Transfer Learning for Video Recognition with Scarce Training Data

Yu-Chuan Su, Tzu-Hsuan Chiu, Chun-Yen Yeh, Hsinfu Huang, Winston H. Hsu

Abstract—Unconstrained video recognition and Deep Convolution Network (DCN) are two active topics in computer vision recently. In this work, we apply DCNs as frame-based recognizers for video recognition. Our preliminary studies, however, show that video corpora with complete ground truth are usually not large and diverse enough to learn a robust model. The networks trained directly on the video data set suffer from significant overfitting and have poor recognition rate on the test set. The same lack-of-training-sample problem limits the usage of deep models on a wide range of computer vision problems where obtaining training data are difficult. To overcome the problem, we perform transfer learning from images to videos to utilize the knowledge in the weakly labeled image corpus for video recognition. The image corpus help to learn important visual patterns for natural images, while these patterns are ignored by models trained only on the video corpus. Therefore, the resultant networks have better generalizability and better recognition rate. We show that by means of transfer learning from image to video, we can learn a frame-based recognizer with only 4k videos. Because the image corpus is weakly labeled, the entire learning process requires only 4k annotated instances, which is far less than the million scale image data sets required by previous works. The same approach may be applied to other visual recognition tasks where only scarce training data is available, and it improves the applicability of DCNs in various computer vision problems. Our experiments also reveal the correlation between meta-parameters and the performance of DCNs, given the properties of the target problem and data. These results lead to a heuristic for meta-parameter selection for future researches, which does not rely on the time consuming meta-parameter search.

Index Terms—Video Recognition, Deep Convolution Network, Transfer Learning

I. INTRODUCTION

Unconstrained video recognition has become an important topic in terms of both research interest and application needs [1]. Traditional video recognition research mostly focuses on either recognition within specific domains such as human action recognition [2] or videos from limited sources such as news videos or movies [3], [4], because the majority of video collections were comprised of them, and videos from other sources are scarce for both research and applications. With the rise of the Internet and social media, however, people now generate and share a large volume of videos over the internet everyday, which surpasses videos generated from professional production in terms of both number and total length. These videos are hard to manage and analyze due to the volume and lack of annotation, and automatic recognition of these videos is becoming a promising solution for these problems.

Despite the importance of video recognition, less progresses are made compared with visual recognition on images. One possible reason is the lack of data; while there exist many different public data sets designed for different aspects of image recognition [5], [6], there are less video data sets with complete ground truth, and they are usually much smaller in scale compared with image data sets. The additional modality of audio and dimension of time also complicate the problem and introduce additional storage and computation overhead. One common approach for video recognition is to utilize the techniques in image recognition, which may be applied on either the video level or frame level [1]. Although some efforts are made on developing video specific features that consider the temporal dimension [7], [8], these features usually introduce significant computation overhead and are mostly used in action recognition, because the difference between consecutive frames embeds important information for action. General video recognition, on the other hand, does not neces-
sarily benefit from these temporal information. For example, we can identify a “Birthday Party” video by simply looking at non-consecutive keyframes, and the differences between consecutive frames obviously have little help.

Deep Learning as a learning paradigm has received much attention in the past decade and has been shown effective in various domains including natural language processing [9], speech recognition [10], etc. In computer vision, the great success of applying DCNs on the ImageNet data set has made it one of the most promising approaches for visual recognition [11], [12]. However, the usage of Deep Learning in general visual recognition problems are still rare compared with traditional methods, such as Bag-of-Words (BoW) [13] or Fisher Vector (FV) [14] with linear Support Vector Machine (SVM). In fact, many Deep Learning methods on visual recognition are only evaluated on data sets such as MNIST [15] or CIFAR [16] that contain images with much lower resolution than real photos, and the experiences obtained on these data sets are not directly applicable to real images.

There are several difficulties for applying DCNs on visual recognition in practice. The first one is its dire need for extremely large amount of labeled training data, while obtaining data with ground truth may be difficult in certain domains. This problem can be clearly seen in our experiments (cf. Section VI-C). The second one is the extensive requirement of computation power in the training phase. Previous results on ImageNet were made available only by high end parallel computation platforms such as advanced GPU or clusters with thousands of nodes [11], [12], [17]. Therefore, the evaluation of DCN on image recognition is limited, and many of its properties are unknown to the majority of the research community, which makes the usage of Deep Learning on visual recognition even more difficult.

In this work, we apply DCN on unconstrained consumer videos using the Columbia Consumer Video (CCV) data set [1]. Contrast to [18], [19], which extends DCN into time domain for action recognition, we treat the DCN as a frame-based recognizer and recognizes each frame independently. The recognition results of each frame are then aggregated using late fusion to determine the label of the video. The reason why we use only 2D convolution is that we target on video category recognition rather than action recognition, and the temporal dimension does not provide much information but introduces much higher computational cost as mentioned before. To avoid DCNs from overfitting the CCV training set, we perform transfer learning from images to videos where existing image data sets can provide enough diversity to avoid overfitting. The work flow is illustrated in Fig. [1] where we use the image data set to pre-train a network for initialization and augment the video data set. The result shows that, although being obtained from different sources and having a different visual appearance, the image data set helps DCNs to learn important visual patterns and greatly improves the performance on video recognition.

