Video2Shop: Exact Matching Clothes in Videos to Online Shopping Images

Zhi-Qi Cheng[1], Xiao Wu[1], Yang Liu[2], Xian-Sheng Hua[2]
[1]Southwest Jiaotong University, [2]Alibaba Group
{zhiqicheng,huaxiansheng}@gmail.com; wuxiaohk@swjtu.edu.cn; panjun.ly@alibaba-inc.com

Abstract

In recent years, both online retail and video hosting service have been exponentially grown. In this paper, a novel deep neural network, called AsymNet, is proposed to explore a new cross-domain task, Video2Shop, targeting for matching clothes appeared in videos to the exactly same items in online shops. For the image side, well-established methods are used to detect and extract features for clothing patches with arbitrary sizes. For the video side, deep visual features are extracted from detected object regions in each frame, and further fed into a Long Short-Term Memory (LSTM) framework for sequence modeling, which captures the temporal dynamics in videos. To conduct exact matching between videos and online shopping images, LSTM hidden states for videos and image features extracted from static images are jointly modeled, under the similarity network with reconfigurable deep tree structure. Moreover, an approximate training method is proposed to achieve the efficiency when training. Extensive experiments conducted on a large cross-domain dataset have demonstrated the effectiveness and efficiency of the proposed AsymNet, which outperforms the state-of-the-art methods.

1. Introduction

With the exponential growth of e-commerce, online clothing shopping becomes more and more popular, which takes up a significant portion of the retail. Driven by the huge profit potential, clothing item retrieval has been received a great deal of attention in multimedia and computer vision communities. Meanwhile, online video streaming service becomes increasingly popular. When watching idol drama or TV shows, such as Korean TV drama *My Love From the Star*, where beautiful girls wear fashion clothes, the viewers, especially the females, are more easily attracted by those beautiful clothes and stimulated to buy the identical ones shown in the video. In this paper, we consider a new scenario of such online clothing shopping, called *Video2Shop*: finding the clothes appeared in videos to the identical items in online shops.

Although the street-to-shop clothing matching has been explored previously [9, 10, 16, 25, 29], which searches the online clothing by street fashion photos, finding clothes appeared in videos to the exactly same items in online shops has not been well studied yet. The diverse appearance of clothes, cluttered background, viewpoint change, occlusion, different light condition and motion blur in videos, make Video2Shop a challenging task. More specifically, the clothing items appeared in videos and online shopping websites demonstrate significant visual discrepancy. On one hand, the clothes in the videos are usually captured from different viewpoints (from front, side or back), which lead to great varieties in visual appearance. The complex scenes and common motion blur in videos make the situation even worse. On the other hand, the online clothing images are not always with clean background, since the clothes are often worn by fashion models in outdoor scenes to show the real wearing effect. The cluttered background imposes difficulties for clothing localization and analysis. These problems make the *Vidoe2Shop* task more challenging than the street-to-shop search.

The architecture of the proposed deep neural network, AsymNet, is illustrated in Fig. 1. When users watch videos through web pages or set-top box devices, the system will retrieve the exactly matched clothing items from online shops and return them to the users. Clothing detector is first deployed for both video side and image side, to extract a set of proposals (clothing patches) to identify the potential clothing regions, eliminating the impact of background regions and leading to more accurate clothing localization. For videos, clothing tracker is then conducted to generate clothing trajectories from extracted clothing patches, which contain the same clothing items appeared in continuous frames. Intuitively, the clothing patches with inconsistent viewpoints can be preserved. Due to the promising performance and stability, Faster-RCNN [22] and Kernelized Correlation Filters (KCF) tracker [8] are adopted in this paper as the clothing detector and clothing tracker, respectively. Deep visual features are generated for shopping images and clothing trajectories in videos, which are achieved with image feature network (IFN) and video fea-

