Kobe University, NICT and University of Siegen on the TRECVID 2016 AVS Task

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

***Abstract*—We submit the following three runs for the TRECVID 2016 AVS task.**

1) *kobe_nict_siegen_D_M_1:* **This combines the results of several small-scale multi-layer neural networks, called micro neural networks (microNNs). Although a large number of concepts are necessary for treating various queries, the computational cost of preparing detectors for all these concepts is huge. Thus we use microNNs as lightweight concept detectors. The input of each microNN is a vector of outputs extracted by a pre-trained convolutional neural network (CNN), which is fine-tuned to a target concept using ImageNet and/or IACC video data. Fine-tuning is carried out using imbalanced numbers of positive and negative data.**
2) *kobe_nict_siegen_D_M_2:* **This is identical to kobe_nict_siegen_D_M_1 except that fine-tuning is performed using the same numbers of positive and negative data.**
3) *kobe_nict_siegen_D_M_3:* **This combines the results of several long short-term memory (LSTM) networks for different concepts. The input of each LSTM is a vector of outputs extracted by a pre-trained CNN, which is fine-tuned to a target concept using ImageNet and/or IACC video data. The numbers of positive and negative data are imbalanced for fine-tuning.**

The results of these runs validate the efficiency of training microNNs for various concepts and their usefulness for achieving reasonable retrieval performances. Furthermore, LSTMs significantly improve the retrieval results for some queries.

I. Introduction

The TREC Video Retrieval Evaluation (TRECVID) is an annual worldwide competition where large-scale benchmark video data are used to evaluate methods developed around the world [1]. At TRECVID 2016 [2], we participated in the ad-hoc video search (AVS) task, where an end-user searches shots containing people, objects, activities, locations and so on, as well as combinations of these. This paper presents our methods developed for the AVS task.

In recent years, deep learning, in particular convolutional neural networks (CNNs), have often been used for video analysis. A CNN is a type of forward propagation neural network that achieves excellent performance in many tasks and has attracted much research attention. Also, a CNN can be used effectively as a feature extractor, where a feature vector for an image is formed by the output values from a hidden layer in the CNN. In this feature extraction, we need to consider which layer should be used. A CNN constructs a feature hierarchy where features from one layer are recursively abstracted into higher-level features in the next layer. The features from lower layers represent primitive but generic visual characteristics, while features from upper layers describe content that is semantic but specialized to the target domain. We have showed that using the output of the seventh layer is the most effective choice for the IACC dataset in TRECVID 2015 [3].

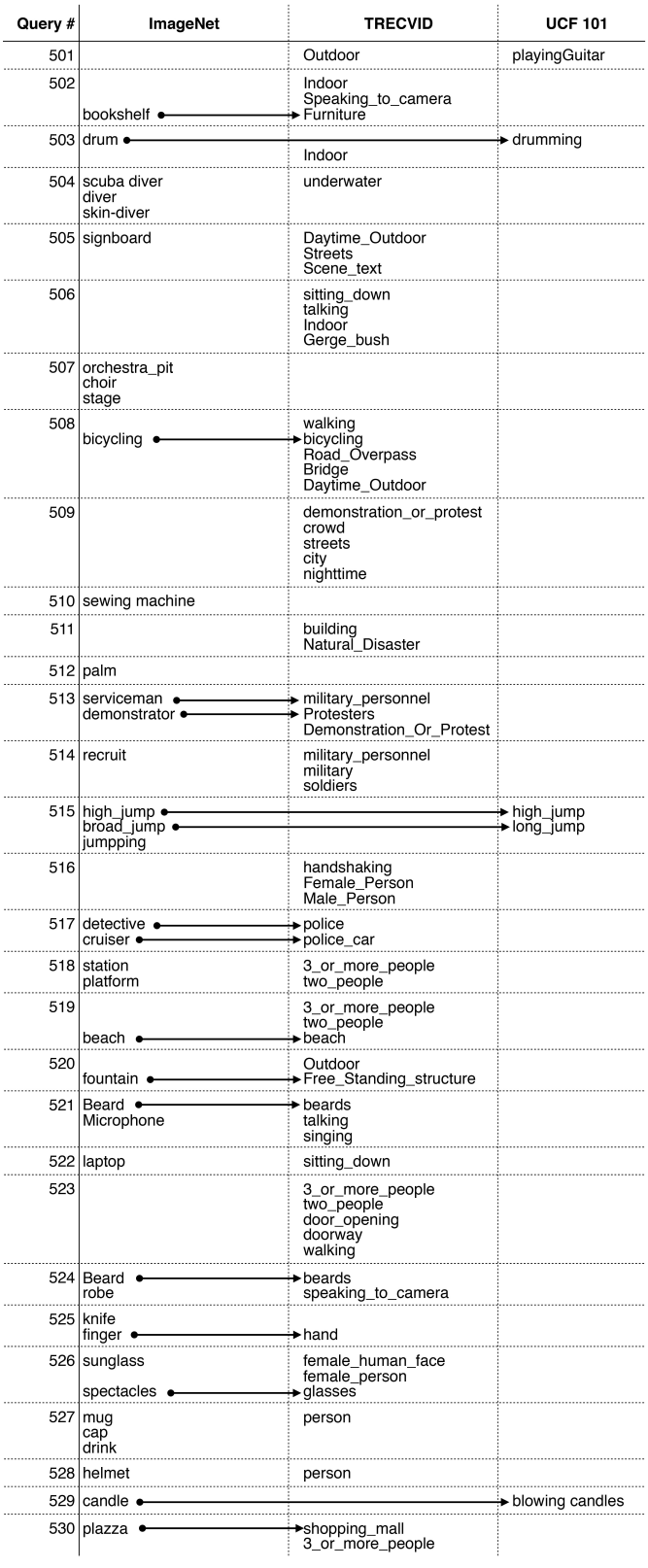
Using the features obtained from a CNN, we can easily implement transfer learning and use these features to build classifiers that work in different domains. However, this approach has an expensive computational cost when building several classifiers. In the AVS task, an accurate search for one query requires combining detection results for several related concepts. Thus, to respond to various queries, it is necessary to build a large number of classifiers that detect diverse concepts. Therefore, we propose learning small networks that use features obtained from the learned network (CNN) as the input for efficient transfer learning. Each small-scale network is referred to as a micro neural network (microNN). A microNN is small in terms of the number of layers and nodes, so it can be trained quickly. In the following, we present retrieval methods using microNNs and demonstrate their usefulness in the AVS task.