Our main contribution is that we successfully learn a DCN with reasonable performance using only scarce training data. Existing works on DCN rely on a large and diverse training set, while the available training data in many problems are actually scarce to the point that DCN may totally fail. With the transfer learning approach, we can apply DCN in more general visual recognition problems and data sets. We show that transfer learning can be done not only in the generative DBN [20], [21], but also in the supervised discriminative DCN. Other contributions include that we point out that lack of training data is one of the most important problems for DCN, and our systematic study on the meta-parameters (e.g., image resolution, depth, training data diversity) of DCN provide future research the hints on how to configure the network.

The rest of the paper is organized as follows. In Section II we discuss the concept of DCN and related works. In Section III we describe the proposed method for transfer learning. In Section IV we describe the data sets we used in this work and their properties, and we summarize the DCN architectures we used in Section V. Our preliminary studies on the DCN architecture and the importance of training data diversity are in Section VI. Experiment results for transfer learning are in Section VII. Finally, we summarize the work in Section VIII.

II. DEEP CONVOLUTION NETWORK FOR VISUAL RECOGNITION

The main challenge of Deep Neural Network is that it is hard to train, due to the complexity introduced by the depth and the large number of learnable parameters. To overcome the difficulty, a greedy layerwise pre-training is applied before training on the labeled data [22]. Pre-training is an unsupervised learning process that learns the hidden layers one-by-one, where each layer is learned by either maximizing the input likelihood or minimizing reconstruction error. The entire network is then fine-tuned with the labeled training data using gradient descent. The pre-training is believed to learn a better representation of the input, which facilitates the following supervised learning and leads to better generalizability [23]. The unsupervisedly learned networks can even be used for transfer learning [20], [21], which indicates that the networks are capable of learning universal features that are not specific to the training set.

In computer vision, the most popular Deep Learning architecture is the DCN [15]. It can be viewed as a specialized Multi-Layer Perceptron (MLP) with a manually crafted architecture and regularization. MLP can be formulated as a series of affine transform followed by non-linear dimension-wise transform:

$$h^{l} = \begin{cases} g(W^{(l)}x) & \text{if layer } l = 1 \\ g(W^{(l)} f^{(l-1)}) & \text{if layer } l > 1. \end{cases}$$

(1)

Given the activation function $g$ and loss function, the learnable parameters $W^{(l)}$ are learned by gradient descent. One significant problem of applying MLP directly to general images is that the image signal usually contains tens of thousands of dimensions, which leads to extremely large affine transform matrices $W$. Without a clever learning process or regularization, the model will be extremely prone to overfitting.

To avoid overfitting, convolution was introduced as a regularization on $W$ based on the prior knowledge on the human
visual system and the image signal [13], [24]. Convolutions are usually formulated as
\[ h_{k,i,j}^l = g \left( \sum_{k', r, s=1}^{N_W} \tilde{W}_{l,k,i,j}^{l,k,r,s} h_{k',i,j}^{l-1} + \frac{\sum_{r,s} \tilde{W}_{l,k,i,j}^{l,k,r,s} - \sum_{r,s} \tilde{W}_{l,k,i,j}^{l,k,r,s}}{N_W} \right), \] (2)
where both the input and output are three dimensional tensors, and \( \tilde{W}_{l,k,i,j}^{l,k,r,s} \) stands for the \( k \)-th convolution kernel in the \( l \)-th layer. The width of the kernel is denoted by \( N_W \). It can be reformulated as
\[ h_{k,i,j}^l = g \left( \sum_{k', i', j'=1}^{N_H} \tilde{W}_{l,k,i,j}^{l,k,i',j'} h_{k',i',j'}^{l-1} \right), \] (3)
where \( N_H \) stands for the width of \( h_{k,i,j}^{l-1} \) (assume squared input). The convolution kernel \( \tilde{W}_{l,k}^{l,k} \) is redefined over the entire input space at each position \((i,j)\) as \( W_{l,k,i,j}^{l,k} \) with the following two constraints:
\[ W_{l,k,i,j}^{l,k,i',j'} = 0, \text{ if } \| i - i' \| > \frac{N_W}{2} \text{ or } \| j - j' \| > \frac{N_W}{2}, \] (4)
and
\[ W_{l,k,i,j}^{l,k,i',j'} = W_{l,k,i',j'}^{l,k,i,j'} \forall r, s. \] (5)
By vectorizing \( h_{k,i,j}^l \) over \( i, j, k \) and \( W_{l,k,i,j}^{l,k,i',j'} \) over \( i, j, k \) and \( i', j', k' \), respectively, \( W_{l,k,i,j}^{l,k,i',j'} \) will reduce to the two dimensional tensor \( \mathbf{W}^{(l)} \) in Eq. 1 as shown in Fig. 2 with two additional constraints: local response as in Eq. 4 and tied weight as in Eq. 5. The local response constraint enforces that only a small part of \( \mathbf{W}_{l,k,i,j}^{l,k,i',j'} \) can be non-zero, which means that \( h_{k,i,j}^l \) can only depend on a small fraction of \( h_{l}^{l-1} \) while it can have arbitrary dependency in MLP. This reduces the number of learnable parameters with the heuristic that local patterns are important for both visual recognition and human vision system. Tied weight further reduces the number of learnable parameters by enforcing kernels \( W_{l,k,i,j}^{l,k,i,j} \) with same \( l,k \) to share the same value.

