Learning Neural Networks with Ranking-based Losses for Action Retrieval

Md Atiqur Rahman and Yang Wang
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada
{*atique, ywang*}*@cs.umanitoba.ca*

***Abstract*—We consider the problem of learning image/video retrieval using a neural network based approach that optimizes the ROC loss function. Neural network is one of the most widely used techniques in computer vision. Standard neural network uses simple loss functions, such as the softmax loss or hinge loss over labels. Such loss functions are suitable for standard classification problems where the performance is measured by the overall accuracy. For image/video retrieval, the performance is usually measured by some ranking-based loss that is not well captured by the softmax loss or hinge loss. In this paper, we develop a learning approach that incorporates the ranking-based loss function in neural network. We apply our approach in the problem of action retrieval in static images and videos. The experimental results show that our proposed approach outperforms standard neural networks trained with softmax loss as well as an SVM-based approach that also optimizes the ROC loss function.**

***Keywords*-deep learning; image/video retrieval; ROC area optimization**

I. Introduction

Neural network is one of the most widely used machine learning algorithms in computer vision nowadays. In particular, multi-layer convolutional neural networks have been shown to be very effective in a variety of computer vision problems, e.g., image classification [1], object detection [2], etc. Classic neural networks use the softmax loss as the loss function. The learning of neural networks involves optimizing this loss function using stochastic gradient descent and backpropagation.

For standard multi-class classification, the softmax loss is a reasonable choice as the loss function, since the performance of multi-class classification is usually measured by the overall accuracy. But for many computer vision applications, the performance is measured by some complex losses that do not decompose into a simple sum of individual terms measured over each training instance. Examples of such complex losses include the area under ROC curve, the F_1-score etc., which are commonly used in information retrieval. For these applications, using the softmax loss as the loss function is often suboptimal, since the learning will end up optimizing the wrong performance measure. Ideally, we would like the learning algorithm to directly optimize the right performance measure.

Previous work [3] has developed methods for optimizing such complex losses in the case of linear classifiers (e.g., linear SVM). But optimizing such complex losses in neural networks is more challenging since neural networks are nonlinear classifiers. In this paper, we propose a learning algorithm based on backpropagation to optimize complex loss functions. In particular, we consider the application in image and video retrieval and use our approach to optimize the area under ROC curve, which is commonly used to measure the performance of image/video retrieval methods. We apply our approach on the problem of action retrieval in static images and videos.

II. Related Work

Our work overlaps with two lines of research – one involves directly optimizing application specific performance measures, which in this case is ROC area, and the other direction is the image/video retrieval using Deep Neural Networks (DNNs). Among the early approaches that used neural networks for learning ranking functions with applications to information retrieval, RankProp [4] is one which employs point-wise training, i.e.; training on individual observations only, and therefore, offers better running time. But, it lacks a probabilistic model, and does not provide a well-defined convergence condition. RankNet [5], on the other hand, provides a probabilistic model for ranking by training a neural network using gradient descent with a relative entropy based general cost function. Like ours, RankNet is a pairwise approach, which trains on pairs of relevant-irrelevant examples and gives preference ranking. RankOpt [6], on the other hand, provides a linear model that optimizes ROC area by approximating it using a sigmoid function.

The work of Joachims et al. [3] is most closest to ours. Based on the structural SVM framework of [7], it provides efficient algorithms to directly optimize a range of nonlinear performance measures including ROC area and accounts for linear running time in terms of total number of observations. However, it offers a linear model, whereas, our approach provides a nonlinear model that can directly optimize ROC area. The superiority of our approach over this work is demonstrated in the Results subsection.

