Deep Learning from Temporal Coherence in Video

Hossein Mobahi HMOBAHI2@UIUC.EDU
Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA

Ronan Collobert COLLOBER@NEC-LABS.COM
Jason Weston JASONW@NEC-LABS.COM
NEC Labs America, 4 Independence Way, Princeton, NJ 08540

Abstract

This work proposes a learning method for deep architectures that takes advantage of sequential data, in particular from the temporal coherence that naturally exists in unlabeled video recordings. That is, two successive frames are likely to contain the same object or objects. This coherence is used as a supervisory signal over the unlabeled data, and is used to improve the performance on a supervised task of interest. We demonstrate the effectiveness of this method on some pose invariant object and face recognition tasks.

1. Introduction

With the availability of ever increasing computing power, large-scale object recognition is slowly becoming a reality. Several massive databases have been introduced for that purpose. Huge hand-labeled datasets like NORB (LeCun et al., 2004) can be constructed e.g. by considering a few objects and varying the pose (using a rotating platform) as well as lighting conditions, the presence of clutter and the background. Scaling to more classes has been shown to be possible using click-through data from search engines, like in the recent 80 million tiny images (Torralba et al., 2008) database. Nevertheless, labeling images remains expensive, which motivates research into ways of leveraging the cheap and basically infinite source of unlabeled images available in the digital world.

Classical semi-supervised learning (Chapelle et al., 2006) and transduction (Vapnik, 1995) are machine-learning classification techniques able to handle labeled and unlabeled data, which assume each unlabeled example belongs to one of the labeled classes that are considered. If the unlabeled data is coming from heterogeneous sources then in general no class membership assumptions can be made (the unlabeled image might not belong to any of the considered classes), and one cannot rely on these methods. There are however cases where unlabeled data has a useful underlying structure which can be exploited. One example is video, a source of images constrained by *temporal coherence*: two successive frames are very likely to contain similar contents and represent the same concept classes. Each object in the video is also likely to be subject to small transformations, such as translation, rotation or deformation over neighboring frames. This sequential data thus provides a signal from which to learn a representation invariant to these changes. In this paper we propose a learning method that can leverage temporal coherence in video to boost the performance of object recognition tasks.

We choose a deep convolutional network architecture (LeCun et al., 1998), well suited for large-scale object recognition. We propose a training objective with a temporal coherence regularizer added to a typical training error minimizing objective, resulting in a modified backpropagation rule. While this paper focuses on object recognition and the use of video, the proposed algorithm can be applied to other sequential data with temporal coherence, and when minimizing other choices of loss.

We report experimental results on a visual object recognition task, COIL100 (Nayar et al., 1996), where the goal is to recognize objects regardless of their pose. We show that in this case, temporal coherence in video acts as a very good regularizer, and that, without using hand-crafted or strongly engineered features one can produce models that compete with state-of-the-art hand-designed methods like VTU (Wersing & Körner,

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009.

2003). We then go on to investigate how the choice of video source can affect this result, i.e. we measure the relative performance depending on the source of the video. To do this we recorded our own datasets of real rotating 3D objects, of differing degrees of similarity to COIL, and show how this unlabeled supplementary video also improves performance on our supervised classification task. We also report experiments on the ORL face dataset (Samaria & Harter, 1994), with similar results.

From a biological point of view, several authors (Hinton & Sejnowski, 1999; Becker, 1996a) argue that pure supervised learning is a poor model of how animals really learn. Learning from temporal coherence in sequence data, e.g. in audio and video, on the other hand provides a natural, abundant source of data which seems a more biologically plausible signal than used in most current machine learning tasks. In this respect, we believe this is an important direction of research, and this work proposes a simple and intuitive solution to this problem.

The rest of the paper is organized as follows. Section 2 describes our neural network architecture for incorporating temporal coherence for object recognition, and Section 3 describes previous related work. Section 4 presents experimental results, and Section 5 concludes.

2. Exploiting Video Coherence with CNNs

Our choice of architecture is a convolutional neural network (CNN) (LeCun et al., 1998). A CNN performs a chain of filters and resolution reduction steps as shown in Figure 1. This structure imposes a hardwired prior knowledge which is advantageous for visual recognition tasks compared to fully connected networks due to several reasons. First, it takes the topology of 2D data into account, as opposed to converting it to a long 1D vector. Second, the locality of filters significantly reduces the number of connections and henceforth parameters to be learned, reducing overfitting problems. Finally, the resolution reduction operation provides better tolerance against slight distortions in translation, scale or rotation. CNNs have justified themselves in many visual recognitions tasks including handwritten digit recognition (LeCun et al., 1998) and face detection (Osadchy et al., 2005).