Despite the convolution regularizations, DCNs still have a large amount of learnable parameters and are still prone to overfitting. While unsupervised pre-training is popular in DBN to improve the generalizability, DCN as a fully supervised learning algorithm does not utilize similar learning techniques. Therefore, a large training set with high diversity is necessary to learn a DCN, yet such data sets are not easily obtainable in the past. Also, the computation cost of convolution made training on moderate resolution images (200x200, etc.) a formidable task until very recently. The most impressive breakthrough of DCN on visual recognition comes from its superior performance on the ImageNet data set, where A. Krizhevsky et al. [11] and Q.-V. Le et al. [12] independently report a significant performance improvement over traditional image features. While the network architecture used by each group is significantly different from each other, the key factors for the success of both groups are the extremely large training set as well as the parallel acceleration that makes learning possible. This partially explains the resurgence of Neural Networks, which had been developed long before they received much attentions: it is only recently that such large training sets as well as computation power have become available for the network to be learnable.

III. TRANSFER LEARNING WITH DEEP CONVOLUTION NETWORK

In this section, we describe the transfer learning approaches we apply to utilize image information in video recognition. Transfer learning helps to learn a more generalizable DCN. This is important because DCNs are prone to overfitting, especially when only scarce training data is available. While increasing training data helps to solve the problem, there are cases where collecting new data with complete ground truth is difficult. Transfer learning solves the problem by using labeled data from other domains where a large number of training data is available to improve the network.

The goal is similar to the pre-training process in DBN and Stacked Auto-Encoder (SAE) in the sense that it improves the generalizability by learning a better intermediate representation [25]. A good representation does not necessarily optimize the loss during training, which is done by supervised backpropagation in Deep Neural Network; instead, it should capture important patterns that are general to all data. While DBN and SAE achieve this by performing unsupervised pre-training before supervised training, we learn the representations from other domains and then optimize the representation through transfer learning.

A. Mixing Data Sets

The first approach for transfer learning is to mix image data sets with the video data set. Or equivalently, we train a DCN that simultaneously recognizes images from the image data sets and frames from the video data set, as illustrated in Fig. 3. This is made possible since the intermediate layers are shared by all output units in Neural Networks and may
benefit from the training data of other classes. If the lower layers can learn general visual patterns that are shared across different data sets, the additional classes and training data can help to learn better low and middle level features and avoid overfitting.

B. Transfer Mid-level Features

The second approach for transfer learning is to transfer the learned feature from images to videos by initializing the learnable parameters of DCNs using the network pre-trained on image domain. The approaches can also be considered as a supervised pre-training, which is analogous to the unsupervised pre-training in DBN or SAE. The network is then fine-tuned using the target data set to optimize the features for the target data set or domain. This approach is motivated by the fact that image features rely on important visual patterns that are shared across all natural images, so they can be used to characterize images outside the training set. Because the convolution kernels in DCN learn important low level visual patterns in natural images [12], these convolution kernels serve as the low level features used in traditional visual recognition, and they may also be similar across all natural images and data sets. This can be seen in Fig. 4 where some of the first layer kernels are very similar even if they are learned from two non-overlapping data sets. Therefore, the network parameters learned from one data set may be useful for another data set.

The fine-tuning step is performed to further optimize the feature. This is especially important for the higher layers in DCN, such as the fully connected layer, because these layers capture more complex patterns [12] that may not generalize well to other domains. For example, while lines and corners are common to all natural images, a pattern of face will appear in only specific domains and may not be useful in all problems. Because training neural networks as an optimization problem is non-convex, different initial values of the learnable parameters will lead to different networks. In fact, it is known that good initialization will lead to better performance [25]. Therefore, by initializing the learnable parameters with learned visual patterns, we should be able to learn more generalizable networks, as suggested by the pre-training process.

While previous research focus on unsupervised pre-training for better representation learning, several parallel works have utilize the supervised pre-training approaches to address the same lack-of-training-data problem [19], [26]–[28]. In most of these works, DCNs learned on the ILSVRC data set are applied on other static visual recognition problems such as object recognition on the Pascal VOC data set. Our work is complementary to them in that we show the same approach can be well generalized in both source and target domain. The data set we used for supervised pre-training is a weakly labeled data set that uses image tags as ground truth, and the DCNs are then apply to Youtube videos rather than static images. Also, some of our discoveries and conclusions are consistent to those in [19], [28] as described below.

