University of Siegen, Kobe University and NICT at TRECVID 2015 SIN and MED Tasks

Kimiaki Shirahama*, Takashi Shinozaki‡, Yasuyuki Matsumoto†, Marcin Grzegorzek* and Kuniaki Uehara†
* Pattern Recognition Group, University of Siegen
kimiaki.shirahama@uni-siegen.de, marcin.grzegorzek@uni-siegen.de
† Graduate School of System Informatics, Kobe University
matsumoto@ai.cs.kobe-u.ac.jp, uehara@kobe-u.ac.jp
‡ Brain Imaging Technology Laboratory, CiNet, National Institute of Information and Communications Technology (NICT)
tshino@nict.go.jp

***Abstract*—For the SIN task, we have submitted the following three runs:**

1) *2C_M_D_siegen_kobe_nict.15_1:* **This run uses the Convolutional Neural Network (CNN) trained on ImageNet dataset. For each shot, a feature is extracted as outputs of the fifth, sixth or seventh layer in the CNN. SIN is carried out by linearly fusing SVMs each of which is trained on one feature using IACC.1.tv10.training and IACC.1.A-C.**
2) *2C_M_D_siegen_kobe_nict.15_2:* **To examine the effectiveness of the above fusion approach, this run performs SIN only using an SVM which is trained on the feature corresponding to outputs of the sixth layer in the CNN.**
3) *2C_M_D_siegen_kobe_nict.15_3:* **We attempt to use the motion information in the video obtained by a separated Deep Neural Network (DNN) with Motion Receptive Field (MRF) inputs. The motion information is extracted as a 900-dimensional vector, and combined with the feature corresponding to outputs of the sixth layer in the CNN. SIN is examined by an SVM with the feature vector.**

Results by these runs indicate that we could achieve a reasonable performance only using image-based features (the CNN is applied to video frames independently), but could not use motion information effectively. Moreover, other types of features like audio are necessary for accurate concept detection.

For the MED task, we have submitted the following five runs:

1) *SiegenKobeNict_MED15_MED15EvalSub_PS_10Ex_SML_psvm10_1:* **Each video is divided into shots for which** 346 **concepts defined in SIN and** 1000 **objects defined in ImageNet are detected. The video is then represented as a** 1346**-dimensional vector by max-pooling of** 1346 **concepts/objects detection results over shots. Based on this, an event is detected by building an SVM using** 10 **example videos and background training videos.**
2) *SiegenKobeNict_MED15_MED15EvalSub_PS_100Ex_SML_csvm100_1:* **This run is the same to the above one except that** 100 **example videos are used.**
3) *SiegenKobeNict_MED15_MED15EvalSub_PS_100Ex_SML_chcrfseq100_1:* **To discriminate between relevant and irrelevant shots to an event and consider their temporal structures, this runs builds a Hidden Conditional Random Field (HCRF) using** 100 **example videos and background training videos. Each shot is assigned to a hidden state representing the relevance to the event by considering the hidden state assigned to the previous shot. The occurrence of the event is examined based on the assignment of hidden states to shots in the video.**
4) *SiegenKobeNict_MED15_MED15EvalSub_PS_100Ex_SML_chcrftree100_1:* **To treat long-range temporal structures of shots, this run constructs an HCRF by representing a video as a tree structure, where shots are hierarchically merged into nodes based on their visual similarities and temporal distances. That is, nodes express shot sequences with different time lengths.**
5) *SiegenKobeNict_MED15_MED15EvalSub_PS_100Ex_SML_csvm-hcrftree100_1:* **This run fuses the second and fourth runs by simply averaging outputs by them.**

Although we validated the effectiveness of using tree structures, at present, non-linear SVMs based on max-pooling outperform HCRFs due to the low discrimination power of hidden states characterised by linear functions.

I. Introduction

TREC Video Retrieval Evaluation (TRECVID) is an annual worldwide competition where large-scale benchmark video data are used to evaluate methods developed all over the world [1]. At TRECVID 2015 [2], we participated in the Semantic INdexing (SIN) task where methods for detecting concepts like *Airplane*, *Boat_Ship* and *Computers* in shots are evaluated, and the Multimedia Event Detection (MED) task where the assessment addresses methods which identify videos containing certain events like "Bike trick", "Marriage proposal" and "Beekeeping". This paper presents our methods developed for the SIN and MED tasks.

For the SIN task, our purpose is to examine the effectiveness of *deep learning* which constructs a feature hierarchy with higher-level features formed by the composition of lower-level ones [3], [4]. Traditional hand-crafted features like SIFT and HOG are insufficient for representing diverse visual appearances. The reason is that these are built upon pre-specified representations which are not necessarily optimal to represent various visual appearances. Thus, much research attention has recently put on *feature learning* (or representation learning) to extract useful features from data [3], and deep learning is the most representative approach. One big advantage of deep learning is its discrimination power. A feature hierarchy can represent up to $O(2^N)$ visual appearances only using $O(N)$ parameters [3]. Intuitively, the discrimination power of features at one layer is exponentially increased based on numerous combinations of features at the previous layer. Based on such feature hierarchies, deep learning has showed remarkable

performance improvements on several worldwide competitions on image, video and audio classification [4], [5], [6].

We use a feature hierarchy represented by a Convolutional Neural Network (CNN) which is built on 1.2 million training images in ImageNet dataset [7]. Since each layer in the CNN represents a distinct abstraction of (a video frame in) a shot, we consider that the performance of concept detection is improved by using multiple features derived from different layers. Based on this idea, we compute the following three features for the shot: The first, second and third features consist of neuron outputs at the fifth, sixth and seventh layers in the CNN, respectively. Then, a concept is detected by fusing SVMs each of which is built on one feature. Moreover, we also compute another feature vector for motion information by a separate deep neural network. The network has a similar structure to 'Google Brain' [8], and spontaneously obtains the motion related features with unsupervised learning rule. We combine the motion feature vector with the feature vector of the CNN, and detect a concepts with a SVM.