We now formally describe this architecture.

2.1. Convolution and Subsampling

Each convolution layer C_l in our architecture described in Figure 1 operates a linear $K^l \times K^l$ filtering over N^{l-1} image input planes $\boldsymbol{z}^{l-1}_{1 \dots N^{l-1}}$ of size $D^{l-1} \times D^{l-1}$. It outputs an arbitrary chosen number N^l of planes $\boldsymbol{z}^l_{1 \dots N^l}$, where the value at position (i, j) in the p^{th} plane is computed as follows:

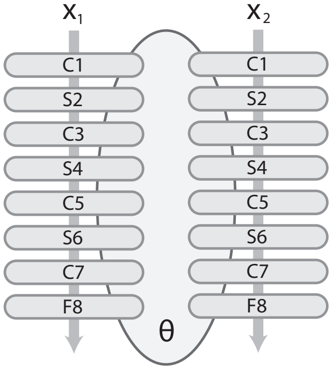

Figure 2. Siamese Architecture: the parameters θ are shared between the two identical copies of the network. Two examples are input, one for each copy, and the comparison between them is used to train θ, i.e. using the temporal coherence regularizer in equation (4).

$$z^l_p(i,j) = b^l_p + \sum_q \sum_{s=1}^{K^l} \sum_{t=1}^{K^l} w^l_{p,q,s,t}\, z^{l-1}_q(i-1+s, j-1+t)\,,$$

where the biases b^l_p and the filter weights $w^l_{p,q,s,t}$ are trained by backpropagation. The output plane size is $D^l \times D^l$, where $D^l = D^{l-1} - K^l + 1$.

Subsampling layers S_l simply apply a $K^l \times K^l$ smoothing over each input planes:

$$z^l_p(i,j) = b_p + w_p \sum_{s=1}^{K^l} \sum_{t=1}^{K^l} z^{l-1}_p(i-1+s, j-1+t)\,,$$

where the parameters b^l_p and w^l_p are also trained by backpropagation.

A non-linearity function like $\tanh(\cdot)$ is applied after each convolution and subsampling layer. A final classical fully-connected layer outputs one value per class in the considered task. To interpret these values as probabilities, we add a "softmax" layer which computes:

$$\tilde{P}_p = \frac{\exp(z^{l-1}_p)}{\sum_q \exp(z^{l-1}_q)}\,. \tag{1}$$

We suppose we are given a set of training examples $\{(\boldsymbol{x}_n, y_n)\}_{n=1 \dots N}$, where $\boldsymbol{x}_n$ represents a two-dimensional input image, and y_n a label. We then minimize the negative log-likelihood $L(\boldsymbol{\theta}) = \sum_{n=1}^N L(\boldsymbol{\theta}, \boldsymbol{x}_n, y_n)$ over the data with respect to all

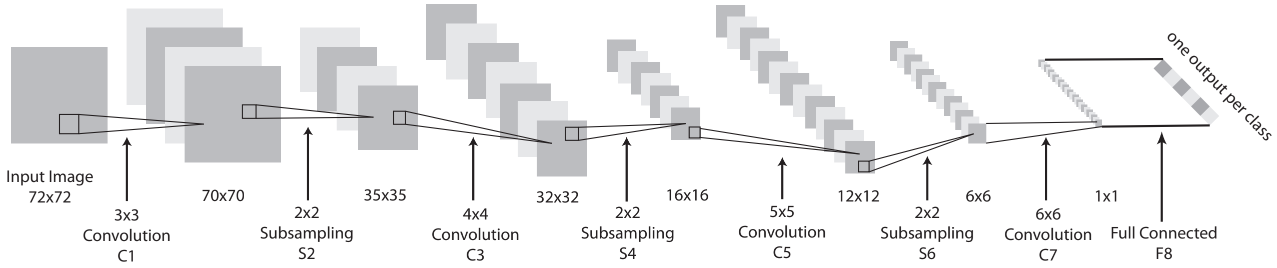

Figure 1. A Convolutional Neural Network (CNN) performs a series of convolutions and subsamplings given the raw input image until it finally outputs a vector of predicted class labels.