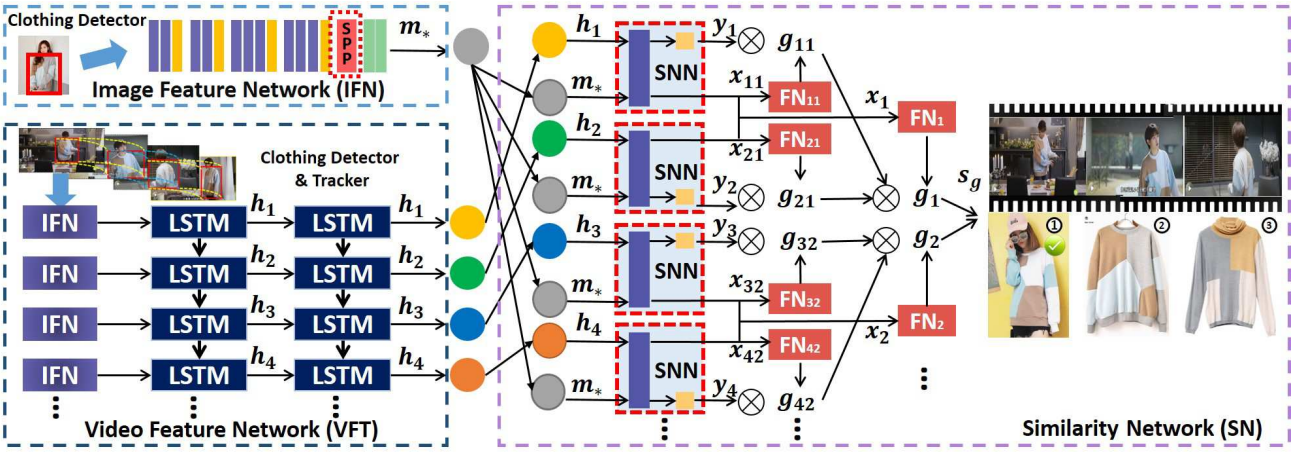

Figure 1. Framework of the proposed AsymNet. After clothing detection and tracking, deep visual features are generated by image feature network (IFN) and video feature network (VFN), respectively. They are then fed into the similarity network to perform pair-wise matching.

ture network (VFN), respectively. For videos, deep visual features are further fed into a Long Short-Term Memory (LSTM) framework [4] for sequence modeling, which captures the temporal dynamics in videos. Then the problem of Video2Shop is formulated as an asymmetric (multiple-to-single) matching problem, i.e., exact matching a sequence of clothes appeared in videos to a single online shopping clothing. These features are then fed into the similarity network to perform pair-wise matching between clothing regions from videos and shopping images, in which a reconfigurable deep tree structure is proposed to automatically learn the fusion strategy. The top ranked results are then returned to users.

The main contributions of the proposed work are summarized as follows:

- A novel deep-based network, AsymNet, is proposed for cross-domain Video2Shop application, which is formulated as an asymmetric (multiple-to-single) matching problem. It mainly consists of two components: image/video feature representation and similarity measure.
- To conduct exact matching, LSTM hidden states for clothing trajectories in videos, and image features extracted from online shopping images, are jointly modeled under the similarity network with a reconfigurable deep tree structure.
- To train AsymNet, an approximate training method is proposed to improve the training efficiency. The proposed method can handle large-scale online search.
- Experiments have been conducted on the first and the largest Video2Shop dataset, which consists of 26,352 clothing trajectories in videos and 85,677 clothing shopping images. Experiments demonstrate the effectiveness of the proposed method, which outperforms the state-of-the-art approaches.

The rest of our paper is organized as follows: related works are first reviewed in Section 2. The details of feature extraction networks and similarity networks are elaborated in Sections 3 and 4, respectively. The approximate training of the network is presented in Section 5. Finally, experiments are introduced in Section 6.

2. Related Work

2.1. Clothing Retrieval

Clothing retrieval has wide applicability for commercial systems. Extensive efforts have been focused on similar clothing retrieval [1, 2, 3, 10, 15, 18, 19] and exactly same clothing retrieval [16, 25].

For similar clothing retrieval, clothing recognition and segmentation techniques are used in [15, 18] to retrieve similar clothing. In order to tackle the domain discrepancy between street photos and shop photos, sparse representations are utilized in [19]. With the adoption of deep learning, an attribute-aware fashion-related retrieval system is proposed in [10]. A convolutional neural network using the contrastive loss is proposed in [1] to learn the visual similarity between products. Based on the Siamese network, a Dual Attribute-aware Ranking Network (DARN) is proposed in [9] to retrieve the similar clothing.

For exactly same clothing retrieval, exact matching street clothing photos in online shops is firstly explored in [16]. A robust deep feature representation is learned in [25] to bridge the domain gap between the street and shopping images. A new deep model, namely FashionNet, is proposed in [20], which learns clothing features by jointly predicting clothing attributes and land-marks. Despite recent advances in exact street-to-shop retrieval, there are few studies focused specifically on exact matching clothes in videos to online shops.

2.2. Deep Similarity Learning

As deep convolutional neural networks are becoming ubiquitous, there has been growing interest in similarity

learning with deep models [1, 6, 20, 26, 28]. Several convolutional neural networks are proposed for image patch-matching [6, 26, 28]. For object retrieval, an neural network with contrastive loss function is designed in [1]. A novel Deep Fashion Network architecture is proposed in [20] for efficient similarity retrieval. These techniques are coupled with either pre-defined distance functions or multi-layer neural networks to learn the similarity. Inspired by these works, we propose a tree structure similarity learning network to match clothes appeared in videos to the exactly same items in online shops.

3. Representation Learning Networks

When clothing regions are detected in images and then tracked into clothing trajectories for videos, feature extraction networks are then conducted to obtain the deep features.

3.1. Image Representation Learning Network

Image Feature Network (IFN) is implemented based on VGG16 [24], in which the input image patches are scaled to 256x256 and then cropped to a random 227x227 region, so that it meets the output requirement for the last convolutional layer. In our Video2Shop matching task, Faster-RCNN [22] is adopted to detect clothing regions in shopping images. Unfortunately, the detected clothing regions are with arbitrary sizes, which violate the requirement of the input size. Enlightened by the idea of recently proposed spatial pyramid pooling (SPP) architecture [7], which pools features in arbitrary regions to generate fixed-length representations, a spatial pyramid pooling layer is inserted between the convolutional layers and the fully-connected layers of the network, as shown in Fig. 2. It aggregates features of the last convolutional layer through spatial pooling, so that the size of the pooling regions is independent of the size of the input.

