Shallow Classification or Deep Learning: An Experimental Study

Xu-Cheng Yin, Chun Yang, Wei-Yi Pei
Department of Computer Science and Technology
School of Computer and Communication Engineering
University of Science and Technology Beijing
Beijing 100083, China
Email: xuchengyin@ustb.edu.cn
ych.learning@gmail.com, peiweiyi.ustb@gmail.com

Hong-Wei Hao
Institute of Automation
Chinese Academy of Sciences
Beijing 100190, China
Email: hongwei.hao@ia.ac.cn

Abstract—After being kick-started with major breakthrough in 2006 by Hinton, LeCun and Bengio respectively, deep learning has been becoming the mainstream for challenging classification systems, which, however, always were with “shallow” discriminative classifiers in the past. In this paper, we argue that in common classification cases with plenty but not enough training examples, mixed-quality examples for dozens of categories, deep learning and shallow classification may have complementary performance. Then, we design a hybrid recognition strategy with classification switching to adaptively fuse deep learning and shallow classification technologies. Finally, we present a variety of experiments with visual recognition tasks, i.e., USPS character recognition, Caltech101 visual object classification, and ICDAR scene text recognition. Specifically, we perform word recognition by dynamically combining the conventional open source OCR engine with the present popular convolutional neural networks, and construct an effective end-to-end scene text recognition system with open-vocabulary. This end-to-end system is evaluated on ICDAR 2011 Robust Reading Competition (Challenge 2) dataset, the f measure of which is 54.5%, much better than 45.2% of the latest state-of-the-art performance.

I. INTRODUCTION

Before deep learning was kick-started with major breakthrough in 2006 by three research groups (Hinton’s [1], [2], LeCun’s [3] and Bengio’s [4]), shallow pattern recognition systems had been “mainstream” [5]. Even now, shallow technologies based on Support Vector Machines (SVM), AdaBoost, Random Forests, etc. are still used in many commercial classification systems.

In recent years, deep learning techniques have had important empirical successes in a number of traditional AI applications, e.g., speech recognition [6], computer vision [7], natural language processing [8], [9], object recognition [10], transfer learning and domain adaptation [11]. The detailed survey of deep learning can be referred to [12], [13], [14]. One core concept behind all deep learning techniques is to automatically discovery the abstract representations of data, with the belief that more abstract representations of data such as audio signals, images, and video tend to be more useful. Current deep learning algorithms, e.g., convolutional neural networks (CNN), learn abstract representation by deep architectures, since more abstract concepts can often be constructed in terms of less abstract ones. Many web giants have been moving down the road for Deep Learning researches and applications, e.g., Google, Microsoft, Baidu, and Facebook. More and more challenging classification systems prefer to deep learning technologies (e.g., Google and Baidu Image Search). Most winning methods for recent vision classification competitions are deep learning based systems, e.g., “PhotoOCR” [15] for word recognition in ICDAR 2013 Robust Reading Competition, and “SuperVision” [10] for Task 1 and 2 in ImageNet Large Scale Visual Recognition Challenge 2012.

Besides a very high computation complexity for the training procedure, there are also several other noted issues for the deep learning system. First, it should have enough examples for training in order to learn reliable features and (distributed) representations, especially for classification with large-scale categories. For example, in the ImageNet Competition 2012, “SuperVision” is based on about 1.2 million training images and 50,000 validation images [10]. Similarly, deep learning based classification techniques for recognizing scene characters and texts should be trained on a lot of examples, e.g., “PhotoOCR” is trained on 40 million character samples [15]. It is time-consuming, very difficult and even impossible to collect enough training samples in many cases. However, shallow “mainstream” classification always can compute discriminative features and construct a fairly robust classifier based on plenty training examples. Second, compared to shallow classification, generally deep learning should be more powerful in learning useful representations from challenging examples, e.g., very blur and small images for character recognition. It is always overfitting, and should require trickily tuning numerous network parameters with a lot of time-consuming and error-pruning experiments [10]. However, shallow classification techniques have a very efficient and effective performance on high-quality examples for easily extracting conventional discriminative features.