parameters $\boldsymbol{\theta}$ of the network:

$$L(\boldsymbol{\theta}) = -\sum_{n=1}^{N} \log P_{\boldsymbol{\theta}}(y_n|\boldsymbol{x}_n) = -\sum_{n=1}^{N} \tilde{P}_{\boldsymbol{\theta},\, y_n}(\boldsymbol{x}_n) \quad (2)$$

We use stochastic gradient descent (Bottou, 1991) optimization for that purpose. Random examples $(\boldsymbol{x},\, y)$ are sampled from the training set. After computation of the gradient $\partial L(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$, a gradient descent update is applied:

$$\boldsymbol{\theta} \longleftarrow \boldsymbol{\theta} - \lambda \frac{\partial L(\boldsymbol{\theta},\, \boldsymbol{x},\, y)}{\partial \boldsymbol{\theta}}\,, \quad (3)$$

where λ is a carefully chosen learning rate (e.g., choosing the rate which optimizes the training error).

2.2. Leveraging Video Coherence

As highlighted in the introduction, video coherence ensures that consecutive images in a video are likely to represent the same scene. It is also natural to enforce the representation of input images in the deep layers of the neural network to be similar if we know that the same scene is represented in the input images.

We consider now two images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, and their corresponding generated representation $\boldsymbol{z}^l_{\boldsymbol{\theta}}(\boldsymbol{x}_1)$ and $\boldsymbol{z}^l_{\boldsymbol{\theta}}(\boldsymbol{x}_2)$ in the l^{th} layer. We exploit the video coherence property by enforcing $\boldsymbol{z}^l_{\boldsymbol{\theta}}(\boldsymbol{x}_1)$ and $\boldsymbol{z}^l_{\boldsymbol{\theta}}(\boldsymbol{x}_2)$ to be close (in the L_1 norm) if the two input images are consecutive video images. If the two input images are not consecutive frames, then we push their representations apart. This corresponds to minimizing the following cost:

$$L_{coh}(\boldsymbol{\theta},\, \boldsymbol{x}_1, \boldsymbol{x}_2) = \quad (4)$$

$$\begin{cases} ||\boldsymbol{z}^l_{\boldsymbol{\theta}}(\boldsymbol{x}_1) - \boldsymbol{z}^l_{\boldsymbol{\theta}}(\boldsymbol{x}_2)||_1, & \text{if } \boldsymbol{x}_1,\, \boldsymbol{x}_2 \text{ consecutive} \\ \max(0,\, \delta - ||\boldsymbol{z}^l_{\boldsymbol{\theta}}(\boldsymbol{x}_1) - \boldsymbol{z}^l_{\boldsymbol{\theta}}(\boldsymbol{x}_2)||_1), & \text{otherwise} \end{cases}$$

where δ is the size of the margin, a hyperparameter chosen in advance, e.g. $\delta = 1$.

Algorithm 1 Stochastic Gradient with Video Coherence.

```
Input: Labeled data $(\boldsymbol{x}_n,\, y_n)$, $n = 1, ...N$, unlabeled video data $\boldsymbol{x}_n$, $n = N+1, ...N+U$
repeat
    Pick a random labeled example $(\boldsymbol{x}_n,\, y_n)$
    Make a gradient step to decrease $L(\boldsymbol{\theta},\, \boldsymbol{x}_n,\, y_n)$
    Pick a random pair of consecutive images $\boldsymbol{x}_m, \boldsymbol{x}_n$ in the video
    Make a gradient step to decrease $L_{coh}(\boldsymbol{\theta},\, \boldsymbol{x}_m,\, \boldsymbol{x}_n)$
    Pick a random pair of images $\boldsymbol{x}_m, \boldsymbol{x}_n$ in the video
    Make a gradient step to decrease $L_{coh}(\boldsymbol{\theta},\, \boldsymbol{x}_m,\, \boldsymbol{x}_n)$
until Stopping criterion is met
```

In our experiments, we enforced video coherence as described in (4) on the $(M-1)^{th}$ layer of our M-layer network, i.e. on the representation yielded by the successive layers of the network just before the final softmax layer (1). The reasoning behind this choice is that the L_1 distance we use may not be appropriate for the log probability representation in the last layer, although in principle we could apply this coherence regularization at any layer l. In practice, minimizing (4) for all pairs of images is achieved by stochastic gradient descent over a "siamese network" architecture (Bromley et al., 1993): two networks sharing the same parameters $\boldsymbol{\theta}$ compute the representation for two sampled images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ as shown in Figure 2. The gradient of the cost (4) with respect to $\boldsymbol{\theta}$ is then computed and updated in the same way as in (3).

