Hybrid CNN-HMM Model for Street View House Number Recognition

Qiang Guo, Dan Tu, Jun Lei, Guohui Li

Department of Information System and Management
National University of Defense Technology

Abstract. We present an integrated model for using deep neural networks to solve street view number recognition problem. We didn't follow the traditional way of first doing segmentation then perform recognition on isolated digits, but formulate the problem as a sequence recognition problem under probabilistic treatment. Our model leverage a deep Convolutional Neural Network(CNN) to represent the highly variabe appearance of digits in natural images. Meanwhile, hidden Markov model(HMM) is used to deal with the dynamics of the sequence. They are combined in a hybrid fashion to form the hybrid CNN-HMM architecture. By using this model we can perform the training and recognition procedure both at word level. There is no explicit segmentation operation at all which save lots of labour of sophisticated segmentation algorithm design or finegrained character labeling. To the best of our knowledge, this is the first time using hybrid CNN-HMM model directly on the whole scene text images. Experiments show that deep CNN can dramaticly boost the performance compared with shallow Gausian Mixture Model(GMM)-HMM model. We obtaied competitive results on the street view house number(SVHN) dataset.

1 Introduction

Though the research of recognizing handwritten and machine printed characters have been last for several decades [1, 2], recognizing text in natural scene still remains a difficult computer vision problem. Text captured in unconstrained natural scene show quite large appearance variablity. They have different fonts, scales, rotations, lightening conditions etc.

Tradition methods for doing this task always fall in a seprated pipeline of first segmenting the image to extract isolated charactes then perform recognition on the extracted characters. Segmentation and recognition are long been considered to couple with each other which make it difficult to do either in a seperated way. Meanwhile, hand designed segmentation algorithm often became fragile when facing natural scene images. Also, we need large amount of isolated characters to cover the characters variablity. If the captured image is the whole text, that would need to label the bounding box of each character for training, which is a highly labor consuming work.

Our aim is to make the training and recognizing processs performed directly on the whole text image. In this paper, we address the problem on a special

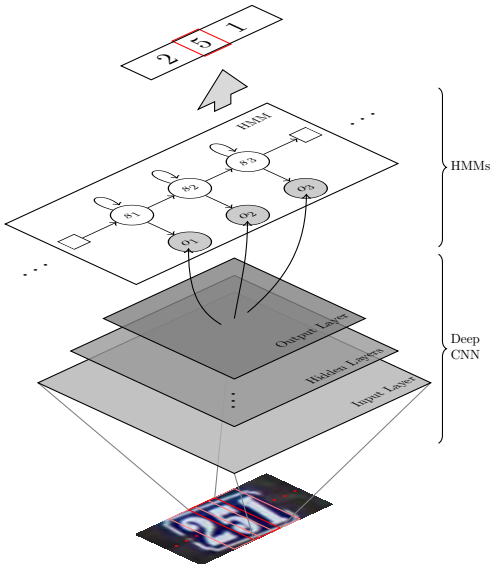

Fig. 1. Hybrid CNN-HMM Model. This image demonstrates the recognition process of our model. Input image is processed by sliding window to extract a sequence of frames. Each frame is normalised to the same scale and fed into the deep CNN which produce its posterior probability belonging to a category. This probability, after normalization, is used as the output probability of HMM. HMM is used to infer the most probable digits out of the frame sequence.

natural scene text dataset of street view number images. Those images are collected from street view imagery contain only digits which make us focus on the the image modeling problem other than language model.

Digits in street view number images are generally arranged in a sequential manner though somethimes in different vertical position. So we formulate the problem as a sequence recognition problem. To this end, the image is preprocessed to extract a sequence of frames which is seemed as observations at different time. In a generative way, the frames are seemed to be generated by an underlying state sequence which in this situation are digits from 0 to 9 and background clutter.

We propose a method in an unified framework integrating segmentation and recognition by hybriding hidden Markov model with deep convolutional neural networks. Fig.1 shows the architecture of our model. Sliding window is performed on the input image to extract a sequence of frames. Hiddem Markov model (HMM) is used to handle time variablity and deep convolutional neural networks (CNN) deal with all kinds of character variations. Deep CNN is critical to the performance of the model for its great representation capability.