The work of Mcfee et al. [8] is also based on structural SVM framework. It uses gradient descent for metric learning interpreted as an information retrieval problem and can optimize a set of ranking measures including ROC area. Having modified the 1-slack margin-rescaling cutting-plane algorithm of [9] by incorporating a new constraint for the metric and replacing the ℓ_2 penalty with an ℓ_1 penalty in the regularization term, it ensured more sparsity and low-rank solutions. Another pair-wise ranking approach was proposed by Cao et al. [10] which was based on the ranking SVM but was targeted for document retrieval. They modified the loss function of ranking SVM to include two new cost parameters called rank parameters and query parameters. Rank parameters try to offset the bias due to the imbalance in the number of instance pairs from different ranks, thereby intensifying training on the top rankings. Query parameters are intended to offset the bias introduced as the number of relevant instances varies over the queries.

Regarding the use of DNNs to address the image retrieval problem, recent advancement in learning high-level representation of images via the use of Convolutional Neural Networks (CNNs) has made it possible to close the semantic gap between high-level representation of textual queries and low-level representation of images. For example, Bai et al. learned high-level representations of images by using a multi-tasking transfer learning DNN architecture [11] and trained a set of binary classifiers for different textual queries based on these representations. Since it is very difficult for such an approach to scale with a massive number of queries, a bag-of-words (BoWs) based DNN model was proposed in [12]. Here, the DNN learned high-level representations of input images are mapped into BoWs space where visual similarity between images is computed, whereas, relevance between textual query and image is measured by the cosine similarity between BoW representations of the two. To further improve the results, a page rank algorithm was used to consider the visual similarity of the retrieved images.

The work of Razavian et al. [13] is related to our work as it also exploits image representations obtained from a classification CNN for the task of image retrieval. Their method does not require fine-tuning the classification CNN with target domain data, still can deliver high retrieval accuracy when compared to retrieval techniques not based on CNN image representations. A very recent work [14] in this direction of exploiting classification CNN for image retrieval showed that image representations obtained from the lower layers of the classification CNN performs better than that obtained from the last layer as is usually done with other approaches. Based on the recent successful classification CNNs like GoogleNet [15] and OxfordNet [16], they leverage the benefit of using VLAD encoding of the local convolutional features obtained from the lower layers of these classification nets for instance level image retrieval.

III. Proposed Approach

In this work, we aim to learn binary ranking functions that directly optimize ROC area. Since ROC area is a nonlinear performance measure that cannot be decomposed over individual instances of a training sample, we use the multivariate structured SVM formulation to predict the ranking of the whole sample instead of individual instances as described in [3]. Unlike this SVM approach, we use a neural network having the architecture as shown in Fig. 1, with a view to learning complex nonlinear ranking functions.

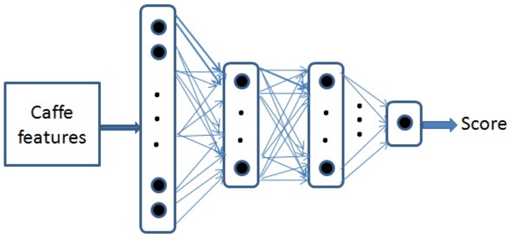

Figure 1. Architecture of the neural network for the proposed approach.

To formally describe, let S represent a training sample of n examples $S = ((x_1, y_1), \dots, (x_n, y_n))$, where $x_i \in \mathbb{R}^d$ represents the feature vector for a single example and $y_i \in \{-1, +1\}$ represents one of two possible ranks of the example, namely, irrelevant or relevant. Instead of predicting the rank of each example individually, we try to learn a mapping function $h : X \times \cdots \times X \rightarrow Y$ that takes all n examples $X = (x_1, \dots, x_n)$ at once and maps them to a vector of n labels $y = \{y_1, \dots, y_n\} \in Y = \{-1, +1\}^n$. In order to obtain the best label vector y that gives the optimal ordering of the sample giving the best ROC area measure, we use a nonlinear discriminant function p as follows:

$$p(X) = \arg\max_{y \in Y} F_W(X, y) \tag{1}$$

Here, $F_W(X, y)$ is a scoring function which in turn is defined as follows:

$$F_W(X, y) = W_M^T \Psi(\phi(X), y) \tag{2}$$

Here, $\phi(X)$ is a transformation function that performs a sequence of nonlinear transformations on the sample X. To be specific, each example $x_i \in X$ is passed through $m = 1, 2, \dots, (M-1)$ layers of nonlinear transformations in a neural network, where the output of the m^{th} layer is given by -

