

Learning Semantic Feature Map for Visual Content Recognition

Rui-Wei Zhao¹, Zuxuan Wu², Jianguo Li³, Yu-Gang Jiang^{1*}

¹ Shanghai Key Lab of Intelligent Information Processing, School of Computer Science, Fudan University

² University of Maryland, ³ Intel Labs China

rwzhao14@fudan.edu.cn, zxwu@cs.umd.edu, jianguo.li@intel.com, ygj@fudan.edu.cn

ABSTRACT

空间的 线索
The spatial relationship among objects provide rich clues to object contexts for visual recognition. In this paper, we propose to learn 语义 Semantic Feature Map (SFM) by deep neural networks to model the spatial object contexts for better understanding of image and video contents. Specifically, we first extract high-level semantic object features on input image with convolutional neural networks for every object proposals, and organize them to the designed SFM so that spatial information among objects are preserved. To fully exploit the spatial relationship among objects, we employ either Fully Convolutional Networks (FCN) or Long-Short Term Memory (LSTM) on top of SFM for final recognition. For better training, we also introduce a multi-task learning framework to train the model in an end-to-end manner. It is composed of an overall image classification loss as well as a grid labeling loss, which predicts the objects label at each SFM grid. 广泛的 大量的
The results on CCV benchmark demonstrate the robustness and generalization capability of the proposed approach. 鲁棒性和泛化能力

CCS CONCEPTS

• Computing methodologies → Visual content-based indexing and retrieval; Image representations;

KEYWORDS

语境融合
image representation; contextual fusion; image classification; video classification

1 INTRODUCTION

Visual recognition is a fundamental and challenging topic for understanding image and video contents. Recent years have witnessed major breakthroughs in this field with high capacity Convolutional Neural Networks (CNN) models that learn discriminative feature representations in an end-to-end fashion [5, 15, 30]. A typical CNN

model consists of several convolutional layers topped by a few fully-connected layers for classification, operating on normalized images with single objects [15, 34]. Its success mainly stems from its ability to progressively learn discriminative features from low-level (e.g., texture and edges) to high-level (e.g., parts). However, when there are multiple objects in an image, directly applying CNN models though feasible is not optimal since rich underlying contextual information among objects is not utilized during the reasoning procedure. 源于 推理过程

To explore these contextual relationships of multiple objects, current deep learning frameworks usually couple CNN models with semantic region proposals. In this way, multiple object-level features are generated for a single image. Unlike the object detection tasks which operate on these separate object features to localize objects, it is desirable that these features are aggregated into unified representations for recognition tasks. Currently, most existing works adopt simple mean/max pooling of object features across feature dimensions to integrate multiple features, or employ certain popular feature encoding methods like Fisher Vector [43] or VLAD [12]. However, the original rich spatial contextual clues like the shapes and locations of different objects are discarded in the process of generating an image-level feature for recognition. This is clearly not sufficient for effective visual recognition. We argue that a good image representation should not only be discriminative and holistic but also contain details about how objects in the image interact with each other serving as context. 联结 冗余的

To address the above limitations, in this paper, we propose to learn a generic semantic representation in a multi-task framework for general visual recognition tasks such as image and video classification. Figure 1 provides an overview of the proposed framework. Given an input image, we first generate dense object proposals with algorithms like edge box [47] and compute high-level semantic features with a CNN model for each proposal. Then, we reorganize these object-level semantic features into a contextual representation named Semantic Feature Map (SFM), in the form of a three dimensional array that preserves the spatial layout and interaction information among multiple objects. We design two modules (i) a Fully Convolutional Networks (FCN) and (ii) a Long-Short Term Memory (LSTM) operating on the SFM for image recognition. The FCN model bridges the gap between object-level features and class labels using stacked convolutional layers, which not only increases the discriminative power of features but also preserves spatial layout information. The LSTM model explicitly considers SFM grids as a sequence and take each spatial grid as an individual time step. By learning the mapping between grids in a recurrent manner, the spatial relationships among objects can thus be exploited. Furthermore, to fully explore the spatial information of objects, we additionally incorporate the grid labeling loss, which trains the network to learn object labels at pre-defined spatial grids by creating a grid label map, 个别时间步 反复的方式

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
MM'17, October 23–27, 2017, Mountain View, CA, USA.
© 2017 ACM. ISBN 978-1-4503-4906-2/17/10...\$15.00
DOI: <https://doi.org/10.1145/3123266.3123379>

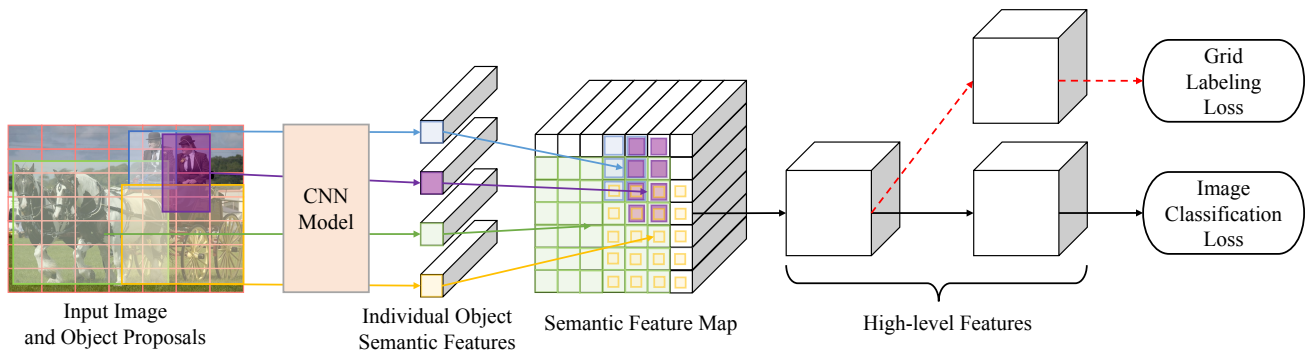


Figure 1: Illustration of the proposed framework. Object proposals are first extracted on the input image. Differently colored boxes represent different high-level object semantic features extracted by deep CNN models. They are spatially pooled into the feature map grids in the middle, where spatial contextual information of the original objects are preserved. Additional FCN/LSTM layers can be appended to learn higher-level features and make final predictions. Multi-task loss functions are employed to train the classification system in an end-to-end style. (Best viewed in color.)

in a multi-task learning framework. Finally, the whole architecture is trained in an end-to-end fashion.