3.2. Video Representation Learning Network

Video Feature Network (VFN) is illustrated in Fig. 1. For videos, the aforementioned Image Feature Network (IFN) is also used to extract convolutional features. Since the temporal dynamics exist in videos, traditional average pooling strategy becomes invalid. Recurrent Neural Network (RNN) [4] is a perfect choice to solve this problem. Recently, due to its long short-term memory capability for modeling sequential data, Long Short-Term Memory (LSTM) [4] has been successfully applied to a variety of sequence modeling tasks. In this paper, it is chosen to characterize the clothing trajectories in videos.

Based on the LSTM unit proposed in [27], a typical LSTM unit consists of an input gate i_t, a forget gate f_t, an output gate o_t, as well as a candidate cell state g_t. The interaction between states and gates along the time dimension

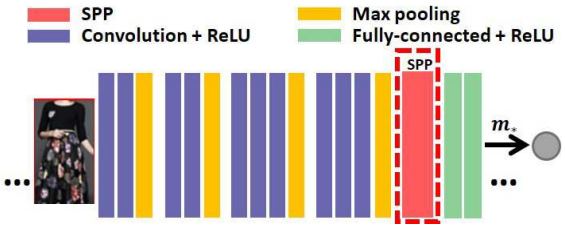

Figure 2. The Architecture of Image Feature Network

is defined as follows:

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} M \begin{pmatrix} \mathbf{h}_{t-1} \\ \mathbf{m}_t \end{pmatrix}, \quad \mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \quad \mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t). \tag{1}$$

Here, c_t encodes the cell state, h_t encodes the hidden state, and m_t is the convolutional feature generated by Image Feature Network. The operator $\odot$ represents element-wise multiplication. Given convolutional features $M(m_1, ..., m_n)$ of a clothing trajectory in videos, a single LSTM computes a sequence of hidden states $(h_1, ..., h_n)$. Further, we find that the temporal variety cannot be fully learned by a single LSTM, so we stack LSTM network to further increase the discriminative ability of the network, by using the hidden units from one layer as the inputs for the next layer. After experimental validation, a two-level LSTM network is utilized in this work.

4. Similarity Learning Networks

4.1. Motivation

To conduct pair-wise similarity measure between clothing trajectories from videos and shopping images, a similarity network is proposed. The inputs are several LSTM hidden states $(h_1, h_2, ..., h_n)$ from Video Feature Network and a convolutional feature m_* from Image Feature Network. The output is a similarity score s_g. This problem is formulated as an asymmetric (multiple-to-single) matching problem. Traditionally, this problem is solved by conducting average or max pooling on whole clothing trajectories to obtain the global similarity or directly select the similarity of the last one in trajectories. More recently, a key volume detection method [30] is proposed to solve the similar problem. However, these methods will fail in our Video2Shop application due to the large variability and complexity of video data. The average or max values cannot completely represent the clothing trajectory. Although key volume is able to learn the most critical parts, it is still too simple to solve this task.

Based on the statistical theory [11, 14], these learning problems are formulated as a mixture estimation problem,

which attacks a complex problem by dividing it into simpler problems whose solutions can be combined to yield a solution to the complex problem. Enlightened by this idea, we novelly extend the generalized mixture expert model to Recurrent Neural Networks, and modify the strategy of mixture estimation to gain a global similarity. The proposed approach attempts to allocate fusion nodes to summarize the single similarity located in different viewpoints.

4.2. Network Structure

Because there are multiple inputs and only one output, a tree structure is proposed to automatically adjust the fusion strategy, which is illustrated in Fig. 1. There are two types of nodes involved in the tree structure, i.e., similarity network node (SNN) and fusion node (FN), corresponding to the leaves and the branches in the tree. The similarity network node acts as the leaves of the tree, which calculates the similarity between a single LSTM hidden state and a convolutional feature. After that, these results are passed to fusion nodes, which generate a scalar output controlling the weight of similarity fusion. These fusion nodes will be passed layer by layer to fuse the results of internal results. In this work, a five-layer structure is adopted. Finally, a final global similarity will be given. Details of each substructure are given as follows.

Similarity Network Node (SNN) To facilitate understanding, we will first introduce the one-to-one similarity measure between a LSTM hidden state h_i and a convolutional feature m_*. As indicated in [16], cosine similarity is too general to capture the underlying differences between features. Therefore, the similarity between h_i and m_* is modeled as a network with two fully-connected layers, denoted as the red dotted box shown in Fig. 1. Specifically, the first two fully-connected layers have 256 (fc1) and 1 (fc2) outputs, respectively. The output of the last fully-connected layer for the i-th SNN is a real value z_i. On the top of the network, logistic regression is used to generate the similarity y_i between h_i and m_*:

$$y_i = \frac{1}{1 + e^{-z_i}} \tag{2}$$

Fusion Node (FN) Since SNN is piece-wisely smoothed, which is analogous to corresponding generalized linear models (GLIM) [5]. Once the individual SNN is calculated, the fusion scores of all fusion nodes will be generated with a a tree-like structure. In this network, multiple low-level fusion nodes are connected to a higher-level fusion node, which forms a tree-like structure. For simplicity, here we use a 2-level structure (in Fig. 1) as an example. The low-level fusion nodes refer to the leaves while the top-level is the side of the root. For a low-level fusion node FN_{ij}, which indicates the i-th low-level node connecting to the j-th neighboring top-level node, an intermediate variable $\varepsilon_{\mathbf{ij}}$ is defined as:

$$\varepsilon_{\mathbf{ij}} = \mathbf{v_{ij}}^T (\mathbf{x_{ij}}) \tag{3}$$

where $\mathbf{v_{ij}}$ is the parameters of this FN and $\mathbf{x_{ij}}$ is the feature vector from the fc1 layer of corresponding SSN. Here, each low-level fusion node is connected to a specific SNN. The output of the low-level fusion node g_{ij} is a weighted score normalized by the scores of all fusion nodes connecting to the same top-level fusion node:

$$g_{ij} = \frac{e^{\varepsilon_{i,j}}}{\sum_i e^{\varepsilon_{i,j}}} \tag{4}$$

Similarly, for the top-level fusion node FN_j, an intermediate variable $\varepsilon_{\mathbf{j}}$ is computed as: $\varepsilon_{\mathbf{j}} = \mathbf{v_j}^T (\mathbf{x_j})$, where $\mathbf{x_j}$ is an average pooling vector from multiple low-level fusion nodes, which are connected to FN_j, $\mathbf{v_j}$ is the parameters of this fusion node. The fusion score g_j is normalized by the scores of all top-level fusion nodes as: $g_j = \frac{e^{\varepsilon_j}}{\sum_j e^{\varepsilon_j}}$. With such a tree structure, for each mini-batch, the parameters of fusion nodes are updated in the forward pass. Once the similarity network converges, the fusion strategy is obtained.

4.3. Learning Algorithm

In this subsection, we will introduce the learning method of our similarity network. The learning is implemented in a two-step iteration approach, where similar network nodes and fusion nodes will be mutually enhanced. The feature representation network and similar network nodes are first learned, and then the fusion nodes are learned when similar network nodes are fixed.

Learning of Similarity Network Node: The learning problem of SNN is formulated as minimizing a Logarithmic Loss. Suppose that we have N convolutional features from the first fully-connected layer fc1 as $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}\}$ and each has a label $\hat{y_k} \in \{0, 1\}$, where $k \in [0, N]$, and 0 means "does not match" while 1 denotes "matches". The loss function $\bigtriangledown(SNN)$ is defined as:

$$L = \frac{1}{N} \sum_{k=1}^{N} (\hat{y_k} log\,(y_k) + (1 - \hat{y_k})\, log\,(1 - y_k)) + \lambda \left\| W_i \right\|^2 \tag{5}$$

where W_i is the parameters of i-th SNN, y_k is the output of single similarity network with x_k as the input, which is defined in Eqn. 2.

Learning of Fusion Node: For a given mini-batch feature set of the fc1 layer, when SNN is fixed, the global similarity s_g can be defined as the mixture of the similarity of each SNN:

$$p(s_g) = \sum_j g_j \sum_i g_{ij} p_i(y) \tag{6}$$

where $p(s)$ and $p_i(y)$ are similarities of global and i-th SNN. g_j and g_{ij} are the fusion scores of higher and low-level fusion nodes. The meaning of the Eqn. 6 is that the similarity of all similar network nodes are passed to multiply layers of fusion nodes to generate the results of global similarity.

In order to implement the learning algorithms of Eqn. 6, the posterior probabilities of fusion nodes are defined. The fusion scores of top-level g_j and low-level g_{ij} are referred as prior probabilities, since they are computed without the knowledge of corresponding output of SNN y_i (as calculated in Eqn. 3 and Eqn. 4). With Bayes' rule, the posterior probabilities at the top-level fusion nodes and low-level nodes are denoted as follows:

$$h_j = \frac{g_j \sum_i g_{ij} p_i(y)}{\sum_j g_j \sum_i g_{ij} p_i(y)} \tag{7}$$

and

$$h_{ij} = \frac{g_{ij} p_i(y)}{\sum_i g_{ij} p_{ij}(y)} \tag{8}$$

With these posterior probabilities, a gradient descent learning algorithm is developed for Eqn. 6. The log likelihood function of a training sample is obtained as:

$$l = \ln \sum_j g_j \sum_i g_{ij} p_i(y) \tag{9}$$

In this case, by differentiating l with respect to the parameters, the following gradient descent learning rules for the parameters of top-level and low-level fusion nodes are obtained as:

$$\bigtriangledown \mathbf{v_j} = \alpha(h_j - g_j)\mathbf{x_j}^{(t)} \tag{10}$$

$$\bigtriangledown \mathbf{v_{ij}} = \alpha h_j(h_{ij} - g_{ij})\mathbf{x_{ij}} \tag{11}$$

where α is a learning rate. v_j and v_{ij} are the parameters of high-level and low-level fusion nodes, respectively. These equations denote a batch learning algorithm to train fusion nodes (i.e. tree structure). To form a deeper tree, each SNN is expanded recursively into a fusion node and a set of sub-SNN networks. In our experiment, we have five-level deep tree structure and the number of fusion nodes in each level is 32, 16, 8, 4, 2, respectively.