$$v_i^{m+1} = s(W_m^T v_i^m + b_m) \tag{3}$$

with the initial case of $v_i^1 = x_i$. Here, W_m and b_m are the set of weights and biases respectively, at layer m and $s : \mathbb{R} \mapsto \mathbb{R}$ is a nonlinear activation function, which in our case is the $sigmoid$ function. Therefore, the whole sample $X = (x_1, \dots, x_n)$ is transformed to a nonlinear representation V^M, such that $V^M = \phi(X) = (v_1^M, \dots, v_n^M)$.

Now, referring back to Eq. 2, $\Psi(\phi(X), y)$ is a compatibility function that measures the compatibility between the transformed input V^M and output label vector y. Following [3], we used a simple compatibility function Ψ of the following form that depends only on individual transformed training example v_i^M and its rank label y_i.

$$\Psi(\phi(X), y) = \Psi(V^M, y) = \sum_{i=1}^{n} v_i^M y_i = V^M y \qquad (4)$$

Finally, the $(M-1)$ nonlinear layers of the neural network are followed by a linear scoring layer (the M^{th} layer) with weights W_M (and no biases) to prduce the scores $F_W(X, y)$ as shown in Eq. 2. Therefore, putting everything together, the optimal labeling sequence for the training sample X would be –

$$p(X) = \arg\max_{y \in Y} W_M^T V^M y \qquad (5)$$

Once the scores for the whole sample is predicted, we can simply sort the scores in descending order to get a total ranking of the sample. A perfect ranking requires the scores for all relevant examples to be higher than that of the irrelevant ones. In order to learn the retrieval function that minimizes ROC area loss of the training sample, the neural network tries to optimize an objective function of the following form:

$$\begin{aligned} \arg\min_{W_m, b_m} O = \;& O_1 + O_2 \\ =\;& F_W(X, y') + \Delta(y, y') - F_W(X, y) \\ &+ \frac{\lambda}{2}\left(\sum_{m=1}^{M} ||W_m||_F^2 + \sum_{m=1}^{M-1} ||b_m||_2^2\right) \\ =\;& W_M^T V^M y' + \Delta(y, y') - W_M^T V^M y \\ &+ \frac{\lambda}{2}\left(\sum_{m=1}^{M} ||W_m||_F^2 + \sum_{m=1}^{M-1} ||b_m||_2^2\right) \end{aligned} \qquad (6)$$

The objective function O includes two terms – the loss term O_1 and the regularization term O_2. Minimizing O_1 actually leads to maximizing $F_W(X, y)$, the score for the correct label vector y, while minimizing $F_W(X, y')$, the score for any incorrect label vector y'. Instead of an example-based loss, O_1 is having a sample-based loss $\Delta(y, y')$ which is actually an application specific loss and thus measures the ROC area loss in this case. The regularization term O_2 tries to keep the parameters of the neural network small. Here, $||A||_F$ represents the Frobenius norm of the matrix A and λ is a regularization parameter. Like [3], we are using pair-wise ranking to learn retrieval functions. Therefore, the ROC area loss in this setting can be simply measured by the number of misranked pairs as follows:

$$\Delta(y, y') = \frac{\text{total misranked pairs}}{P \times N} \qquad (7)$$

Here, P is the total number of relevant examples and N is the total number of irrelevant examples in the training sample. To calculate the total misranked pairs for the current parameters, we use Algorithm 3 as described in [3].