The optimization of the object recognition task (2) and the video coherence (4) is done simultaneously That is, we minimize:

$$\sum_{n=1}^{N} L(\boldsymbol{\theta},\, \boldsymbol{x}_n,\, y_n) + \gamma \sum_{m,n} L_{coh}(\boldsymbol{\theta},\, \boldsymbol{x}_m,\, \boldsymbol{x}_n)$$

with respect to $\boldsymbol{\theta}$.

In order to limit the number of hyper-parameters, we gave the same weight to each task, i.e. $\gamma = 1$, and minimization is then achieved by alternating stochastic updates from each of the two tasks. Further, the distribution of consecutive versus non-consecutive frames used to minimize $L_{coh}(\cdot)$ presented during stochastic gradient descent will also affect learning. Here, again we simplify things by presenting an equal number of each, as described in Algorithm 1.

3. Previous and Related Work

Temporal Coherence Learning Ramanan, Forsyth and Barnard (2006) have used temporal coherence to build models of animals. Their system detects possible segments and learns a spatial model from clusters coherent over time. An annotator is required to clip portions where the object of interest is not present. In contrast, our work allows one to use a single video containing the entire set of objects and each object may disappear and reappear arbitrarily.

Noor et al (2006) learn a graph with each node being a different appearance of an object connected to temporally proximal nodes. A test image is reconstructed by a linear combination of some node and its neighbors. However, transformations even as simple as translation or scaling are highly nonlinear in image space. Therefore, we use a deep convolutional network architecture that can tolerate partial translation, scaling and rotation (Lawrence et al., 1997).

(Wiskott & Sejnowski, 2002) learn invariant (slowly varying) features from unsupervised video based on a reconstruction loss. This can then be used for a supervised task but is not trained at the same time.

The work of Becker (1996a; 1999), which is probably the most related work to ours, explores the use of temporal context using a fully connect neural network algorithm which introduces extra neurons, called contextual gating units, and a Hebbian update rule for clustering based on context ("competitive learning") This method was applied to rotating objects (faces) and showed improvements over not taking into account the temporal context. In comparison, our work does not introduce new network architectures, we instead introduce a natural choice of regularizer for taking advantage of temporal coherence that can be applied to any choice of network. We chose to apply our method to a state-of-the art deep convolutional network.

Becker and Hinton (1992; 1996b) also introduced the IMAX method that maximizes the mutual information betwen different output units which can be applied to learning spacial or temporal coherency. However, this method has a number of drawbacks including a "tendency to become trapped in poor local minima" and that "learning is very slow" (unless specific tricks are used) due to the small gradients induced by their criterion, as reported by the authors. In contrast, our method is highly scalable and can be easily trained on millions of examples, and we observe improved generalization whenever we applied it.

Semi-Supervised Learning Classical Semi-supervised learning methods utilize unlabeled examples coming from the same distribution (and hence classes) as the labeled data, and can be realized by either shallow architectures (e.g. kernelized methods) or deep ones. There are many variants of each type, see e.g. (Chapelle et al., 2006).

Two main methods are transductive inference and graph-based approaches. Transductive methods like TSVMs (Vapnik, 1995) involve maximizing the margin (confidence) on a set of unlabeled examples which come from the same distribution as the training data. Several authors argue that this makes an assumption that the decision rule lies in a region of low density (Chapelle & Zien, 2003). Graph-based methods use the unlabeled data by constructing a graph based on a chosen similarity metric, e.g. one builds edges between k-nearest neighbors. For example, Laplacian SVM (Belkin et al., 2005) works by directly regularizing for a two-class SVM that $||f(x) - f(x')||^2$ should be small for two examples x and x' connected in the graph. (Weston et al., 2008) presents a similar approach for neural networks. Further, (Chopra et al., 2005) applied a siamese network similar to ours but for a fully supervised (not semi-supervised) face similarity task (not using video). See also (Bowling et al., 2005) for an embedding algorithm using the actions of a robot which seems related to our work. Finally, we note that many graph-based approaches are also used in an unsupervised rather than supervised setup (Tenenbaum et al., 2000; Roweis & Saul, 2000).

In contrast to TSVMs, our method does not make a strong assumption that the class labels of objects in the unlabeled video have to belong to the training classes, and we show experimentally in Section 4 that our method takes advantage of examples coming from differing classes.