5. Approximate Training

Intuitively, to achieve good performance, different models should be trained independently for different clothing categories. To achieve this goal, a general AsymNet is first trained, followed by fine-tuning for each clothing category to achieve category specific models. There are 14 models to be trained. In this section, we will introduce the approximate training of AsymNet.

To train a robust model, millions of training samples are usually needed. It is extremely time-consuming to train the AsymNet using traditional training strategy. Based on an intrinsic property of this application, that is, many positive and negative samples (i.e. shopping clothes) share the same clothing trajectories in the training stage, an efficient training method is proposed, which is summarized in Alg. 1.

Algorithm 1 Approximate Training Method.

Input: An AsymNet containing IFN, VFN and SNN, LSTM hidden states L, Convolutional features C.

Output: AsymNet

1: Sample N clothing trajectories;
2: Get the L=net_foward(VFN), C=net_foward(IFN);
3: Copy L $2 \times S$ times as $\hat{L}$, sent C and $\hat{L}$ to SNN;
4: Train SNN and compute $\bigtriangledown(SNN)$ as Eqn. 5;
5: Compute h_i and h_{ij} as Eqn. 7-8;
6: Train fusion nodes as Eqn. 10-11;
7: Train IFN using $\bigtriangledown(SNN)$;
8: Train VFT using $\bigtriangledown(VFN_u)$ as Eqn. 12;

Assume that the batch size of training is N and $2 \times S$ shopping images are sampled for a trajectory, where the numbers of positives and negatives are equal to S. In total, we have N clothing trajectories of videos and $2 \times S \times N$ clothing shopping images in each batch. To accelerate the similarity network training, the LSTM hidden states of N trajectories are copied $2 \times S$ times and sent them to the similarity network. To train the video feature networks, the gradient of clothing trajectory can be approximated as:

$$\bigtriangledown (VFN) = \frac{1}{2 \times S} \bigtriangledown (SNN) \tag{12}$$

Meanwhile, the gradient of image feature networks is $\bigtriangledown(SNN)$ as defined in Eqn. 5.

6. Experiment

In this section, we will evaluate the performance of individual component of AsymNet, and compare the proposed method with the state-of-the-art approaches.

6.1. Dataset and Metrics

Without proper datasets available for Video2Shop application, we collect a new dataset to evaluate the performance of identical clothing retrieval through videos. To the best of our knowledge, this is the first and the largest dataset for Video2Shop application. There are a number of online stores in e-commerce websites Tmall.com and Taobao.com, which sell the same styles of clothes appeared in movies, TV and variety shows. Accordingly, the videos and corresponding online clothing images are also posted on these stores. We download these videos from Tmall MagicBox, a set-top-box device from Alibaba Group, and the shots containing the corresponding clothes are manually extracted as

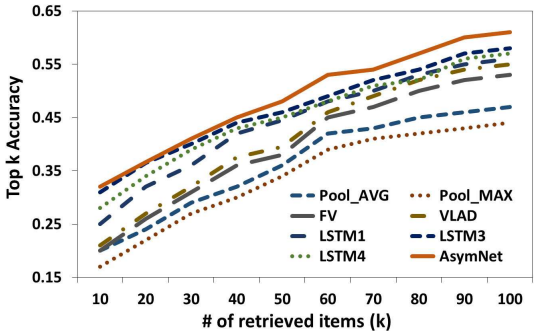

Figure 3. Performance Comparison of Representation Networks

the clothing trajectories. In total, there are 85,677 online clothing shopping images from 14 categories, 26,352 clothing trajectories extracted from 526 videos, and 39,479 exact matching pairs. The dataset information is listed in Table 1. We also collect similar matching pairs for the evaluation of similar retrieval algorithms.

In order to train the clothing detector, 14 categories of clothes are manually labeled, in which 2000 positive samples are collected per category from online images. Faster-RCNN [22] is utilized as the clothing detector, and the clothing trajectories are generated by Kernelized Correlation Filters (KCF) tracker [8]. The parameters used in Faster-RCNN and KCF are the same as the original version. Duplicate clothing trajectories are removed. Each clothing trajectory in our dataset is linked to exactly matched clothing images and they are manually verified by annotators, which form the ground truth. With an approximate ratio of 4:1, these exact matching video-to-shop pairs are split into two disjoint sets (training and testing sets), which are non-overlapped. Meanwhile, in order to reduce the impact of background and lead to more accurate clothing localization. Faster-RCNN is also used to extract a set of clothing proposals for online shopping images.

Evaluation Measure: Since the category is assumed to be known in advance, the experiments are performed within the category. Followed by the evaluation criterion of [16, 25], the retrieval performance is evaluated based on *top-k accuracy*, which is the ratio of correct matches within the top k returned results. Notice that once there is at least one exactly same product among the top 5 results as the query, which is regarded as a correct match in our setup. For simplicity, the weighted average is used for evaluation.