For the MED task, we focus on the following two problems: The first is the *weakly supervised setting* where each training video is annotated only with the occurrence or absence of an event, despite the fact that this video may include several semantically different shots [9]. For simplicity, we call training videos annotated with the occurrence of the event and those annotated with its absence *positive videos* and *negative videos*, respectively. In particular, because of the weakly supervised setting, positive videos include several irrelevant shots to the event. For example, while shots showing kissing and hugging are relevant to the event "marriage proposal", shots displaying conversation and surrounding situations are included in positive videos for this event. Such irrelevant shots clearly have adverse influences on building a classifier which can accurately identify videos where the event occurs. Hence, we need to analyse training videos to discriminate between relevant and irrelevant shots to the event.

The second problem is the extraction of *temporal structures* of an event. For example, for the event "marriage proposal", a shot where a man and woman are talking to each other is often followed by a shot where they are hugging. Such temporal structures are useful for disambiguating the event presented in a video. However, we target 'unconstrained' videos where shots are taken by any camera technique and connected by any editing technique. As a result, videos where the event occurs do not show apparent temporal structures. For the event "marriage proposal", in one video the conversation and hugging of a man and woman are presented in two consecutive shots, while in another video they are separated by additional shots like a shot where a man gives a ring to a woman, and a shot showing the surrounding situation. To effectively mine temporal structures from unconstrained videos, we need to examine shots and shot sequences in different temporal abstraction levels. For the above example, we aim to extract a temporal structure where a shot sequence including the conversation between a man and woman is followed by the one including their hugging.

To address the weakly supervised setting and temporal structure extraction, we firstly use *time-constrained shot clustering* method [10] to represent a video as a tree structure where nodes express shot sequences with different time lengths. Then, given positive and negative videos for an event, our method constructs a *Hidden Conditional Random Field* (HCRF) which is a probabilistic discriminative classifier with a set of hidden states [11]. Here, the occurrence of the event in a video is predicted by assigning nodes in the tree structure to hidden states. In other words, the weakly supervised setting is handled using hidden states as the intermediate layer to discriminate between relevant and irrelevant nodes (i.e., shot sequences) to the event. In addition, transitions among hidden states indicate temporal structures specific to the event. To examine the effectiveness of the above-mentioned approach, we compare it to an SVM which has no mechanism to discriminate between relevant and irrelevant shots, and an HCRF which assigns shots to hidden states without considering their temporal abstraction into nodes [9].

II. Semantic Indexing

Fig. 1 illustrates an overview of three types of SIN methods that we have examined. All of three use the output of a Convolutional Neural Network (CNN).

For the first two methods, we only use static image information in a video. In order to utilise recent progress of deep neural network, we employed 'Caffe' which is one of the most popular CNN framework developed by Berkeley Vision and Learning Centre [4], and use the reference model accompanying with it. The discrimination style of the reference model is based on the output category of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), and is different from 60 SIN concepts. For remapping ILSVRC 1000 concepts to SIN concepts, we apply SVM to the neural output of the fifth, sixth and seventh layers, which are 4096-, 4096- and 1000-dimensional vectors, respectively. In order to merge several types of learning representation effectively, we also perform 'linear fusion' which unifies SVM results of three layers with optimized weights for each SIN concept.

For the third method, we attempt to utilise the motion information in addition to the static information of the video. We prepare another deep neural network which processes only motion information in a video. The network consists four layers and is trained by unsupervised learning. It outputs 900-dimensional feature vectors which expresses a learning representation of short term motion information of a scene in a video. The feature vector of the motion network is combined with the output vector from sixth layer of CNN, and is applied to SVM.

A. Methods

1) Middle layer output for static image vectors: This method uses SVM to the neural output in a middle layer of CNN which trained with a huge amount of static images beforehand, and performs the discrimination of SIN concepts in a video. We use 'Caffe' for the framework of CNN, and BVLC

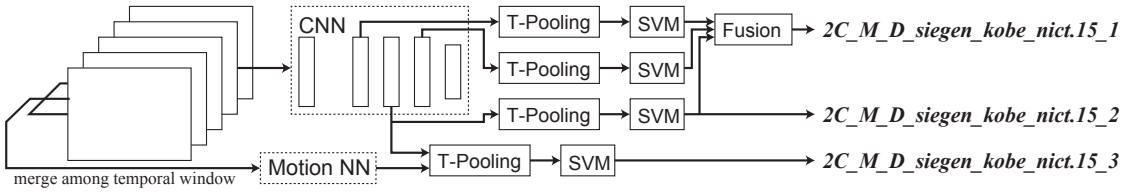

Fig. 1. An overview of our three SIN methods. 'T-Pooling' means temporally max-pooling.

reference model for the pre-trained network model [4]. The model consists of 650,000 neurons, 60,000,000 parameters, 630,000,000 connections (Fig. 2), and is trained with 310,000 iterations of 1,200,000 images in ImageNet.

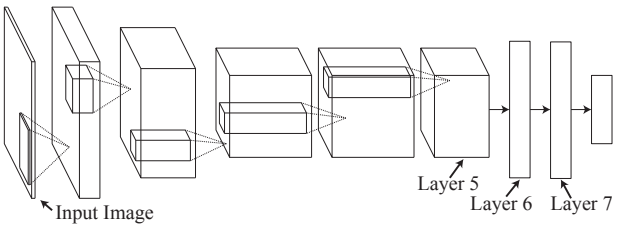

Fig. 2. An overview of the structure of the Convolutional Neural Network (CNN). Modified from Krizhevsky *et al.*, 2012 [4].

We use OpenCV (version 3.0) to extract a single frame from a video [12]. Frames are captured every one second in the video, and an additional frame is captured at the centre of the shot to stabilise the discrimination performance for shots with short duration. Each extracted frame is resized to 227x227 pixel and applied to CNN. We use the standard mean image of the reference model for the normalisation of the image. The neural outputs of the fifth, sixth and seventh layers which have 4096, 4096, 1000 dimension are retrieved for each frame. The sequence of the vectors is merged with temporal max-pooling, resulting in 4096, 4096, 1000 dimensional feature vectors for each shot. SVMs are trained for each feature vector and SIN concept with IACC.1.tv10.training and IACC.1.A-C datasets. We use two types of SVMs: one is *libsvm* [13], and the other is originally implemented SVM coded by C and MATLAB [16]. We scaled the training data from -1 to 1 for *libsvm*, but do not scaled it for the originally implemented SVM. We use the probability estimates of the SVM for concept discrimination, and select top 2000 shots for each concept.