IV. DATA SET

In this section, we describe the data sets we used in this work. Because DCN has many meta-parameters entangled in a single learning process, instead of performing cross validation and grid search for these parameters, we select them based on the empirical experience from other data sets. We use three different data sets including the CCV [1] video data set, and we will describe their properties in this section. We also show the example images in Fig. 5.

ILSVRC2012, ILSVRC2012 [29] is an 1k classes subset of ImageNet used in ImageNet Large Scale Visual Recognition Challenge. While the 1k classes may be either internal or leaf nodes of the ImageNet ontology, they are guaranteed to be non-overlapping. The entire data set comprises of 1.5 million manually labeled images, where 1.2 million of them are used as the training set and 50,000 images are used as the validation
set; we report the performance on the validation set. The data set is characterized by its large number of classes and is often used to evaluate visual recognition system with large semantic space. Although images in the data set are usually clear and un-occluded, which is often considered an over simplification of real images, the ILSVRC subset is nevertheless the most widely adopted data set for large scale visual recognition.

**Yahoo!-Flickr.** Yahoo!-Flickr is a data set released by Yahoo!. The data set contains images with 10 different classes, where the classes are defined by the popular tags in Flickr and the ground truths are obtained directly from Flickr tags. Therefore, the data set is a weakly labeled data set constructed without any human annotation. This makes the data set very different from most visual recognition benchmark such as ImageNet, which requires intensive human involvement during construction. Each class contains 150k images for training and 50k images for testing. While being another million-scale visual recognition benchmark other than ILSVRC, the Yahoo!-Flickr data set is characterized by its large intra-class variations. The images in each class may be visually very diverse or even share no visual similarity at all. Also, the Yahoo!-Flickr data set is known to be noisy because of user tagging behavior [30].

**CCV.** Columbia Consumer Video (CCV) Database [1] is a video data set that targets on unconstrained consumer video content analysis. The data set contains 9,317 manually labeled Youtube videos with 20 semantic categories, where each video may belong to arbitrary number of categories; in other words, the data set is a multi-label data set instead of a multi-class data set. The 9,317 videos are divided into an equal-sized train and test set, and the general evaluation protocol adopts binary relevance multi-label classification on the 20 semantic categories with mean average precision (MAP) criterion.

To learn a DCN for the video data set, we first perform uniform sampling on the training set to transform the video data set into an image data set. We treat each frame as an independent sample, and we define the 21st class as the null class that contains samples without any semantic category. In the testing phase, we perform recognition on the keyframes independently and use late fusion to combine the confidence score of different frames.

### V. Network Architecture

In this section, we describe the network architecture and other parameters we used for our experiments. We will refer to these architectures based on the input image resolution and number of convolution layers in the following sections.

**2-convolution-layer network.** For networks with two convolution layers, we adopt an architecture similar to that of LeNet5 [15]. The network is composed of 5 layers; the first two layers are convolution layers, and the third and fourth layer are fully connected layers. The last layer is a classification layer. The structure of convolution layers depend on the image resolution. For 256x256 and 128x128 input image resolution, the first convolution layer has 64 kernels of size 11x11x3 with a stride of 4 pixels. The second convolution layer has 128 kernels of size 5x5x64. Max pooling with 2x2 region is performed after each convolution layer. The third layer, or the first fully connected layer has 4,096 neurons; the second fully connected layer has 1,024 neurons for Yahoo!-Flickr data set and 2,048 for ILSVRC2012. The structure of fully connected layers are kept the same over all input image resolution and other parameters we used for our experiments. We will refer to these architectures based on the input image resolution and number of convolution layers in the following sections.
convolution layers have stride 1 and are followed by 2x2 max pooling, and Rectified Linear Unit (ReLU) activation is used in all the network layers.

3-convolution-layer network. The 3-convolution-layer network is modified from 2-convolution-layer network, with the following differences: for 256x256 and 128x128 input image resolution, the first convolution layer now uses kernel of size 7x7x3 with a stride of 3; the third convolution layer has 320 kernels of size 3x3x128, and no max pooling is performed on the output of the third convolution layer. The second convolution layer and fully connected layers are the same as the 2-convolution-layer network with different input size. For 64x64 input image resolution, the first convolution layer consists of 32 kernels of size 5x5x3, the second layer consists of 64 kernels of size 5x5x32, and the third layer has 120 kernels of size 3x3x64; max pooling is performed after the first and second convolution layer. For 32x32 input image resolution, the convolution layer structures are similar to that of 64x64 input image resolution, except we perform max pooling only after the first layer, otherwise the third layer will be too small.

4-convolution-layer network. The 4-convolution-layer network is similar to the architecture used in [11] with some simplification. The first convolution is the same as that in the 2-convolution-layer network. The second convolution layer has 128 kernels of size 5x5x64. The third convolution layer has 192 kernels of size 3x3x128, and the fourth convolution layer has 192 kernels of size 3x3x192. Note that we do not perform response normalization and grouping on the convolution kernels as in [11], and our max pooling is performed over 2x2 regions without overlapping.