We conduct extensive experiments on Pascal VOC and MSCOCO benchmarks to evaluate the proposed method for image classification and very competitive results are obtained. To further investigate the effectiveness and generality of our approach, we transfer our learned semantic feature map to video data and test its performance in video classification. We observe that the learned feature representation successfully outperforms state-of-the-art methods on the CCV video dataset.

优于最新方法

The major contributions of this work are summarized as follows.

- (1) We propose to learn semantic feature map, a 紧凑的 and discriminative image representation directly derived from deep object features, which exploits the rich contextual information among multiple objects within an image.
- (2) We utilize both FCN and LSTM modules to model the SFM representation for recognizing targeted classes. For the LSTM module, we consider the SFM grids to be a sequence and each grid as a time step, upon which the LSTM is applied to explicitly model spatial context information.
- (3) We introduce the grid labeling loss, which trains the network to learn object labels at each spatial grids with the created grid label map, as an additional supervision in the multi-task learning framework.
- (4) We conduct extensive experiments to evaluate the proposed feature representation on popular image classification benchmarks. The results showed that our method can obtain competitive results. We also conduct experiments to verify that our learned SFM can be transferred to benefit video classification.

The rest of this paper is organized as follows. Section 2 briefly reviews the related works. Section 3 elaborates the proposed feature representation and the related multi-task learning method. In Section 4, we conduct experiments to evaluate our proposed framework. Conclusions are drawn in Section 5.

2 RELATED WORKS

Image and video classification problems have been extensively studied with much efforts paid to the design of high quality image feature representations. We focus the review on recent works related to our proposed approach.

Deep CNN features. Deep CNN networks are the most popular and successful models for various visual recognition tasks in recent years. Since the famous AlexNet [15] appeared in 2012, multiple improved network structures have emerged, including VGGNet [32], GoogLeNet [34] and ResNet [10]. It was found that the deeper models managed to generate better feature representations with higher discriminative power. These models continuously increased the evaluation results on popular image classification benchmarks. The powerful CNN models can be naturally applied to frame feature extraction for video classification task as well. In [45], the authors systematically studied the performance of image-based video recognition using features from different layers of deep models together with multiple kernels for classification. They demonstrated that the off-the-shelf CNN features coupled with SVMs could obtain decent recognition performance. Simonyan and Zisserman [31] proposed a two stream CNN model to extract features on RGB video frames and optical flows for action recognition. Wu et al. [41] proposed a hybrid deep learning framework combining CNN frame features and LSTM temporal modeling. Note that only global image CNN features are considered in these methods.

Deep object features. Recently, combining advanced CNN models with semantic regions is widely used in visual recognition systems since image or video classes are strongly related to the objects within it. Gong et al. [8] performed orderless pooling over all extracted CNN region features sampled at differently scaled sliding windows. Quite a number of object proposals methods were proposed in the past few years, such as selective search [35], edge box [47], BING [3], etc. Based on these region proposals, Girshick et al. [7] invented R-CNN to successfully combine CNN with local region information for object detection. In [6], an improved

Fast R-CNN model was invented where ROI pooling was used to more efficiently extract fix sized object features from objects of various original sizes, which could be regarded as a kind of SPP on the ROI convolutional layer feature maps [9]. After that, Ren et al. [29] invented the RPN to generate the object proposals within the detection networks in an end-to-end manner, where the ROI pooling module was also employed. Besides, there are other CNN models proposed in pursuit of extreme efficiency like YOLO [28] and SSD [21]. The inside-outside networks (ION) [1] is also related. It uses RNN to learn the contextual feature representation on the detected regions at multiple scale. Note that in ION, the contextual information is learned for individual detected objects, while our proposed method tries to exploit the contextual information on unified feature map describing all objects semantic information. These methods are mainly designed for image object detection while we aim to utilize the rich object clues for holistic image recognition.

Combining object features. The detected object-level features are found to be useful for the whole image classification problem. Generally, various extracted object features are required to be combined together into a unified image-level representation so that it could be fed into the classifiers. When combining the object features for classification, most current methods lacks of paying enough attention to the contextual information utilization among different object features. For instance, Oquab et al. [27] used average pooling method to combine the individual object scores. Yang et al. [43] used fisher vector to encode the object CNN features. Luo et al. [23] used a pre-trained SVM to select important image regions and fused the selected region scores with computed region weights. Mettes et al. [25] fused the image parts results with weighted sum pooling in an boosted manner. [37] fused the predictions of image crops with max pooling scheme. Wei et al. [38] proposed a CNN based method called Hypotheses-CNN-Pooling (HCP) which removed some regions according to their sizes, aspect-ratios and confidence scores. Then it clustered the regions into groups and keeps only one representative region in each group, which could also be regarded as a variant of max pooling scheme. Zhao et al. [46] adopted a multi-scale max pooling scheme on different sized objects selected by gating networks. Wang et al. [36] used RNN coupled with CNN feature to exploit label interactions for multi-label classification. Note that although these methods exploited the object-level features, rich context information like shape, location, overlaps were not well considered in the fusion process. Another related work is the ObjectBank [18]. It applied multi-scale SPP on object detection responses derived from traditional features on the global image. Our method is different from ObjectBank in that we spatially pool the individual extracted deep region features back into a fix sized tensor map.

Object features in video. For video data, Sun et al. [33] adopted the prediction scores of CNN models pre-trained from ImageNet dataset as the object detection scores. They proved in their experiments that the object clue is helpful to improve the video classification results. In [39], the authors further extended the object features with scene features in video for zero-shot learning problems. However, the object scores were generated based on holistic input image in these works. They did not rely on the region features extracted

from explicitly detected objects, while our proposed approach fuses deep feature on multiple image regions.

3 OUR APPROACH

In this section, we first describe the generation of the semantic feature map, and then introduce how to utilize it in FCN and LSTM based classification networks structures. Finally, we present a multi-task training scheme to learn the best parameters.