6.2. Performance of Representation Networks

In this subsection, we compare the performance of representation networks with other baselines. 1) Average pooling, 2) Max pooling, 3) Fisher Vector [21] and 4) VLAD [12]. We utilize 256 components for Fisher vectors and 256 centers for VLAD as common choices in [12, 23]. The PCA projections, GMM components of Fisher vectors, and K-means centers of VLAD are learned from approximately 18,000 sampled clothing regions in the training set. For these baselines, average pooling and max pooling are directly used on the CNN features of clothing trajectories. Fisher vector and VLAD are used to encode the CNN features of shopping images and clothing trajectories, respectively. The similarity is then estimated by single similarity network. In addition, the impact of different levels (1, 3 and 4 levels) of LSTM network is also investigated, denoted as LSTM1, LSTM3 and LSTM4, respectively. For LSTM based networks, the final output from the similarity feature network is used as the final matching result. The performance comparison is shown in Fig. 3.

From Fig. 3, we can see that the general performance is increased as k becomes larger, which means that it will be treated as a correct match once there is at least one exactly same item with the top k returned results. But we can also noticed that the performance of top 10 is still far from satisfactory, since it still a challenging task to match clothes appeared in videos to the online shopping images. There exists significant discrepancy between these cross-domain sources, including diverse visual appearance, cluttered background, occlusion, light condition, motion blur in the video, and so on.

The performance of average pooling is better than max pooling. Both Fisher Vector and VLAD have better performance than the average pooling representation. VLAD has slightly better performance than Fisher Vector. Overall, all LSTM based networks outperform pooling based methods. The proposed AsymNet achieves the best performance, which has significantly better performance than the other two pooling approaches. As the increase of the levels of LSTM network, the performance is firstly increased and then dropped when the number of levels is more than two. Our AsymNet adopts the two-level LSTM structure.

6.3. Structure Selection of Similarity Network

To investigate the structure of similarity network, we vary the number of levels and the fusion nodes in similarity network, while keeping all other common settings fixed. We evaluate two types of architectures: 1) Homogeneous branches: all fusion nodes have the same number of branches; 2) Varying branches: the number of branches is inconsistent across layers. For homogeneous setting, one-level flat structure with 32 fusion nodes to hierarchical structure with five levels (62 fusion nodes) are tested. For the varying temporal branches, we compare six networks with branches in increasing order: 4-8, 2-4-4, 2-2-2-4 and decreasing order: 8-4, 4-4-2, 4-2-2-2, respectively.

The performance of these architectures is shown in Fig. 4, in which the structure is represented in the form: #Level:#Branches in each level from leaves to root of the tree, connected with hyphen. From this figure, we can see that the overall performance is significantly improved as the

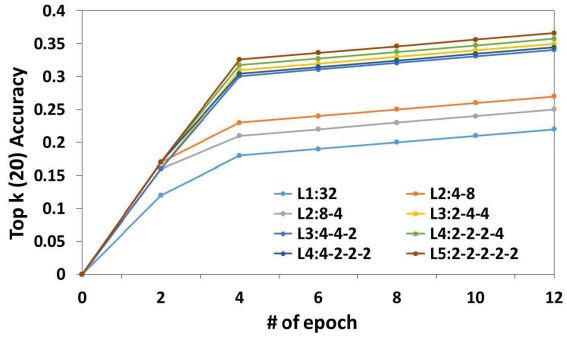

Figure 4. The top-20 retrieval accuracy (%) of the proposed AsymNet with different structures.

number of epoch increases. As the training proceeds, the parameters in the fusion nodes begin to grow in magnitude, which means that the weights of fusion nodes are becoming more and more reasonable. Meanwhile, the performance is significantly improved as the number of epoch increases. However, the improvement is not obvious after 4 epochs, since the weights of fusion nodes tend to be stable. The weight adjustment becomes subtle because the overall weights are optimized.

When one-level flat structure is adopted, it only has the leaves in the tree structure. The entire similarity network is reduced to a single averaged generalized linear models at the root of the tree. As the training proceeds, the parameters in the fusion nodes begin to grow in magnitude. When the fusion notes begin to take action, the performance of the system is boosted. We also notice that the general performance is increased when more levels of fusion nodes are involved. The boosting is pretty conspicuous for the first three layers. The improvement becomes minor when multi-level structure is formed. It indicates that the similar network becomes stable when the levels of fusion nodes are more than three.

6.4. Performance of Similarity Learning Network

In order to verify the effectiveness of our similarity network, we compare the performance of the proposed method with other methods when fusion nodes are not included. These baselines include: the final matching result is determined by the average (Avg) and the maximum (Max) of all similar networks, or the last (Last) similar network. In addition, the latest work KVM [30] is also considered, in which the key volume proposal method used in KVM is directly utilized to fuse the fc1 features in SNN. We formulate the similarity learning task as a binary classification problem. With that, the same loss function in KVM can still be used.

