TRECVid 2013 Semantic Video Concept Detection by NTT-MD-DUT

Yongqing Sun, Kyoko Sudo, Yukinobu Taniguchi
NTT Media Intelligence Laboratories, Japan
1-1 Hikarinooka Yokosuka Kanagawa, 239-0847 Japan
Haojie Li,Yue Guan, Lijuan Liu
School of Software, Dalian University of Technology, China
yongqing.sun @lab.ntt.co.jp

ABSTRACT

In this report, we describe the approaches and experiments on TRECVid 2013 video concept detection conducted by NTT Media Intelligence Laboratories in collaboration with Dalian University of Technology. For this year's task, we focused our efforts on two aspects. For the first aspect, we investigated the state-of-the-art machine learning algorithm and feature representation for large-scale concept classifiers construction. Specifically, we first evaluated a newly developed powerful image representation which has been successfully adopted in other visual classification task, i.e., Fisher Vector, for concept detection. Meanwhile, we are also interested in the using of deep learning technique for video classification, and to this end, we have tested various settings of deep learning and the results are reported. For the second aspect, we followed the subspace partition based framework we proposed in our last year work and to balance the precision and efficiency, we proposed a sparse soft-clustering method for ensemble learning, which can get the optimal replication parameter. We conducted experiments on TRECVid SIN task evaluation dataset and submitted 4 runs based on the above methods.

Keywords

Concept Detection, Video Retrieval, Subspace Partitioning, Deep Learning, Fisher Vector

1. Introduction

The goal of concept detection, or high-level feature extraction, is to build mapping functions from the low-level features to the high-level concepts with machine learning techniques [1]. The main building modules of state-of-the-art concept detection systems include feature extraction and fusion, and classifier training. Thus, what kind of features and what classifier models are adopted have critical impacts on the performance of concept detection. Most of efforts of current systems on TRECVid Semantic Indexing (SIN) task are focusing on the above two issues [2][3][4][8], and many powerful image features and advanced classifying schemes have been proposed.

Image features representation is a hot research topic in computer vision and multimedia domains. Bag-of-visual-words (BOV), which transforms local image descriptors into fixed-size sparse feature vector based on keypoints extraction, is the dominantly used approach for image representation in various image classification or retrieval tasks [5][6]. Though impressive results have been achieved, the performance of BOV is degraded due to the information loss in the quantization of local feature descriptors. As an extension of BOV, Fisher Vector is a newly proposed image representation which aggregates local descriptors into a global descriptor using a huge vector and, have been shown to outperform BOV for image/video classification [7][9][10] and retrieval[11]. At the same time, as a similar huge-vector represented global descriptor, Supper Vector [8], won the championships of the last two years TRECVid SIN task. Thus, it is worthy and demanding to study the utility of Fisher Vector in concept detection, i.e., the SIN task.

*TRECVID'13, November26 – 28, 2013, Gaithersburg, Maryland USA.

In the computer vision research community, deep leaning method, especially convolutional neural networks have attracted lots of interests recently. Deep learning is first proposed by G. E. Hinton in 2006 [12] to represent data (including image, audio and text) by mimicking the multilayer abstracting mechanism of human brain, which combined the feature learning and classifier into a single learning framework. Afterwards, deep learning methods have been widely studied and applied to image classification [13], human action recognition [14], gesture recognition [15], and scene parsing [16] and achieved state-of-the-art performances. Motivated by these successful applications, we intent to evaluate the practicability of deep learning methods in large-scale concept detection with diverse training examples.

In the rest of the paper, we will detailed describe and analyze our efforts in using Fisher Vector and deep learning for TRECVid concept detection in Section 2 and Section 3 respectively. In Section 4, we will present our new approach to enhance the ensemble-based classification method proposed in our last year work. Section 5 will give the experimental results and finally, we conclude our work in Section 6.

2. Concept Detection Based on Fisher Vector

The Fisher vector is a rich image representation, which extends the BOV by encoding high-order statistics instead of the 0-order, i.e., the occurring frequency of visual words encoded by BOV. The basic idea is to represent a set of data by gradient of its log likelihood with respect to model parameters.