In order to obtain the set of weights W_m (for $m = 1, 2, \dots, M$) and biases b_m (for $m = 1, 2, \dots, M-1$), we solve Eq. 6, using stochastic gradient descent. The gradient G_M^W of the objective function O with respect to the weights of the M^{th} layer (i.e; W_M) can then be written as follows:

$$\begin{aligned} G_M^W = \frac{\partial O}{\partial W_M} &= \Psi(\phi(X), y') - \Psi(\phi(X), y) + \lambda W_M \\ &= V^M y' - V^M y + \lambda W_M \end{aligned} \qquad (8)$$

For the other layers of the neural network, i.e.; for $m = (M-1), \dots, 1$, the gradients G_m^W and G_m^b with respect to the weights W_m and biases b_m, respectively can be computed using the chain rule of derivates as follows:

$$G_m^W = \frac{\partial O}{\partial V_M} \frac{\partial V_M}{\partial V_{M-1}} \cdots \frac{\partial V_{m+1}}{\partial W_m} = \delta_m V_m + \lambda W_m \qquad (9)$$

$$G_m^b = \delta_m + \lambda b_m \qquad (10)$$

where, δ_m is defined as follows:

$$\delta_m = \begin{cases} W_{m+1}^T (y' - y) \odot s'(Z_m) & \text{, if } m = M-1 \\ \\ W_{m+1}^T \delta_{m+1} \odot s'(Z_m) & \text{, otherwise} \end{cases} \qquad (11)$$

Here, $s'(a)$ is the derivative of the $sigmoid$ function $s(a) = \frac{1}{1+e^{-a}}$ and $\odot$ is an element-wise multiplication operator. Z_m is defined as -

$$Z_m = W_m^T V_m + b_m \qquad (12)$$

The update rule for the I^{th} iteration of weight update then becomes $W_m^I = W_m^{(I-1)} + \eta G_m^W$, where η is the learning rate. Update rule for the biases are similar. Details about setting the hyper-parameters η and λ are discussed in section IV.

IV. Experiments

A. Setup and Datasets

In order to predict a binary ranking with a view to retrieving images or videos from a repository, we directly optimize the ROC area using a neural network. We compare our approach with an structural SVM formulation called $\text{SVM}_{\text{multi}}$ [3] as implemented in $\text{SVM}^{\text{light}}$ [17] that can also directly optimize ROC area. Moreover, to support our hypothesis that directly optimizing application specific loss (in this case, ROC area loss) gives better performance than optimizing some surrogate loss, we compare our approach with a standard neural network having the same architecture and parameters as ours, but optimized for general classification loss, more specifically, softmax loss. For the rest of the paper, we call our neural network approach directly optimizing for ROC area loss as NN_{ROC} and the other one optimizing for softmax loss as NN_{Gen}.

To evaluate our approach, we conducted experiments on two different datasets – the Stanford 40 actions dataset [18] and the UCF101 actions dataset [19]. The Stanford 40 actions dataset contains 4,000 training images and 5,532 test

Table I
RETRIEVAL PERFORMANCE COMPARISON OF THE PROPOSED APPROACH WITH THE BASELINE APPROACHES.

Dataset	Average ROC area (%)			Improvement over the baselines (%)		# of classes showing performance gain		# of classes showing performance decline	
	NN_{Gen}	$\text{SVM}_{\text{multi}}$	NN_{ROC}	NN_{Gen}	$\text{SVM}_{\text{multi}}$	NN_{Gen}	$\text{SVM}_{\text{multi}}$	NN_{Gen}	$\text{SVM}_{\text{multi}}$
Stanford 40 actions Train/Test: 4000/5532 Features: 4096 Total Class: 40	84.65	88.00	91.11	6.46	3.11	all	35	none	5
UCF101 actions Train/Test: 9537/3723 Features: 4096 Total Class: 101	95.12	98.66	99.13	4.01	0.47	all	51	none	34

images covering a total of 40 different human action categories. The UCF101, on the other hand, is a video dataset containing videos of 101 action classes with a train and test split of 9,537 and 3,783 videos respectively, summing up to a total of 13,320 videos. For all the experiments, we used the train/test splits as suggested by the datasets.

