, the correlation can make it clearer what kind of salient regions are handled by each function. To sum up, the following optimisation is performed to estimate F and C [14]:

$$\min_{F,C} \frac{1}{PN} \sum_{i=1}^{P} \sum_{n=1}^{N} l\left(\sum_{t=1}^{T} c_{it} f_t(\boldsymbol{x}_{in}) + \alpha_{ib} b_{in},\ y_{in} \right), \quad (7)$$

where $l(\cdot)$ indicates the loss (error) computed as the squared difference between the refined saliency value ($\sum c_{it} f_t(\boldsymbol{x}_{in}) + \alpha_{ib} b_{in}$) and the annotated one (y_{in}). Equation (7) aims to extract F and C that minimise the average refinement error for $P \times N$ macro-blocks. This optimisation can be solved by an EM-like algorithm, which iteratively switches between the estimation of C keeping F fixed, and the one of F keeping C fixed (see [14] for more details).

The bottom-up saliency map of each test shot is refined using the estimated F and C. Let us assume the jth test shot where the nth macro-block is characterised by $\boldsymbol{x}_{jn}$ based on 54 Gaussian images and b_{jn} of the bottom-up saliency map. Based on the contextual cueing in Section II, the same refinement mechanism is used for shots with similar spatial layouts. Thus, we first find the $\hat{i}$th positive shot that has the most similar spatial layout to the jth test shot. Then, the saliency value of the nth macro-block is refined into s_{jn} using F and the weight vector $\boldsymbol{c}_{\hat{i}}$ for the $\hat{i}$th positive shot [14]:

$$s_{jn} = \sum_{t=1}^{T} c_{\hat{i}t}\ f_t(\boldsymbol{x}_{jn}) + \alpha_{\hat{i}b}\ b_{jn}. \quad (8)$$

The computation of similarities regarding spatial layouts requires to consider the global visual characteristic of a shot. To this end, for each of six feature maps in Figure 4, a histogram is created by quantising the value of every pixel into eight bins. This histogram represents the overall distribution of values in the feature map with respect to intensity, red-green contrast, blue-yellow contrast, or so forth. We use the concatenation of such histograms as the feature of the shot, and compute the similarity between two shots as their cosine similarity.

C. Weakly Supervised Learning

Motivated by the success of Support Vector Machines (SVMs) in object detection/recognition and SIN [20], [27], we employ the WSL method that is an extended SVM for WSL [16]. Usually, an SVM is trained using training shots associated with binary labels, that is, the presence or absence of a target concept. Then, it is used to predict the same type of binary labels for test shots. On the other hand, the method in [16] uses training shots with binary labels to build an SVM that can identify regions of the target concept. The main idea is that the method simultaneously localises the most distinctive regions and builds an SVM to distinguish those regions. More specifically, the SVM is trained so as to characterise regions that are contained in every positive shot, but are not contained in any negative shot. These regions are likely to contain the target concept.

First of all, we explain how regions of a target concept are localised by the method in [16]. Let x be an arbitrary shot without specifying it is positive or negative. We define the localisation as the problem to find the best 'rectangular' region $\hat{r}$ from the set of all possible regions $\mathcal{R}(x)$ in x. With respect to this, one rectangular region is defined by four parameters, the top-left, top-right, bottom-left and bottom-right positions. Thus, simply speaking, $\mathcal{R}(x)$ includes W^2H^2 rectangular regions if the frame size of x is $W \times H$ pixels. Since efficient search of $\hat{r}$ will be discussed later, we here concentrate on the localisation mechanism. Assuming that a feature vector $\boldsymbol{\varphi}(r)$ can be computed for any region $r \in \mathcal{R}(x)$ using descriptors in r, a linear SVM with the discrimination function $\boldsymbol{w}\boldsymbol{\varphi}(r) + b$ is used to examine whether r contains the target concept. Here, b is a bias term and $\boldsymbol{w}$ is an weight vector in which each dimension represents the relevance to the presence of the target concept. As $\boldsymbol{\varphi}(r)$ has larger values on more relevant dimensions, the target concept is more likely to appear in r. Therefore, $\hat{r}$ is determined as the region that maximises the discrimination function [16]:

$$\hat{r} = \operatorname*{argmax}_{r \in \mathcal{R}(x)} (\boldsymbol{\varphi}(r) + b). \quad (9)$$