Graph methods on the other hand suffer from two further problems: (1) building the graph is computationally burdensome for large-scale tasks, (2) they make an assumption that the decision rule lies in a region of low density *with respect to the distance metric chosen for k-nearest neighbors.*

Our method does not rely on the low density assumption at all. To see this, consider uniform two-dimensional data where the class label is positive if it is above the y-axis, and negative if it is below. A nearest-neighbor graph gives no information about the class label, or equivantly there is no margin to optimize for TSVMs. However, if sequence data (analogous to a video) only has data points with the same class label in consecutive frames then this *would* carry information. Further, no computational cost is associated with collecting video data for computing (4), in contrast to building neighbor graphs. Realistically, in high dimensional spaces nearest neighbors can also perform poorly, e.g. in the pixel space of images.

4. Experiments

We consider two types of experiment, object and face recognition, detailed in Sections 4.1 and 4.2.

4.1. Object Recognition

4.1.1. Datasets

COIL100 We use the COIL100 image dataset developed at Columbia University as our main recognition task (Nayar et al., 1996). This set contains color pictures of 100 objects, each 72x72 pixels; some examples are shown in Figure 3. There are 72 different views for every object, i.e. there are 7200 images in total. The images were obtained by placing the objects on a turntable and taking a shot for each 5 degree turn. The images were clipped and passed through illumination normalization.

COIL100-Like We created a video dataset of auxiliary images to complement COIL100. The idea is to provide images that are similar to, but not the same objects as, the ones in COIL100. This will enable us to measure the success of our method when the unlabeled video comes from a different distribution. For this purpose we collected 4 types of objects that are also present in COIL100: namely fruits, cars, cups, and cans with 10 objects per type (see Figure 4).

Similar to COIL100, we recorded video utilizing a turntable. As we wanted to record a continuous stream of video, in real-time we removed and placed each object, where the operator is wearing a black glove which can be seen in the video. The idea is that our setup mimics that of a child holding a toy or other object and rotating it in her hand, and in this way learning about image transformations. The turntable makes 4 revolutions per minute and the recording rate is 24 fps. We downsampled the video so that two successive frames give a 5 degree rotation of the objects.

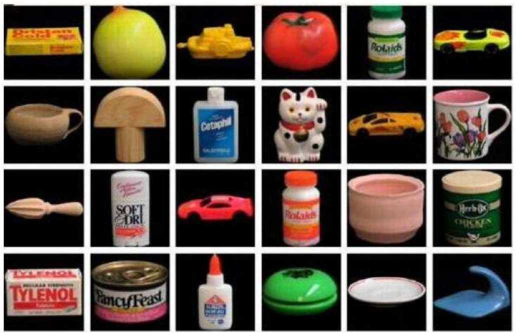

Figure 3. Dataset 1: Examples of the 100 objects from **COIL100**, each of which has 72 different poses.

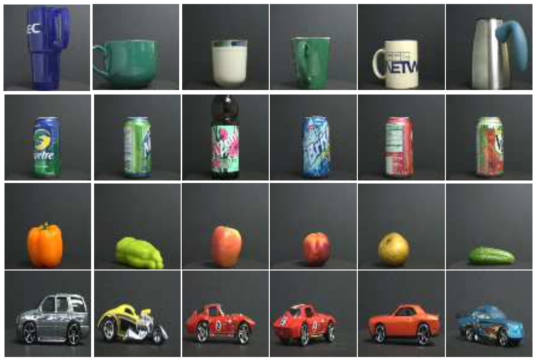

Figure 4. Dataset 2: Examples of 40 **COIL100-Like** objects, each of which is provided with 72 different poses, as a video stream. This video was collected to provide similar sensory data as provided in the COIL dataset.

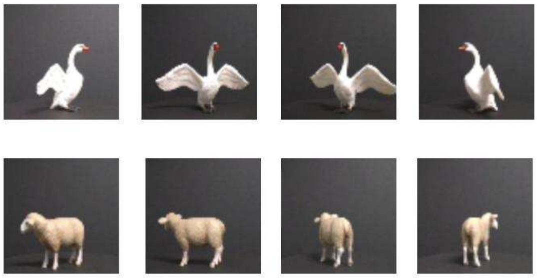

Figure 5. Dataset 3: Examples of 60 animals from our **Animal Set**, comprising of animals such as horses, ducks, deer and rabbits. Again, 72 poses are provided for each animal as a video stream.