We conduct a bootstrap process to get initial labels of frames. In the bootstrap stage we train a Gaussian mixture model(GMM)-HMM model to get the

initial frame assignments, then switch to train the hybrid CNN-HMM model. We show that starting with a not-so-well GMM-HMM model, we gain dramatic performance improvements after using CNN as the character model.

Similar methods have been used in speech and handwriting recognition communities, but to our knowledge, no one had investigated this model on scene text recognition task. Experiments show that deep CNN can dramaticly boost the performance compared with shallow Gausian Mixture Model(GMM)-HMM model. We obtaied competitive results on the street view house number(SVHN) dataset.

2 Related Work

Street view number recognition can fall in the category of natural scene text recognition problem. Research on scene text recognition already started in mid-90s [3], but still a not solved problem. Different from machine-printed character or handwriting recognition problem, recognizing text in natural scene has its special difficulties. Text captured in unconstrained natural scene show quite large appearance variablity. They have different fonts, scales, rotations, lightening conditions and also orgnized in various kind of layouts.

Traditional methods deal with this problem are based on sequential character classification by either sliding window [4,5] or connected components [6,7], after which a word prediction is made by grouping character classifier predictions in a left-to-right manner. More recent works [8–10] make use of over-segmentation methods, guided by a supervised classifier to generate candidates. Words are recognized through a sequential beam search optimization over character candidates.

Often a small fixed lexicon as language model to constrain word recognition [4, 5] in scene text recognition. However, the problem of recognizing street view numbers could not benifit from language model, which makes the problem more difficult.

In all these systems, character model is the the most critical component. Nowadays, as Convolutional Neural Networks are showing more and more powerful capability in object recognition tasks [11–14]. Some works have used CNN to tackle scene text recognition problem [5, 9, 13, 15]. Among these research, CNN shows great capability to represent all kinds of character variation in natural scene and still holding highly discrimitivity. Convolutional networks has succefully applied to recognition of handwritten numbers in 90s [16, 17].

Another important issue is how to infer the characters from the whole image when it's difficult to isolate each single character from the image. Most of previous work on SVHN dataset recognize isolated digits, except Goodfellow etc [15]. They solve the problem by directly using a deep large CNN to model the whole image and with a simple graphical model as the top inference layer. Inference of the position of digits in the number image is done by the network implicitly. Segmentation and recognition is performed in a unified way. However, as ad-

dressed by the authors [15], the method rest heavily on the assumpation that the sequence is of bounded length, with a small number length.

Alsharif *etc* [8] also used hybrid model, but only used HMM for segmentation. The whole process is still a combinition of explicitly segmentation and character candidates recognition.

In this work, we proposed a model to combine segmentation and recognition. Our model is different from all other methods to this problem. Our character model is build upon the success of large CNN architecture on large scale image recognition task and with our tuning for the special problem. And, we use a effecitive graphical model – Hidden Markov Model – to inference the digits of in the image. The two parts are combined in a hybrid fashion [18].

Our model is motivated by recently successiful application of deep neural networks on speech recognition [19, 20]. In speech recognition community, deep learning methods have take over traditional shallow GMM-HMM model and achieve significant improvements [19–22].

3 Problem Formulation

We formulate the street view number recognition task as a sequence recognition problem.

Let I represent a street view number image which contains unknown amount of digits. We treat the whole number image as a concatenated sequence of frames arranged horizontally. The frame squence is represented as $O = \{o_1, o_2, \cdots, o_T\}$, in which o_i corresponds to feature of the ith frame. The length of the sequence is T. Define $Y = \{y_1, y_2, \cdots, y_L\}$ as the label of the image. L is the amount of digits in the image, y_i is the ith digit's label.

We treat the frames as being generated sequentially from a Markov process that trainsit between states $S = \{s_1, s_2, \cdots, s_K\}$. That brings the use of hidden Markov model.

In our setting, the frames are categorized to 11 categories which contains 10 digits and a nul category, $\{0, \cdots, 9\} \cup \{\mathsf{nul}\}$. The nul category represents non-digit frames which contain pre- or post-digit background, inter-digit interval and clutter frames.