3.1 Semantic Feature Map

Given an input image I whose height and width are H_I and W_I respectively, we utilize a region proposal algorithm to generate a collection of N bounding boxes. Let us denote the collection of generated bounding boxes for the image as

$$\mathcal{B} = \{B_1, \dots, B_N\} \quad (1)$$

where each bounding box B_n is described by the 4 dimensional tuple as

$$B_n = (y_n, x_n, h_n, w_n) \quad (2)$$

Here the tuple values (y_n, x_n) are the top-left coordinates of the bounding box and (h_n, w_n) are its height and width. Note that the quantity of the generated region proposals (N) depends on the image content and the adopted proposal generation methods. For example, commonly used object proposal algorithms like selective search [35], edge boxes [47] and RPN [29] can usually generate several thousand box proposals on an image. In this work, we adopted the top 1000 ranked proposals on the image returned by the edge box algorithm, which was widely used in recent related works [38, 46]. Given the proposal bounding boxes, we can further denote the collection of the cropped region image data as

$$\mathcal{R} = \{R_1, \dots, R_N\} \quad (3)$$

where each item $R_n = I[y_n : y_n + h_n - 1, x_n : x_n + w_n - 1]$ stands for the rectangle area indexed from row y_n to $y_n + h_n - 1$ and column x_n to $x_n + w_n - 1$ of the original image data.

Then we need to extract deep features from these image ROIs. We follow [6] to forward the whole image data with the collection of bounding boxes through shared convolutional layers to produce fix sized feature maps for every ROIs, in form of 3D arrays of shape $H \times W \times C$, by the ROI max pooling operation. Here H is the object feature map height; W is the feature map width; and C is the feature map channel. Take the VGG net based model as an example, the shape of the extracted object feature size is $7 \times 7 \times 512$. Conceptually, ROI pooling divides the $h \times w$ ROI window into an $H \times W$ grid of sub-windows of approximate size $h/H \times w/W$ and then max-pooling the values in each sub-window into the corresponding output grid cell [6]. Note that a couple of other R-CNN related alternatives [7, 29] can be used to extract to object features. We adopt the method in [6] due to its advantages at efficiency and capabilities of working with any kind of object proposals. These extracted object features can be further transformed into higher-level semantic descriptions $\mathbf{x}_n \in \mathcal{R}^{C'}$ by the appended fully connected layers. Like in VGG net, the $C' = 1000$ dimensional attribute features can be obtained by the 8th fully connected layer. Different from the ROI pooling in generating individual region features of fixed shape, our approach aims to merge all the ROI features back into a unified high-level semantic feature map of fixed shape with the contextual information

among objects preserved. We formulate our generated image-level feature representation as a 3D feature map of shape $H' \times W' \times C'$, where H' , W' and C' are the height, width and channels (or number of semantic attributes) of the new feature map. Ideally, we hope that the spatial and contextual information are stored in the spatial plane and the semantic information is kept in the C' channels.

To achieve the above purpose, the idea used in our approach is inspired by SPP [9]. Take an arbitrary bounding box B_n and its extracted semantic features \mathbf{x}_n as an example. We hope to map the high-level semantic feature values of \mathbf{x}_n into the target unified feature map \mathbf{z} according to the object location. Mathematically, the mapped response area is a scaled box $B'_n = (y'_n, x'_n, h'_n, w'_n)$ calculated by

$$y'_n = H'/H_I y_n \quad (4)$$

$$x'_n = W'/W_I x_n \quad (5)$$

$$h'_n = H'/H_I h_n \quad (6)$$

$$w'_n = W'/W_I w_n \quad (7)$$

For each column (y', x') of the box area B'_n in the unified feature map \mathbf{z} , we fill in the column values $\mathbf{z}[y', x', :]$ with the object semantic features \mathbf{x}_n . When multiple object features are mapped into a same location with overlaps, we keep the element-wise max values of the features in every feature dimension across all corresponding object features. Mathematically, the feature mapping function over all object regions can be expressed as

$$\mathbf{z}[y', x', c'] = \max \{\mathbf{x}_n[c']; (y', x') \in B'_n, n = 1, \dots, N\} \quad (8)$$

The illustration of the feature map generation is shown in the center of Figure 1, where bins of different colors represent high-level object semantic features of different objects. They are spatially pooled into the semantic feature map (shown in the center of the figure), where spatial context information of the original objects are preserved.

3.2 Contextual Feature Modeling

To further bridge the semantic gap for recognizing targeted classes, transforming these object features \mathbf{x}_n to higher space is required, which can be simply done by appending convolutional or fully connected layers. For example, in [45] the object features were further encoded into 4096 dimensional mid-level features and 1000 dimensional attribute features by fully connected layers. However, this will result in the loss of spatial information of objects and incur a large amount of parameters; thus we choose convolutional layers to improve the discriminative power of the semantic feature, upon which a classifier can be appended. It is worth noting that the classifier is expected to fully exploit the contextual information embedded in the feature map. Therefore, we introduce FCN and LSTM to address this problem in the following.

FCN. The FCN structure consists of several stacked convolutional layers which increase the size of receptive fields and preserve spatial layout information [22]. It was originally used in image segmentation tasks to output pixel-level prediction. The FCN based classifier in our framework is illustrated in Figure 2. When the filter size in the appended convolution layer is greater than one, the reception field of the convolutional filters can cover a block area on the feature map so that the contextual information under the reception field are jointly learned. In order to learn higher-level

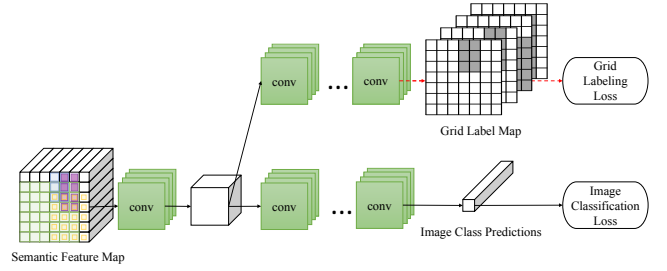


Figure 2: The FCN based classification networks with multi-task losses. (Best viewed in color.)

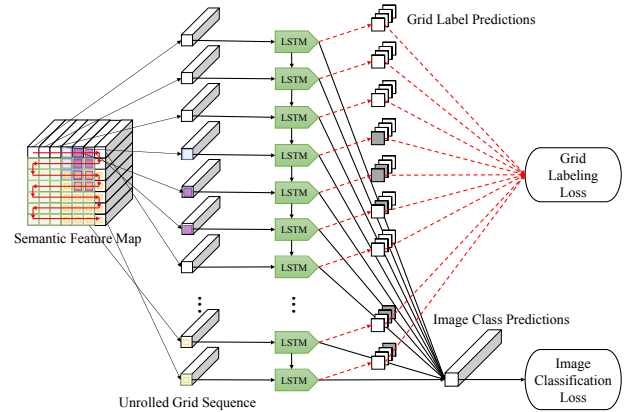


Figure 3: The LSTM based classification networks with multi-task losses. (Best viewed in color.)

feature representation specific to the classification task, multiple convolution layers are stacked. The number of last layer convolutional filters in the classification stream is set to equal the number of image classes, so that the outputs of the network can be used for class prediction.