2) Motion feature vectors: In the previous section, we only use temporally static features in a video. However, those features have no enough information to distinguish a concept which highly depends on detailed motion information. To tackle the problem, we attempt to use motion information involved in a temporal sequence of video frames. Since the motion information is processed on a separate information pathway from the static images in the real human brain [14], we prepared an deep neural network separated from the CNN which only processes the temporally static visual information. The network is constituted of many Self Organizing Maps (SOMs) [15] which involve 100 neurons, enabling to form the learning representation for complex motions in a video by unsupervised learning. The layers consist of 91, 49, 25 and 9 SOMs, and each SOM receives 9 SOMs which are located spatially neighboured in the previous layer, resulting in 16,400 neurons and 7,988,400 connections (Fig. 3). The network is implemented by a custom C code with OpenMP multi threading, and python interface scripts.

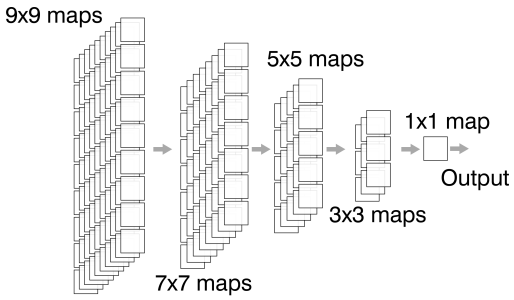

Fig. 3. An overview of the structure of deep neural network for motion detection. Each small rectangle means an SOM involving 100 neurons.

A frame captured from a video is firstly converted to grayscale, then applied to Laplacian filter. The filtered image is shrunk to 160x120 pixels, cropped 90x90 pixels on the centre region and extracted 9x9 sets of sub-region images with 8x8 pixels. For each sub-region image, temporally successive three images are combined, and form motion representing matrix which represents local motion characteristics with 8x8x3 dimension. The training has two stages: the first stage is local level training with motion representing matrix only in the first layer of the network, and the second stage is network level training using learning representation of motion obtained by the first stage. In the first stage, a single layer SOM is trained by normal SOM learning rule (traditional competitive learning rule) with 1,000,000 motion representing matrix which is randomly sampled from IACC.1.A-C for training data. As a result, motion receptive field (MRF) is organised as the weight matrix of the SOM (Fig. 4).

In the second stage, we use the weight of MRF to initial weight of the all SOMs in the first layer, and train whole DNN with SOM-based unsupervised learning. The training is performed with 5,000,000 sequence sets which have three frames and randomly sampled from IACC.1.A-C. The 900-

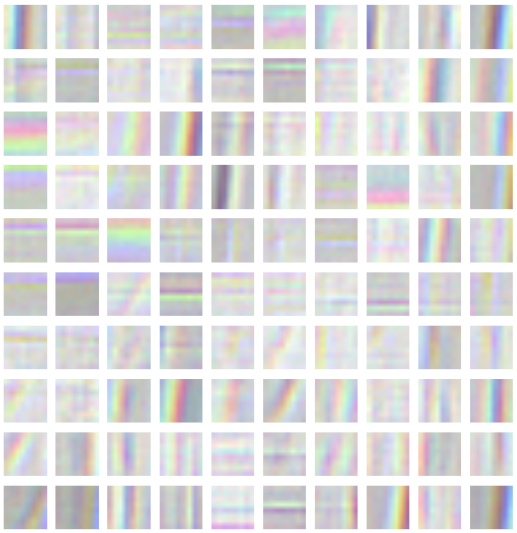

Fig. 4. Motion Receptive Field obtained by initial unsupervised learning. Red, green, and blue colour represent the receptive field of the temporally successed first, second, and third frame respectively.

dimensional vector is combined with the output of sixth layer of CNN, resulting 4996-dimensional feature vector. SVM is applied in the same way as static images.

B. Results

Using IACC.2.C video dataset, we first examine whether our basic method which uses CNN, temporal max-pooling and SVM could work for the detection of SIN concepts. Fig. 5 shows the performance comparison among the results using outputs of layer 5, 6 and 7, respectively. The result shows the optimal layer is different among respective concepts. Thus, we perform a 'linear fusion' to unify the effective information among outputs of three layers. The result of the linear fusion exhibit better discrimination performance among all concepts than the result of any other single layer. We use the result of the linear fusion as our primary result (*2C_M_D_siegen_kobe_nict.15_1*). On the other hand, we use the result which uses only the sixth layer as the baseline of our runs *2C_M_D_siegen_kobe_nict.15_2*. Although the result of the linear fusion improved the robustness of the discrimination, the improvement is not so intensive in comparison to our previous 'Fusion' study [16]. It may be due to the fact that outputs from three layers contain almost the same information because the outputs of later layer is linear transformation of the earlier layer. Thus, other types of features like motion and audio are necessary for more accurate concept detection.

Fig. 6 presents the performance comparison between our primary run *2C_M_D_siegen_kobe_nict.15_1* and other methods developed on IACC.2.C video dataset. In Fig. 6, each method is represented by a bar depicting its MAP. Actually, we have took a fatal mistake for the SIN submission, and the submitted results are totally valueless. The MAPs of mistaken runs are indicated by yellow bars, representing quite low MAPs. Our corrected primary result, which corresponds to *2C_M_D_siegen_kobe_nict.15_1*, is coloured by red. The result is better than our mistaken one. However, it indicates that there is much room for improving our SIN method.

Fig. 7 shows the effect of the usage of motion information for the detecting SIN concepts. The detection result with the motion information delivered from the separated deep neural network is coloured in blue, which corresponds our run *2C_M_D_siegen_kobe_nict.15_3*. The result without the motion information which uses the same SVM and parameters is coloured in green. Although those results indicate no significance improvement by the usage of motion information on the total result, there is a certain amount improvement in some concepts (38: Dancing, 41: Demonstration Or Protest, 95: Press Conference, 478: Traffic). A common feature of those concepts is a vast amount of motions all over the frame, meaning that our method could retrieve some but very limited motion information. Actually, the MRF which we use in the run has very limited spatial and temporal characteristics (Fig 4). More variation of those characteristics in the MRF, and related network implementation might be required for the effective usage of the motion information.