Define $\mathcal{M}$ as the HMM model. The key parameters of $\mathcal{M}$ are the initial state probability distribution $\pi = \{p(q_0 = s_i)\}$, the transition probabilities $a_{ij} = p(q_t = s_j | q_{t_1} = s_i)$, and a model to estimate the observation probabilities $p(o_t|s_i)$. Training samples are given in the form of $\{O_i, S_i\}$, where $i \in \{1, \cdots, N\}$, N is the size of training set. Note there's no need of digits boundary information, but only their digit labels.

We first use GMM as the character model, then switch to CNN. The labels for training CNN are got by performing *forced alignment* on the frame sequence. The learning process involves learning the transition probability of $\mathcal{M}$, the mean m_i, variance σ_i of each GMM components and the parameters of CNN.

Recognition of the numbers is performed by *maximum a posteriori*(MAP) estimation. That is, given O we want to find S which satisfize

$$\hat{S} = \underset{S}{\operatorname{argmax}} \log P(S|O).$$

We use *Viterbi algorithm* [23] to get the most probable state sequence. After that, we eliminate all nul states to get the final recognition results.

There is no need of explicit segmentation in both training and recognition. Acturally, during recognition and forced alignment, segmentation is implicitly performed within the inference process. To the user, the supplied data are only the whole number images and labels of all digits in the image, no position information is needed.

4 Hybrid CNN-HMM model

4.1 Model architecture

The whole architecture of the hybrid CNN-HMM model is showed in Fig.1.

We use HMM to model the dynamics of frame sequences and deep Convolutional Neural Network to model the frame appearance. Specifically, CNN is used to approximate the emission probability $p(o_t|s_t)$ of each frame under a given state. These two components constitute the hybrid CNN-HMM model.

To train CNN we need to assign each frame a label to tell which state it belongs to. So we conduct a bootstrap process. A GMM-HMM model is trained to initilize the transition probabilities among HMM states and give an initial label assignment to each frame. This is achieved by doing forced alignment with the initial GMM-HMM model. The GMM-HMM model could give a not-perfect, but somehow, reasonable alignment.

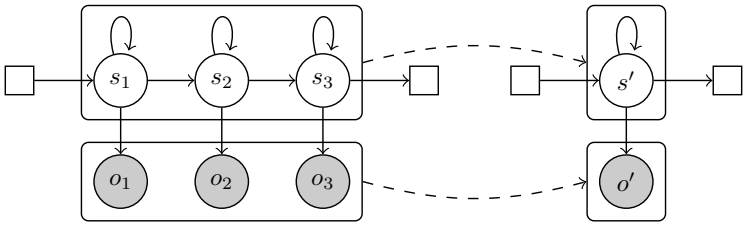

Fig. 2. HMM models. The left is 3-state HMM $\mathcal{M}$, the right is 1-state HMM $\mathcal{M}'$. Dashed arrows demonstrate the convertion of $\mathcal{M}$ to $\mathcal{M}'$.

Each frame category has a corresponding HMM, that means we have 11 HMMs in total. We use 3-state HMM to model each digit and 1-state HMM for background clutter frames. The 3-state HMM models pre-digit, mid-digit and post-digit frames. Fig.2 shows the structure of the HMM. Circle nodes represent