II. The proposed method

Given an ad-hoc query, we begin by manually selecting relevant concepts for each shot. A simple rule is used so that this process can be easily automated in the future. A list of the concepts selected for each query in this study is shown in Fig. 1. An arrow indicates that the model on the left-hand side is transferred into the model on the right-hand side.

Fig. 2 shows an overview of three methods that are used to construct microNNs for the concepts in Fig. 1. All of them use the outputs of a hidden layer in a CNN.

First, the feature extraction process uses the model learned by VGGNet [4], which achieved the second-highest performance at ILSVRC (ImageNet Large Scale Visual Recognition Competition) 2014. VGGNet is a CNN with a very deep architecture consisting of 16 or 19 layers and with very



Query #	ImageNet	TRECVID	UCF 101
501		Outdoor	playingGuitar
502	bookshelf	Indoor Speaking_to_camera Furniture	
503	drum	Indoor	drumming
504	scuba diver diver skin-diver	underwater	
505	signboard	Daytime_Outdoor Streets Scene_text	
506		sitting_down talking Indoor Gerge_bush	
507	orchestra_pit choir stage		
508	bicycling	walking bicycling Road_Overpass Bridge Daytime_Outdoor	
509		demonstration_or_protest crowd streets city nighttime	
510	sewing machine		
511		building Natural_Disaster	
512	palm		
513	serviceman demonstrator	military_personnel Protesters Demonstration_Or_Protest	
514	recruit	military_personnel military soldiers	
515	high_jump broad_jump jumpping		high_jump long_jump
516		handshaking Female_Person Male_Person	
517	detective cruiser	police police_car	
518	station platform	3_or_more_people two_people	
519	beach	3_or_more_people two_people beach	
520	fountain	Outdoor Free_Standing_structure	
521	Beard Microphone	beards talking singing	
522	laptop	sitting_down	
523		3_or_more_people two_people door_opening doorway walking	
524	Beard robe	beards speaking_to_camera	
525	knife finger	hand	
526	sunglass spectacles	female_human_face female_person glasses	
527	mug cap drink	person	
528	helmet	person	
529	candle		blowing candles
530	plazza	shopping_mall 3_or_more_people	

Fig. 1. A list of concepts selected for each query.

small receptive fields (3x3). We use VGGNet with 16 layers. However, the outputs of VGGNet are specific to concepts defined in ImageNet, and do not match the concepts for the AVS task. Thus, we use a 4096-dimensional feature vector that consists of neuron outputs from the second fully-connected layer "fc7" of VGGNet. Based on this, two approaches are used for building microNNs. The difference between these approaches is the balance between the number of positive and negative examples. The last approach uses a long short-term memory (LSTM) network on top of the microNNs in the previous two approaches to integrate the temporal information in a video.