Based on this localisation mechanism, let x_i^+ and x_j^- be the ith positive and jth negative shots for a target concept, respectively. The parameters of the SVM (i.e., $\boldsymbol{w}$ and b) is estimated by solving the following optimisation problem [16]:

$$\min_{\boldsymbol{w},b} \left(\frac{1}{2}||\boldsymbol{w}|| + C\sum_i \alpha_i + C\sum_j \beta_j \right), \quad (10)$$

$$\text{s.t.} \quad \max_{r\in\mathcal{R}(x_i^+)} (\boldsymbol{w}\boldsymbol{\varphi}(r) + b) \geq +1 - \alpha_i \ (\alpha_i \geq 0), \quad (11)$$

$$\max_{r\in\mathcal{R}(x_i^-)} (\boldsymbol{w}\boldsymbol{\varphi}(r) + b) \leq -1 + \beta_i \ (\beta_i \geq 0), \quad (12)$$

where α_i (or β_j) is a slack variable representing the degree of mis-classification for the region in the x_i^+ (or x_j^-). In addition, C is a parameter to control the effect of mis-classification. The optimal $\boldsymbol{w}$ and b yields the situation where at least one region in x_i^+ is classified as positive (Equation (11)), while all regions in x_j^- are classified as negative (Equation (12)). The optimisation is solved using a coordinate descent approach, which iterates examining each training shot to find the best region that maximises the current discrimination function, and updating this function using newly found best regions [16].

For efficient optimisation, it is important to quickly find the best region for each training shot. To this end, we employ the region search method developed in [16], [28]. First, we use 'Bag-of-Visual-Word' (BoVW) representation to express the feature $\boldsymbol{\varphi}(r)$ by quantising 'Scale-Invariant Feature Transform' (SIFT) descriptors in r. Each SIFT descriptor represents the edge shape in a patch, reasonably invariant to changes in scale, rotation, viewpoint and illumination [4]. As pre-processing, SIFT descriptors are extracted from patches, which have the radius of 10 pixels and are located at every sixth pixel in each training shot. Then, one million SIFT descriptors are randomly sampled and grouped into 1000 clusters, where each cluster centre is a 'Visual Word' (VW) representing a characteristic SIFT descriptor. Afterwords, by assigning each SIFT descriptor in r to the most similar VW, $\boldsymbol{\varphi}(r) = (\varphi_1(r), \cdots, \varphi_D(r))$ ($D = 1000$) is created where $\varphi_d(r)$ represents the frequency of the dth VW.

With the BoVW representation, the discrimination function of a linear SVM can be transformed as follows [28]:

$$\boldsymbol{w}\boldsymbol{\varphi}(r) + b = \sum_{d=1}^{D} w_d\varphi_d(r) + b = \sum_{n=1}^{N} w(\text{VW}_n) + b, \quad (13)$$

where N is the number of SIFT descriptors in r, and $w(\text{VW}_n)$ is the weight in $\boldsymbol{w} = (w_1, \cdots, w_D)$ corresponding to the VW associated with the nth SIFT descriptor. For example, $w(\text{VW}_n) = w_1$ if the nth SIFT descriptor is assigned to the first VW. Equation (13) means that the discrimination function can be computed by simply adding weights of VWs linked to SIFT descriptors in r. This enables us to estimate the 'upper bound' for a set of regions [28]. No region in this set takes the discrimination function value larger than the upper bound. Hence, the best region $\hat{r}$ maximising the discrimination function can be efficiently found by discarding many sets of regions for which upper bounds are small.

Finally, $\hat{r}$ that is detected in the ith positive shot x_i^+ using the optimised $\boldsymbol{w}$ and b, is used as the annotated salient region in the top-down process. Note that since the top-down process is based on 22×18 macro-blocks (pixels), the video frame in x_i^+ is resized to this size by preserving the relative spatial relation between $\hat{r}$ and the frame. Then, if the nth macro-block falls in $\hat{r}$, $y_{in} = 1$ otherwise $y_{in} = 0$.