LSTM. Besides the FCN classifier, we also propose to use LSTM to further encode the contextual information for classification, as illustrated in Figure 3. LSTM is an RNN variant, which is designed to encode or decode information in a long time sequence [11]. Compared with standard RNN, non-linear multiplicative gates and a memory cell are introduced to avoid the vanishing gradient effect. These included input, output and forget gates govern the information flow into and out of the memory cell. The use of temporal model is inspired from [1, 48], in which RNN are used to encode contextual information.

In our LSTM, we unroll our unified $H' \times W' \times C'$ feature map into a sequence of length $H' \times W'$, where each item in the sequence is a C' dimensional semantic feature vector at grid $\mathbf{z}[y, x]$. Considering the spatial proximity property of the feature map, we organize the sequence into the LSTM according to a zigzag path. More specifically, we can feed into the LSTM every grids of feature in \mathbf{z} row by row ordered from top to bottom, as is indicated by the red arrows over the feature map on the left of Figure 3. For every odd rows in the feature map, the sequence is scanned in the left

to right direction; while for every even rows, it is ordered from right to left. Mathematically, the input sequence to the LSTM for $t = [1, \dots, H'W']$ can be expressed as

$$\mathbf{z}_t = \begin{cases} \mathbf{z}[\text{quo}(t, W') + 1, \text{rem}(t, W'), :], & \text{if } \text{mod}(t, W') \text{ is odd} \\ \mathbf{z}[\text{quo}(t, W') + 1, W' - \text{rem}(t, W') + 1, :], & \text{otherwise} \end{cases} \quad (9)$$

where $\text{quo}(\cdot)$ and $\text{rem}(\cdot)$ respectively refer to the quotient and remainder operations. In the classification stream, the sequenced LSTM outputs at each time step are max pooled into the final class prediction.

3.3 Multi-task Learning

Multi-task learning strategy has shown its advantages in existing object feature based image classification tasks [46]. We design the grid labeling loss as an extra clue in the network to help the training of FCN and LSTM classifiers in addition to the original image classification loss.

The multi-task learning scheme for FCN classifier is shown in Figure 2. The bottom row of the figure displays the image classification stream. In this stream, the unified semantic feature map is transformed by several convolutional layers into an L dimensional vector for image class prediction. Here L is the label size. For the optimization (error back-propagation) in the image classification stream, we adopted the same balanced cross entropy loss used in [46] considering the class imbalance in the training data. It is formulated as

$$L_{img} = \frac{1}{M} \sum_{i=1}^M \left\{ \frac{1}{|y_i = 1|} \sum_{j=1}^L y_{ij} \log \hat{y}_{ij} + \frac{1}{|y_i = 0|} \sum_{j=1}^L (1 - y_{ij}) \log(1 - \hat{y}_{ij}) \right\} \quad (10)$$

where M is the mini-batch size; y_{ij} is the j -th ground truth label for the i -th image in the batch; and \hat{y}_{ij} is the probabilistic output of the classifier.

In the grid labeling stream displayed in the upper row of Figure 2, we need to build a target grid label map in order to set the learning target for this stream as the extra supervision knowledge. To solve this problem, we directly fill the image pixels with their belonging objects annotations by an L dimensional 0/1 valued vectors. Then we spatial pool the ground truth vectors for each objects in the same way as described in Section 3.1. The only difference is that we replace the high-level object semantic features with the L dimensional ground truth object labels. For simplicity, we set the grid label map to the same shape as the semantic feature map. Consequently, we obtain a spatial label map of shape $H' \times W' \times L$ filled with 0/1 label indicators as the learning target for this stream. Similarly, convolutional layers can be stacked to fit the internal feature map to the learning targets. In this grid labeling stream, we set the overall loss function L_{grd} to be the same balanced cross entropy loss function L_{img} used in image classification stream.

The two losses L_{img} and L_{grd} are combined to form the multi-task loss function

$$L_{all}(x) = L_{img}(x) + \lambda L_{grd}(x) \quad (11)$$

where λ controls the loss weight. With this multi-task loss, we are able to jointly tune the network for multi-label image classification with the supervision from object annotations when available.

For the LSTM based classifier, we simply unroll the spatial label map according to the same sequence as the contextual feature map and treat them as the learning targets at each time step in the grid labeling stream, as illustrated in Figure 3. Therefore, the target at each time step is a L dimensional label vector filled with 0/1 values. The loss function at each time step is also set to the balanced cross entropy loss. The combined multi-task loss function is in the same form as equation 11.

Note that during inference, object proposals are generated and high-level region features are spatially pooled into the semantic feature map in the same way as during training. However, only the classification stream is kept to produce predicted image classes and the grid labeling stream is not used. The grid labeling stream is mainly deployed to assist better learning of the high-level mid-layer feature representation during training.

4 EXPERIMENTS

In this section, we report our method on image and video classification tasks. The experiments are designed to study the effectiveness and generality of the feature representation, classification networks and multi-task learning strategies.

4.1 Datasets and Experimental Settings

For image classification, we first evaluated the proposed method on the widely used PASCAL VOC 2007 and 2012 benchmarks. Both benchmarks contain images from 20 categories including animals, handmade objects and natural objects at wild locations and scales. In VOC 2007/2012 there are 5011/11540 images in training & validation set and 4952/10991 images in test set, annotated with multiple class labels and object-level bounding boxes. In addition, we also evaluated on the MSCOCO dataset [19], which has a higher positive label density on a much larger label set of 80 common objects. It contains 82,783 images in the training set and 40,504 images in the validation set.

For video classification, we evaluated our feature representation on the Columbia Consumer Videos (CCV) benchmarks. The CCV dataset [14] contains 9,317 YouTube videos and 20 classes such as dog, basketball, graduation ceremony and wedding dance, *etc.* We adopted the training and test split as suggested in [14]. On all image and video datasets, we reported results by using mean average precision (mAP) as the evaluation metric.