The Caffe package [31] is used for learning the networks. The learning process is similar to that in [11], where we crop and mirror the images to produce more training samples. Images of size 256x256 are cropped to a size of 227x227, 128x128 to 116x116, 64x64 to 56x56 and 32x32 to 28x28. The same dropout ratio, initial learning rate and momentum as in [11] are used.

VI. EXPERIMENT – NETWORK CONFIGURATION SELECTION

One difficulty of applying DCN in visual recognition is the large amount of meta-parameters in the model. While learning algorithms such as SVM usually have only 1~2 meta-parameters in practice and can be selected using cross-validation, DCN has at least an order of magnitude more meta-parameters and these meta-parameters can’t be determined independently. Therefore, the time consuming training process makes it computationally infeasible to perform cross validation for the meta-parameters. Although [32] suggests random search over grid search, it still relies on cross validation for parameter search. Instead, we believe that by studying the correlation between the properties of the target problem and the optimal or sub-optimal parameters, we can learn a heuristic that gives us a reasonable range of meta-parameters based on the problem itself, rather than the computation intensive parameter-search procedure.

To determine the meta-parameters for the frame-based video recognizer, we study the effect of meta-parameters on performance using Yahoo!-Flickr and ILSVRC2012 data sets. We set the meta-parameters using these experiences instead of performing cross-validation. In particular, we study the image resolution for input image and the required depth of the network, which have the greatest effect on computation cost of DCN. We also study how to sample the frames from video, which manifests the fact that a large training set with high diversity is necessary for training DCN.

A. Image Resolution

Early experiments of DCN are mostly based on data sets with very low resolution images. While it is claimed that these thumbnail images are still human recognizable [33], it is obvious that images with higher resolution contain more detailed information that may be helpful for visual recognition. In fact, most existing visual recognition benchmarks as well as real image corpora such as Flickr are comprised of images with much higher resolution [5]. [6], and experiments on thumbnail images do not approximate real images. However, higher resolution introduces higher computational cost, which grows roughly quadratically in DCN. Therefore, we try to investigate whether high resolution images are necessary for DCN in general visual recognition.

To study the performance of DCN under different image resolutions, we resize the images of Yahoo!-Flickr and ILSVRC2012 into four different resolutions ranging from 256x256 to 32x32. We then train DCNs with either 2 or 3 convolution layers on each resolution. The results are in Table I, where the two data sets show different responses with

| Res.     | Depth | ILSVRC2012 | Yahoo!-Flickr |
|----------|-------|------------|---------------|
|          |       | Top-1      | Top-5         | Accuracy | MAP     |
| 32x32    | 2-conv.| 0.26       | 0.47         | 0.48     | 0.46    |
|          | 3-conv.| 0.23       | 0.41         | 0.43     | 0.43    |
| 64x64    | 2-conv.| 0.31       | 0.55         | 0.51     | 0.50    |
|          | 3-conv.| 0.31       | 0.55         | 0.49     | 0.47    |
| 128x128  | 2-conv.| 0.39       | 0.61         | 0.51     | 0.51    |
|          | 3-conv.| 0.39       | 0.61         | 0.51     | 0.51    |
| 256x256  | 2-conv.| 0.40       | 0.64         | 0.54     | 0.53    |
|          | 3-conv.| 0.46       | 0.71         | 0.56     | 0.56    |
respect to image resolution. For the ILSVRC2012 data set, the performance is very sensitive to the image resolution, and we can consistently obtain 10% ~ 15% relative improvement by doubling the resolution. Also note that 3-convolution-layer network has worse performance than 2-convolution-layer network with 32x32 input image resolution, which implies the limit on depth imposed by image resolution. In fact, we have to abandon max pooling after the second convolution layer of the network to have a large enough hidden layer, otherwise the network will be nearly unlearnable and have extremely bad performance. The Yahoo!-Flickr data set, on the other hand, shows only moderate performance degradation when reducing the resolution, and we can achieve reasonable performance even with 32x32 thumbnail images.

The difference of the two data sets stems from the fact that they are designed for different purpose and have different properties. In particular, while the ILSVRC2012 data set is designed for object recognition where the object may be present in only a small part of the image, the Yahoo!-Flickr data set is designed for tag prediction, where the tags are mostly high level concepts that describe the entire image. The results indicate that we may use smaller images without significant loss of performance yet reduce the computational cost quadratically. Nevertheless, increasing the resolution consistently yields better performance and enables the usage of deeper networks, so we use 256x256 resolution in following experiments.

B. Depth of Architecture

In this section, we compare the performance of DCNs with different numbers of convolution layers. Previous works on ILSVRC use single depth for their networks [11], [12]. Although they mentioned a significant performance degradation with less layers [11], it is unclear whether the same conclusion holds across all situations. Since adding layers in the network significantly increases the computational cost, we would like to use networks as shallow as possible if the additional layers have no or even negative contributions on the performance.