In recent years, many frameworks for deep learning, such as Caffe [5], Chainer [6], and Tensor Flow [7], have become available. In this study, microNNs are constructed using Chainer, which is a neural network framework developed by Preferred Networks. Chainer supports various network architectures, including feed-forward nets, convnets and recurrent nets, and is flexible enough to build networks for different purposes. It also supports CUDA computation and only requires few lines of code to leverage a graphics processing unit (GPU).

A. Feature extraction using CNN

In general, it is not realistic to learn a deep neural network for video recognition from scratch, because of the computational cost of training using a large amount of data. Pre-trained networks, such as AlexNet [8], VGGNet and GoogLeNet [9], are usually transferred to a classifier suitable for a target problem. In our case, we convert the version of VGGNet released in the model zoo of Caffe so that it can be used in Chainer.

However, VGGNet is specific to concepts defined in ILSVRC 2014, so it cannot be used for accurate detection of concepts that are appropriate for the AVS task. To overcome this, we focus on a phenomenon called representation learning [10], where lower layers in a deep neural network characterize visual features that can be used universally for various images or videos. Based on this, we extract features from a middle-level layer of VGGNet trained on natural images, and use those features to build microNNs for concepts that are suitable for the AVS task. In this study, we perform experiments using the output of the second fully connected layer "fc7" in VGGNet. This feature extraction is applied to the images in the ImageNet dataset, the videos in the TRECVID dataset, and the videos in the UCF101 dataset [11]. For the video datasets, we use VGGNet on one out of every 30 frames in each shot. An overall feature is extracted by aggregating the features extracted from the frames based on max-pooling.

B. Micro Neural Networks structure

In *kobe_nict_siegen_D_M_1* and *kobe_nict_siegen_D_M_2*, a microNN is constructed for each concept based on features extracted by VGGNet. This microNN is a binary classifier that outputs two values for the presence and absence of the concept. The microNN consists of the input, hidden and output layers,

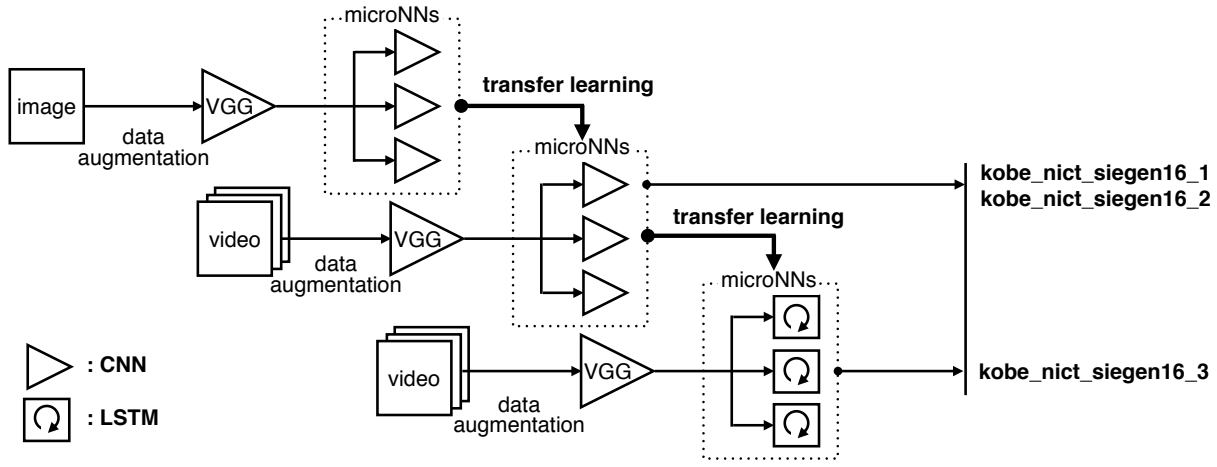


Fig. 2. An overview of our three AVS methods.

which are fully-connected and contain 4096, 32 and two nodes, respectively. This small-scale structure allows the microNN to be efficiently trained. During learning, we apply Dropout [12] to improve learning by ignoring randomly selected nodes. Dropout can avoid overfitting by reducing the degrees of freedom and raising the generalization performance.

Finally, an important issue in training a microNN for a concept is the balance between the number of positive examples and the number of negative examples. The former is usually much smaller than the latter, because any kind of image or video that does not contain the concept is negative. However, using too many negative examples may bias the microNN to preferentially produce high output values for the absence of the concept. To check this, the microNNs in *kobe_nict_siegen_D_M_1* are trained in an imbalanced setting where positive examples are significantly outnumbered by negative examples. In particular, we use all the available positive examples, and then randomly select negative examples until the total number of training examples is 30000. On the other hand, the microNNs in *kobe_nict_siegen_D_M_2* are trained by balancing the numbers of positive and negative examples. Specifically, the numbers of positive examples and of negative examples are both 15000.