III. Multimedia Event Detection

Fig. 8 illustrates an overview of three types of MED methods that we have developed. First of all, an event is 'highly-abstracted' in the sense that various objects interact with each other in different situations. In consequence, visual appearances of videos where the event occurs have got a huge variance in the space of low-level features like colour, edge, and motion. Hence, we adopt a *concept-based approach* which uses detection results of concepts as features to detect the event [17]. Compared to the space of low-level features where each dimension just represents the physical value of a shot, in the space of concept detection results, each dimension represents the appearance of a human-perceivable meaning. Thus, the variation of videos containing the event becomes smaller and can be modelled more easily.

We detect concepts in each shot in a video where shot boundaries are detected as significant differences of colour histograms between two consecutive video frames. In Fig. 8, a shot is represented by one video frame and arranged from front to back based on its shot ID. Then, we detect 346 concepts in each shot using the method described in the previous section. In addition, since our concept-based approach defines an event based on appearances of concepts, it is important to use a rich concept vocabulary to cover a variety of events. Thus, to enlarge the concept vocabulary, we use 1000 objects (e.g., *Castle*, *Car_Wheel* and *Bee*) which are defined in ImageNet dataset and can be detected by the CNN in the previous section [7]. For simplicity, we do not distinguish these 1000 objects from the above 346 concepts, and term both of them just as 'concepts'. That is, we detect 1346 concepts in each shot. This leads to represent the shot as a 1346-dimensional vector where each dimension has a 'detection score' representing a scoring value between 0 and 1 in terms

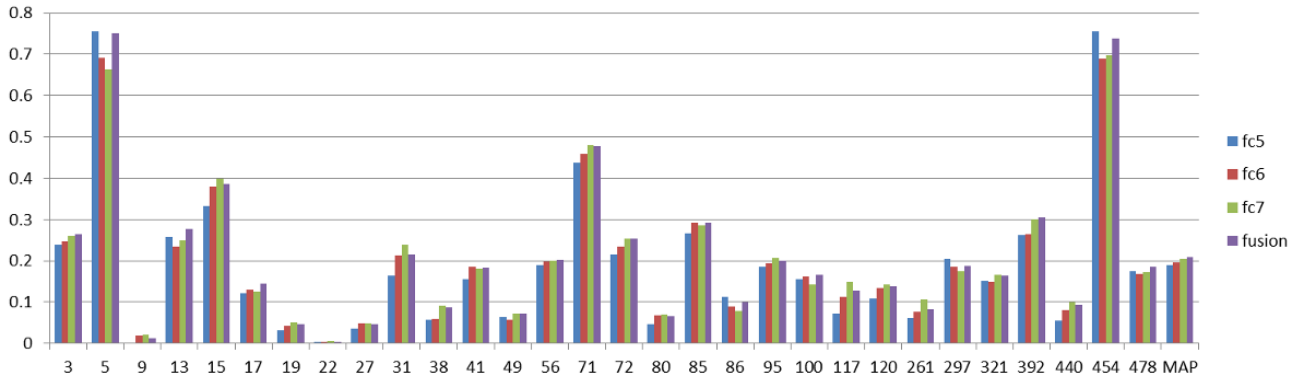

Fig. 5. Performance comparison among layer 5, 6, 7, and fusion.

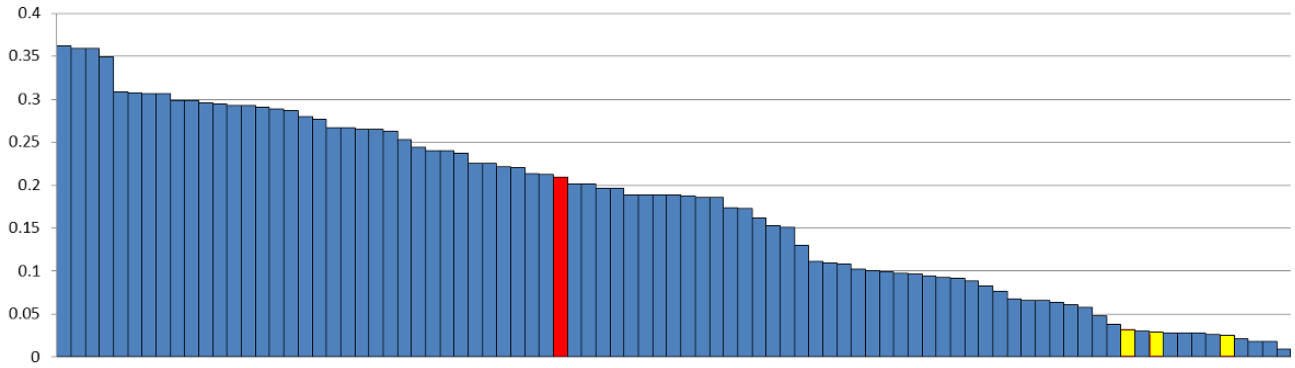

Fig. 6. Performance comparison between *2C_M_D_siegen_kobe_nict.15_1* and the other methods for SIN. The red bar shows our corrected primary result, and yellow bars show submitted our three results.

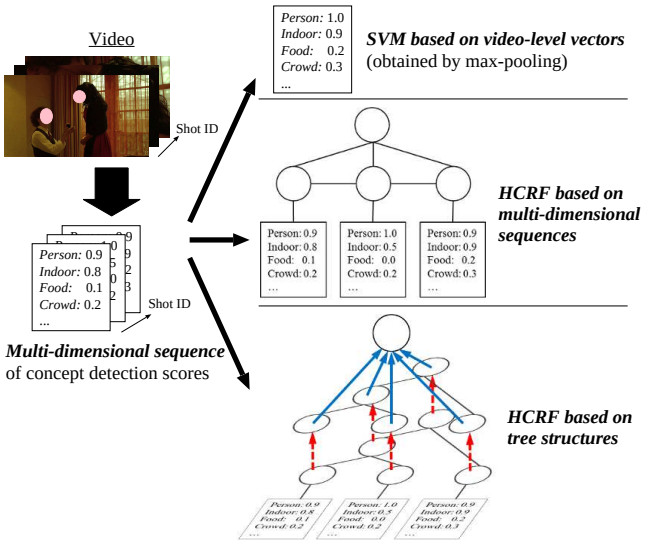

Fig. 8. An overview of our three MED methods.

of the appearance of one concept. A larger detection score indicates more likelihood that the concept appears in the shot. By sequentially aggregating such vectors of detection scores, we represent a video as a *multi-dimensional sequence* as depicted in Fig. 8. Below, based on this representation, we present our three MED methods shown in the right side of Fig. 8.