The top-20 retrieval performance comparison is shown in Fig. 5. From this figure, we can see that the performance of Avg is better than Max. Last has better performance than Avg and Max. The main reason is that the last hidden states learn the whole temporal information in the clothing

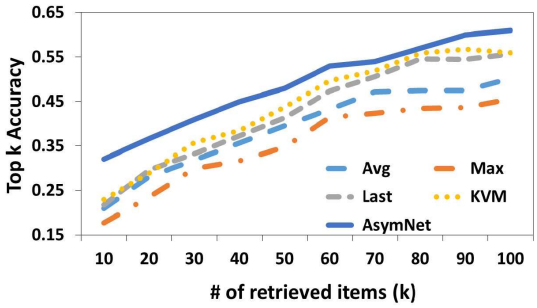

Figure 5. Performance of Similarity Learning Network

trajectories. The noise in clothing trajectories affects the performance of Avg and Max greatly. KVM considers the discriminative information may occur sparsity in a few key volumes, while other volumes are irrelevant to the final result. Although KVM is able to learn the most critical parts from clothing trajectories, it is too simple to consider the whole trajectory, in which different local viewpoints in trajectory are not well considered. The proposed AsymNet outperforms these baselines, which has significantly higher performance.

6.5. Comparison With State-of-the-art Approaches

To verify the effectiveness of the proposed AsymNet, we compare it with the following the state-of-the-art approaches: 1) **AlexNet (AL)** [17]: the activations of the fully-connected layer fc6 (4,096-d) are used to form the feature representation. 2) **Deep Search (DS)** [10]: it is an attribute-aware fashion-related retrieval system based on convolutional neural network. 3) **F.T. Similarity (FT)** [16]: category-specific two-layer neural networks are trained to predict whether two features extracted by the AlexNet represent the same product item. 4) **Contrastive & Softmax (CS)** [1]: it is based on the Siamese Network, where the traditional contrastive loss function and softmax loss function are used. 5) **Robust contrastive loss (RC)** [25]: multi-task fine-tuning is adopted, in which the loss is the combination of contrastive and softmax. For clothing trajectories in videos, we calculate the average similarity to gain the most similar shopping images. The cosine similarity is used for all these methods except FT.

The detailed performance comparison is listed in Table 1. AsymNet achieves the best performance for top-20 retrieval accuracy. It significantly outperforms AlexNet, in which the performance is almost doubled. The performance of AlexNet [17] and Deep Search [10] is unsatisfactory, which only use the convolutional features to retrieve images and do not learn the underlying similarity, The performance of two contrastive based methods (CS [1] & RC [25]) are slightly better than FT [16], since contrastive loss has a stronger capability to identify minor differences. RC has better performance than CS because it exploits the category information of clothing. For some categories having

Table 1. The top-20 retrieval accuracy (%) of the proposed AsymNet compared with state-of-the-art approaches. The notations represent the numbers of images (# I), video trajectories (# TJ), queries (# Q) and its corresponding results (# R).

Category	# I	# TJ	# Q	# R	AL [17]	DS [10]	FT [16]	CS [1]	RC [25]	AsymNet
Outwear	18,144	5,581	1,116	3,628	17.31	22.94	26.97	27.61	31.80	**42.58**
Dress	14,128	4,346	869	2,825	22.93	24.90	25.56	29.33	34.34	**49.58**
Top	7,155	2,201	440	1,431	17.45	24.83	25.26	29.14	32.94	**35.12**
Mini skirt	6,571	2,021	404	1,314	23.35	24.83	27.47	29.50	31.30	**32.48**
Hat	6,534	2,010	402	1,306	15.82	13.98	20.19	25.87	33.81	**35.12**
Sunglass	6,133	1,886	377	1,226	11.85	7.46	11.35	11.83	**12.26**	12.16
Bag	5,257	1,617	323	1,051	23.78	27.63	27.47	25.67	25.48	**36.82**
Skirt	4,453	1,370	274	890	19.79	25.06	22.44	24.50	24.43	**41.75**
Suit	3,906	1,201	240	781	18.65	25.18	19.72	25.29	26.60	**42.08**
Shoes	3,358	1,033	206	671	11.45	24.10	23.92	25.03	**27.58**	26.95
Shorts	3,249	999	199	649	11.15	5.99	13.90	14.84	**16.62**	13.74
Pants	2,738	842	168	547	17.57	22.54	25.77	29.49	28.36	**32.13**
Breeches	2,044	628	125	408	23.45	22.99	25.03	28.52	28.76	**48.28**
High shoots	2,007	617	123	401	12.05	13.11	14.57	15.46	**16.04**	14.94
Overall	85,677	26,352	5,266	17,128	18.36	21.44	23.47	25.73	28.73	**36.63**

Figure 6. Example with top-5 retrieval results of the proposed AsymNet. The difference in terms of detailed decorative patterns are labelled with red boxes.

no obvious difference in clothing trajectories, RC performs slightly better than AsymNet. Overall, our proposed approach demonstrates clearly better performance than these approaches. This is mainly because AsymNet can handle the temporal dynamic variety existing in videos, and it integrates discriminative information of video frames by automatically adjusting the fusion strategy.