D. Semantic Indexing with Filtering

As a result of FoA with WSL, saliency maps have been computed for all shots. As illustrated in Figure 3, our SIN method extracts the feature of a shot as an extended BoVW representation by weighting descriptors based on its saliency map. Let $\{\text{VW}_n\}_{n=1}^N$ be a set of VWs that are associated with N SIFT descriptors extracted from the whole of the shot. Also, let us denote by $\{p_n\}_{n=1}^N$ a set of centre positions of patches from which the N SIFT descriptors are extracted. That is, VW_n is the VW associated with the nth SIFT descriptor, which is extracted from the nth patch having the centre position p_n. Since the size of the saliency map is 22×18 pixels (macro-blocks), it is resized to the same size as the video frame of the shot. By checking this resized saliency map, we obtain $\{s_n\}_{n=1}^N$ where s_n represents the saliency of p_n. Then, an 'weighted' D-dimensional vector $\boldsymbol{\phi} = (\phi_1, \cdots, \phi_D)$ is created. Regarding this, in the normal BoVW representation, the value of the dimension corresponding to VW_n is incremented, so that the resulting feature represents the frequency of each VW. Different from this, in our extended BoVW representation, the value of the dimension is increased by s_n. Thereby, if VW_n is extracted from the patch in a salient region where the target concept probably appears, VW_n's effect is large, otherwise small (see red dotted arrows in Figure 3). Like this, the weighted vector $\boldsymbol{\phi}$ emphasises the appearance of the target concept while suppressing effects of other concepts. Finally, using positive and negative shots represented by such $\boldsymbol{\phi}$s, a detector is constructed as a non-linear SVM with Radial Basis Function (RBF) kernel [29].

Before constructing the detector, filtering is performed to eliminate positive shots where the target concept appears in non-salient regions, because their weighted vectors falsely emphasise other concepts. To this end, we make a simple assumption that the target concept is salient if its region is large. Hence, positive shots are filtered out if regions detected by the WSL method are smaller than a threshold. Also, this filtering is executed when applying the detector to test shots. But, the purpose is to distinguish test shots where salient regions certainly include the target concept from the other ones. For the latter, we take into account FoA failures where falsely detected salient regions would cause weighted vectors undesirably emphasising non-target concepts. Thus, weighted vectors are extracted only from test shots where regions detected by the WSL method are larger than the threshold. For the other shots, non-weighted vectors are extracted based on the normal BoVW representation. Finally, the list of sorted test shots in terms of the detector's outputs is returned as the SIN result .

IV. Experimental Results

In this section, we firstly examine the effectiveness of our FoA method extended by WSL, and then evaluate the performance of SIN based on this extended FoA method.

A. Evaluation of FoA based on WSL

To examine the adequacy of incorporating WSL into FoA, we target three concepts *Person*, *Car* and *Explosion_Fire*. For each concept, we use 1000 positive shots and 5000 negative shots in TRECVID 2009 video data [2]. The performance is

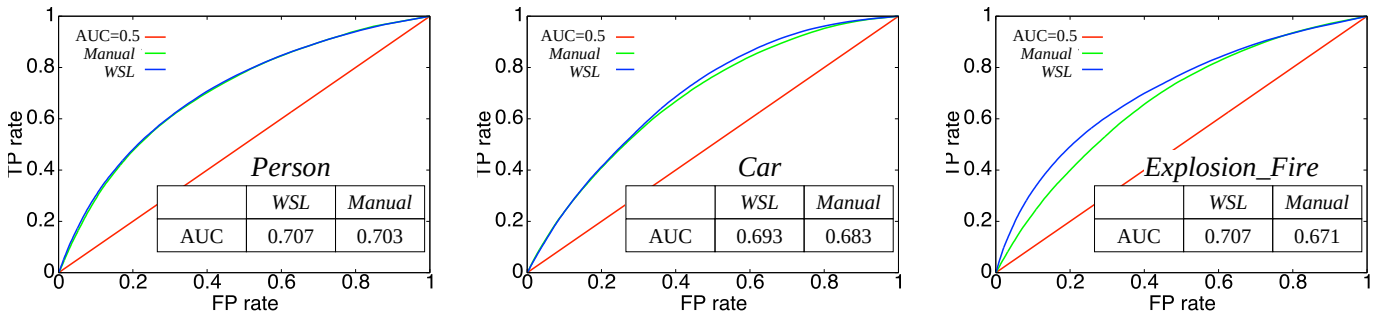

Figure 6. Performance comparison between *WSL* and *Manual*.