A. Methods

1) SVM based on video-level vectors: We represent each video as a 'video-level vector' by performing max-pooling on detection scores of shots. That is, each dimension in this vector represents the maximum detection score for a concept over shots in the video. Based on this vector representation, we build the following two types of SVMs using example videos as positive, and near-miss and background training videos as negative: The first is a linear SVM SVM_{linear}. Note that HCRF-based approached described below use hidden states characterised by linear combinations of concept detection scores. Hence, we use SVM_{linear} as the baseline to evaluate improvements by HCRF-based approaches which precisely handle shot sequences and their temporal structures in a video. Also, the performance of non-linear SVMs is generally higher than that of linear ones. Thus, in order to investigate their performance difference, we build SVM_{rbf} which is a non-linear SVM using RBF kernel.

2) HCRF based on multi-dimensional sequences: We present an HCRF $HCRF_{\mathrm{seq}}$ for videos represented as multi-dimensional sequences [9]. That is, $HCRF_{\mathrm{seq}}$ only targets

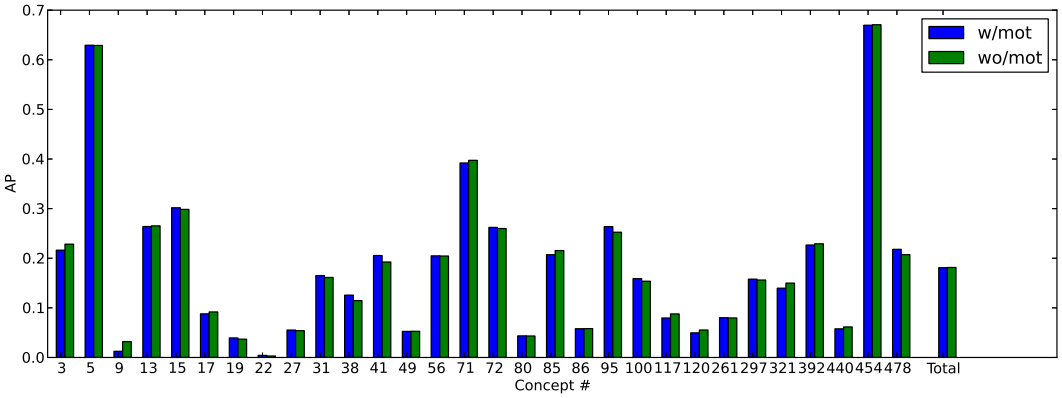

Fig. 7. Performance comparison between theresult with and without motion information supplied from the separate DNN only for motion.

temporal structures of shots and does not count the ones of shot sequences. $HCRF_{\text{seq}}$ is characterised by a set of hidden states $\mathcal{H}$. Each hidden state $h \in \mathcal{H}$ has the following three types of parameters: The first is the 'label relevance' $\theta_{\text{label}}(y, h)$ representing the relevance of h to the occurrence of an event $y = 1$ or its absence $y = 0$. The second type of parameter is the 'weight vector' $\boldsymbol{\theta}_{\text{weight}}(h)$ where each dimension indicates the weight of a concept. That is, $\boldsymbol{\theta}_{\text{weight}}(h)$ signifies characteristic concepts of h. The last type of parameter is the 'transition relevance' $\theta_{\text{trans}}(y, h, h')$ indicating the relevance of transition from h to another state h' conditioned on $y = 1$ or $y = 0$. To sum up, $\theta_{\text{label}}(y, h)$ is used to discriminate between relevant and irrelevant shots under the weakly supervised setting, while $\theta_{\text{trans}}(y, h, h')$ characterises temporal structures of shots for the event. In addition, $\boldsymbol{\theta}_{\text{weight}}(h)$ is used to check the suitability of assigning h to a shot.

We explain how $HCRF_{\text{seq}}$ estimates the occurrence of an event based on the multi-dimensional sequence of a video $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_M)^{\mathrm{T}}$. Here, the j-th shot $\boldsymbol{x}_j$ $(1 \leq j \leq M)$ is represented as a C-dimensional vector $(x_{j,1}, \ldots, x_{j,C})^{\mathrm{T}}$ (i.e., $C = 1346$), and $x_{i,c}$ $(1 \leq c \leq C)$ represents the c-th concept detection score for $\boldsymbol{x}_j$. Let $\boldsymbol{h}(\boldsymbol{x}) = (h(\boldsymbol{x}_1), \cdots, h(\boldsymbol{x}_M))^{\mathrm{T}}$ be a sequence of hidden states assigned to M shots in $\boldsymbol{x}$. That is, $h(\boldsymbol{x}_j)$ denotes the hidden state assigned to $\boldsymbol{x}_j$. Assuming the event occurrence $(y = 1)$ or absence $(y = 0)$, this assignment is evaluated as follows:

$$\Psi(y, \boldsymbol{h}(\boldsymbol{x}), \boldsymbol{x}; \boldsymbol{\theta}) = \sum_{j=1}^{M} \theta_{\text{label}}(y, h(\boldsymbol{x}_j)) + \sum_{j=1}^{NM} \boldsymbol{x}_j \cdot \boldsymbol{\theta}_{\text{weight}}(h(\boldsymbol{x}_j)) + \sum_{j=2}^{M} \theta_{\text{trans}}(y, h(\boldsymbol{x}_{j-1}), h(\boldsymbol{x}_j)), \quad (1)$$

where $\boldsymbol{\theta}$ is the whole set of parameters of all hidden states. The first term is the sum of label relevances of hidden states assigned to all shots, and represents the overall relevance of these states to $y = 1$ or $y = 0$. The second term accumulates the product between $\boldsymbol{x}_j$ and $\boldsymbol{\theta}_{\text{weight}}(h(\boldsymbol{x}_j))$, and indicates the overall degree of how much shots match with assigned hidden states. The last term sums transition relevances and represents how relevant transitions of hidden states in $\boldsymbol{h}(\boldsymbol{x})$ are to $y = 1$ or $y = 0$. Assuming that $\boldsymbol{h}(\boldsymbol{x})$ is appropriately selected for each training video, $\boldsymbol{\theta}$ should be optimised so that $\Psi(y = 1, \boldsymbol{h}(\boldsymbol{x}), \boldsymbol{x}; \boldsymbol{\theta})$ is large for positive videos, while for negative ones $\Psi(y = 0, \boldsymbol{h}(\boldsymbol{x}), \boldsymbol{x}; \boldsymbol{\theta})$ is large.