4.2 Implementation Details

In our implementation, our model was built upon the VGG16 based CNN network. We spatially transformed the 1000 dimensional high-level attribute features in the 8th fully connected layer into a 7×7 grid semantic feature map by the approach described in Section 3.1. Thus, our generated feature map had the same size as the pool5 layer output of the original VGG model. We also experimented with different FCN/LSTM structures and single/multi-task loss functions to explore their effectiveness, as discussed in Section 4.3. The weights in the new convolutional and LSTM layers were initialized with Gaussian distribution with standard deviation set to $1E-5$ and bias set to 0. Adam solver was used with the base learning rate set to $10E-4$. During the training stage, we first tuned FCN and LSTM sub networks for 20 epochs and then globally fine-tuned the

Table 1: Ablation studies of network structure settings VOC 2007 datasets.

Model Settings	mAP (%)
(a) FCN with single layer 7×7 convs	93.8
(b) FCN with two layer $1 \times 1 + 7 \times 7$ convs	94.0
(c) FCN with two layer $3 \times 3 + 7 \times 7$ convs	94.1
(d) LSTM 1 direction	94.2
(e) LSTM 2 directions	94.4
(f) LSTM 4 directions	94.5

whole network for another 20 epochs. For testing, our method took about 410ms on single Titan X GPU for each image or video frame, including the region proposal generating time by edge box (around 200ms per image).

4.3 Ablation Studies

Effects of FCN and LSTM settings. We first investigate the performance of different network structures of FCN and LSTM on VOC 2007 dataset and discuss their capabilities of contextual information utilization on the generated semantic feature map. The networks parameters were trained under multi-task loss functions. For the FCN based classifiers, we explored on three different convolution filter sizes applied to the generated $7 \times 7 \times 1000$ semantic feature map. Model (a) is to apply 7×7 filters covering the whole semantic feature map. Model (b) is to apply 1×1 filters with pad size 0 and stride size 1, followed by another convolution layer with 7×7 filters. Model (c) is to apply 3×3 filters with pad size 1 and stride size 1, also followed by another convolution layer with 7×7 filters. In these three cases, we further concatenated FCNs to transform the generated features for class predictions. The number of convolution filters was set to 500 in all mid FCN layers. The results are shown in the first 3 rows of Table 1. As we can see, as the depth increases, better classification results are obtained, which confirms the fact that deeper networks possess higher discriminative power [17]. With the same layer depth, model (c) is slightly better than model (b), which indicates that 3×3 filters in the first layer is better at progressively exploiting the contextual information in the feature map since their reception fields cover blocks of areas.

For the LSTM based classifiers, we experimented with unrolling paths on the semantic feature map in multiple directions when organizing the input sequences. The results are shown in the lower rows in Table 1. More specifically, row (d) is the result by applying single directional LSTM, whose sequence is unrolled row by row in from the semantic feature map as described in equation 9. Result (e) is from a bi-directional LSTM by combining forward and backward sequences in model (d) as inputs. The final row shows the performance from a 4-directional LSTM, where unrolled sequence major in column directions and its reversed version are further included. The results show that higher performance can be achieved with more sequences unrolled from multiple directions serving as inputs for the LSTM model, which implies that better contextual information in the feature map are learned.

Effects of multi-task loss functions. In this part, we investigate the effects of the single and multi-task loss functions on the

Table 2: Ablation studies of multi-task loss functions on VOC 2007 datasets.

Loss Function Settings	mAP (%)
(a) FCN with single-task loss function	93.5
(b) FCN with multi-task loss functions	94.1
(c) LSTM with single-task loss function	94.0
(d) LSTM with multi-task loss functions	94.5
(e) FCN + LSTM with multi-task loss functions	94.6

VOC 2007 benchmark. The results are presented in Table 2. As we can see from row (a) and (c), the FCN and LSTM based classifiers trained with single image classification loss ($\lambda = 0$ in equation 11) offer an mAP of 93.5% and 94.0% respectively. This shows that our models can already achieve competitive results even without further object annotations provided. When the multi-task learning strategy ($\lambda = 1$ in equation 11) is introduced, the results with FCN and LSTM based classifiers in row (b) and (d) are both improved over their single task counterparts. These results verify that our multi-task loss functions coupled with the FCN and LSTM models are able to help more effectively exploit the extra supervised information in the object annotations to further improve the classification performance. In row (e), we report the mAP of fused the predictions from multi-task FCN and LSTM results. The result is further slighted improved.

4.4 Comparison with State of the Arts on Image Classification Datasets

Here we compare the proposed method with most recent state-of-the-art methods on PASCAL VOC 2007 and 2012 benchmarks. The compared methods and their property descriptions are listed in Table 3. Note that most of the methods utilize object information in the image by simple max/mean pooling to combine separate object features, while our approach explicitly model the contextual relationships among objects. The AGS [4] method uses subcategory mining by graph shift based on traditional image features. It was shown to be helpful when fused with deep learning based algorithms [37] to improve the performance.

Table 4 and 5 respectively display the detailed per-category performance on VOC datasets. As shown in Table 4, all our methods achieve better classification performance than state-of-the-art methods on VOC 2007 dataset. The FCN classifier follows the setting in Table 1 (c); the LSTM based classifier follows the setting in Table 1 (f). We can find that both FCN and LSTM based classification networks outperform existing methods in mAP. Comparatively, the LSTM based classifier yields higher mAP score than the FCN based one. Finally, combining these two results together can further slightly improve the performance, offering a better mAP score of 94.6%.

Our FCN and LSTM based results on VOC 2012 benchmark shown in Table 5 are better than most of the single model based state-of-the-art competitors, only expect for the RGNN-RL method. Again, fusing the FCN and LSTM results can improve the performance to mAP 93.2%. Note that by combining our result with RGNN-RL and AGS can significantly boost the performance to

Table 3: List of compared methods on VOC datasets.

Method	Models	Object Features Fusion
PRE-1000C [27]	AlexNet + sliding windows.	Late mean pooling.
CNN S TUNE [2]	Fine-tuned global CNN-S.	Global image features only.
VGG-16+19 [32]	Fusion VGG16 + VGG19.	Global image features only.
FV+LV-Fusion [43]	Regional CNN + VGG16 + VGG19.	Encoded by fishier vectors.
DA-Fusion [23]	Fusion of deep attributes.	Early max pooling grouped by region sizes.
HCP-VGG [38]	HCP with VGG16.	Late max pooling.
HCP++ [38]	HCP-VGG + subcat. model.	Late max pooling.
SPD [25]	GoogleNet + part based fusion.	Weighted by boosting.
CRP [37]	VGG16 + random crop pooling.	Category-wise max pooling.
CRP+AGS [37]	CRP + fusion with AGS.	Category-wise max pooling.
RGNN-RL [45]	VGG16 + region selection.	Early max pooling grouped by region sizes.
CNN-RNN [36]	VGG16 + LSTM in label sequence.	Global image feature only.

