OBJECT AND TEXT-GUIDED SEMANTICS FOR CNN-BASED ACTIVITY RECOGNITION

*Sungmin Eum [†§], *Christopher Reale, Heesung Kwon[†], Claire Bonial [†], Clare Voss[†]

[†]U.S. Army Research Laboratory [§]Booz Allen Hamilton Inc.

ABSTRACT

Many previous methods have demonstrated the importance of considering semantically relevant objects for carrying out video-based human activity recognition, yet none of the methods have harvested the power of large text corpora to relate the objects and the activities to be transferred into learning a unified deep convolutional neural network. We present a novel activity recognition CNN which co-learns the object recognition task in an end-to-end multitask learning scheme to improve upon the baseline activity recognition performance. We further improve upon the multitask learning approach by exploiting a text-guided semantic space to select the most relevant objects with respect to the target activities. To the best of our knowledge, we are the first to investigate this approach.

***Index Terms*—** text-guided, CNN, activity recognition, object recognition, word2vec

1. INTRODUCTION

In recent years, a significant amount of research in the computer vision community has focused on human activity recognition. The objective of this research is to be able to automatically recognize and understand what humans depicted in a video are doing. Among many approaches, a group of authors have shown that recognizing human activities can be better performed by incorporating various external information. One of the popular and effective external information has been proved to be exploiting relevant objects which reside in different human-involved activities or events [1, 2, 3, 4, 5].

Following the similar spirit, we also attempt to tackle the problem of human activity recognition by incorporating the relevant object information, but with a novel text-guided semantics to constrain the relevant objects with respect to the target activities. Technically, we seek to train a deep convolutional neural network (CNN) using a multitask learning [6, 7, 8, 9] scheme, where activity recognition and object recognition is learned concurrently. This approach helps the overall network training in two ways. First, it allows us to exploit a large object recognition dataset to boost the amount of training data we have. Second, it allows us to incorporate general knowledge from text about the target activities that may not be fully apparent from the training videos, which in turn, improves the overall activity recognition performance.

One advantageous aspect of our approach in learning the task of object recognition alongside the main task (i.e., activity recognition) is that we do not seek to localize the objects within the human activity scenes, but instead harvest the object information by simply using a shared network between the two tasks. As we are enforcing two separate loss functions, although using a shared network, we can make use of the vast amount of object recognition dataset (e.g., ImageNet [10]) and save the hassle of having to provide object bounding box ground truth. Moreover, this enables us to use much deeper networks without overfitting, thus achieving higher recognition rates.

In order to effectively incorporate the "relevant" objects information into the training of activity/object recognition network, we devise a novel approach called 'Text-guided Relevance Analysis (TRA)' where we analyze the relationship between the activities and the objects within a text-guided semantic space. We make use of the textual labels for the activities as well as the objects such as "tennis swing", "blow candles", "ball" which are provided by the original dataset. In order to project these textual labels into a common semantic space, we use Word2Vec [11] word embedding. Based on the fact that semantically similar words are likely to be embedded in close locations in the semantic vector space, we approximate the relevance between the objects and the target activities within this space.

We experimentally show that incorporating the objects and applying TRA for relevant object selection are both effective in outperforming the baselines for human activity recognition.

2. OUR METHOD

2.1. Incorporating Object Recognition with Activity Recognition in Multitask Learning

Previous approaches have demonstrated that being able to detect or recognize objects within an image can improve recognition of relevant events and activities in that image [4, 2]. We take a similar approach in exploiting the object information but with two major novel aspects. First, we introduce a practical way of training and enhancing the activity recog-

*These authors contributed equally to this work

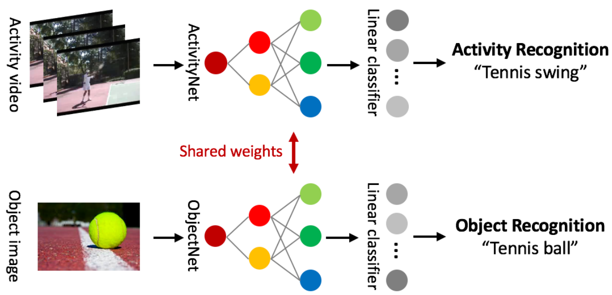

Fig. 1: **Object-incorporated activity recognition network architecture.** Colored region of the network is being shared between the ActivityNet and the ObjectNet, while the grayscale portions (softmax linear classifiers) are learned separately to handle the specific tasks of activity and object recognition.