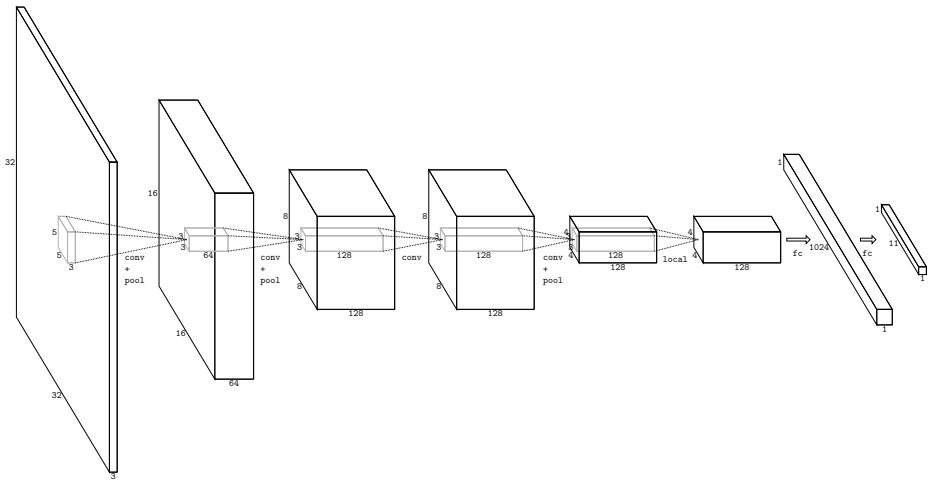

Fig. 3. Deep CNN architecture used in this work

hidden (without filling) and observation states (with filling). Note we also add non-emission states, which is represented as rectangle nodes, at the head and tail for the convenience of concatenating HMMs.

When using generative model, *e.g.* GMM, and training with MLE criterion, its better to divide the feature space into finer categories. That makes each GMM easier to represent its feature space. However, this treatment is not good for discrimitive models, for that would introduce ambiguities among inter- and intra class variations. Its properer to treat different variations of a digit as the same category. So, after bootstrap training we convert the 3-state HMMs to 1-state HMMs, denoted as $\mathcal{M}'$, to leverage the strong invariance representation capability of deep CNN. Frames belong to all states within a 3-state HMM are assigned to one category corresponding to the single state of $\mathcal{M}'$. Experiment demonstrates that CNN-HMM dramaticly improve the performance of GMM-HMM.

Our CNN architecture is implemented upon Alex Krizhevsky's ConvNet [12]. We investigated different configurations of architecture. Our best architecture consists of 4 convolutional layers, 3 of which have consecutive pooling layers, 1 locally connected layer and 2 fully connected layers. The top layer is an 11-way softmax layer. All connections are feedforward from one layer to the next. The filter size of each convolution layer is showed in Fig.3. Each convolutional layer includes local response normalization across maps. The max pooling window size is 2×2. The stride alternates between 2 and 1 at each layer. All convolution and locally connected layers contain rectifier units [24]. Details of the architecture's configuration is showed in Fig.3.

4.2 Training procedure

Feaure extraction We first extract frames of each image by sliding window with overlapping of consecutive windows. After that, we perform PCA on the frames data. Features are extracted by projecting each frame on the principal components. Then the feature is passed to next stage as observations.

Bootstrap training of GMM-HMM Then we use GMM-HMM as the bootstrap model to train the initial transition probability and get frame-state assignment. The extracted PCA features are fed to GMM-HMM model. We initialize the mixture components by using global data mean and variance with random noise added.

Each HMM conresponds to a digit category and contains three states to adapt pre-digit, mid-digit and post-digit frames. Background clutter frames are modeled with 1-state HMMs for there are not much obvious sequential difference among them.

The GMM-HMM model is trained with Baum-Welch algorithm [25,26] under maximum-likelihood (ML) criterion until the likelihood converges.

To use CNN, we need to supply the label of each frame for CNN is trained in a supervised way. After training of GMM-HMM, we perform forced alignment on all the images. That assigns each frame to a corresponding HMM state. This category is used as the label for CNN. To make CNN robust to intra category variations while keeping discrimitive to inter category variations, we merge the three states in a HMM to one state and observation frames corresponding to these states are fall in the same category of this state. The switching process is demonstrated in Fig.2.

After the bootstrap training we obtain the label of each frame. CNN is then trained with these generated data. We split the dataset into training and validation set. The network is trained under cross entropy objective by stochastic gradient descent (SGD) until the validation error stops to drop.

Note that the output of softmax layer is an estimation of the state posterior probability $p(s_t|o_t)$ [27], while the observation probability in HMM is

$$p(o_t|s_t) = \frac{p(s_t|o_t)p(o_t)}{p(s_t)}$$

where $p(s_t)$ is the prior probability of each state estimated from the training set, and $p(o_t)$ is independent of the true digit label and thus can be ignored. Actually, the output probability used in hybrid model is the posterior probability divided by prior probability, $\frac{p(s_t|o_t)}{p(s_t)}$, which is called *scaled likelihood* [18].