Table 4: Classification results (average precision in %) comparison with state of the arts on VOC 2007.





















																					mAP
PRE-1000C	88.5	81.5	87.9	82.0	47.5	75.5	90.1	87.2	61.6	75.7	67.3	85.5	83.5	80.0	95.6	60.8	76.8	58.0	90.4	77.9	77.7
CNN S TUNE	95.3	90.4	92.5	89.6	54.4	81.9	91.5	91.9	64.1	76.3	74.9	89.7	92.2	86.9	95.2	60.7	82.9	68.0	95.5	74.4	82.4
CNN-RNN	96.7	83.1	94.2	92.8	61.2	82.1	89.1	94.2	64.2	83.6	70.0	92.4	91.7	84.2	93.7	59.8	93.2	75.3	99.7	78.6	84.0
VGG-16+19	98.9	95.0	96.8	95.4	69.7	90.4	93.5	96.0	74.2	86.6	87.8	96.0	96.3	93.1	97.2	70.0	92.1	80.3	98.1	87.0	89.7
FV+LV-Fusion	92.8	96.9	97.1	95.8	74.3	94.2	96.7	97.7	76.7	90.5	88.0	96.9	97.7	95.9	98.6	78.5	93.6	82.4	98.4	90.4	92.0
SPD	98.7	97.0	97.9	94.8	78.3	91.4	96.4	97.3	75.0	85.0	82.4	95.4	96.1	94.7	98.5	75.9	90.9	82.1	97.3	89.7	90.7
DA-Fusion	99.4	97.5	96.8	96.6	81.3	92.9	96.8	97.1	75.6	93.7	84.5	95.8	96.8	96.0	98.6	81.9	97.7	80.2	99.0	91.5	92.5
HCP-VGG	98.6	97.1	98.0	95.6	75.3	94.7	95.8	97.3	73.1	90.2	80.0	97.3	96.1	94.9	96.3	78.3	94.7	76.2	97.9	91.5	90.9
RCP	93.3	97.6	98.0	96.4	79.3	93.8	96.6	97.1	78.0	88.7	87.1	97.1	96.3	95.4	99.1	82.1	93.6	82.2	98.4	92.8	92.5
RGNN-RL	99.3	97.0	97.5	98.1	80.6	95.5	97.2	98.0	82.1	96.5	86.3	97.5	97.9	95.6	98.8	84.0	97.2	82.7	99.1	93.3	93.7
Ours (FCN)	99.7	97.8	98.5	97.7	81.5	96.6	97.3	98.5	82.8	98.7	86.9	98.6	98.9	96.2	98.1	82.9	99.3	79.4	99.2	93.1	94.1
Ours (LSTM)	99.9	97.5	98.5	98.6	82.6	96.5	97.7	98.6	82.9	98.2	87.1	98.7	99.0	96.7	98.9	85.8	99.3	81.2	99.4	93.3	94.5
Ours (FCN+LSTM)	99.8	97.6	98.6	98.5	82.7	96.6	97.7	98.6	83.1	98.4	87.9	98.6	99.1	96.6	98.8	85.9	99.2	81.1	99.3	93.5	94.6

Table 5: Classification results (average precision in %) comparison with state of the arts on VOC 2012.





















																					mAP
PRE-1000C	93.5	78.4	87.7	80.9	57.3	85.0	81.6	89.4	66.9	73.8	62.0	89.5	83.2	87.6	95.8	61.4	79.0	54.3	88.0	78.3	78.7
CNN S TUNE	96.8	82.5	91.5	88.1	62.1	88.3	81.9	94.8	70.3	80.2	76.2	92.9	90.3	89.3	95.2	57.4	83.6	66.4	93.5	81.9	83.2
VGG-16+19	99.1	89.1	96.0	94.1	74.1	92.2	85.3	97.9	79.9	92.0	83.7	97.5	96.5	94.7	97.1	63.7	93.6	75.2	97.4	87.8	89.3
FV+LV-Fusion	98.9	93.1	96.0	94.1	76.4	93.5	90.8	97.9	80.2	92.1	82.4	97.2	96.8	95.7	98.1	73.9	93.6	76.8	97.5	89.0	90.7
DA-Fusion	99.2	93.7	96.0	95.2	81.7	94.3	91.6	98.1	81.9	91.7	83.5	96.3	95.6	96.0	98.2	77.8	93.6	74.7	97.6	91.9	91.4
HCP-VGG	99.1	92.8	97.4	94.4	79.9	93.6	89.8	98.2	78.2	94.9	79.8	97.8	97.0	93.8	96.4	74.3	94.7	71.9	96.7	88.6	90.5
HCP++	99.8	94.8	97.7	95.4	81.3	96.0	94.5	98.9	88.5	94.1	86.0	98.1	98.3	97.3	97.3	76.1	93.9	84.2	98.2	92.7	93.2
RCP	99.3	92.2	97.5	94.9	82.6	94.1	92.4	98.5	83.8	93.5	83.1	98.1	97.3	96.0	98.8	77.7	95.1	79.4	97.7	92.4	92.2
RCP+AGS	99.8	94.5	98.1	96.1	85.5	96.1	95.5	99.0	90.2	95.0	87.8	98.7	98.4	97.5	99.0	80.1	95.9	86.5	98.8	94.6	94.3
RGNN-RL	99.3	95.7	97.7	95.4	84.5	96.2	94.6	98.4	84.6	95.6	84.1	97.9	98.0	96.7	98.7	82.9	96.1	79.6	98.6	93.4	93.4
Ours (FCN)	99.4	95.2	97.0	94.8	82.5	95.3	93.8	98.1	84.0	94.0	83.7	97.6	97.0	96.8	98.5	81.8	94.3	77.4	98.0	92.0	92.5
Ours (LSTM)	99.4	95.6	97.7	94.8	84.4	95.7	94.9	98.2	83.5	94.8	83.2	97.8	97.2	96.9	98.8	82.2	95.3	79.6	98.2	93.0	93.1
Ours (FCN+LSTM)	99.4	95.6	97.7	94.9	85.0	95.6	94.9	98.2	83.8	94.8	83.6	97.9	97.1	97.1	98.9	82.5	95.3	79.7	98.2	93.0	93.2
Ours (all combined)	99.8	96.3	98.2	96.4	87.1	97.0	96.3	98.9	89.3	96.1	87.5	98.6	98.3	97.9	98.7	82.8	96.2	85.3	98.9	95.1	94.7

Table 6: Classification results (average precision in %) comparison on MSCOCO dataset.