Suppose we have a generative probability model $\mathrm{P(X|\theta)}$, where $\mathrm{X = \{x_t | x_t \in R^D, t = 1 \dots T\}}$ is a sample set, and θ is the set of model parameters [10]. We can map X into a vector by computing the gradient vector of its loglikelihood function at the current θ:

$$\mathrm{F_x = \nabla_\theta log P(x|\theta)} \qquad (1)$$

Where $\mathrm{F_X}$ is a Fisher Vector, it can be seen as a measurement of the direction to make θ fit better to X. Since $|\theta|$ is fixed, the dimensions of Fisher Vector for different X are the same, which makes $\mathrm{F_X}$ a suitable alternative to represent a image with its local features.

Assuming that we generate the local descriptors $\mathrm{\{x_t | x_t \in R^D, t = 1 \dots T\}}$ of frame I, we model the visual vocabulary with a Gaussian Mixture Model (GMM)[17] where $\theta = \{w_i, \mu_i, \sum_i \quad, i = 1 \dots N\}$ is the input parameter in which w_i , μ_i and $\sum_i$ denote respectively the weight, mean vector and covariance matrix, and each Gaussian corresponds to a visual word. Let $\mathrm{P_i}$ be the distribution of Gaussian i and we get:

$$\mathrm{P(X|\theta) = \sum_{i=1}^{N} w_i P_i(X|\theta) = \sum_{i=1}^{N} w_i N(x|\mu_i, \sum_i \quad)} (2)$$

We denote by $\gamma_i(x_t)$ the probability for x_t to have been assigned to the $\mathrm{i-th}$ Gaussian. By Bayes formula, we get

$$\gamma_i(x_t) = \frac{w_i P_i(x_t|\theta)}{\sum_{j=1}^{N} w_j P_j(x_t|\theta)} \qquad (3)$$

By assuming the covariance matrices are diagonal, and given the sample $\mathrm{X = \{x_t | x_t \in R^D, t = 1 \dots T\}}$, we denate the $\mathrm{d-th}$ element of μ_i as μ_i^d , and the $\mathrm{d-th}$ element of $\sum_i$ as $(\sigma_i^d)^2$. By assuming that each local feature is independent, the Fisher Vector $\mathrm{F_X}$ of feature points set X is:

$$\mathrm{F_X} = [\frac{\partial \mathrm{L(X|\theta)}}{\partial \mu}, \frac{\partial \mathrm{L(X|\theta)}}{\partial \sum}] \qquad (4)$$

$$\frac{\partial \mathrm{L(X|\theta)}}{\partial \mu_i^d} = \sum_{t=1}^{T} \gamma_t(i) \{\frac{x_t^d - \mu_t^d}{(\sigma_i^d)^2}\} \qquad (5)$$

$$\frac{\partial \mathrm{L(X|\theta)}}{\partial \sigma_i^d} = \sum_{t=1}^{T} \gamma_t(i) \{\frac{(x_t^d - \mu_t^d)^2}{(\sigma_i^d)^3} - \frac{1}{\sigma_i^d}\} \qquad (6)$$

Hence, we get the Fisher Vector representation with the dimension of 2KD, in which, K represents the number of the Gaussian in GMM, and D represents the dimension of each lower feature. The first term in Eq (4) is composed of first order differences of feature points to cluster centers. The second term contains second order terms.

In our experiments, we use dense SIFT as local features of the image and, to reduce the dimension of Fisher Vector, we use Principle Component Analysis to reduce SIFT descriptors into 64. We randomly select about 2M features from the training set and use them to train the PCA projection matrix and then generate the codebooks with GMMs. We use 256 Gaussians in our experiment and only the mean vector (see Eq. (4)) are kept as the Fisher Vector, thus the dimension of our Fisher Vector is 16384.

After extracting the Fisher Vector representation for each image, we perform two kinds of normalization, i.e., L2 normalization and power normalization [7], which have been demonstrated that they can remarkably improve the classification performance. Finally, we train Support Vector Machines (SVM) classifiers with linear kernels.

3. Concept Detection with Convolutional Neural Networks

Deep neural network has emerged as robust supervised feature learning and classification tools for general objection recognition and image classification tasks [19][15]. As a deep hierarchical model, Convolutional Neural Networks (CNNs) are multi-layered NNs specialized on recognizing visual patterns directly from image pixels and, are well-known for robustness to distortion and minimal pre-processing [20]. In our experiments, we adopt CNN with Max-Pooling layer architecture (MPCNNs), since it is found that Max-Pooling can lead to faster convergence, select superior invariant features, and improve generalization [21]. MPCNNs vary in how convolutional and subsampling layers are realized and trained [15]. Figure 1 illustrates our MPCNN architecture used.