Embedded Viterbi training of CNN-HMM After the bootstrap training process we get label frames as the training data for CNN and initial states transition probability of HMM. Then hybrid CNN-HMM model is trained with *embedded Viterbi* algorithm [28]. The main training procedure is summerized in Algorithm.1.

Algorithm 1: Embedded Viterbi Training Algorithm

$t \leftarrow 0$;
$\Delta AP \leftarrow$ inf;
Train a gmm-hmm model, assign each output HMM state a label-id;
Use *Viterbi-decoding* algorithm to recognize each number image and evaluate average precision AP_t;
while $\Delta AP > 0$ **do**
    Use *forced-alignmnet* algorithm to assign each frame a label-id as its category;
    Use labeled frames to train CNN as cnn_t;
    Get the prior probability $p(c_i)$ of category c_i, where $i = 1, 2, \cdots, N_c$;
    Feed each frame to cnn_t to get its posterior probability $p(c_i|x_j)$;
    Compute the scaled likelihood $p_{scaled}(x_j|c_i)$;
    Perform *embeded-training* to re-estimate the transition probability of HMM, denote the new hybrid model as $\mathsf{cnn\text{-}hmm}_t$;
    Use *Viterbi-decoding* algorithm to recognize each number image and evaluate average precision AP_{t+1};
    $\Delta AP \leftarrow AP_{t+1} - AP_t$;
    $t \leftarrow t + 1$;
end
Output cnn_t as the final CNN model and $\mathsf{cnn\text{-}hmm}_t$ as the final CNN-HMM hybrid model;

The main idea of embedded Viterbi algorithm is to alternatively updating CNN and HMM until the average precision stops to improve. The procedure iteratively improve the best assignment of each frame, and can be proved to converge to a local optimum. This makes it possible to use supervised model, such as CNN, without the need of hand-labeled data.

5 Experiments

5.1 Dataset

The Street View House Numbers (SVHN) dataset [29] contains images captured from Google Street View. It includes two kinds of images. One have over $600,000$ cropped digit images, the other is constitute with more than $200,000$ images containing indeterminate number of digits. Our work deal with the later one.

Nearly all previous work are done on isolated digits, except the work by Goodfellow *et al.* [13] which use loosely cropped images. We solve this problem by integrating segmentation and recognition in a different way. Images in SVHN show quite large appearance variability, image blur and unnormalized layouts. Those make the recognition on the whole image quite difficult.

We preprocess the dataset similar with the way Goodfellow *et al.* [13] does. First find the smallest bounding box contain all individual digit bounding box, then expanding the box by 30% then crop out the expanded bounding box. Note, we do not resize all cropped images to the same size.

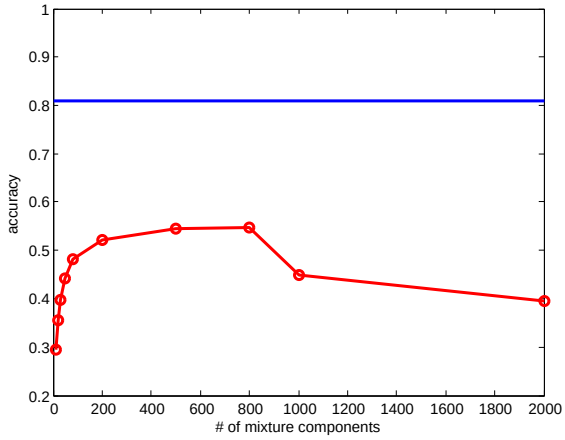

Fig. 4. Relationship between recognition accuracy and the number of mixture components for GMM-HMM (red line). The blue line is the final accuracy of CNN-HMM.

We set the frame width to 10 pixel with 4 pixel overlap to do sliding window, and then randomly selected 100,000 frames to train the PCA feature. We choose 60 principal components which perserve 98% percentage of the data infomation, then project each frame to the 60 eigenvectors to get the coefficients as the PCA feature.

We do not use any isolated digit images to train HMMs and CNN, training is totally performed in an embedded way at word level.