Method	mAP (%)
VGG16 (global)	65.8
RGNN-RL [45]	72.9
Ours (FCN)	73.1
Ours (LSTM)	73.3
Ours (all combined)	74.0

94.7%, which outperforms the best result mAP 94.3% from RCP+AGS method [37]. This indicates that our learned feature representation can successfully provide valuable complementary information to existing state-of-the-art methods to improve the overall classification performance. For the weakness analysis, we observe that our methods work comparatively worse on some small objects (bottle, plant) and objects often under occlusions (sofa).

The evaluated results on MSCOCO are presented in Table 6. The original VGG16 net achieves an mAP of 65.8% and the reported mAP by RGNN-RL model is 72.9%. Our feature representation with FCN based classifier obtains a higher mAP 73.1% and LSTM based classifier obtains mAP 73.3%. Fusing our results with the predictions of RGNN-RL model offers an mAP 74.0%. We can find that on all test image data, the object clue knowledge provided by our method can provide useful information to boost performance when fusing with other methods.

4.5 Transferring to Video Classification Task

In this part, we investigate the transferability of the semantic feature map generation model learned from image data to video classification task. Here we directly use our models pre-trained on VOC 2012 datasets to extract the object-level semantic features on the video frames. All frame features from a single video are averaged and input into the neural networks to train the video classifiers.

We compare our results obtained from the contextual semantic feature map with other state-of-the-art methods on CCV dataset. The evaluation results are shown in Table 7. It is obvious that the results from previous deep learning based methods [33, 40, 41] outperform approaches based on traditional features. Among the compared methods, the most related one is [33], which also leveraged deep object-level features for video recognition. The authors obtained an mAP = 81.7% with global CNN features extracted from video frames and optical flow images. When they extracted object features based on the whole video frames and averaged them into the video representation, the resulted mAP is 64.6%. They also used LSTM to temporally encode the object feature across frames, the resulted mAP is 59.8%. Fusing these temporal and non-temporal object-level results together raises the mAP to 68.4%. After fusing all the results from global and object clues, their method obtained mAP of 85.9%.

We can find that both our FCN based approach (mAP 69.2%) and LSTM based approach (mAP 68.9%) outperform the best pure object clue result in [33]. Note that VGG19 based networks and more than 20K object scores were used in [33], while our method used only 1K object semantic attributes extracted from VGG16

Table 7: Classification results (average precision in %) comparison on CCV dataset.

Method	mAP (%)
Lai et al. [16]	43.6
Jiang et al. [14]	59.5
Xu et al. [42]	60.3
Ma and Yuen [24]	63.4
Jhuo et al. [13]	64.0
Ye et al. [44]	64.0
Liu et al. [20]	68.2
Nagel et al. [26]	71.7
Wu et al. [41]	83.5
Wu et al. [40]	84.0
Sun et al. [33] (global)	81.7
Sun et al. [33] (object non-temporal)	64.6
Sun et al. [33] (object temporal)	59.8
Sun et al. [33] (object temporal + non-temporal)	68.4
Sun et al. [33] (all combined)	85.9
Ours (FCN)	69.2
Ours (LSTM)	68.9
Ours (FCN+LSTM)	69.7
Ours (all combined)	86.5

based networks. And our method did not additionally use LSTM to temporally encode features across sequence frames while [33] did.

Similar to the findings in image data, fusing the FCN and LSTM results together can lead to an improved mAP = 69.7% on CCV dataset. When combining our results with the global CNN results, the final evaluated mAP rises to 86.5%, which means that our method is effective and can provide more valuable complementary clues to traditional CNN models for video classification than the object based method in [33].

5 CONCLUSION

In this paper, we propose to learn a generic semantic feature map to model spatial contextual information among objects for visual content understanding. Optimized by the multi-task loss functions, the generated feature representation can work with FCN and LSTM based classifiers to achieve competitive performance on popular image benchmarks. We also show that the learned feature map in image domain can be transferred to video domain to provide effective object clues to produce improved classification results.

ACKNOWLEDGMENTS

This work was supported by two NSFC projects (#61622204 and #61572134).

REFERENCES

- [1] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross B Girshick. 2016. Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Las Vegas, NV, USA, 2874–2883.
- [2] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In *British Machine Vision Conference*.