To evaluate the effect of different depths on the performance, we learn convolution networks with 2~4 convolution layers on Yahoo!-Flickr data set with 256x256 image resolution. The results are in Table [I] 3-convolution-layer network turns out to have the best performance, although the performance gain of the 3rd convolution layer is relative minor. We also carry out the same experiment on ILSVRC2012. The performance gain of the additional convolution layer is more significant than that in Yahoo!-Flickr, which achieves 10% relative improvement. Because our target CCV data set is more similar to Yahoo!-Flickr data set in terms of image content class number, 2-convolution-layer should have reasonable performance as justified in Table [IV]

C. Training Data Number and Diversity

In this section, we discuss how to sample frames from videos for the DCN. Based on the results on ILSVRC2012 and Yahoo!-Flickr data sets, we train a 2-convolution-layer network as a frame level recognizer. Our first attempt is to use the keyframes for training, where a total of 8,508 keyframes from the CCV training set are used. These keyframes are resized to 256x256 resolution for training. The resultant performance, however, is poor with significant overfitting. Our postulation is that a large training set is necessary for learning a robust DCN. We choose 20k samples per class for comparison because the performance of linear SVM saturates terms of both image content and number of categories and should benefit less from the additional depths.

| Depth | 2-layers | 3-layers | 4-layers |
|-------|----------|----------|----------|
| Yahoo!-Flickr | 0.535 | 0.560 | 0.491 |
| ILSVRC2012 | 0.534 | 0.559 | 0.478 |

| Cycles | 5 | 10 | 20 |
|--------|---|----|----|
| Training Size | 20k | 150k | 20k | 150k | 20k | 150k |
| Loss (Train) | 0.62 | 1.27 | 0.33 | 1.30 | 0.19 | 1.12 |
| Loss (Test) | 1.65 | 1.54 | 1.67 | 1.49 | 1.72 | 1.44 |
| Accuracy | 0.44 | 0.51 | 0.44 | 0.52 | 0.43 | 0.54 |

In this section, we discuss how to sample frames from videos for the DCN. Based on the results on ILSVRC2012 and Yahoo!-Flickr data sets, we train a 2-convolution-layer network as a frame level recognizer. Our first attempt is to use the keyframes for training, where a total of 8,508 keyframes from the CCV training set are used. These keyframes are resized to 256x256 resolution for training. The resultant performance, however, is poor with significant overfitting. Our postulation is that a large training set is necessary for learning a robust DCN. We choose 20k samples per class for comparison because the performance of linear SVM saturates terms of both image content and number of categories and should benefit less from the additional depths.

| Depth | 2-layers | 3-layers | 4-layers |
|-------|----------|----------|----------|
| Yahoo!-Flickr | 0.535 | 0.560 | 0.491 |
| ILSVRC2012 | 0.534 | 0.559 | 0.478 |

To verify the postulation, we study the effect of training set size on the Yahoo!-Flickr data sets. The reason why we choose Yahoo!-Flickr is that the study of performance dependency on training set size is part of the goal the data set is designed for. The ILSVRC2012 data set, on the other hand is not suitable for the study, because reducing the training set may either lead to very few samples for some classes or change the ratio of different classes.

We train 2-convolution-layer networks on Yahoo!-Flickr using both 20k and 150k training samples for each class. We choose 20k training samples for comparison because previous results show that the performance of linear classifier saturates with 20k training samples per-class. The results are shown in Table [III] which clearly shows that smaller training set leads
and video data sets. In practice, we subsample roughly the frame recognizer.

Within a video are then averaged to predict the category of the frame recognition. The recognition results of all the keyframes videos, we extract only keyframes from the video and perform images and train a image recognizer for the frames. For test fps uniform sampling. We treat these frames as independent learning approach to avoid overfitting.

Therefore, we use 1-fps sampling to produce the training data set. The label of each video is much smaller than image data sets such as Yahoo!-Flickr, where there are nearly no high frequency signals in the kernels. By mixing CCV with Yahoo!-Flickr, we introduce high frequency kernels into the network. This may explain why training DCN on CCV only yields significant overfitting and poor performance: it ignores informative signals.

to significant overfitting and therefore poor performance. For the CCV data set, we try to increase the number training data by uniform sampling on the video. We compare the results using 1-fps sampling and 4-fps sampling, which leads to roughly 400k training samples and 1.6 million training samples respectively. Although 4-fps leads to a much larger training set, experiment results show that the performance using 4-fps are nearly identical with 1-fps while it takes much more storage and computation. The reason is that 4-fps sampling results in a large data set with small visual diversity, and the redundant training data are not helpful for learning. Therefore, we use 1-fps sampling to produce the training data from the CCV data set.

VII. EXPERIMENT – VIDEO RECOGNITION WITH TRANSFER LEARNING

In this section, we examine the proposed transfer learning approach on CCV data set. Although uniform sampling on videos generates much more training samples than keyframes, there are still only 0.4 million frames for training, which is much smaller than image data sets such as ILSVRC2012 and Yahoo!-Flickr. Therefore, we adopt the proposed transfer learning approach to avoid overfitting.