The neural network as shown in Fig. 1 was used for both the proposed approach and the baseline approach of NN_{Gen}. Having a biased hyperplane, it consists of four layers (i.e.; $M = 4$) with 100, 50, 50 and 1 units in the layers, respectively. Stochastic gradient descent with momentum 0.9, weight decay 0.0005 and minibatch size of 100 were used for training the network. We used fixed learning rates of 10^{-3} and 10^{-4} for the Stanford 40 and UCF101 datasets, respectively, as selected by line search. All the weights and biases of the network were initialized randomly. We implemented the neural network using a popular deep learning tool called MatConvNet [20].

For the image dataset, we extracted 4,096 dimensional feature vector for each image from the fc6 fully connected layer of the Caffe implementation [21] of AlexNet deep network model as described in [1]. We used activations from fc6 layer, as they have have been reported to produce better results for a variety of visual recognition tasks [22] .

For the UCF101 dataset, we used a deep-learning based video representation tool called Convolutional 3D (C3D) [23]. C3D is a deep 3-D convolutional network that is trained on a large scale of video dataset. It has been reported to provide state-of-the-art video representation used for video analysis. C3D segments a video into chunks of 20 frames. It then passes each chunk of frames through the deep network and extracts 4,096 dimensional deep-learning feature vector from the fully connected layer fc7. Finally, the individual group feature vectors are averaged over each video to produce a single 4,096 dimensional vector representation of the video.

The regularization parameters λ (for NN_{ROC} and NN_{Gen}) and C (for $\text{SVM}_{\text{multi}}$) were set empirically by using a validation set consisting of $\frac{1}{3}$ of the training examples selected at random. For the baseline SVM approach of $\text{SVM}_{\text{multi}}$ both linear and nonlinear kernels were used and the best results are reported.

B. Results

For each of the classes in a dataset, we learn a binary retrieval function considering the examples belonging to the query class as relevant and all other examples as irrelevant. Table I lists the average performance on the two datasets as achieved by our approach NN_{ROC} and the two baseline methods of $\text{SVM}_{\text{multi}}$ and NN_{Gen}. We use ROC area as the measure of retrieval performance.

While comparing our approach NN_{ROC} with the baseline approach $\text{SVM}_{\text{multi}}$, among the 40 classes in the image collection, 35 classes showed performance gain as opposed to 5 showing decline in performance, whereas, for the video collection, 51 showed performance gain, 34 showed decline in performance and the rest were affected by neither approach. On the other hand, while comparing with the other baseline approach of NN_{Gen}, all the classes for both datasets see performance improvement with an average performance gain larger than that achieved over $\text{SVM}_{\text{multi}}$. This is no surprise as NN_{Gen} is not optimizing ROC area loss, rather general classification loss, and therefore, exhibits poor retrieval performance.

Figure 2 shows ROC curves of our approach as well as the two baseline methods for 40 different query classes on the Stanford 40 actions dataset [18]. As depicted in Table I, the ROC curves of our approach (blue curves) are covering the respective ROC curves of the baseline methods for almost all classes.

Since learning binary retrieval functions requires predicting binary ranking that maximizes the scores for the relevant examples while minimizing scores for the irrelevant ones, we can perform classification by these scores. Therefore, to further investigate the effectiveness of our approach, we perform multi-class classification by taking the scores of an example for all the classes and then predicting the class of the example to be one that gives the maximum score. The results are shown in Table II.