- [3] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. 2014. BING: Binarized Normed Gradients for Objectness Estimation at 300fps. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- [4] Jian Dong, Wei Xia, Qiang Chen, Jiashi Feng, ZhongYang Huang, and Shuicheng Yan. 2013. Subcategory-Aware Object Classification. In *IEEE Conference on Computer Vision and Pattern Recognition*. 827–834.
- [5] Mark Everingham, S M Ali Eslami, Luc Van Gool, Christopher K I Williams, John Winn, and Andrew Zisserman. 2014. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision* 111, 1 (2014), 98–136.
- [6] Ross B Girshick. 2015. Fast R-CNN. In *IEEE International Conference on Computer Vision*.
- [7] Ross B Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 580–587.
- [8] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. 2014. Multi-scale Orderless Pooling of Deep Convolutional Activation Features. In *European Conference on Computer Vision*. 392–407.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 9 (2015), 1904–1916.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Las Vegas, NV, USA, 770–778.
- [11] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (1997), 1735–1780.
- [12] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. 2010. Aggregating Local Descriptors Into a Compact Image Representation. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, San Francisco, CA, USA, 3304–3311.
- [13] I-Hong Jhuo, Guangnan Ye, Shenghua Gao, Dong Liu, Yu-Gang Jiang, D T Lee, and Shih-Fu Chang. 2014. Discovering Joint Audio-Visual Codewords for Video Event Detection. *Machine Vision and Applications* 25, 1 (Oct. 2014), 33–47.
- [14] Yu-Gang Jiang, Guangnan Ye, Shih-Fu Chang, Daniel P W Ellis, and Alexander C Loui. 2011. Consumer Video Understanding: a Benchmark Database and an Evaluation of Human and Machine Performance. In *ACM International Conference on Multimedia Retrieval*. ACM Press, Trento, Italy, 1–8.
- [15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*. 1097–1105.
- [16] Kuan-Ting Lai, Felix X Yu, Ming-Syan Chen, and Shih-Fu Chang. 2014. Video Event Detection by Inferring Temporal Instance Labels. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Columbus, OH, USA, 2251–2258.
- [17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.
- [18] Li-Jia Li, Hao Su, Yongwhan Lim, and Fei-Fei Li. 2014. Object Bank: An Object-Level Image Representation for High-Level Visual Recognition. *International Journal of Computer Vision* 107, 1 (2014), 20–39.
- [19] Tsung-Yi Lin, Michael Maire, Serge J Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision*. 740–755.
- [20] Dong Liu, Kuan-Ting Lai, Guangnan Ye, Ming-Syan Chen, and Shih-Fu Chang. 2013. Sample-Specific Late Fusion for Visual Category Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Portland, OR, USA, 803–810.
- [21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. SSD: Single Shot MultiBox Detector. In *European Conference on Computer Vision*. Springer International Publishing, Amsterdam, The Netherlands, 21–37.
- [22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully Convolutional Networks for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Boston, MA, USA, 3431–3440.
- [23] Jianwei Luo, Jianguo Li, Jun Wang, Zhiguo Jiang, and Yurong Chen. 2015. Deep Attributes from Context-Aware Regional Neural Codes. *arXiv.org* (2015). arXiv:1509.02470v1
- [24] Andy J Ma and Pong C Yuen. 2014. Reduced Analytic Dependency Modeling: Robust Fusion for Visual Recognition. *International Journal of Computer Vision* 109, 3 (2014), 233–251.
- [25] Pascal Mettes, Jan C van Gemert, and Cees G M Snoek. 2016. No Spare Parts: Sharing Part Detectors for Image Categorization. *Computer Vision and Image Understanding* 152 (Nov. 2016), 131–141.
- [26] Markus Nagel, Thomas Mensink, and Cees G M Snoek. 2015. Event Fisher Vectors: Robust Encoding Visual Diversity of Visual Streams. In *British Machine Vision Conference*. BMVA Press, Swansea, UK, 178.1–178.12.
- [27] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. 2014. Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. 1717–1724.
- [28] Joseph Redmon, Santosh Divvala, Ross B Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Las Vegas, NV, USA, 779–788.
- [29] Shaoqing Ren, Kaiming He, Ross B Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*. 91–99.
- [30] Olga Russakovsky, Jia Deng, Hao Su, and et al. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.
- [31] Karen Simonyan and Andrew Zisserman. 2014. Two-Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems*. Montréal, Canada, 568–576.
- [32] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large Scale Image Recognition. In *International Conference on Learning Representations*. Dan Diego, CA, USA.
- [33] Yongqing Sun, Zuxuan Wu, Xi Wang, Hiroyuki Arai, Tetsuya Kinebuchi, and Yu-Gang Jiang. 2016. Exploiting Objects with LSTMs for Video Categorization. In *ACM Multimedia*. ACM Press, Amsterdam, The Netherlands, 142–146.
- [34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going Deeper with Convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Boston, MA, USA, 1–9.
- [35] Jasper R R Uijlings, Koen E A van de Sande, Theo Gevers, and Arnold W M Smeulders. 2013. Selective Search for Object Recognition. *International Journal of Computer Vision* 104, 2 (2013), 154–171.
- [36] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. 2016. CNN-RNN: A Unified Framework for Multi-label Image Classification. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Las Vegas, NV, USA, 2285–2294.
- [37] Meng Wang, Changzhi Luo, Richang Hong, Jinhui Tang, and Jianshi Feng. 2016. Beyond Object Proposals: Random Crop Pooling for Multi-Label Image Recognition. *IEEE Transactions on Image Processing* 25, 12 (Dec. 2016), 5678–5688.
- [38] Yunchao Wei, Wei Xia, Min Lin, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. 2015. HCP: A Flexible CNN Framework for Multi-label Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2015), 1–8.
- [39] Zuxuan Wu, Yanwei Fu, Yu-Gang Jiang, and Leonid Sigal. 2016. Harnessing Object and Scene Semantics for Large-Scale Video Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- [40] Zuxuan Wu, Yu-Gang Jiang, Xi Wang, Hao Ye, and Xiangyang Xue. 2016. Multi-Stream Multi-Class Fusion of Deep Networks for Video Classification. In *ACM Multimedia*. ACM Press, Amsterdam, The Netherlands, 791–800.
- [41] Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Xue. 2015. Modeling Spatial-Temporal Clues in a Hybrid Deep Learning Framework for Video Classification. In *ACM MM*. 461–470.
- [42] Zhongwen Xu, Yi Yang, Ivor W Tsang, Nicu Sebe, and Alexander G Hauptmann. 2013. Feature Weighting via Optimal Thresholding for Video Analysis. In *IEEE International Conference on Computer Vision*. Sydney, Australia, 3440–3447.
- [43] Hao Yang, Joey Tianyi Zhou, Yu Zhang, Bin-Bin Gao, Jianxin Wu, and Jianfei Cai. 2015. Can Partial Strong Labels Boost Multi-label Object Recognition? *arXiv.org* (2015).
- [44] Guangnan Ye, Dong Liu, I-Hong Jhuo, and Shih-Fu Chang. 2012. Robust Late Fusion with Rank Minimization. In *IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI, USA, 3021–3028.
- [45] Shengxin Zha, Florian Luisier, Walter Andrews, Nitish Srivastava, and Ruslan Salakhutdinov. 2015. Exploiting Image-trained CNN Architectures for Unconstrained Video Classification. In *British Machine Vision Conference*. Xianghua Xie, Mark W Jones, and Gary K L Tam (Eds.). BMVA Press, Swansea, UK, 60.1–60.13.
- [46] Rui-Wei Zhao, Jianguo Li, Yurong Chen, Jia-Ming Liu, and Yu-Gang Jiang. 2016. Regional Gating Neural Networks for Multi-label Image Classification. In *British Machine Vision Conference*. York, UK, 1–12.
- [47] C Lawrence Zitnick and Piotr Dollár. 2014. Edge Boxes: Locating Object Proposals from Edges. In *European Conference on Computer Vision*. 391–405.
- [48] Zhen Zuo, Bing Shuai, Gang Wang, Xiao Liu, Xingxing Wang, Bing Wang, and Yushi Chen. 2016. Learning Contextual Dependence with Convolutional Hierarchical Recurrent Neural Networks. *IEEE Trans. Image Processing* 25, 7 (March 2016), 2983–2996.