The optimisation of $\boldsymbol{\theta}$ is based on the following conditional probability of y given $\boldsymbol{x}$ [11]:

$$P(y|\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{\forall \boldsymbol{h}(\boldsymbol{x}) \in \mathcal{H}} P(y, \boldsymbol{h}(\boldsymbol{x})|\boldsymbol{x}, \boldsymbol{\theta}) \quad (2)$$

$$= \frac{\sum_{\forall \boldsymbol{h}(\boldsymbol{x}) \in \mathcal{H}} e^{\Psi(y, \boldsymbol{h}(\boldsymbol{x}), \boldsymbol{x}; \boldsymbol{\theta})}}{\sum_{\forall y' \in \mathcal{Y}; \forall \boldsymbol{h}(\boldsymbol{x}) \in \mathcal{H}} e^{\Psi(y', \boldsymbol{h}(\boldsymbol{x}), \boldsymbol{x}; \boldsymbol{\theta})}}. \quad (3)$$

Equation (2) indicates that $\boldsymbol{h}(\boldsymbol{x})$ is marginalised out by taking the sum of $P(y, \boldsymbol{h}(\boldsymbol{x})|\boldsymbol{x}, \boldsymbol{\theta})$s over all possible instances of $\boldsymbol{h}(\boldsymbol{x})$ (i.e., all possible assignments of hidden states to $\boldsymbol{x}$). Equation (2) is further transformed into Equation (3), where the numerator with the fixed y is normalised by the denominator taking the sum of numerators with all $y' \in \{0, 1\}$. Thus, Equation (3) can be considered as a conditional probability.

Suppose N training videos where the i-th training video $\boldsymbol{x}^{(i)}$ is annotated with the event label $y^{(i)} = 1$ if it is positive, otherwise $y^{(i)} = 0$. We estimate $\boldsymbol{\theta}$ which maximises the following log-likelihood based on conditional probabilities for $\boldsymbol{x}^{(i)}$ and $y^{(j)}$:

$$L(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log P(y^{(i)}|\boldsymbol{x}^{(i)}, \boldsymbol{\theta}) - \frac{||\boldsymbol{\theta}||^2}{2\sigma^2}, \quad (4)$$

where the second term is the L2 regularisation term and useful for preventing $\boldsymbol{\theta}$ from being overfit to training videos. The optimal $\boldsymbol{\theta}^*$ is estimated by a gradient ascent method based on the derivative of Equation (4) in terms of each parameter in $\boldsymbol{\theta}$ [11]. Finally, using $\boldsymbol{\theta}^*$, the relevance score of a test video $\boldsymbol{x}$ is computed as the conditional probability of $y = 1$ for $\boldsymbol{x}$, that is, $P(y = 1|\boldsymbol{x}, \boldsymbol{\theta}^*)$ based on Equation (2).

3) HCRF based on Tree Structures: Compared to $HCRF_{\mathrm{seq}}$ based on multi-dimensional sequences, this HCRF $HCRF_{\mathrm{tree}}$ uses tree structures of videos in order to flexibly examine temporal structures in different abstraction levels. The tree structure of a video is extracted by time-constrained shot clustering which is an extended version of bottom-up clustering [10]. Here, shots in the video are iteratively merged into nodes (i.e., shot sequences) based on similarities in terms of their vectors of concept detection score and temporal distances. Specifically, we represent the tree structure of a video $\boldsymbol{x}$ as a set of nodes $\mathcal{N}$. Our time-constrained shot clustering method initialises $\mathcal{N}$ as empty and gradually enlarges it by adding nodes that will be extracted at the subsequent iterations. Since the tree structure can be easily constructed by checking inclusion relations among nodes in $\mathcal{N}$, we omit these relations. Our method starts with assigning each shot $\boldsymbol{x}_i$ to a node n_i and adding it to $\mathcal{N}$. Using this $\mathcal{N}$, we initialise a similarity matrix $\boldsymbol{S}$ where each element represents the similarity between two nodes. Afterwards, the most similar pair of nodes $\hat{n}_1$ and $\hat{n}_2$ are selected based on $\boldsymbol{S}$, and merged into a new node $\hat{n}$. According to this, $\boldsymbol{S}$ is updated by removing elements for $\hat{n}_1$ and $\hat{n}_2$ and adding those for $\hat{n}$. The selection and merge of the most similar nodes and the update of $\boldsymbol{S}$ are iterated until the size of $\boldsymbol{S}$ becomes 1×1. Below, we closely describe the above-mentioned processes.

For the initialisation of $\boldsymbol{S}$, it is only required to compute the similarity between two shots $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ because each node consists only of a single shot. We define this similarity $Sim(\boldsymbol{x}_1, \boldsymbol{x}_2)$ by considering their visual similarity $Sim_{vis}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ and temporal distance $TP(\boldsymbol{x}_1, \boldsymbol{x}_2)$:

$$Sim(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{1}{TP(\boldsymbol{x}_1, \boldsymbol{x}_2)} * Sim_{vis}(\boldsymbol{x}_1, \boldsymbol{x}_2), \qquad (5)$$

where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are represented by vectors of detection scores for 1346 concepts, and $Sim_{vis}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is computed as their cosine similarity. For $TP(\boldsymbol{x}_1, \boldsymbol{x}_2)$, we consider the *temporal locality* of contents in a video [10]. Since contents in the video are sequentially developed shot by shot, two temporally distant shots do not have strong semantic relation even if they are visually very similar. In other words, many other contents should be presented between these shots, so viewers cannot consider them as semantically-related. The temporal locality is used so that $Sim_{vis}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is reduced based on $TP(\boldsymbol{x}_1, \boldsymbol{x}_2)$ representing the temporal distance between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$.