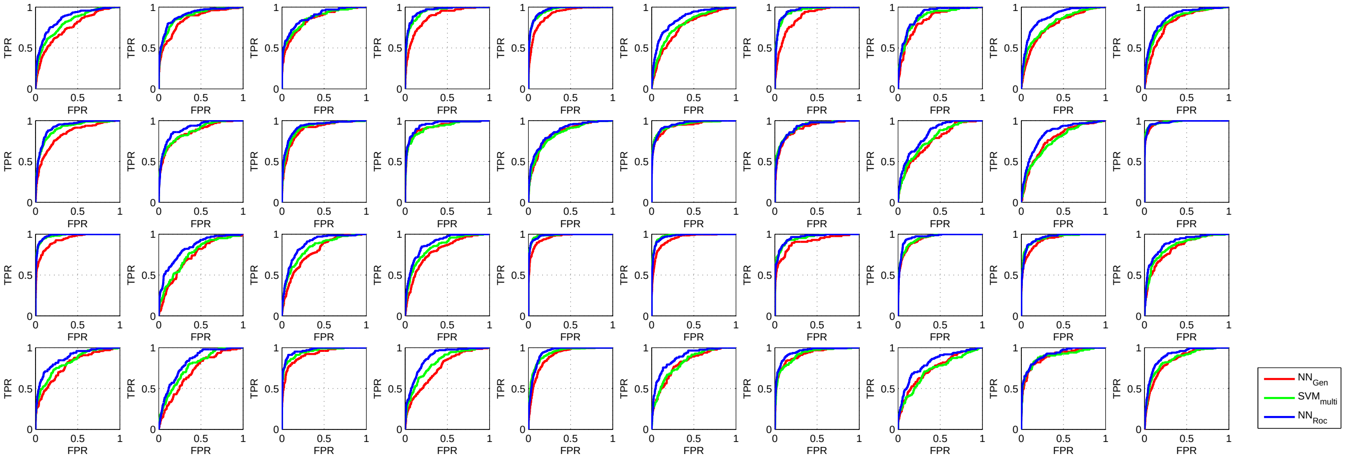

Figure 2. ROC curves of the proposed method and the baseline methods for the Stanford 40 actions dataset [18]. TPR and FPR represent True Positive Rate and False Positive Rate, respectively.

As shown in the table, our proposed approach outperforms the two baseline methods for both datasets. The improvement is more pronounced over the baseline approach of NN_{Gen} than $\text{SVM}_{\text{multi}}$. This is attributed to the fact that NN_{Gen} is optimizing for softmax loss which is a general classification loss, whereas, $\text{SVM}_{\text{multi}}$ and our approach both optimize for the application specific loss, namely ROC area loss. The reason for our approach to demonstrate superiority over the SVM based approach is because, our approach provides a nonlinear model which is able to better handle the higher order nonlinearities inherent in the data. Figure 3 shows the confusion of our approach for the multi-class classification on the Stanford 40 actions dataset [18].

We also show some retrieval examples of our approach and the two baseline methods for three different queries on the Stanford 40 actions [18] dataset as shown in Fig. 4. For each of the three methods, we show the top ten retrieved examples believed to contain humans performing the query action. Qualitatively, better results are produced by our method over the baselines as evidenced from the retrieved examples.

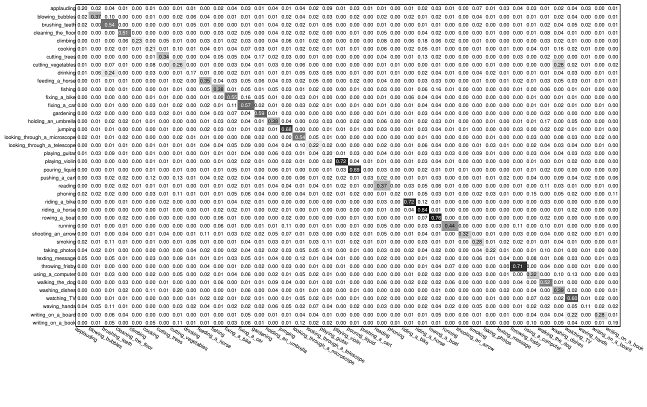

Figure 3. Confusion matrix of our approach on the Stanford 40 actions dataset [18].

Table II
COMPARISON OF MULTI-CLASS CLASSIFICATION ACCURACY OF OUR APPROACH WITH THE BASELINE METHODS.