The entire workflow of video recognition is as follows: first, we sample video frames from the training set video using 1-fps uniform sampling. We treat these frames as independent images and train a image recognizer for the frames. For test videos, we extract only keyframes from the video and perform frame recognition. The recognition results of all the keyframes within a video are then averaged to predict the category of the video. The following experiments are performed by replacing the frame recognizer.

A. Mixing Data Sets

In this section, we examine the approach of mixing image and video data sets. In practice, we subsample roughly the same number of images as video frames (0.4 million) from the Yahoo!-Flickr data set and mix these images with the CCV data set to create a new training set. The label of each sample is kept the same as their original label. The results are shown in Table [V]. Although the two data sets are from very different sources, mixing the image from Yahoo!-Flickr indeed improves the performance. This may be explained by the first convolution layer kernels, as in Fig. [6] which shows that by mixing images into video data set, the network successfully learns the high frequency signals that are ignored when training the network on CCV only. These signal are known to be important, and ignoring them may degrade the generalizability.

B. Transfer Mid-level Features

In this section, we examine the approach of transferring mid-level features. In practice, we initialize the network for CCV by the network trained on Yahoo!-Flickr and ILSVRC2012 respectively. During training, we either update all the parameters in the network or update only the fully connected layers and keep the convolution kernels unchanged. The reason why we keep the convolution kernels unchanged is to reduce the number of learnable parameters to avoid overfitting as motivated by the convolution regularization. If the lower layer convolution kernels do learn important patterns such as lines or corners, they should be similar and reusable over different data sets, so keeping them unchanged should not degrade the performance. The result are in Table [IV]. Initializing the network with pre-trained networks improves the performance significantly, and updating only the fully connected layers is better than updating all parameters. This indicates that when the training data is not enough, avoiding to update the low level features in DCNs will reduce the overfitting problem and lead to better performance. These results are identical with that in [19], which shows that update only the fully connected layers lead to better recognition accuracy, although we use totally different data and different network structures. They are also consistent with [28], which shows that fully connected layers in DCNs are less generalizable than convolution layers and therefore require fine-tuning. Note that the networks learn different sets of convolution kernels using different initialization, as shown in Fig. [7] while both of them achieve good performance.

Finally, we combine the second and third approaches. That is, we initialize the network with the one trained on the Yahoo!-Flickr data set and fine-tune the network with both images and video frames. The recognition performance is further improved by the combination. It is worth noting that the Yahoo!-Flickr data set is a weakly labeled data set that uses only Flickr tags rather than the expensive human annotation for ground truth, and the only annotated data in the entire learning process is the video data set, which requires much less human intervention and annotation overhead. Although the Yahoo!-Flickr data set is known to be noisy, it provides as good initialization as the ILSVRC2012 data sets.
TABLE IV

| Depth | 2-convolution-layers | 5-convolution-layers |
|-------|----------------------|----------------------|
| Initialization | Random | Yahoo!-Flickr | ILSVRC2012 | Random |
| Training Set | CCV | CCV + Yahoo!-Flickr | CCV | CCV + Yahoo!-Flickr | CCV + Yahoo!-Flickr |
| Update Policy | FC + CONV | FC + CONV | FC + FC | FC + FC | FC + FC |
| MAP | 0.445 | 0.469 | 0.484 | 0.497 | 0.494 | 0.499 | 0.489 | 0.490 | 0.469 | 0.482 |

![Yahoo!-Flickr → CCV](image1)

![ILSVRC2012 → CCV](image2)

Fig. 7. The first convolution layer kernels of networks using a pre-trained network as initialization. Although the kernels are both learned from the CCV data set, they show different visual patterns which are more similar to their initialization points respectively. Note the clear 1-to-1 correspondence between the kernels above and those in Fig. 3. In fact, most of the kernels do not change significantly after fine-tuning with CCV data sets, which indicates the patterns learned from either Yahoo!-Flickr or ILSVRC2012 are helpful for recognizing CCV videos, i.e. data sets of different domains.

VIII. CONCLUSIONS

In this work, we apply DCNs on unconstrained video recognition, where the networks are used as frame-based recognizers. Our preliminary study shows that one of the most significant obstacles for training a DCN is the requirement for a large amount of training samples. Without enough training samples, the networks are highly prone to overfitting which leads to poor performance. This problem is especially important for video recognition, because videos with ground truth are scarce and hard to obtain.

To overcome the problem, we train DCNs with transfer learning from images to videos. The image corpus can be weakly labeled, which is widely available in various social media such as Flickr and Instagram. These rich image samples can help to learn more robust recognizers on the video frames, even though they are from different domains. The transfer learning process makes training DCN with scarce training data possible, where we achieve reasonable performance using only 4k videos for training. Because weakly labeled data sets are good enough for supervised pre-training without harming the performance, we are exempt from the requirement of collecting and annotating a large data set as in previous researches in DCN.