5.2 Results and analysis

We investigate the relationship between recognition accuracy and the number of mixture components for GMM-HMM. For each setting the model is trained until accuracy convergence. The result is showed in Fig.4. As shown in the figure, recognition accuracy increases with the number of mixture components but stops at 800, only achieve a not-so-well perfomance of 0.55 accuracy. Continually increasing the amount of mixture components makes the model tend to overfit.

The training of hybrid CNN-HMM model starts with this model. After embedded Viterbi training we get an accuracy of **0.81**, which improve the performance of GMM-HMM model by 47.2%. This demonstrates the represention superiority of deep model vs shallow model.

We also compared the performance of our hybrid with model with traditional methods [29]. These methods [29] first do over segmentation then search over segmentation hypotheses to find the best hypotheses. Different digit classifiers are tried in their work. The precision/recall curves are showd in Fig.5.

Fig.6 shows some selected results by the hybrid CNN-HMM model, note that the model correctly recognize different fonts of digits in street view images

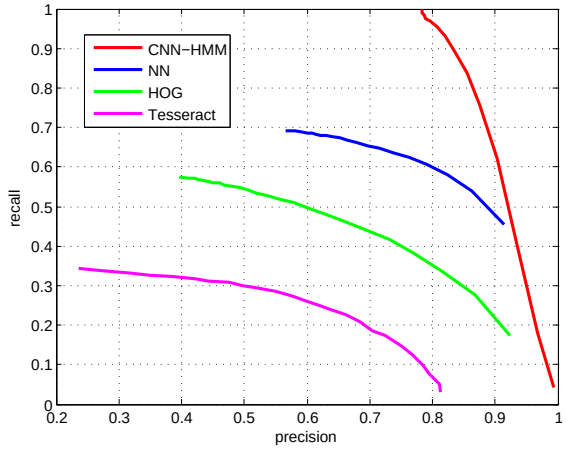

Fig. 5. Performance comparsion with traditional methods

when they are highly touching each other, blured, scaled or even their layout are slanted. Also, we visualize the segmentation lines over the images, which are automatically got by Viterbi decoding. The segmentation demonstrated the resonablity of our model and can be used to harvest large amount of digits directly from the images for training other character models.

We also conducted ablation study for the contribution of different layers of CNN. Table.1 summerized the accuracy after removing different layers. Through the ablation study we can see that depth of the network is imprtant to the system performance. Dropping three layers(two convolutional layers and one locally connected layer) decreases the performance by degree of 13.6%.

Table 1. Performance after removing different layers

ablation layers	accuracy(%)
no ablation	**81.07**
conv3	80.26
full1	80.04
local	78.44
conv3&local	76.94
conv2&conv3&local	70.03

6 Conclusion

We proposed hybrid CNN-HMM model to recognize street view numbers. The method ultilized CNN – a model recently performed promisingly well at many

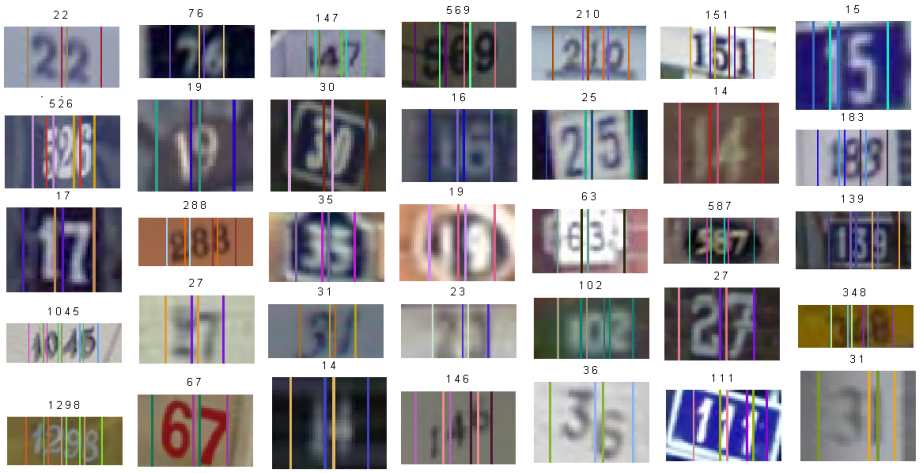

Fig. 6. Recognition results of hybrid CNN-HMM model. Segmentation line cuts generated by Viterbi algorithm are overlapped on the image. Two lines with the same color segment out a digit.