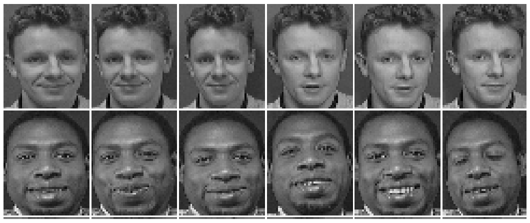

Figure 6. Dataset 4: Examples from the **ORL** face dataset.

Animal Set We also created a video dataset containing objects rather dissimilar to COIL100. This will enable us to measure the success of our method when the unlabeled video shares no objects in common with the supervised task of interest. To do this, we collected a set of toy animals, consisting of 60 toys of different types such as horse, duck, cow, sheep, deer, dog, cat, pig, mouse, rabbit and different types of birds. The data was collected in the same way as for COIL100-Like. Some examples are given in Figure 5.

Although both COIL100 and our video are in color, we convert them to grey scale so that recognition only depends on the shape of the objects. This will enable us to compare our results with other works who also use only the shape information, which is indeed a harder task than using color information as well.

Both the datasets we collected, COIL100-Like and Animal Set are available at `http://ml.nec-labs.com/download/videoembed`.

4.1.2. Methods

We compare our CNN architecture summarized in Figure 1 against previously used methods. These include Support Vector Machines (SVM) using a polynomial kernel (Roobaert & Hulle, 1999), a nearest neighbor classifier on the direct images (Wersing & Körner, 2003) (Nearest Neighbor), an eigenspace plus spline recognition model (Nayar et al., 1996) (Eigen Spline), a SpinGlass Markov Random Field (SpinGlass MRF) (Caputo et al., 2002), and a hierarchical view-tuned network for visual recognition tasks (VTU) (Wersing & Körner, 2003).

Linear SVM and Nearest Neighbor Classifier are well-known, so we briefly describe the other methods. Spin Glass MRF essentially uses an energy function inspired by models of physics of disordered systems. The eigenspace and spline method first obtains the eigenspace of training images (all classes together). It then projects training samples onto a subset of eigenvectors with the top eigenvalues. In the reduced space, a spline interpolation is applied to the samples of each class separately. The recognition phase projects a test point to the reduced space and assigns it to the class with closest spline.

The VTU method builds a hierarchy of biologically inspired feature detectors. The method applies Gabor filters at four orientations, followed by spatial pooling. It then learns receptive field profiles using a special type of sparse coding algorithm with invariance constraints. VTU has shown to achieve very good results, but there is a lot of manual engineering of knowledge in the algorithm as well as a lot of tuning parameters.

4.1.3. Results

The setup of our experiments is as follows. First, we use a standard CNN without utilizing any temporal information to establish a baseline for our contribution. We then explore three scenarios based on the source of the unlabeled video. These sources are COIL100 objects, COIL100-Like, and our Animal Set. In all of these scenarios, the labeled training and testing data for the supervised task belong to COIL.

For comparability with the settings available from other studies on COIL100, we choose two experimental setups. These are (i) when all 100 objects of COIL are considered in the experiment and (ii) when only 30 labeled objects out of 100 are studied (for both training and testing). In either case, 4 out of 72 views (at 0, 90, 180, and 270 degrees) per object are used for training, and the rest of the 68 views are used for testing. All of our reported numbers are based on averaging the classification rate on the test data over 10 training runs. The results are given in Table 1.

The first experiment uses only the labeled examples, and no unlabeled video, for training (Standard CNN). The performance (test set accuracy) is 71.49% and 84.88% for 100 and 30 objects respectively. This is slightly better than SVM but worse than VTU.

Next, we treat COIL100 as a continuous unlabeled video sequence of rotating objects with 72 consecutive frames per each object (after 72 frames the continuous video switches object). For the 100 object result, the test set is hence part of the unlabeled video (a so-called "transductive" setting). Here we obtained 92.25% accuracy (*video*CNN V:COIL100) which is much higher than the best alternative method (VTU).

A natural question is what happens if we do not have access to test data during training, i.e. the setting is a typical semi-supervised situation rather than a "transductive" setting. To explore this, we used 30 objects as the primary task, i.e. 4 views of each object in this set were used for training, and the rest for test. The other 70 objects were treated as an unlabeled video sequence (again, images of each object were put in consecutive frames of a video sequence). Training with 4 views of 30 objects (labeled data) and 72 views of 70 objects (unlabeled video sequence) resulted in an accuracy of 95.03% on recognizing 68 views of the 30 objects (*video*CNN V:COIL"70") This is about 5% above VTU's performance.