C. Learning MicroNNs

Under the settings described above, we perform gradual transfer learning for each concept using the following two steps.

(i) We learn a microNN using images from the ImageNet dataset.

(ii) We refine the microNN using videos from the TRECVID dataset or the UCF101 dataset by regarding the weight parameters learned in the first step as initial values. If the annotation of the concept is available only in the image dataset (ImageNet) or the video dataset (TRECVID or UCF101), the microNN is trained only using that dataset.

In general, CNN learning is strongly affected by the initial values. Especially in the case of little training data, it is important to obtain suitable initial values to prevent overfitting. Therefore, compared with learning a microNN from full scratch, we often obtain better results by using parameters that have been optimized on images as initial values.

In addition, only a few minutes are required to learn a microNN for a concept. This is much faster than learning a support vector machine (SVM) in the Semantic INdexing (SIN) task from TRECVID 2015.

D. Long Short Term Memory

In *kobe_nict_siegen_D_M_3*, an LSTM [13] [14] [15] was used to aggregate the outputs of the microNNs over multiple video frames. The LSTM is a kind of recurrent neural network (RNN) introduced by Hochreiter and Schmidhuber (1997). An LSTM replaces units in a hidden layer of an RNN with LSTM blocks that individually consist of memory functionality and three gates (input, output and forget). With this architecture, an LSTM can maintain long-term dependencies that cannot be captured by an RNN. We consider microNNs for different concepts that produce output values for multiple frames in a shot. Because max-pooling over these output values causes a significant loss of temporal information, we aim to use an LSTM to capture temporal characteristics. (This is designed to avoid the long-term dependence problem and to store the information for a long period of time. Unlike max-pooling, the feature vector obtained by an LSTM reflects the temporal characteristics of a shot.) In the present study, LSTM-based microNNs are trained for 14 concepts for which the temporal relationships among video frames are important (see the bold-font concepts in Fig. 1). Also, each LSTM-based microNN is trained using an imbalanced numbers of positive and negative examples as in *kobe_nict_siegen_D_M_1*.

E. Shot Retrieval based on Selected Concepts

Assume that the concepts related to a given query are selected based on Fig. 1. To balance the output values produced

Fig. 3. A comparison between the retrieval results using the summation of microNN scores and those using multiplication for the query 502.

by microNNs for different concepts, we normalize the output values for each concept so that the maximum and minimum are 1 and -1, respectively. For each shot, we calculate the sum of the output values of the microNNs for the selected concepts to use as the overall score representing the appropriateness of the shot for the query.

In preliminary experiments, we tested how summation and multiplication combine the output values for selected concepts. We use summation because it outperformed multiplication on the data tested. For the query 502, Fig. 3 shows a comparison between the retrieval results using multiplication (upper row) and those using summation (bottom row). Although these results are similar in terms of the top-ranked shots, they are different for lower-ranked shots. In particular, the 50th shot retrieved using multiplication shows a "woman" which is irrelevant to the query 502, while the 50th shot retrieved using summation is still relevant to the query. One reason for this is that multiplication is sensitive to errors in concept detection. More specifically, when using multiplication, the overall score of a shot becomes very small even if only one concept related to the query is not detected (i.e., the corresponding microNN outputs a very low value). Compared to multiplication, summation is more tolerant to errors in concept detection.

III. Evaluation Experiment Results

Fig. 4 shows retrieval results for *kobe_nict_siegen_D_M_1* and *kobe_nict_siegen_D_M_3*. Each row shows the results for one query by displaying key frames of the shots ranked in the first, second, third, 50th, 100th, 500th and 1000th positions. Each key frame is selected as the middle frame in a shot. The results for queries 502, 503, 525 and 529 are obtained using *kobe_nict_siegen_D_M_1*. The first two queries show the effectiveness of transferring microNNs from the image to the video domain, while the last two queries indicate the insufficiency of microNNs that are trained only in the image domain. The result for the query 509 is obtained using *kobe_nict_siegen_D_M_3*.

We demonstrate the utility of microNNs for the AVS task using IACC.3.C. Fig. 5 shows a performance comparison between *kobe_nict_siegen_D_M_1*, *kobe_nict_siegen_D_M_2* and *kobe_nict_siegen_D_M_3* on each of the 30 queries. This figure indicates that, for most of the queries, using imbalanced training examples leads to a higher average precision than using balanced training examples. This implies that, rather than balancing the numbers of positive and negative examples, it is more important to use numerous negative examples to accurately determine the boundary between the presence and the absence of a concept.