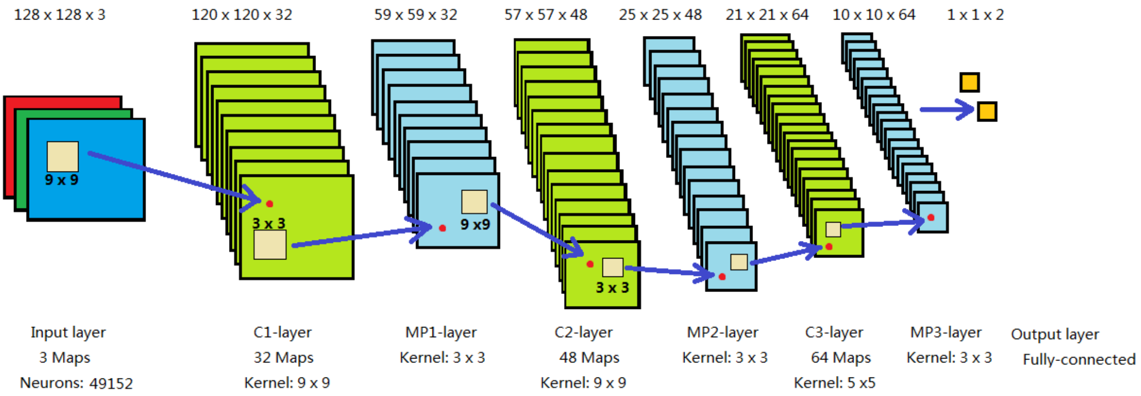

Figure 1. The architecture for MPCNN

Convolutional layer

A convolutional layer is parameterized by: the number of maps, the size of the maps and kernel sizes. Each layer has M maps of equal size (M_x, M_y). A kernel of size (K_x, K_y) (as the rectangles inside the maps shown in Figure 1) is shifted over the valid region of the input image. Each map in layer L^n is connected to all maps in layer L^{n-1}. Neurons of a given map share their weights but have different input fields.

Max-pooling layer

The output of the max-pooling layer is given by the maximum activation over non-overlapping rectangular regions of size (K_x, K_y) . Max-pooling creates position invariance over larger local regions and down-samples the input image by a factor of K_x and K_y along each direction.

Classification layer

After multiple convolutional and max-pooling layers, a shallow Multi-layer Perceptron (MLP) is used to complete the MPCNN. The output layer has one neuron per class. A softmax activation function is used, thus each neuron's output represents the posterior class probability.

In SIN task, we have 60 concepts and each concept has training examples [22]. In our experiments, we conducted two types of testing: binary classification and multi-classification. In the multi-classification setting, beside the 60 classes (concepts), we construct a common negative class by checking the negative keyframes of each concept. Thus for such case, in the architecture shown in Figure 1, the number of neuron in the output layer will be changed to 61.We separated the training examples into training and testing sets to test the above two settings. We used the GPU implementation of CNNs [23] with a NVIDIA GTX 670 GPU. The resolution of each keyframe is resized to 128×128. In the binary classification, the training time for each concept is about 3 hours and the average error rate is below 0.01. Though the error is small, considering the number of negatives is much larger than that of positives, there is some over-fitting in the binary classification training. For multi-classification, it took about 2 days to train the CNNs. The final error rate is 0.63 for the training set and 0.73 for the testing set. We also resized the keyframe resolution to 164×164 and tested the multi-classification, however, the results are not good as 128×128. The reason maybe there is not enough training examples.

4. Ensemble-based Concept Detection

In last year's TRECVid SIN task, we proposed an ensemble learning method based on a scalable clustering method called Clara [3] for the purpose of improving the training efficiency and boosting the classifier performance. We drew the conclusion last year that the classification performance is highly influenced by the value of replication parameter *L*, and the larger *L is*, the higher MAP can be achieved. However, on the other hand, too many replications of samples can inevitably place a heavy burden of the training despite of higher MAP. So how to get the optimal replication parameter *L* becomes a key problem.