nition network by carrying out the multitask learning with the object recognition network. Moreover, unlike the previous approaches, we do not attempt to localize or identify the objects within the target domain (in our case, activity recognition) but train the network to perform the task of object recognition using a totally different dataset (ImageNet). This bolsters the amount of training data for the overall network, and at the same time, removes the need for manually annotating/detecting the relevant objects in the target videos. As shown in Figure 1, we share the weights in all the layers of the network between the two tasks except the task-specific softmax classifiers.

Datasets we use for the training (UCF101 [12] and ImageNet [10]) are only annotated for each single task (i.e., videos frames for activity recognition and ImageNet images for object recognition). Thus, we design the network so that each data sample is directly associated with the loss function for the corresponding task. However, as we ground our method in the relevance of the two tasks, all the layers except the softmax layer are being shared between the two tasks.

We can view our multitask learning approach as an extension of the standard finetuning strategy (Figure 2a). In training our network we learn the parameter weights for both the activity recognition (ActivityNet) and the object recognition (ObjectNet) by finetuning from the network pretrained for the task of the object recognition (ObjectNet) as shown in Figure 2b. The continuation of the incorporation of gradients from the object recognition loss acts as a regularization for the overall network parameters, preventing them from overfitting to the activity recognition task. As our pretrained ObjectNet, we have used the network which was trained to classify 1000 object classes assigned by the ImageNet Challenge [10].

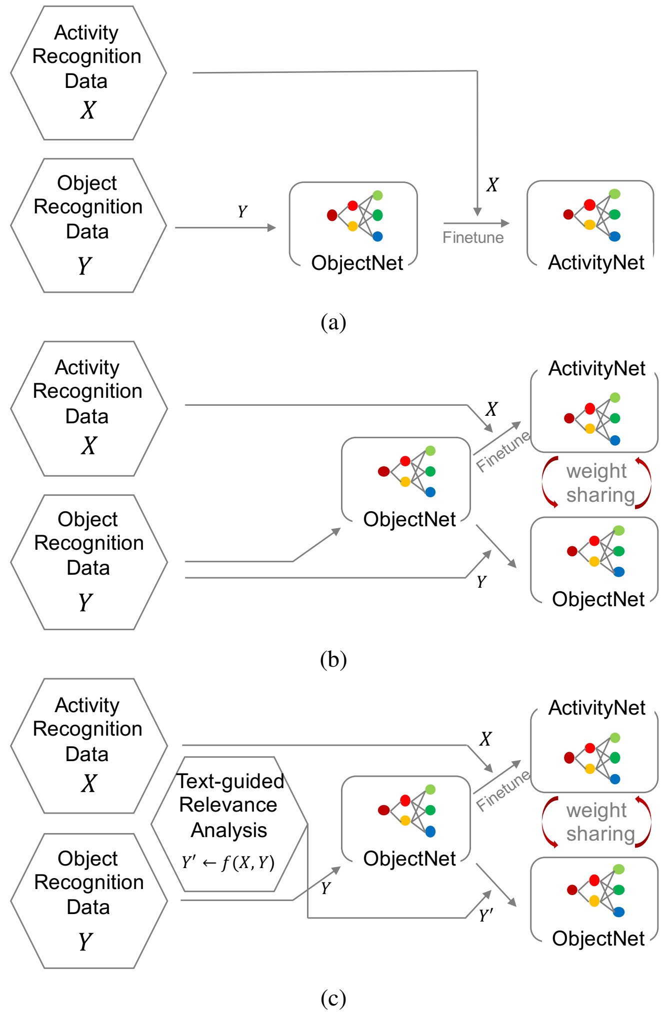

Fig. 2: **Activity recognition network training strategies.** ActivityNet and ObjectNet refer to CNNs for recognizing the activities or objects, respectively. (a) Baseline activity recognition network which finetunes from the pretrained ObjectNet (b) Object-incorporated activity recognition network (c) Text-guided, object-incorporated activity recognition network.