Dataset / Method	Stanford 40	UCF101
NN_{Gen}	28.53%	64.72%
$\text{SVM}_{\text{multi}}$	36.75%	70.10%
NN_{ROC}	40.62%	75.06%

V. Conclusion

Neural network is a very powerful technique for learning complex nonlinear functions that can effectively capture the higher-order nonlinearities inherent in the data. But standard neural network learning algorithms are limited by the simple loss functions being used. In this paper, we have proposed a learning approach that trains a neural network to directly optimize the ROC area loss, a ranking-based loss function. We have demonstrated our approach in retrieving actions from still images and videos. Our experimental results show that our proposed approach is much more effective in retrieving images/videos than traditional neural networks trained with simple softmax loss functions. We also demonstrated superiority of our approach over an SVM-based approach that offers a linear model.

Acknowledgment

This work was supported by NSERC and the University of Manitoba Research Grants Program (URGP). We gratefully acknowledge the support of NVIDIA Corporation with the GPU donations used in this research.

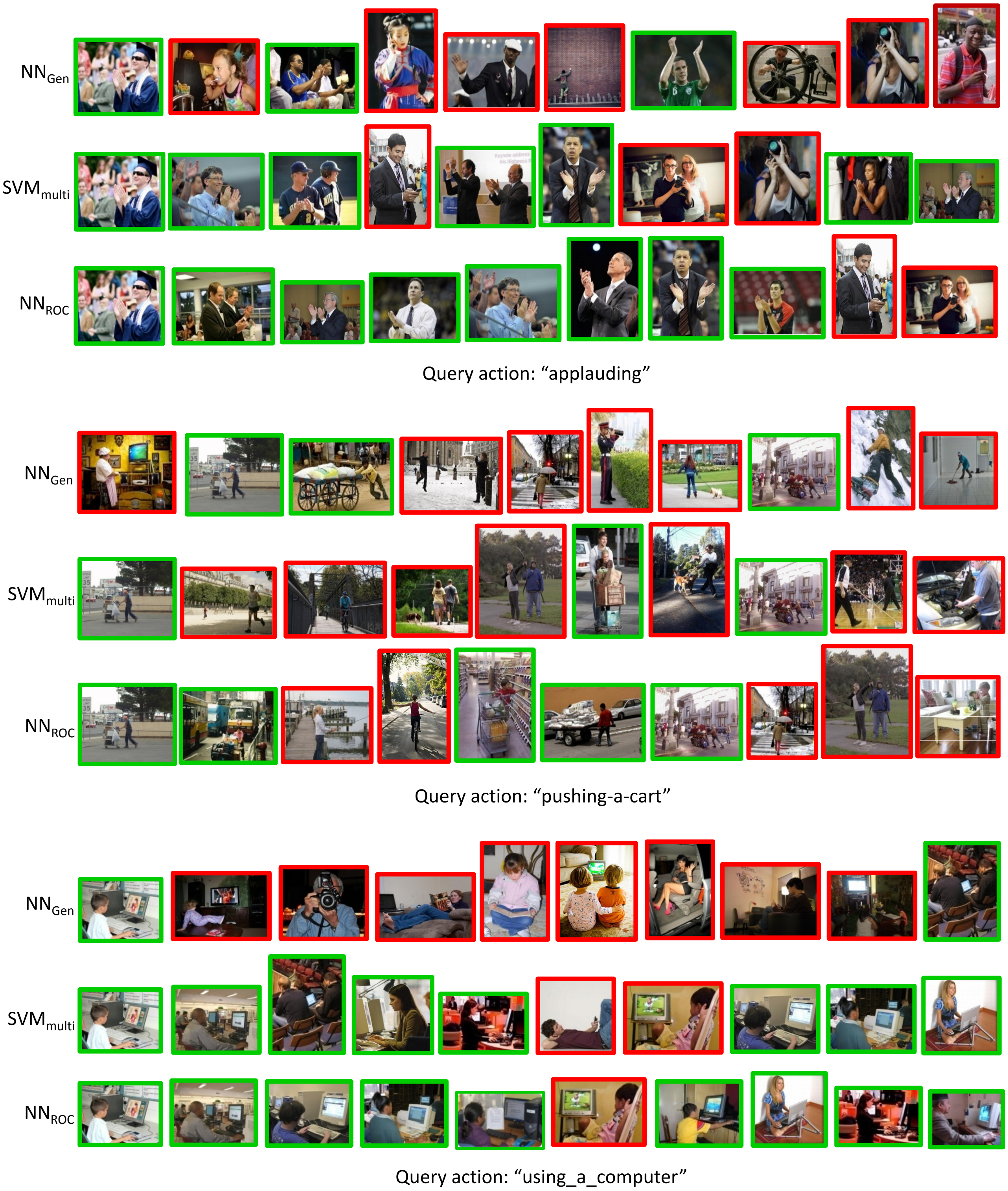
NNGen
SVMmulti
NNROC
Query action: “applauding”
NNGen
SVMmulti
NNROC
Query action: “pushing-a-cart”
NNGen
SVMmulti
NNROC
Query action: “using_a_computer”