To address this problem and further improve the performances, we used sparse coding for ensemble learning. During the process of training, we grouped each training sample to multiple subspaces according to the indices of its non-zero sparse codes. Such sparse soft-clustering method can achieve optimal compromise between precision and efficiency since the definition of sparse coding can ensure that the number of subspaces sample can be soft-clustered into is optimal. Furthermore, during testing, we also used sparse coding for fusion of sub-classifiers to improve testing efficiency. We only used the sub-classifiers corresponding to the non-zero sparse codes of current testing samples for classifier fusion for getting final decision.

5. Experiments

We have submitted 4 runs totally. The description and MAP of each run are shown in the following Tab. 1. The first run is based on the ensemble learning with sparse coding. The second run is based on the ensemble learning with Clara, and the third is the fisher vector run classified by the large linear classification method LIBLINEAR [24], and the last run is the deep learning run.

Tab.1. Description and InfMAP of our 4 SIN runs

Submitted run	InfMAP	Method
13_M_A_NTT_DUT_1_1	0.167	Ensemble learning with sparse coding
13_M_A_NTT_DUT_2_2	0.118	Ensemble learning with Clara
13_M_A_NTT_DUT_3_3	0.105	Linear classificat ion Fisher vector
13_M_A_NTT_DUT_4_4	0.048	Deep learning

From the table, we can see that sparse coding based ensemble learning can improve the performance largely compared with Clara based methods. This is possibly due to sparse coding's advantage of the minimum reconstruction error. However, Clara-based method can also achieve better performances than global linear classification of Fisher vector due to the replication of training samples. Very surprisingly, we found that the last deep learning run got the lowest precision, which is far out of our expectation. The reasons may be due to the problem of under-fitting since the number of parameters in the deep neural network to be determined are much larger than the number of training samples when we normalized the all keyframes to the size of 128×128. Another reason may lie in the high complexity of TRECVid data, which is more diverse than data used in other tasks, such as object classification.

Figure 2 shows the average precisions of our best run 13_M_A_NTT_DUT_1_1. From the figure, we can see that the majority of the concepts can get higher precision than the average. In particular, many concepts such as ***Airplane, Animal, Beach, Boat_Ship, Boy, Bridges, Government_Leader, Instrumental_Musician, Baby, George_Bush*** etc., achieves good average precisions.

Also surprisingly, after checking the top 100 keyframes of our best run, we found that the numbers of hits at depths 100 (top 100) of many concepts are far more than the numbers returned from NIST. For example, the number of hits among the top 100 of the concept

"***Anchorperson***" is almost 100, much greater than 84 returned from NIST as shown in Figure 3. What is the reason which causes the big differences? Through extensive examination, we found that there maybe only one reason: to take full advantages of abundant video