2.2. Leveraging the text-guided semantic space

The object-incorporated activity recognition network introduced in Section 2.1 uses all the objects from the ImageNet dataset to learn the ObjectNet, and thus solely relies on the capability of the multitask network learning process to harvest the necessary information about the objects with respect to the activities. We seek to further improve upon our object-incorporated activity recognition network by exploring the following questions: Which objects are more important and indicative for certain activities? Would selecting this subset of objects help improve activity recognition?

Our strategy is to refine the original object dataset before proceeding into the network training by selecting the most relevant set of objects with respect to the activities in the tar-

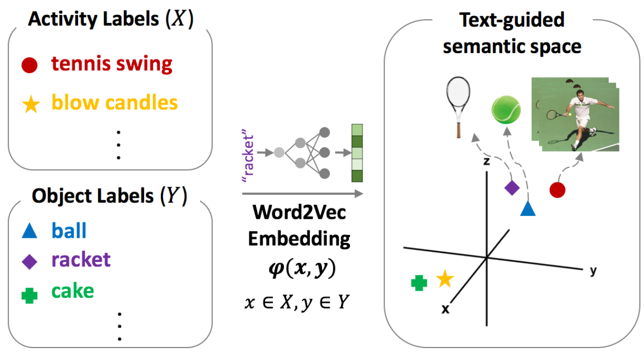

Fig. 3: **Text-guided Relevance Analysis in the semantic space.** Closely related activities and objects are aggregated in the text-guided semantic space using the Word2Vec embedding. In our experiments, object labels correspond to ImageNet class labels.

get domain. To select the most relevant objects, we carry out what we call 'Text-guided Relevance Analysis (TRA)' where we compute the similarity between the textual labels of the activities and those of the ImageNet objects within a semantic vector space. We exploit the the textual labels which are originally provided from both datasets (*UCF101* and *ImageNet*).

In TRA, we use Word2Vec [11] embedding to project the textual labels to the semantic vector space. Word2Vec embeds words and phrases into a vector space based on their usage in a large text corpora. Words that are used in similar contexts will be embedded closer together in the vector space. An illustration of the text-guided semantic space is shown in Figure 3, where the activity label "tennis swing" is closely embedded with the object labels "ball" and "racket".

Assuming $\omega(\cdot)$ as the embedding learned by Word2Vec, we approximate the relevance between a target activity x and an ImageNet class y with the cosine similarity of their vector space representations as follows:

$$\varphi(x, y) = \frac{\omega(x) \cdot \omega(y)}{\|\omega(x)\|_2 \|\omega(y)\|_2}. \tag{1}$$

We then compute the overall relevance κ of an ImageNet class $y \in Y$ to the set of target activities X as the sum of the relevances of y to each activity $x \in X$,

$$\kappa(y|X) = \sum_{x \in X} \varphi(x, y). \tag{2}$$

Once we acquire $\kappa(\cdot)$ for all ImageNet classes, we select the most relevant classes (those whose relevance score is numerically highest) to be used for training the "text-guided, object-incorporated activity recognition network". This overall process of TRA (See Figure 2c), can be considered a dataset refinement procedure $f(\cdot)$ for the original object recognition dataset Y as $Y' = f(X, Y)$:

Table 1: **Highly ranked ImageNet classes using TRA.** Top 3 ImageNet classes for a set of selected activity classes.

Activity (UCF101)	1st	2nd	3rd
ApplyLipstick	lipstick	mascara	nail polish
Biking	bicycling	cycling	motorcycle
Knitting	quilting	needlework	knit
MilitaryParade	soldier	Marine	admiral
cliffDiving	cliff	dive	ledge

$$Y' = f(X, Y) = \{y : rank(\kappa(y|X)) \leq m, y \in Y\}, \tag{3}$$

where $rank(\kappa(y|X))$ indicates the rank in descending order among all $\kappa(y|X)$ such that $x \in X$ and $y \in Y$, while m is the number of selected objects within Y. Based on an empirical analysis, we selected, for our image input dataset (identified as Y in Figure 2c), the images that have text-labels for 1000 objects (m = 1000) for training the final version of the network. In Table 1, we introduce some samples of highly ranked object (ImageNet) classes with respect to the activity (UCF101) classes acquired by the TRA.