Fig. 6 presents a comparison between our methods and other methods developed for the manually-assisted category in the AVS task. Fig. 7 shows a comparison between our methods and all other methods developed for the AVS task. In both figures, the mean average precision (MAP) of each method is represented by a bar. The MAPs of our methods are yellow.

As can be seen in Figs. 6 and 7, our method using LSTM achieves the best accuracy of 0.047. In particular, the MAP for query 509 using LSTM is more than three times higher than the MAP not using LSTM. This means that LSTM can successfully capture temporal characteristics for this query. On the other hand, using LSTM degrades the performance for some queries. One main reason is that our current method has low sampling and only considers a small number of frames in each shot. These frames are clearly insufficient for appropriately capturing temporal characteristics in the shot.

IV. Conclusion and Future Work

In this paper, we have introduced AVS methods that use microNNs to detect concepts related to a query. A microNN has a simple small-scale structure compared to VGGNet, so it can be efficiently learned for the large number of concepts required for the AVS task.

For transfer learning based on a pre-trained CNN, an SVM is often used as a classifier using features obtained from the CNN. In comparison, our proposed micron classifier can be trained more efficiently. In addition, it can be incrementally refined using different training data. For example, the microNNs in this paper are first trained using the ImageNet dataset, and are then refined using the TRECVID and UCF 101 video datasets. Although we manually extract concepts from the query, automatic selection is easy because the rule used in manual selection is very simple.

Our current method only samples one frame from every 30 frames in a shot. The experiment shows that this is clearly insufficient for building an LSTM to capture temporal

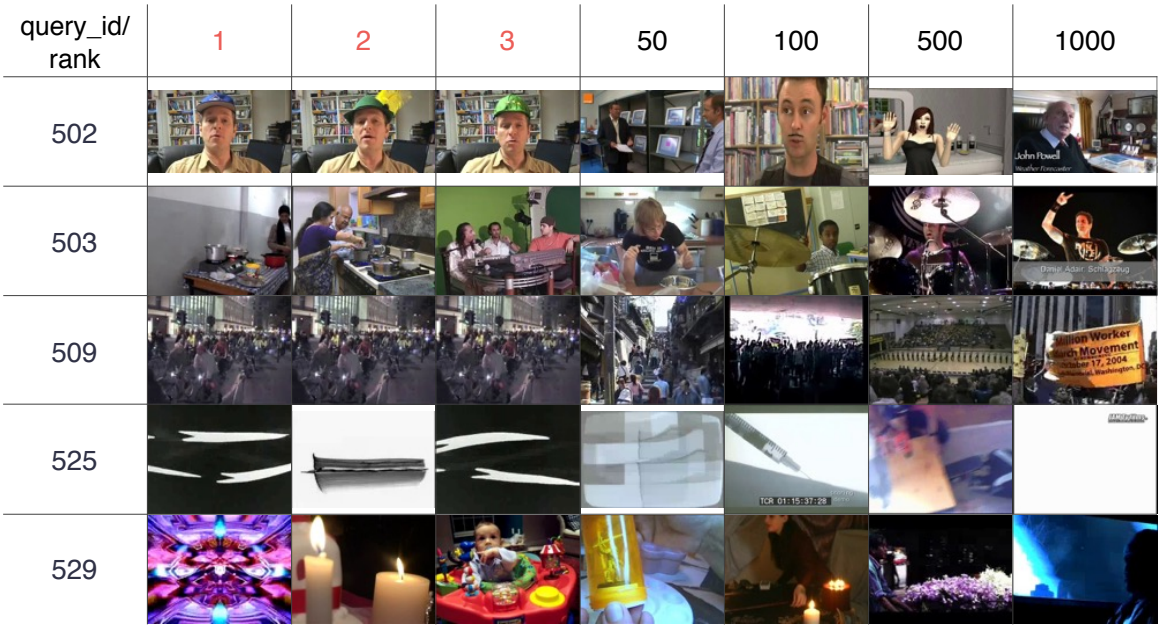


Fig. 4. An illustration of retrieval results for kobe_nict_siegen_D_M_1 and kobe_nict_siegen_D_M_3. Each row shows the results for one query by displaying key frames of the shots ranked in the first, second, third, 50th, 100th, 500th and 1000th positions. Each key frame is selected as the middle frame in a shot.

characteristics. Hence, we will explore training of LSTMs using more densely sampled video frames in the future.