So far the unlabeled sequence was from the same dataset as the training and test. To investigate whether unlabeled video recordings of a large amount of video of some other (possibly similar) objects also improve recognition rates on the supervised task of interest, we then considered our datasets COIL100-Like and Animal Set.

For the following experiment, we consider all 100 objects in COIL (4 training, 68 for testing of each object). The performance when leveraging the COIL-Like video and Animal Set videos are 79.77% (*video*CNN V:COIL100-Like) and 78.67% (*video*CNN V:Animal Set) respectively. Compared to the state-of-the art VTU method, the first number is slightly better and the second slightly worse. However, in both cases we have managed to match the state-of-the-art whilst avoiding a strongly engineered solution for this task by utilizing learning from unlabeled video.

Compared to a plain CNN without using unlabeled video, these are improvements of about 7% and 8%. This indicates that although use of similar objects has a larger improvement, the difference between the gain obtained by COIL-Like and the Animal Set is relatively small. This is important because it opens up the possibility that the CNN structure has the ability to learn pose invariance abstractly, without relying on the actual object set used for training.

The drop in performance when using the COIL-Like set as unlabeled video (*video*CNN V:COIL-Like) rather than objects from COIL100 itself (*video*CNN V:COIL100) is probably partly due to the change in camera and environment parameters. However, our results indicate that using unlabeled auxiliary video is still always beneficial compared to not using video, even when the objects in the auxiliary video are not similar to those of the primary task.

4.2. Face Recognition

We also report a simple experiment on AT&T's ORL face database (Samaria & Harter, 1994), which consists of 10 different gray scale images for each of the 40 distinct subjects, taken at different times and with varying lighting and facial expressions (open / closed eyes, smiling / not smiling). See Figure 6 for examples.

The images were placed in a "video" sequence by concatenating 40 segments, one for each subject, ordering according to the (arbitrary) numbering system in the dataset. Note this is a "transductive" setup, where the labeled train and test images are part of the video (training examples are evenly spaced). We labeled k =1,2 or 5 images per subject and compared to the baselines Nearest Neighbor, PCA, LDA and MRF (Huang et al., 2004). We used the same CNN as in Section 4.1 and rescaled the images from 92 × 112 to 72 × 72 pixels for that purpose, and otherwise performed no special pre-processing. The results given in Table 2 again indicate that learning significantly benefits from unlabeled video through temporal coherence.

Table 1. Test Accuracy on COIL100 in various settings. Both 30 and 100 objects were used following (Wersing & Körner, 2003). Our temporal coherence respecting algorithm *video*CNN, with various choices of video, outperforms a standard CNN and other baselines.

Method	**30 objects**	**100 objects**
Nearest Neighbor	81.8	70.1
SVM	84.9	74.6
SpinGlass MRF	82.79	69.41
Eigen Spline	84.6	77.0
VTU	89.9	79.1
Standard CNN	84.88	71.49
*video*CNN V:COIL100	-	92.25
*video*CNN V:COIL"70"	95.03	-
*video*CNN V:COIL-Like	-	79.77
*video*CNN V:Animal	-	78.67

Table 2. Test Accuracy Performance on ORL Faces with k labeled examples per subject.

Method	**k=1**	**k=2**	**k=5**
Nearest Neighbor	69.07	81.08	94.64
PCA	56.43	71.19	88.31
LDA	-	68.84	88.87
MRF	51.06	68.38	86.95
Standard CNN	71.83	82.58	94.05
*video*CNN V:ORL	90.35	94.77	98.86

5. Conclusion

In this work we proposed a deep learning algorithm for visual object recognition exploiting the temporal coherence in video. Video acts a pseudo-supervisory signal that improves the internal representation of images by preserving translations in consecutive frames. This should be beneficial for many supervised tasks, and huge collections of data can be obtained without human annotation.

In our method, labeled and unlabeled data are trained on simultaneously; temporal coherence of unlabeled data acts as a regularizer for the supervised task. Potentially, one can learn representations that are invariant to pose, illumination, background or clutter, deformations (e.g. facial expressions) or occlusions with appropriate videos.

We conducted several experiments to evaluate our approach by considering several choices of video dataset.