3. EXPERIMENTS

3.1. Experimental details

Preprocessing the data. First, we subtract a mean pixel from each pixel in the image. Then we select a random window from the target frame. The window's width and height are randomly and independently selected (from a uniform distribution) to be between 168 and 256 pixels. Once the width and height are selected, the location of the window within the image is selected at random (again, from a uniform distribution). Finally, the window is resized to 224×224 pixels and fed into the network. The random window selection process helps to generate more variation in the training data to reduce the risk of overfitting. For the ImageNet images, we still subtract the pixel mean, but select a sub-image by simply choosing a random 224×224 window from the image. We can use a simpler window selection with ImageNet because it contains many more images which are uncorrelated unlike the video frames which are highly correlated.

Network architecture setting. We use the ResNet [13] architectures (ResNet 50, 100, and 152) which has recently demonstrated the state-of-the-art performance in various applications. This is in contrast to previous approaches which use shallower networks. Our multitask approach acts as a regularization, enabling us to use the deeper, better-performing

ResNet networks. All networks are initialized by pretraining on the 1000 ImageNet challenge classes.

We incorporate the Temporal Segment Network (TSN) [14] approach in training our networks which is known to capture long-term temporal information. We have empirically determined the optimal number of segments to be three, and thus the size of the activity recognition portion of the batch was set to be a multiple of three. For example, when training our ResNet 50 network, total batch size is 64. Ideally, we would split it evenly between the two network streams (32 each). However, as 32 is not a multiple of three, we use 33 activity recognition samples and 31 ImageNet samples.

Training strategy. We train our networks with stochastic gradient descent on single GPU (NVIDIA TITAN X) system. Due to the depths of the networks used and the memory limitations of the GPU (12 GB), we were forced to use small batch sizes of 64, 48, and 32 frames/images for ResNet 50, 101, and 152, respectively. When training in the multitask setting, we split the batch size between activity recognition frames and ImageNet images. We found that splitting the batch approximately evenly between the two (i.e., giving equal weight to the two objectives) provided the best performance. Note that, in generating each batch for training, activity recognition frames and the ImageNet images are pulled out in a random fashion, i.e., no correlation or relevance metric is used to forcefully pull the images from two different datasets.

When training ResNet 50, we initialize the learning rate to .001. We divide it by 10 after 10k and 13k iterations and train for 15k iterations in total. Due to the smaller batch sizes, we initialize the learning rates for the ResNet 101 and 152 to .0005. For ResNet 101, we divide it by 10 after 13k and 18k iterations and train for 20k iterations in total. For ResNet 152, we divide it by 10 after 28k and 36k iterations and train for 40k iterations in total. Weight decay was set as .0001. During training of all three architectures, we place dropout layers just before the final softmax classifiers. Dropout rate is set as .25.

At test time, we use the standard approach of generating predictions for 25 evenly spaced frames. For each frame, we generate predictions from 10 different 224×224 pixel windows: one from each corner of the frame, one from the center of the frame, and then a horizontally flipped version of each of those. For each video, 250 probability predictions are made for each of the classes. We average them and predict the activity with the highest value. In this work, we use a Word2Vec model which was trained on an internal Google dataset of news articles containing a billion words [11]. One may also experiment with other types of word embedding models such as gloVe [15], fastText [16], or LexVec[17, 18, 19].

3.2. Performance Evaluation

We evaluate the performance of our approach on the UCF 101 benchmark dataset [12]. We have used the ResNet to construct the baseline architecture for both the activityNet and the objectNet (See Figure 1). The experiments were carried out on three different ResNet networks (ResNet 50, 101, and 152) under three different settings (baseline, object-incorporated, text-guided + object-incorporated). The baseline approach is the standard method without multitask learning. For the object-incorporated multitask approach, we randomly selected 1000 ImageNet classes to learn the objectNet. The text-guided + object-incorporated approach uses Word2Vec to select the 1000 most relevant ImageNet classes as described in Section 2.2.