Our current method works as an object recognizer to classify an object located in the center of an input image. We aim to extend this to a scene recognizer by considering correspondences across an entire image. This will allow us to acquire a more detailed meaning of an image by combining the object and scene recognizers. In addition, we plan to incorporate optical flows acquired from image sequences into the microNNs so that they can capture both spatial and temporal characteristics.

REFERENCES

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

[2] G. Awad, J. Fiscus, M. Michel, D. Joy, W. Kraaij, A. F. Smeaton, G. Qunot, M. Eskevich, R. Aly, and R. Ordelman, "Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking," in *Proc. of Proceedings of TRECVID 2016*, 2016.

[3] 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.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[5] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[6] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proc. of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[7] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. of CVPR 2015*, 2015, pp. 1–9.

[10] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.

[11] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *CoRR*, vol. abs/1212.0402, 2012.

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3104–3112.

[15] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. of CVPR 2015*, 2015, pp. 3156–3164.

V. APPENDIX

A. Comparison between microNN and SVM

We compare microNN and SVM on the dataset of TRECVID 2015 Semantic indexing (SIN) task. Fig. 8 shows APs and MAPs of microNN and SVM over 30 concepts. In addition, For each of microNN and SVM, Table.I shows the MAP and the average computation time taken for learning a detector of each concept.

TABLE I
PERFORMANCE COMPARISON BETWEEN SVM AND MICRONN

approach	MAP(%)	Average learning time (second/concept)
microNN	0.1626	0.1385
SVM	0.2148	110.44

Fig. 8 indicates that for some concepts, APs of microNN are slightly lower than those of SVM. On the other hand, Table I presents that the learning speed of microNN is more than 1,000 times faster than that of SVM.

B. Optimization of microNN

We check the effect of unit number in a hidden layer of microNN. Fig. 9 shows APs and MAPs over 30 concepts in TRECVID 2015 SIN task using different unit numbers. Although a clear tendency is not observed, 32 units yield the best result and are used in our actual implementation.

Next, we examine the influence of the number of hidden layers in microNN. Fig. 10 shows APs and MAPs obtained using one hidden layer and two hidden layers. By increasing the number of layers in the hidden layer, no improvement is obtained, so we use only one layer of microNN in this method.

C. Scene recognition

While some concepts like *Car* and *Airplane* are related to objects, others like *Indoor* and *Beach* are related to scenes. Although most of CNNs are trained for object recognition, we consider that the accuracy of concept detection can be improved by additionally using CNNs trained for scene recognition. Two-stream CNN is an independent learning method of multiple information sources. In this research, scene recognition and object recognition are performed by Two-stream CNN by miroNN. Two-stream CNN of scene recognition advances learning with microNN which concatenates features extracted by VGGNet and features extracted by Place_CNDS (Place) into a single higher-dimensional feature vector.

Using the dataset of TRECVID 2015 SIN task, we compare the performance by microNN learned only with features extracted from VGGNet to the performance by microNN learned with the combination of features extracted from Place and VGGNet. In Fig. 11, the former is represented by the dotted line while the latter is depicted by the solid line. In Fig. 11, for concepts such as “planes” and “news casters”, the addition of features extracted from Place significantly degrades the performance. On the other hand, , it improves the performance for concepts related to scenes such as “bridge”, “office”, “hill” are greatly improved. Place has an adverse effect on the identification of a concept expressing an object, but has a beneficial effect on detecting concepts for scenes. Therefore, it is expected that an overall performance can be improved using an ensemble learning model that dynamically change the combination of features extracted from VGGNet and Place.