Our result suggests that strong improvements can be achieved when the unlabeled data comes from the same dataset that labeled data comes from. When the two sets come from different datasets, use of unlabeled data is still beneficial, where probably the more similar the objects are, the more beneficial the data is.

References

Becker, S. (1996a). Learning Temporally Persistent Hierarchical Representations. *NIPS* (pp. 824–830).

Becker, S. (1996b). Mutual information maximization: models of cortical self-organization. *Network: Computation in Neural Systems*, *7*, 7–31.

Becker, S. (1999). Implicit Learning in 3D Object Recognition: The Importance of Temporal Context. *Neural Computation*, *11*, 347–374.

Becker, S., & Hinton, G. (1992). Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, *355*, 161–163.

Belkin, M., Niyogi, P., & Sindhwani, V. (2005). On manifold regularization. *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTAT 2005)*.

Bottou, L. (1991). Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nîmes 91*. Nimes, France: EC2.

Bowling, M., Ghodsi, A., & Wilkinson, D. (2005). Action respecting embedding. *ICML 2005*.

Bromley, J., Guyon, I., LeCun, Y., Sackinger, E., & Shah, R. (1993). Signature verification using a siamese time delay neural network. In J. Cowan and G. Tesauro (Eds.), *Advances in neural information processing systems*, vol. 6. Morgan Kaufmann.

Caputo, B., Hornegger, J., Paulus, D., & Niemann, H. (2002). *A spin-glass markov random field for 3-d object recognition* (Technical Report LME-TR-2002-01). Institut fur Informatik, Universitat Erlangen Nurnberg.

Chapelle, O., Schölkopf, B., & Zien, A. (2006). *Semi-supervised learning*. Adaptive computation and machine learning. Cambridge, Mass., USA: MIT Press.

Chapelle, O., & Zien, A. (2003). Semi-Supervised Classification by Low Density Separation. *Advances in Neural Information Processing Systems*, *17*, 1633–1640.

Chopra, S., Hadsell, R., & LeCun, Y. (2005). Learning a Similarity Measure Discriminatively, with Application to Face Verification. *Proc. Computer Vision and Pattern Recognition Conference (CVPR05)*.

Hinton, G., & Sejnowski, T. (1999). *Unsupervised Learning: Foundations of Neural Computation*. MIT Press.

Huang, R., Metaxas, D. N., & Pavlovic, V. (2004). A hybrid face recognition method using markov random fields. *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3* (pp. 157–160).

Lawrence, S., Giles, C., Tsoi, A., & Back, A. (1997). Face recognition: a convolutional neural-network approach. *IEEE Transactions on Neural Networks*, *8*, 98–113.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, *86*.

LeCun, Y., Huang, F. J., & Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. *Proceedings of CVPR'04*.

Nayar, S. K., Watanabe, M., & Noguchi, M. (1996). Real-time focus range sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*, 1186–1198.

Noor, H., Mirza, S. H., Sheikh, Y., Jain, A., & Shah, M. (2006). Model generation for video-based object recognition. *Proceedings of the 14th annual ACM international conference on Multimedia* (pp. 715–718). New York, NY, USA: ACM.

Osadchy, R., Miller, M., & LeCun, Y. (2005). Synergistic face detection and pose estimation with energy-based model. *Advances in Neural Information Processing Systems (NIPS 2004)*. MIT Press.

Ramanan, D., Forsyth, D., & Barnard, K. (2006). Building Models of Animals from Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1319–1334.

Roobaert, D., & Hulle, M. M. V. (1999). View-based 3d object recognition with support vector machines. *In IEEE International Workshop on Neural Networks for Signal Processing* (pp. 77–84).

Roweis, S., & Saul, L. (2000). Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, *290*, 2323–2326.

Samaria, F., & Harter, A. (1994). Parameterisation of a stochastic model for human face identification. *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*.

Tenenbaum, J., Silva, V., & Langford, J. (2000). A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, *290*, 2319–2323.

Torralba, A., Fergus, R., & Freeman, W. T. (2008). 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, *30*, 1958–1970.

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer. Second edition.

Wersing, H., & Körner, E. (2003). Learning optimized features for hierarchical models of invariant recognition. *Neural Computation*, *15*, 1559–1599.

Weston, J., Rattle, F., & Collobert, R. (2008). Deep learning via semi-supervised embedding. *ICML 2008*.

Wiskott, L., & Sejnowski, T. (2002). Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, *14*, 715–770.