Table 2: **Performance comparison.** Accuracy on UCF 101 Dataset. See Figure 2 for the different training strategies.

	Baseline	object incorp.	object incorp. + text-guided
Multitask?	No	Yes	Yes
ResNet 50	81.3	84.0	85.1
ResNet 101	82.6	85.3	86.9
ResNet 152	83.1	86.0	**87.5**
TSN [14]	85.7	-	-

From the results shown in Table 2, it is clear that using the ResNet networks with the baseline approach provides worse performance than the state-of-the-art method (TSN [14]). This is because the architecture used in [14] uses shallower networks which are not as prone to overfitting. When we incorporate the object information in a multitask learning scheme (object-incorporated), the performance increases close to the current state-of-the-art. And finally, when we exploit the text-guided supervision on top of the object incorporation, we are able to outperform the state-of-the-art.

4. CONCLUSION

We have introduced a novel way of constructing an object-incorporated and text-guided CNN to better handle the task of video-based human activity recognition. We do this by leveraging the text-guided semantic space to select the most commonly associated objects with respect to the target activities. We then train the network to recognize the target activities as well as the selected set of objects by exploiting a shared network and a multitask learning approach. We have experimentally verified that the strategies of incorporating objects for activity recognition and text-guided object selection are both effective in improving the performance for the human activity recognition. In the future, we are seeking to incorporate the background scenes into our framework as it also carries significant semantic information for the activities.

5. REFERENCES

[1] Nazli Ikizler-Cinbis and Stan Sclaroff, "Object, scene and actions: Combining multiple features for human action recognition," in *ECCV 2010*, Kostas Daniilidis, Petros Maragos, and Nikos Paragios, Eds., Berlin, Heidelberg, 2010, pp. 494–507, Springer Berlin Heidelberg.

[2] Sungmin Eum, Hyungtae Lee, Heesung Kwon, and David Doermann, "IOD-CNN: Integrating object detection networks for event recognition," in *International Conference on Image Processing (ICIP)*, 2017.

[3] Tanvi S. Motwani and Raymond J. Mooney, "Improving video activity recognition using object recognition and text mining," in *European Conference on Artificial Intelligence (ECAI)*, 2012.

[4] Victor Escorcia and Juan Carlos Niebles, "Spatio-temporal human-object interactions for action recognition in videos," in *International Conference on Computer Vision Workshop*, 2013.

[5] Hyungtae Lee, Sungmin Eum, Joel Levis, Heesung Kwon, James Michaelis, and Michael Kolodny, "Exploitation of semantic keywords for malicious event classification," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[6] Rich Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, July 1997.

[7] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D Castillo, and Rama Chellappa, "An all-in-one convolutional neural network for face analysis," in *Automatic Face & Gesture Recognition (FG), 2017 12th IEEE International Conference on*. IEEE, 2017, pp. 17–24.

[8] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja, "Robust visual tracking via multi-task sparse learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2012, pp. 2042–2049.

[9] Xiao-Tong Yuan, Xiaobai Liu, and Shuicheng Yan, "Visual classification with multitask joint sparse representation," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 4349–4360, 2012.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2009, pp. 248–255.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems (NIPS)*, 2013, pp. 3111–3119.

[12] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 770–778.

[14] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 20–36.

[15] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[16] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov, "Bag of tricks for efficient text classification," *CoRR*, vol. abs/1607.01759, 2016.

[17] Alexandre Salle and Aline Villavicencio, "Incorporating subword information into matrix factorization word embeddings," *CoRR*, vol. abs/1805.03710, 2018.

[18] Alexandre Salle, Marco Idiart, and Aline Villavicencio, "Enhancing the lexvec distributed word representation model using positional contexts and external memory," *CoRR*, vol. abs/1606.01283, 2016.

[19] Alexandre Salle, Marco Idiart, and Aline Villavicencio, "Matrix factorization using window sampling and negative sampling for improved word representations," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016, pp. 419–424.