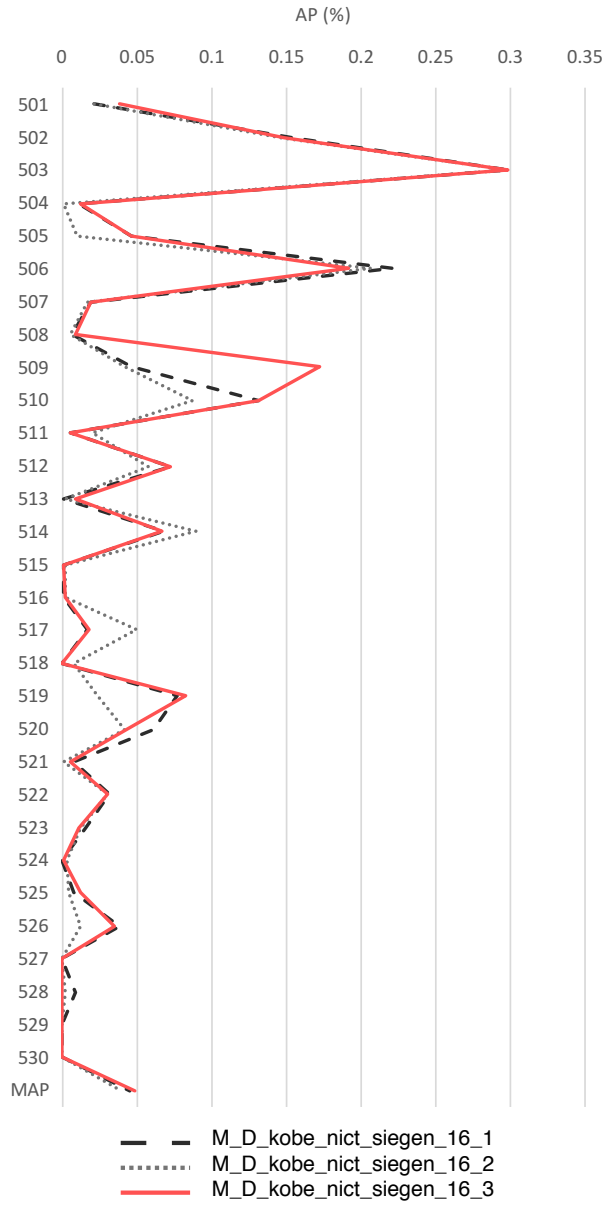


Fig. 5. Performance comparison between our methods.

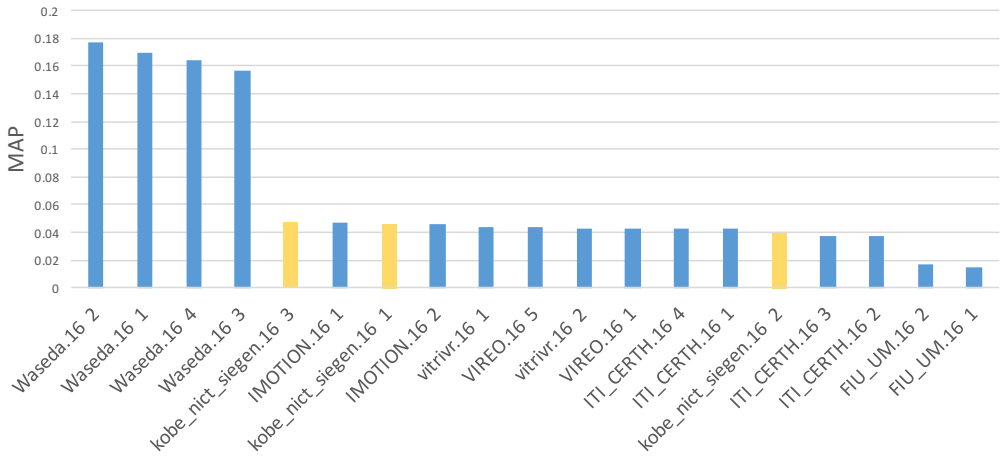


Fig. 6. Performance comparison between our method and other methods developed for the manually-assisted category in the AVS task. The yellow bars indicate our three submitted results.

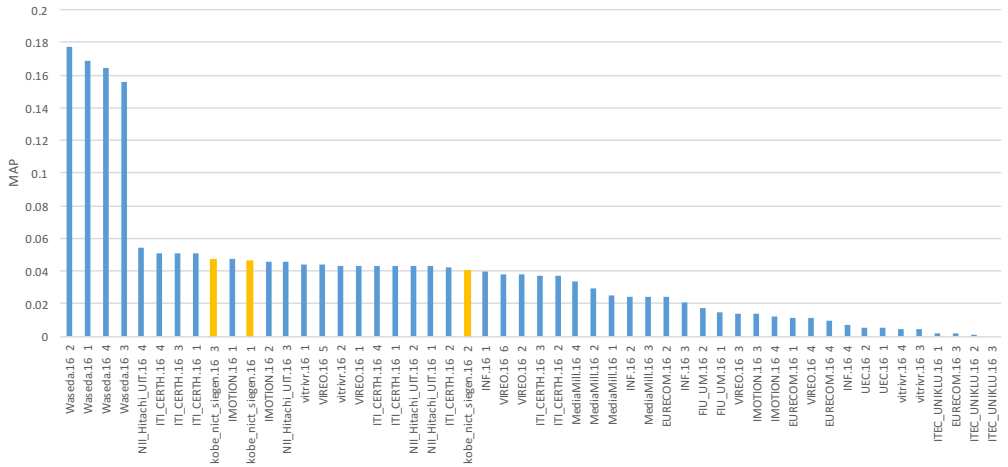


Fig. 7. Performance comparison between out method and all other methods developed for the AVS task. The yellow bars indicate our three submitted results.