evaluated on 1000 test shots where the ground truth of salient regions is manually provided. We compare two FoA methods, *WSL* and *Manual*, which use positive shots where salient regions are annotated by WSL and by manual, respectively. Using manually annotated salient regions can be considered as the best approach. Hence, the comparison between *WSL* and *Manual* aims to investigate how useful salient regions obtained by WSL are, compared to those provided by the best manual approach.

Figure 6 shows Receiver Operating Characteristic (ROC) curves for *WSL* and *Manual*. Each curve is created by calculating True Positive (TP) and False Positive (FP) rates using different thresholds. Here, a macro-block in a saliency map is regarded as salient if its saliency is larger than a threshold. A TP is the number of macro-blocks that are correctly detected as salient, and an FP is the number of macro-blocks falsely detected as salient. A high performance is depicted by an ROC curve biased towards the top-left. In Figure 6, ROC curves of *WSL* and *Manual* are nearly the same for all concepts. As another evaluation measure, an Area Under Curve (AUC) represents the area under an ROC curve. A larger AUC indicates a superior performance where a high TP is achieved with a small FP. Figure 6 presents that *WSL*'s AUCs are nearly the same or even larger than those of *Manual*. The results described above verifies that salient regions annotated by WSL lead to the FoA performance that is comparable to the one based on the best manual approach.

It should be noted that several regions where a target concept does not appear are falsely detected by WSL, and used as annotated salient regions in the top-down process. For example, the red rectangular region in Figure 7 (a) is falsely regarded as showing a car. However, as seen from the bottom-up saliency map in Figure 7 (b), the saliency of this region is very low. More specifically, marco-blocks in this region have very small $\boldsymbol{x}_{jn}$ and b_{jn} in Equation (8). Thus, they cannot be regarded as salient even with the refinement by the top-down process, as shown in Figure 7 (c). Like this, errors in WSL are alleviated based on saliencies obtained by the bottom-up process. In other words, FoA works appropriately as long as regions obtained by WSL are mostly correct.

B. Evaluation of SIN using FoA

We evaluate the effectiveness of SIN utilising FoA using video data in TRECVID 2011 SIN light task [2]. According to the official guideline, 23 target concepts shown in Figure 8 are selected. For each target concept, a detector is constructed with 30000 training shots collected from 240918 shots in 11485 development videos. Here, positive shots are collected based on the result of web-based collaborative annotation

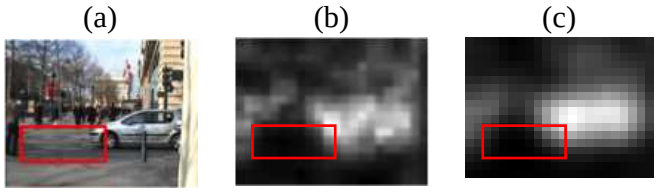

Figure 7. An example of alleviating errors in WSL based on bottom-up saliency maps.

effort where many users on the web collaboratively annotate shots in development videos [30]. Negative shots are collected as randomly selected shots in development videos. This is because the concept usually appears only in a small number of shots, so almost all of randomly selected shots can serve as negative [31]. Although annotation data collected by [30] contain negative shots, our preliminary experiment showed that they lead to worse performance than randomly selected shots. One main reason is the 'biased' shot selection based on active learning, where users are asked to only annotate shots similar to already collected positive shots [30]. In contrast, negative shots by 'non-biased' random selection yield more accurate concept detection. The constructed detector is tested on 125880 shots in 8215 test videos.