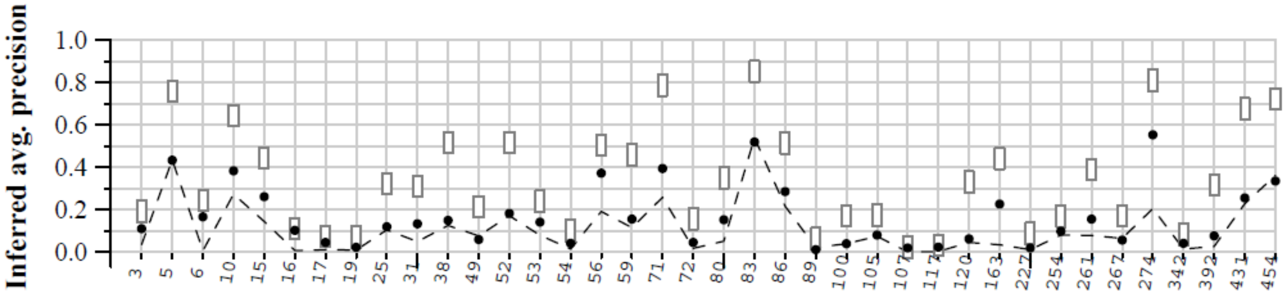

Figure 2. The Average Precisions of our best run 13_M_A_NTT_DUT_1_1

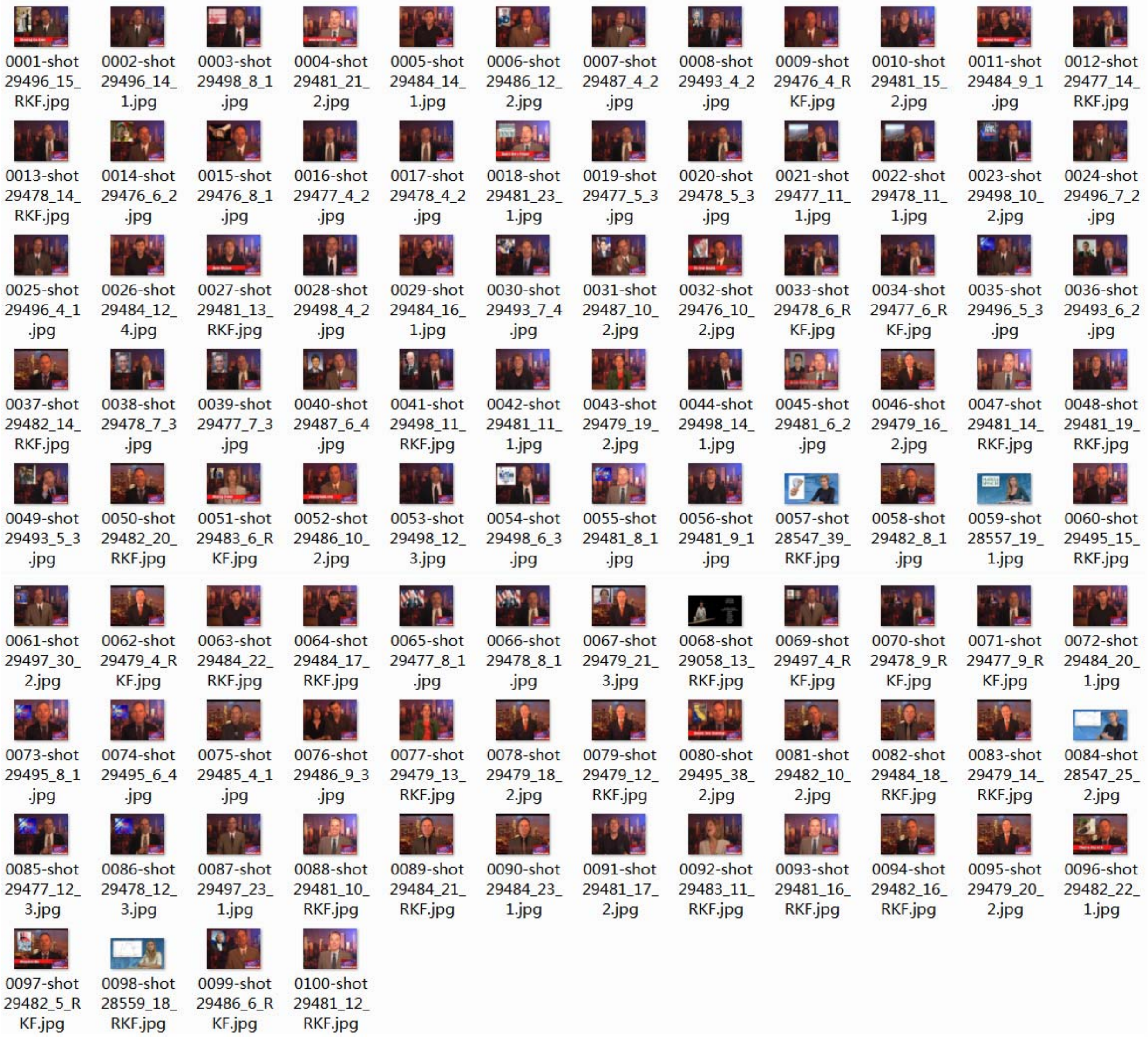

Figure 3. The top 100 key frames returned by our best run13_M_A_NTT_DUT_1_1

(The non-RKF frames are our new added frames) information, we extracted about double number of keyframes than those from TRECVID commonly used keyframes which were released by the LIG collaboration annotation. However, the ground truth from NIST may exist inconsistencies with our newly added key frames which may cause the difference.