Three examples with top-5 retrieval results of the proposed AsymNet are illustrated in Fig. 6, where the exact matches are marked with green tick. Relatively, it is easier to obtain the visually similar clothes, but it is much challenging to obtain the identical ones, especially the query is from videos. For the first two rows, these returned results are visually similar. However, some detailed decorative patterns are different, which are labelled with red boxes. In the last row, although the clothing style is the same, the color is different, so it will not be treated as the correct match.

6.6. Efficiency

To investigate the efficiency of the approximate training method, we compare it with traditional training procedure. All these experiments are conducted on a server with 24 Intel(R) Xeon(R) E5-2630 2.30GHz CPU, 64GB RAM and one NVIDIA K20 Tesla Graphic GPUs. In our experiment, only one sample is performed in inference, the image feature network processes 200 images/sec. The video feature network conducts 0.5 trajectories/sec and the similarity network performs 345 pairs/sec. The computation can be further pipelined and distributed for large-scale applications. The approximate training only costs 1/25 of the training time of traditional way. Meanwhile, the effectiveness of AsymNet is not influenced with the approximate training method. The training of our AsymNet model only takes around 12 hours to converge.

7. Conclusion

In this paper, a novel deep neural network, AsymNet is proposed to exact match clothes in videos to online shops. The challenge of this task lies in the discrepancy existing in cross-domain sources between clothing trajectories in videos and online shopping images, and the strict requirement of exact matching. This work is the first exploration of Video2Shop application. In our future work, we will integrate clothing attributes to further improve the performance.

8. Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61373121), and Program for Sichuan Provincial Science Fund for Distinguished Young Scholars (Grant No. 13QNJJ0149).

References

[1] S. Bell and K. Bala. Learning visual similarity for product design with convolutional neural networks. *ACM TOG*, 34(4):98:1–98:10, 2015.

[2] Z.-Q. Cheng, Y. Liu, X. Wu, and X.-S. Hua. Video ecommerce: Towards online video advertising. In *ACM MM*, pages 1365–1374, 2016.

[3] Z.-Q. Cheng, X. Wu, Y. Liu, and X.-S. Hua. Video ecommerce++: Towards large scale online video advertising. *IEEE Trans. on Multimedia*, 2017.

[4] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv*, 2014.

[5] G. Enderlein. Mccullagh, p., j. a. nelder: Generalized linear models. chapman and hall london new york 1983, 261 s., 16,. *Biometrical Journal*, 29(2):206–206, 1987.

[6] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. Matchnet: unifying feature and metric learning for patch-based matching. In *CVPR*, pages 3279–3286, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *TPAMI*, 37(9):1904–1916, 2015.

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *TPAMI*, 37(3):583–596, 2015.

[9] J. Huang, R. S. Feris, Q. Chen, and S. Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *ICCV*, pages 1062–1070, 2015.

[10] J. Huang, W. Xia, and S. Yan. Deep search with attribute-aware deep network. In *ACM MM*, pages 731–732. ACM, 2014.

[11] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[12] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311, 2010.

[13] M. I. Jordan. Hierarchical mixtures of experts and the em algorithm. In *Advances in Neural Networks for Control and Systems*, pages 1/1–1/3, 1994.

[14] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(2):181–214, 1994.

[15] Y. Kalantidis, L. Kennedy, and L.-J. Li. Getting the look: clothing recognition and segmentation for automatic product suggestions in everyday photos. In *ICMR*, pages 105–112, 2013.

[16] M. H. Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg. Where to buy it: Matching street clothing photos in online shops. In *ICCV*, pages 3343–3351, 2015.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, pages 1097–1105, 2012.

[18] X. Liang, L. Lin, W. Yang, P. Luo, J. Huang, and S. Yan. Clothes co-parsing via joint image segmentation and labeling with application to clothing retrieval. *TMM*, 18(6):1175–1186, 2016.

[19] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan. Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In *CVPR*, pages 3330–3337, 2012.

[20] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *CVPR*, pages 1096–1104, 2016.

[21] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, pages 1–8, 2007.

[22] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[23] J. Sánchez, F. Perronnin, T. Mensink, and J. J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 105(3):222–245, 2013.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[25] X. Wang, Z. Sun, W. Zhang, Y. Zhou, and Y. Jiang. Matching user photos to online products with robust deep features. In *ICMR*, pages 7–14, 2016.

[26] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, pages 4353–4361, 2015.

[27] W. Zaremba and I. Sutskever. Learning to execute. *arXiv*, 2014.

[28] J. Zbontar and Y. LeCun. Computing the stereo matching cost with a convolutional neural network. In *CVPR*, pages 1592–1599, 2015.

[29] B. Zhao, X. Wu, Q. Peng, and S. Yan. Clothing cosegmentation for shopping images with cluttered background. *IEEE Trans. on Multimedia*, 18(6):1111–1123, 2016.

[30] W. Zhu, J. Hu, G. Sun, X. Cao, and Y. Qiao. A key volume mining deep framework for action recognition. In *CVPR*, pages 1991–1999, June 2016.