To examine the effectiveness of weighting descriptors based on FoA and that of filtering, we compare three methods *Baseline*, *Weight* and *Weight+Filter*. *Baseline* and *Weight* use features that are extracted as BoVW representations without and with weighting, respectively. *Weight+Filter* extends *Weight* by adding the filtering process. Figure 8 shows the performance comparison among *Baseline*, *Weight* and *Weight+Filter* in form of a bar graph. For each concept, the top, middle and bottom bars represent Average Precisions (APs) of *Baseline*, *Weight* and *Weight+Filter*, respectively. An AP approximates the area under a recall-precision curve. Regarding its computation, a SIN result for a target concept is a list of 2000 test shots ranked in terms of the detector's outputs. The AP is the average of precisions, each of which is computed at a position where a 'correct' test shot showing the target concept is ranked. A larger AP means a better SIN result where correct test shots are ranked at higher positions. Also, each of three bars at the bottom of Figure 8 presents the Mean of APs (MAP) over 23 concepts as an overall evaluation measure. Figure 8 indicates that *Weight* outperforms *Baseline* for many concepts. The MAP of the former (0.0708) is about 5% higher than that of the latter (0.0676). This validates the effectiveness of using FoA in SIN. In addition, *Weight+Filter*'s MAP (0.0731) exhibits that adding the filtering process improves *Weight*'s MAP by about 3%. This verifies the effectiveness of filtering.

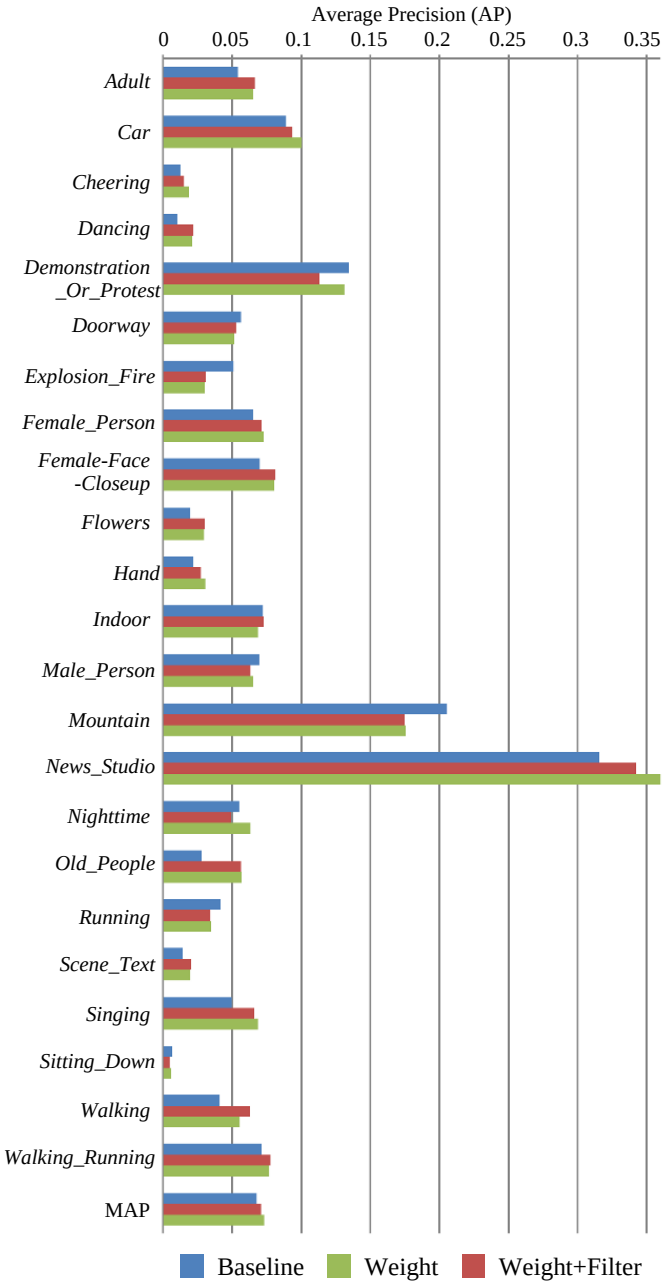

Figure 8. Performance comparison among *Baseline*, *Weight* and *Weight+Filter*.