6. Conclusions and Future Works

To summarize, the introduction of sparse coding into ensemble learning can improve the performance largely due to its advantage of the minimum reconstruction error and the optimal compromise between precision and efficiency. Very surprisingly, deep learning got lowest precision, possibly due to the problem of under-fitting and huge diversity of TRECVid data. However, due to the great success of deep learning in visual recognition [18], we will further investigate the exact factors which cause the big difference in the future.

REFERENCES

[1] Cees G. M. Snoek, Marcel Worring, Concept-Based Video Retrieval, Foundations and Trends in Information Retrieval archive, pp.215~322, 2008

[2] Sheng Tang, Jin-Tao Li, Ming Li, Cheng Xie, Yi-Zhi Liu, Kun Tao, Shao-Xi Xu; "TRECVID 2008 High-Level Feature Extraction By MCG-ICT-CAS"; Proc. TRECVID 2008 Workshop, Gaithesburg, USA , Nov 2008.

[3] Yongqing Sun, Kyoko Sudo, Yukinobu Taniguchi, Haojie Li, Lei Yi, Yue Guan, TRECVid 2012 Semantic Video Concept Detection by NTT-MD-DUT, in Proc. TRECVID 2012 Workshop

[4] Y. Sun, G. Irie, T. Satou, A. Kojima, K. Sudo, M. Morimoto, A. Kimura, TRECVID 2011 Semantic Indexing Task By NTT-SL-ZJU, in Proc. TRECVID 2011 Workshop

[5] Haojie Li, Xiaohui Wang, Jinhui Tang, Chunxia Zhao, Combining global and local matching of multiple features for precise item image retrieval, Multimedia Systems, 2012

[6] Y.-G. Jiang, J. Yang, C.-W. Ngo, and A. G. Hauptmann, Representations of keypoint-based semantic concept detection: A comprehensive study, IEEE Transactions on Multimedia, vol. 12, pp. 42–53, 2010

[7] Jorge Sánchez, Florent Perronnin, Thomas Mensink, Improved Fisher Vector for Large Scale Image Classification, ECCV, 2010

[8] Paul Over, Jonathan Fiscus, Greg Sanders, Barbara Shaw, TRECVID 2013 - An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics

[9] F. Perronnin and C. Dance. Fisher kernels on visual vocabulariesfor image categorization.In CVPR, 2006.

[10] Chen Sun, and Ram Nevatia, Large-scale Web Video Event Classification by use of Fisher Vectors, WACV, 2013

[11] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier. Large-scale image retrieval with compressed Fisher vectors. In CVPR, 2010

[12] G. E. and Salakhutdinov, R. R Reducing the dimensionality of data with neural networks. Science, Vol. 313. no. 5786, pp. 504 – 507

[13] Alex Krizhevsky, IlyaSutskever, Geoffrey E 12, ImageNet Classification with Deep Convolutional Neural Networks, NIPS, 2012

[14] Ho-Joon Kim and Joseph S. Lee and Hyun-Seung Yang,Human Action Recognition Using a Modified Convolutional Neural Network, International Symposium on Neural Networks, 2007, pp.715-723

[15] Nagi et al.,Max-Pooling Convolutional Neural Networks for hand gesture recognition, International Conference on Signal and Image Processing Applications, 2011

[16] Pedro H. O. Pinheiro, Ronan Collobert, Recurrent Convolutional Neural Networks for Scene Parsing

[17] Perronnin, F., Dance, C.R., Csurka, G., Bressan, M.: Adapted vocabularies for generic visual categorization. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006

[18] Alex Krizhevsky, IlyaSutskever, Geoffrey E 12, ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012.

[19] D. Ciresfan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, Flexible, high performance convolutional neural networks for image classification, in Proc. of 22nd Intl. Joint Conf. on Artificial Intelligence,2011, pp. 1237–1242

[20] Y. LeCun, K. Kavukcuoglu, and C. Farabet, Convolutional networksand applications in vision,in Proc. of the IEEE Intl. Symp. on Circuitsand Systems, Jun. 2010, pp. 253–226.

[21] Dominik Scherer, AdreasM ¨ uller, andSven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In International Conference on Artificial Neural Networks, 2010

[22] Stephane Ayache and Georges Quénot, Video Corpus Annotation using Active Learning, In ECIR 2008

[23] https://code.google.com/p/cuda-convnet/wiki/TrainingNet

[24] http://www.csie.ntu.edu.tw/~cjlin/linlinear/