Figure 4. Top ten retrieval results (from left to right) of our approach and the baseline methods for three different queries on the Stanford 40 actions dataset [18]. For each query, first row and second row refer to the retrieval results of the two baseline methods of NN_{Gen} and $\text{SVM}_{\text{multi}}$, respectively, while the third row refers to the retrieval results of our proposed approach NN_{ROC}. Images bounded in green boxes indicate relevant examples, while those bounded in red boxes are irrelevant.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1097–1105.

[2] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[3] T. Joachims, "A support vector method for multivariate performance measures," in *Proceedings of the 22Nd International Conference on Machine Learning*, ser. ICML '05, 2005, pp. 377–384.

[4] R. Caruana, S. Baluja, and T. M. Mitchell, "Using the future to sort out the present: Rankprop and multitask learning for medical risk evaluation," in *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, November 27-30*, 1995, pp. 959–965.

[5] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, "Learning to rank using gradient descent," in *Proceedings of the 22Nd International Conference on Machine Learning*, ser. ICML '05. ACM, 2005, pp. 89–96.

[6] A. Herschtal and B. Raskutti, "Optimising area under the roc curve using gradient descent," in *Proceedings of the Twenty-first International Conference on Machine Learning*, ser. ICML, 2004.

[7] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *J. Mach. Learn. Res.*, vol. 6, 2005.

[8] B. Mcfee and G. Lanckriet, "Metric learning to rank," in *Proceedings of the 27th annual International Conference on Machine Learning (ICML)*, 2010.

[9] T. Joachims, T. Finley, and C.-N. J. Yu, "Cutting-plane training of structural svms," *Mach. Learn.*, 2009.

[10] Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon, "Adapting ranking svm to document retrieval," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '06. ACM, 2006, pp. 186–193.

[11] Y. Bai, K. Yang, W. Yu, W. Ma, and T. Zhao, "Learning high-level image representation for image retrieval via multi-task DNN using clickthrough data," *CoRR*, vol. abs/1312.4740, 2013.

[12] Y. Bai, W. Yu, T. Xiao, C. Xu, K. Yang, W.-Y. Ma, and T. Zhao, "Bag-of-words based deep neural network for image retrieval," in *Proceedings of the ACM International Conference on Multimedia*, ser. MM '14, 2014.

[13] A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "A baseline for visual instance retrieval with deep convolutional networks," *CoRR*, vol. abs/1412.6574, 2014.

[14] J. Y.-H. Ng, F. Yang, and L. S. Davis, "Exploiting local features from deep networks for image retrieval," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2015.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014.

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

[17] T. Joachims, *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 2002.

[18] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. J. Guibas, and F.-F. Li, "Human action recognition by learning bases of action attributes and parts," in *ICCV'11*, 2011, pp. 1331–1338.

[19] 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.

[20] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," *CoRR*, vol. abs/1412.4564, 2014.

[21] 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.

[22] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *CoRR*, vol. abs/1310.1531, 2013. [Online]. Available: http://arxiv.org/abs/1310.1531

[23] T. Du, B. Lubomir, F. Rob, T. Lorenzo, and P. Manohar, "Learning spatiotemporal features with 3d convolutional networks," *arXiv preprint arXiv:1412.0767*, 2015.