computer vision problems, and HMM – a classical graphical model which can effecitively do infference on sequence data. In this work, they are integrated in a hybrid fashion. By using this model, we can perform training and recognition both directly at word level, not need any labeling labour or design sophisticated segmentation algorithm. We achieve promising result on SVHN dataset. Experiments shows that, by using deep CNN substitute GMM with HMM, the hybrid model can significantly increaes the recognition performance. And, this model can easily extended to general scene text recognition by adopting a language model, whether a constraint lexicon or n-gram model.

Yet, HMM have a few shortcomings, for example, lack of context consideration, inproperly independece hyporsis and short at discrimination. Future work should be done by training HMM discrimitively under MMI [30] or MCE [31] criterion or using Conditional Random Field(CRF) [32]. On the other hand, adjacent frames should be considered when training CNN to incroperate context information. The architecture of CNN we used is pretty off-the-shelf. Better architecture need to be investigated.

References

1. G. Nagy, "Twenty years of document image analysis in pami," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 38–62, 2000.
2. M. Cheriet, M. El Yacoubi, H. Fujisawa, D. Lopresti, and G. Lorette, "Handwriting recognition research: Twenty years of achievement and beyond," *Pattern recognition*, vol. 42, no. 12, pp. 3131–3135, 2009.

3. J. Ohya, A. Shio, and S. Akamatsu, "Recognizing characters in scene images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 2, pp. 214–220, Feb 1994.
4. K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1457–1464.
5. T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 3304–3308.
6. L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Computer Vision–ACCV 2010*. Springer, 2010, pp. 770–783.
7. ——, "Real-time scene text localization and recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3538–3545.
8. O. Alsharif and J. Pineau, "End-to-end text recognition with hybrid hmm maxout models," *arXiv preprint arXiv:1310.1811*, 2013.
9. A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "Photoocr: Reading text in uncontrolled conditions." ICCV, 2013.
10. L. Neumann and J. Matas, "Scene text localization and recognition with oriented stroke detection." ICCV, 2013.
11. D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," in *ICDAR*, 2011, pp. 1250–1254.
12. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks." in *NIPS*, vol. 1, no. 2, 2012, p. 4.
13. I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," *arXiv preprint arXiv:1302.4389*, 2013.
14. M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *CoRR*, vol. abs/1311.2901, 2013.
15. I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," *arXiv preprint arXiv:1312.6082*, 2014.
16. 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.
17. O. Matan, C. J. C. Burges, Y. LeCun, and J. S. Denker, "Multi-digit recognition using a space displacement neural network." in *NIPS*, 1991, pp. 488–495.
18. H. Bourlard and N. Morgan, "Connectionist speech recognition: A hybrid approach," 1993.
19. G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.
20. G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.
21. A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6645–6649.

22. T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A.-r. Mohamed, "Making deep belief networks effective for large vocabulary continuous speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 30–35.
23. J. Forney, G.D., "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.
24. K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *ICCV*, 2009, pp. 2146–2153.
25. L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *The Annals of Mathematical Statistics*, vol. 41, no. 1, pp. 164–171, 02 1970.
26. L. E. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of a Markov Process," *Inequalities*, vol. 3, 1972.
27. M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate bayesian a posteriori probabilities," *Neural Computation*, vol. 3, pp. 461–483, 1991.
28. N. Morgan and H. Bourlard, "Continuous speech recognition," *Signal Processing Magazine, IEEE*, vol. 12, no. 3, pp. 24–42, 1995.
29. Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, vol. 2011, 2011.
30. S. Kapadia, V. Valtchev, and S. Young, "Mmi training for continuous phoneme recognition on the timit database," in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, vol. 2, April 1993, pp. 491–494 vol.2.
31. B.-H. Juang, W. Hou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 5, no. 3, pp. 257–265, May 1997.
32. J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01, 2001, pp. 282–289.