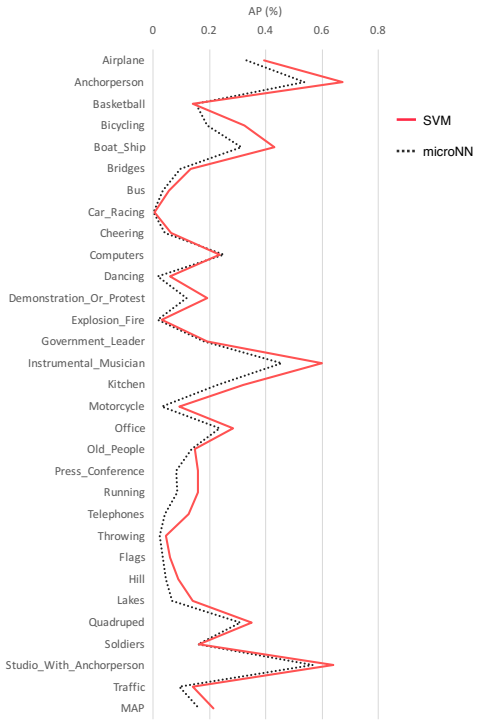


Fig. 8. Performance comparison between microNN and SVM

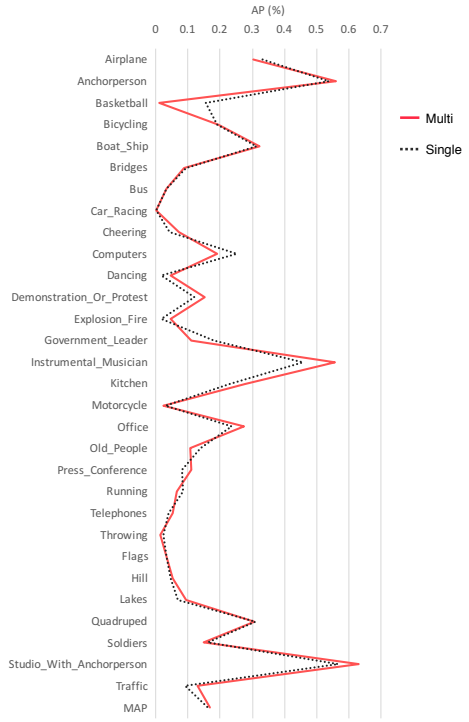


Fig. 10. Performance comparison due to number of layer

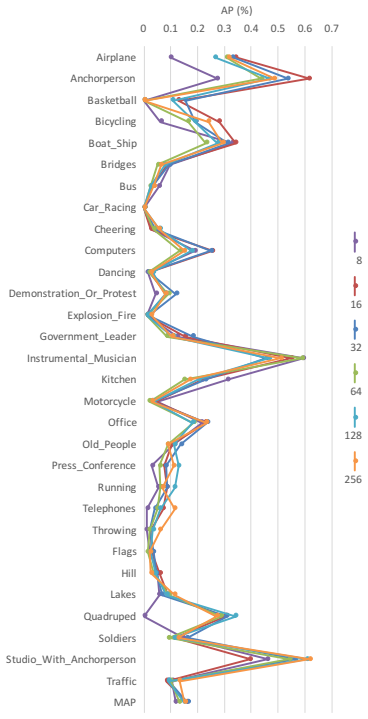


Fig. 9. Performance comparison due to number of unit in hidden layer

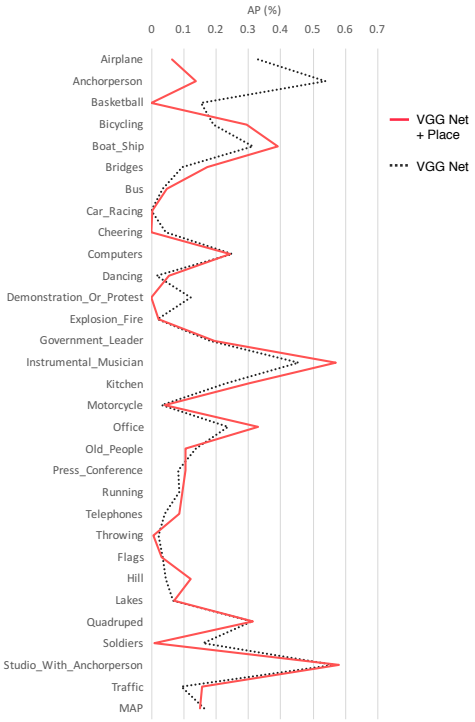


Fig. 11. Performance comaprison between microNN learned only with features extracted from VGGNet and features combining features extracted from Place and VGGNet.