V. Conclusion and Future Work

In this paper, we introduced a SIN method that detects a target concept based on FoA. Our method extracts the feature of a shot by weighting descriptors based on saliencies of patches, from which these descriptors are derived. This enables us to suppress adverse effects of regions irrelevant to the target concept, and emphasise its appearance. For effective integration of SIN and FoA, WSL is employed so that salient region annotation required for the top-down process can be generated from shots labelled only with the presence or absence of the target concept. In addition, filtering is conducted to eliminate shots where non-target concepts are emphasised, by examining the coincidence between salient regions detected by FoA and the target concept's regions identified by WSL. Experimental results validated the effectiveness of all the three contributions, that is, using FoA in SIN, extending FoA with WSL, and filtering.

We will investigate the following two issues in the future: The first is that we used the most standard feature (i.e., BoVW representation) to justify the framework of using FoA in SIN. But, it is relatively straightforward to extend this framework to more sophisticated features, such as the ones based on GMM [5], [6], Fisher vector [7] and deep learning [7] described in Section II. Our ideas for this are summarised below. First, the extraction of GMM-based features starts with estimating a reference GMM using randomly sampled descriptors. Then, the GMM for a shot is computed by modifying the reference GMM based on descriptors extracted from the shot. FoA can be used to control the degree of modification based on the saliency of each descriptor, so that descriptors extracted from patches in salient regions have large influences on the resulting GMM. Second, the reference GMM is also used for Fisher vector-based features. Here, the feature of a shot is computed by averaging first (or second) order differences of descriptors to the mean of each Gaussian component in the reference GMM [32]. This averaging can be improved by considering the saliency of each descriptor. Last, one key factor in deep learning is how to define receptive fields, each of which represents a region that a neuron uses to extract a feature. FoA can be used to prioritise or select receptive fields of neurons by checking saliencies of regions. We will test each of the above-mentioned extensions.

The second issue is that FoA causes the performance degradation for some concepts such as *Explosion_Fire* and *Mountain* in Figure 8. One main region is non-rectangular shapes of these concepts, because our current WSL method can only identify rectangular regions. However, rectangular regions are too coarse to precisely localise non-rectangular concepts, and inevitably include other concepts. As a result, the top-down process does not work well. Hence, we will extend our WSL method by adopting an efficient search algorithm for regions with arbitrary shapes [33].

Appendix
List of Abbreviations

The list below shows abbreviations used in this paper. Each line presents an abbreviation, its full name, and the section where it appears for the first time.

SIN	:	Semantic INdexing (Section I)
FoA	:	Focus of Attention (Section I)
WSL	:	Weakly Supervised Learning (Section I)
SOD	:	Salient Object Detection (Section II)
DSD	:	Discriminant Saliency Detection (Section II)
GMM	:	Gaussian Mixture Model (Section II)
SVM	:	Support Vector Machines (Section III-C)
BoVW	:	Bag-of-Visual-Word (Section III-C)
VW	:	Visual Word (Section III-C)
SIFT	:	Scale-Invariant Feature Transform (Section III-C)
RBF	:	Radial Basis Function (Section III-D)
ROC	:	Receiver Operating Characteristic (Section IV-A)
TP	:	True Positive (Section IV-A)
FP	:	False Positive (Section IV-A)
AUC	:	Area Under Curve (Section IV-A)
AP	:	Average Precisions (Section IV-B)
MAP	:	Mean of Average Precision (Section IV-B)

REFERENCES

[1] K. Shirahama, T. Matsumura, M. Grzegorzek, and K. Uehara, "Empowering semantic indexing with focus of attention," in Proc. of MMEDIA 2015, 2015, pp. 33–36.

[2] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVid," in Proc. of MIR 2006, 2006, pp. 321–330.

[3] C. G. M. Snoek and M. Worring, "Concept-based video retrieval," Found. Trends Inf. Retr., vol. 2, no. 4, 2009, pp. 215–322.

[4] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, 2010, pp. 1582–1596.

[5] N. Inoue and K. Shinoda, "A fast and accurate video semantic-indexing system using fast MAP adaptation and gmm supervectors," IEEE Trans. Multimed., vol. 14, no. 4, 2012, pp. 1196–1205.

[6] K. Shirahama and K. Uehara, "Kobe university and muroran institute of technology at TRECVID 2012 semantic indexing task," in Proc. of TRECVID 2012, 2012.