Using $\boldsymbol{S}$ initialised above, we select the most similar pair of nodes $\hat{n}_1$ and $\hat{n}_2$ and merge them into a new node $\hat{n}$. At this point, $\hat{n}_1$ and $\hat{n}_2$ may not be temporally continuous. That is, some nodes (i.e., shot sequences) may exist between $\hat{n}_1$ and $\hat{n}_2$. Considering the sequential development of contents in a video, it is natural to merge all of $\hat{n}_1$, $\hat{n}_2$ and nodes between them into $\hat{n}$. Then, we update $\boldsymbol{S}$ by removing elements for these merged nodes, and adding elements for $\hat{n}$ where its similarity to each non-merged node n is the maximum among similarities between n and the merged nodes. This update assumes that a viewer tends to connect two shot sequences if he/she finds that a shot in one sequence is related to a shot in the other. In other words, the viewer does not remember several shots in these shot sequences to deduce their connection. Also, the update of $\boldsymbol{S}$ is computationally efficient because we can re-use similarities computed at the initialisation. This means that the similarity between two nodes is the maximum among similarities computed for shots included in those nodes. Finally, the tree structure of a video is constructed by checking merge records of nodes in $\mathcal{N}$.

The application of an HCRF to tree structures only requires two modifications on Equation (1). The first is that when assigning a node $n \in \mathcal{N}$ to a hidden state $h \in \mathcal{H}$, we need its vector representation to compute the suitability of assigning n to h based on $\boldsymbol{\theta}_{\mathrm{weight}}(h)$. We define the vector representation by performing max-pooling over shots contained in n. The second modification is the computation of transition relevances. For this, we can translate the transition from $h(\boldsymbol{x}_{j-1})$ to $h(\boldsymbol{x}_j)$ in Equation (1) into the transition from the hidden state for a parent node to the one for a child node. The HCRF can be trained and tested without any other modification.

B. Results

Using MED14-Test video dataset [18], we first examine whether HCRF-based approaches ($HCRF_{\mathrm{seq}}$ and $HCRF_{\mathrm{tree}}$) can appropriately handle the weakly supervised setting and temporal structures, compared to SVM-based approaches based on video-level vectors (SVM_{linear} and SVM_{rbf}). For all of these four methods, each event is detected using 100 example videos as positive, and near-miss and background training videos as negative. Fig 9 shows their performance comparison. For each event, the first, second, third and fourth bars from the left indicate APs of SVM_{linear}, $HCRF_{\mathrm{seq}}$, $HCRF_{\mathrm{tree}}$ and SVM_{rbf}, respectively. As can be seen from Fig 9, $HCRF_{\mathrm{seq}}$ and $HCRF_{\mathrm{tree}}$ outperforms the baseline SVM_{linear}. This means that HCRF-based approaches which precisely model shots (or shot sequences) in a video appropriately discriminate between relevant and irrelevant shots to an event and exploit their temporal structures. In particular, $HCRF_{\mathrm{tree}}$ is superior to $HCRF_{\mathrm{seq}}$, indicating the effectiveness of tree structures for flexibly examining shot sequences with different time lengths. However, HCRF-based approaches are outperformed by SVM_{rbf}. This suggests the insufficient discrimination power of hidden states which linearly combine concept detection scores, compared to non-linear functions. Thus, we plan to enhance hidden states by adopting non-linear functions.

Fig. 10 shows the performance comparison between our primary run *SiegenKobeNict_MED15_MED15EvalSub_PS_10Ex_SML_p-svm10_1* and the other methods developed on MED15EvalSub video dataset. Our primary run is implemented as SVM_{rbf} which uses 10 example videos as positive, and near-miss and background training videos as negative. In Fig. 10, each method is represented by a bar depicting its MAP. The MAP of our method is coloured by red and indicated by the arrow. In addition, the first, third, fourth and fifth methods coloured by green use much larger computational resources (about 1000 CPU cores and 10000 GPU cores) than the other

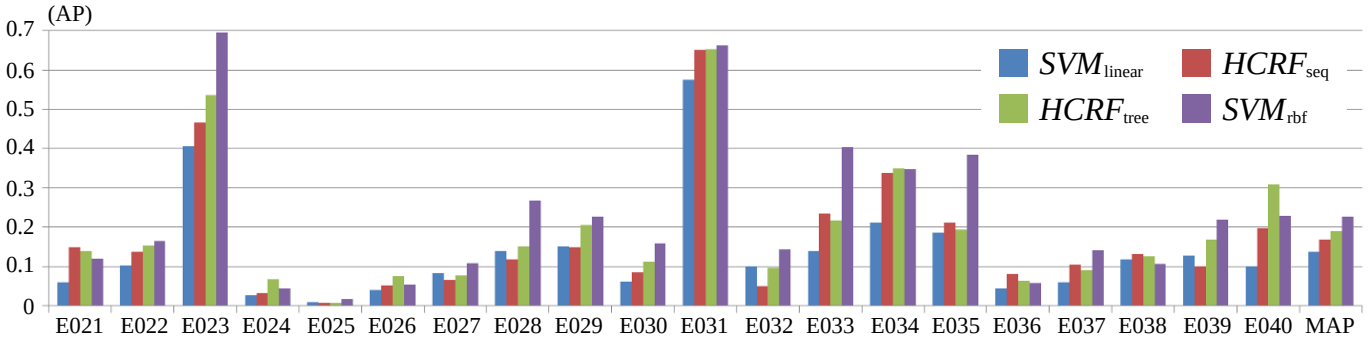

Fig. 9. Performance comparison among SVM_{linear}, $HCRF_{\mathrm{seq}}$, $HCRF_{\mathrm{tree}}$ and SVM_{rbf} on MED14-Test video datasets.

methods (using about 100 CPU cores and 1000 GPU cores). Fig. 10 indicates that there is much room for improving our MED method.

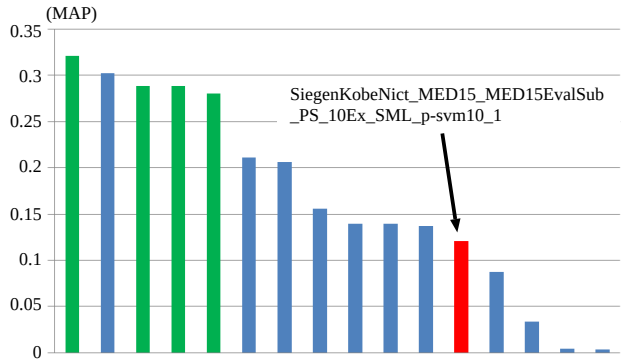

Fig. 10. Performance comparison between *SiegenKobeNict_MED15_MED15EvalSub_PS_10Ex_SML_p-svm10_1* and the other methods on MED15EvalSub video dataset.