Our preliminary study also reveals the correlation between meta-parameters and performance given different data set properties. In particular, we study the effect of depth and image resolution, because these factors have significant impact on the computation cost. The results indicate that high resolution images always yield better performance, but it is more important for object level recognition compared with scene level recognition. The results also show that additional depths in the network may not always be helpful for performance; in fact, deep networks may sometimes perform worse than shallow ones. These studies not only help us to select the meta-parameters used for video recognition, but also provide hints for future researcher and facilitate the research in DCN.

In our future work, we would like to investigate the possibility of unsupervised pre-training for DCN. Our current approach still requires a large labeled corpus, and we would like to reduce the requirement. We are also interested in how to reduce the computational cost of DCN, where the current learning algorithm may take weeks or even months to train a single network and is unrealistic for real applications.

REFERENCES

[1] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui, “Consumer video understanding: A benchmark database and an evaluation of human and machine performance,” in ICMR, 2011.

[2] J. M. Chauvet, E. J. Carmona, and A. Fernández-Caballero, “A survey of video datasets for human action and activity recognition,” Comput. Vis. Image Underst., vol. 117, no. 6, pp. 633–659, 2013.

[3] A. F. Smeaton, P. Over, and W. Kralj, “Evaluation campaigns and trecvid,” in MIR, 2006.

[4] I. Lattev, M. Marsalek, C. Schmid, and B. Roszenfeld, “Evaluation campaigns and trecvid,” in MIR, 2006.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, “Imagenet: A large-scale hierarchical image database,” in CVPR, 2009.

[6] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba, “Sun database: Large-scale scene recognition from abbey to zoo,” in CVPR, 2010.
[7] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional sift descriptor and its application to action recognition,” in *Proceedings of the 15th International Conference on Multimedia*, 2007.

[8] A. Kläser, M. Marszalek, and C. Schmid, “A spatio-temporal descriptor based on 3d-gradients,” in *BMVC*, 2008.

[9] T. Deselaers, S. Hasan, O. Bender, and H. Ney, “A deep learning approach to machine transliteration,” in *Proceedings of the Fourth Workshop on Statistical Machine Translation*, 2009.

[10] A. Mohamed, G. E. Dahl, and G. Hinton, “Acoustic modeling using deep belief networks,” *Trans. Audio, Speech and Lang. Proc.*, vol. 20, no. 1, pp. 14–22, 2012.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *NIPS*, 2012.

[12] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng, “Building high-level features using large scale unsupervised learning,” in *ICML*, 2012.

[13] J. Sivic and A. Zisserman, “Video google: a text retrieval approach to object matching in videos,” in *ICCV*, 2003.

[14] F. Perronnin et al., “Improving the fisher kernel for large-scale image classification,” in *ECCV*, 2010.

[15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[16] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” *Computer Science Department, University of Toronto, Tech. Rep.*, 2009.

[17] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q. V. Le, and A. Y. Ng, “Large scale distributed deep networks,” in *NIPS*, 2012.

[18] S. Ji, W. Xu, M. Yang, and K. Yu, “3d convolutional neural networks for human action recognition,” *PAMI*, vol. 35, no. 1, pp. 221–231, 2013.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *CVPR*, 2014.

[20] X. Glorot, A. Bordes, and Y. Bengio, “Domain adaptation for large-scale sentiment classification: A deep learning approach,” in *ICML*, 2011.

[21] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in *ICML*, 2009.

[22] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[23] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, “Why does unsupervised pre-training help deep learning?” *JMLR*, vol. 11, pp. 625–660, 2010.

[24] H. Lee, C. Ekanadham, and A. Y. Ng, “Sparse deep belief net model for visual area v2,” in *NIPS*, 2007.

[25] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” in *ICML*, 2013.

[26] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep convolutional activation feature for generic visual recognition,” *arXiv preprint arXiv:1310.1531*, 2013.

[27] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and transferring mid-level image representation using convolution neural networks,” in *CVPR*, 2014.

[28] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *CVPR*, 2014.

[29] J. Deng et al., “Large scale visual recognition challenge 2012.” [Online]. Available: [http://www.image-net.org/challenges/LSVRC/2012/](http://www.image-net.org/challenges/LSVRC/2012/)

[30] Y.-C. Su, T.-H. Chiu, G.-L. Wu, C.-Y. Yeh, F. Wu, and W. Hsu, “Flickr-tag prediction using multi-modal fusion and meta information,” in *ACM MM*, 2013.

[31] Y. Jia, “Caffe: An open source convolutional architecture for fast feature embedding,” 2013. [Online]. Available: [http://caffe.berkeleyvision.org/](http://caffe.berkeleyvision.org/).

[32] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *JMLR*, vol. 13, pp. 281–305, 2012.

[33] A. Torralba et al., “80 million tiny images: A large data set for nonparametric object and scene recognition,” *PAMI*, vol. 30, no. 11, pp. 1958–1970, 2008.