[7] C. G. M. Snoek et al., "Mediamill at TRECVID 2014: Searching concepts, objects, instances and events in video," in Proc. of TRECVID 2014, 2014.

[8] S. Frintrop, E. Rome, and H. I. Christensen, "Computational visual attention systems and their cognitive foundations: A survey," ACM Trans. Appl. Percept., vol. 7, no. 1, 2010, pp. 6:1–6:39.

[9] A. Borji and L. Itti, "State-of-the-art in visual attention modeling," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 1, 2013, pp. 185–207.

[10] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: A survey," 2014. [Online]. Available: http://arxiv.org/abs/1411.5878

[11] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 12, 2000, pp. 1349–1380.

[12] A. Oliva, A. Torralba, M. Castelhano, and J. Henderson, "Top-down control of visual attention in object detection," in Proc. of ICIP 2003, 2003, pp. 253–256.

[13] S. Chikkerur, T. Serre, C. Tan, and T. Poggio, "What and where: A bayesian inference theory of attention," Vis. Res., vol. 50, no. 22, 2010, pp. 2233–2247.

[14] J. Li, Y. Tian, T. Huang, and W. Gao, "Probabilistic multi-task learning for visual saliency estimation in video," Int. J. Comput. Vis., vol. 90, no. 2, 2010, pp. 150–165.

[15] W. Kienzle, M. O. Franz, B. Schölkopf, and F. A. Wichmann, "Center-surround patterns emerge as optimal predictors for human saccade targets," J. Vis., vol. 9, no. 5, 2009, pp. 1–15.

[16] M. Nguyen, L. Torresani, F. De la Torre, and C. Rother, "Weakly supervised discriminative localization and classification: a joint learning process," in Proc. of ICCV 2009, 2009, pp. 1925–1932.

[17] G. Kootstra, A. Nederveen, and B. d. Boer, "Paying attention to symmetry," in Proc. of BMVC 2008, 2008, pp. 111.1–111.10.

[18] P. Jiang, H. Ling, J. Yu, and J. Peng, "Salient region detection by UFO: Uniqueness, focusness and objectness," in Proc. of ICCV 2013, 2013, pp. 1976–1983.

[19] H. Jiang, J. Wang, Z. Yuan, T. Liu, and N. Zheng, "Automatic salient object segmentation based on context and shape prior," in Proc. of BMVC 2011, 2011, pp. 110.1–110.12.

[20] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah, "High-level event recognition in unconstrained videos," Int. J. Multimed. Inf. Retr., vol. 2, no. 2, 2013, pp. 73–101.

[21] T. Liu et al., "Learning to detect a salient object," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 2, 2011, pp. 353–367.

[22] H. Jiang, H. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in Proc. of CVPR 2013, 2013, pp. 2083–2090.

[23] L. Elazary and L. Itti, "Interesting objects are visually salient," J. Vis., vol. 8, no. 3, 2008, pp. 1–15.

[24] D. Gao, S. Han, and N. Vasconcelos, "Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 6, 2009, pp. 989–1005.

[25] G. Sharma, F. Jurie, and C. Schmid, "Discriminative spatial saliency for image classification," in Proc. of CVPR 2012, 2012, pp. 3506–3513.

[26] A. Treisman and G. Gelade, "Feature integration theory of attention," Cogn. Psychol., vol. 12, no. 1, 1980, pp. 97–136.

[27] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, 2010, pp. 1627–1645.

[28] C. Lampert, M. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in Proc. CVPR 2008, 2008, pp. 1–8.

[29] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, 2011, pp. 27:1–27:27.

[30] S. Ayache and G. Quénot, "Video corpus annotation using active learning," in Proc. of ECIR 2008, 2008, pp. 187–198.

[31] A. Natsev, M. R. Naphade, and J. Tešić, "Learning the semantics of multimedia queries and concepts from a small number of examples," in Proc. of MM 2005, 2005, pp. 598–607.

[32] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in Proc. of BMVC 2011, 2011, pp. 76.1–76.12.

[33] S. Vijayanarasimhan and K. Grauman, "Efficient region search for object detection," in Proc. of CVPR 2011, 2011, pp. 1401–1408.