Table I presents the performance comparison among our submitted runs. The correspondence between these runs and methods explained above is as follows:

1. *p-svm10* submitted as *SiegenKobeNict_MED15_MED15EvalSub_PS_10Ex_SML_p-svm10_1* is implemented as SVM_{rbf} using 10 example videos as positive, and near-miss and background training videos as negative.
2. *c-hcrfseq100* submitted as *SiegenKobeNict_MED15_MED15EvalSub_PS_100Ex_SML_c-svm100_1* is imple-mented as $HCRF_{\mathrm{seq}}$ using 100 example videos as positive, and near-miss and background training videos as negative (all the following runs use the same training videos).
3. *c-hcrftree100* submitted as *SiegenKobeNict_MED15_MED15EvalSub_PS_100Ex_SML_c-hcrftree100_1* is imple-mented as $HCRF_{\mathrm{tree}}$.
4. *c-svm100* submitted as *SiegenKobeNict_MED15_MED15EvalSub_PS_100Ex_SML_c-svm100_1* is imple-mented as SVM_{rbf}.
5. *c-svm-hcrftree100* submitted as *SiegenKobeNict_MED15_MED15EvalSub_PS_100Ex_SML_c-svm-hcrftree100_1* simply takes averages of outputs by *c-svm100* and *c-hcrftree100*.

TABLE I
PERFORMANCE COMPARISON AMONG OUR SUBMITTED RESULTS ON MED15EVALSUB VIDEO DATASET.

p-svm10	c-hcrfseq100	c-hcrftree100	c-svm100	c-svm-hcrftree100
0.110	0.157	0.155	0.217	0.206

In Table I, *c-hcrftree100* is slightly outperformed by *c-hcrfseq100*. But, considering the advantage of the former over the latter in Fig. 9, it is still valid that tree structures are useful for capturing temporal structures by flexibly examining shot sequences with different time lengths. Also, similar to Fig. 9, the performance of *c-svm100* (i.e., non-linear SVM) is the best. This confirms the necessity of enhancing hidden states in HCRFs with non-linear functions.

IV. CONCLUSION AND FUTURE WORK

In this paper, we presented methods that we developed for the SIN and MED tasks. For the SIN task, we apply CNN to a video discrimination task with temporal max-pooling and linear fusion among information from three CNN layers. Temporal max-pooling uses temporal sparseness and independence of the feature vector, and effectively extracts the concept in a video, resulting good discrimination results. Moreover, the linear fusion successfully unifies image information which distributes among layers with various learning representation, and enhances the discrimination results sufficiently, However, since the unified information is originated from the same feed-forward network, the improvements is moderate. To address the problem, we also try to apply a motion feature vector from the motion specific deep neural network, combined with the outputs of CNN. The network output trained by unsupervised learning improved the discrimination results in some types concept, but no significant difference among overall concepts. To improve the effectiveness of the motion information, we will treat a network with more diverted types of MRF and/or semi-supervised learning related to moving objects.

For the MED task, we addressed the weakly supervised setting and the extraction of temporal structures. To manage these, we developed an MED method which builds an HCRF on videos represented as tree structures. The tree structure of a video facilitates flexibly examining shot sequences in

different abstraction levels, and hidden states in the HCRF work as the intermediate layer to discriminate between relevant and irrelevant shot sequences to an event. Our experimental results showed that although the effectiveness of our HCRF-based method has validated in comparison to the baseline method (linear SVM), its performance based on hidden states characterised by linear functions is inferior to the one of non-linear SVM. To overcome this, we will develop an HCRF where each hidden state is characterised by a non-linear function based on a Multi Layer Perceptron (MLP) [19].

REFERENCES

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

[2] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, G. Quenot, and R. Ordelman, "Trecvid 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proc. of TRECVID 2015*, 2015.

[3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[4] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. of NIPS 2012*, 2012, pp. 1106–1114.

[5] H. Lee, P. Pham, Y. Largman, and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Proc. of NIPS 2009*, 2009, pp. 1096–1104.

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

[7] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. of MM 2014*, 2014, pp. 675–678.

[8] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE Interna tional Conference on*. IEEE, 2013, pp. 8595–8598.

[9] K. Shirahama, M. Grzegorzek, and K. Uehara, "Weakly supervised detection of video events using hidden conditional random fields," *International Journal of Multimedia Information Retrieval*, vol. 4, no. 1, pp. 17–32, 2015.

[10] M. Yeung and B.-L. Yeo, "Segmentation of video by clustering and graph analysis," *Computer Vision and Image Understanding*, vol. 71, no. 1, pp. 94–109, 1998.

[11] A. Quattoni, S. Wang, L. Morency, M. Collins, and T. Darrell, "Hidden conditional random fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1848–1852, 2007.

[12] G. Bradski, *Dr. Dobb's Journal of Software Tools*, 2000.

[13] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[14] J. J. Nassi and E. M. Callaway, "Parallel processing strategies of the primate visual system," *Nature Reviews Neuroscience*, vol. 10, no. 5, pp. 360–372, 2009.

[15] T. Kohonen, *Self-Organization and Associative Memory*, 3rd ed. Berlin: Springer-Verlag, 1989.

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

[17] C. G. M. Snoek and M. Worring, "Concept-based video retrieval," *Foundations and Trends in Information Retrieval*, vol. 2, no. 4, pp. 215–322, 2009.

[18] S. Strassel, A. Morris, J. Fiscus, C. Caruso, H. Lee, P. Over, J. Fiumara, B. Shaw, B. Antonishek, and M. Michel, "Creating havic: Heterogeneous audio visual internet collection," in *Proc. of LREC 2012*, 2012.

[19] Y. Fujii, K. Yamamoto, and S. Nakagawa, "Hidden conditional neural fields for continuous phoneme speech recognition," *IEICE TRANSACTIONS on Information and Systems*, vol. E95, no. 8, pp. 2094–2104, 2012.