We argue that in common classification cases with plenty but not enough training examples, mixed-quality examples for dozens of categories, deep and shallow learning may have complementary performance, and should be fused in an adaptive way. For example, in the scene text recognition, if the extracted text is clear with a fairly high resolution,\footnote{In this paper, we call the conventional (shallow) “mainstream” classification technology as “shallow classification” or “shallow learning” in application systems, e.g., character recognition in document analysis and recognition, and object classification in computer vision.}
it is easily recognized correctly by conventional OCR engines. In contrast, small and burred texts are challenging for conventional OCR engines, but it is possibly classified correctly by deep learning (e.g., convolutional neural networks for character recognition [16], [17]) trained on a plenty of examples. Consequently, according to varied image qualities, we should use different classifiers, i.e., deep or shallow. One rational way is to dynamically combine and switch these two kinds of classification systems.

In this paper, we design a hybrid recognition strategy with classifier switching to adaptively fuse deep learning and shallow classification technologies. Then, we present a variety of experiments with visual recognition tasks, i.e., USPS character recognition, Caltech101 visual object classification, and ICDAR scene text recognition. Specifically, based on the dynamic switching strategy between shallow classification and deep learning, we construct an effective end-to-end scene text recognition system with open-vocabulary. This end-to-end system is evaluated on ICDAR 2011 Robust Reading Competition (Challenge 2) dataset, the $f$ measure of which is 54.5%, much better than 45.2% of the latest state-of-the-art performance.

The rest of this paper is organized as follows. The hybrid recognition strategy between shallow classification and deep learning is presented in Section II. In Section III, we describe a experimental study of our combination strategy on a variety of visual recognition tasks in detail. Final remarks are presented in Section IV.

II. DYNAMIC SWITCHING BETWEEN SHALLOW CLASSIFICATION AND DEEP LEARNING

According to the above descriptions, in common classification cases with plenty but not enough training examples, mixed-quality examples for dozens of categories, deep and shallow learning may have complementary performance, and should be fused in an adaptive way. We propose to dynamically combine shallow classification and deep learning in a sequential style, i.e., we first perform “mainstream” shallow classification, then check the necessity for deep learning by a discriminative switcher; if necessary, we will conduct deep learning for classification. This hybrid recognition strategy with classification switching is shown in Figure 1.

Generally, switching between two classification systems can occur for two main reasons, i.e., the dissatisfaction with the former system’s results, and the adaptation with the next system. For the dissatisfaction, we study the behaviors of the outputs from the “mainstream” shallow classification; the confidence of the recognition output. For the adaptation, we investigate the behaviors of the image itself: the image quality (mainly for blur) with low-level vision features. We simply compute the features with Laplacian of Gaussian (LOG) filters to represent the image quality. All of the features used are describe in Table I.

LOG features are computed in a common and simple way as follows. We use simple $3 \times 3$ Gaussian and Laplacian kernels. After LOG filters, we normalize the LOG response map by the absolute value function. Then, we calculate the histogram with 100 intervals for the absolute resulting map, which directly consists of the 100-dimension features for the image quality.

Actually, this switching is a general two-class classification problem. Considering both the dissatisfaction features for the shallow classification system and the adaptation features for deep learning, we want to construct a switcher based on all these features. However, it is difficult to estimate probability (confidence) of LOG features, because these 100 features with continuous values are always relevant. Moreover, in general cases without enough prior knowledge, the discriminative model is more powerful than the generative one. Consequently, we select SVM to discriminatively learn and construct the classification switcher $^2$. Then, the switcher becomes

$$Y = f_{SVM}(X_1, X_2)$$

where $f_{SVM}$ is the SVM classifier. In each experiment (see Section III), we prepare the data for SVM training with the corresponding training set, where the switching target, 1 or 0, for each example is manually determined by checking $X_1$ and the image quality. SVM ($\epsilon$-SVM) is performed with the Gaussian kernel using default parameters in LIBSVM [18].

III. EXPERIMENTAL STUDY

In the pervious section, we propose an initial strategy, a hybrid recognition system with classifier switching to adaptively fuse deep learning and shallow classification technologies. In this paper, we focus on an experimental study for the above guess and declaration. Consequently, we perform a variety of experiments with visual recognition tasks, i.e., USPS character recognition, Caltech101 visual object classification, and ICDAR 2011 end-to-end scene text recognition with open-vocabulary, all of which are with plenty but not enough training examples.

\[ ^2 \text{We experiment different algorithms, i.e., SVM, AdaBoost and Logistic Regression. The performance of these methods is similar, and we report the results by SVM which has a slightly better classification accuracy.} \]

---

TABLE I. FEATURES FOR SWITCHER LEARNING.

<table>
<thead>
<tr>
<th>Features</th>
<th>dimension</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dissatisfaction</td>
<td>$X_1$: Confidence</td>
<td>1</td>
</tr>
<tr>
<td>Adaptation</td>
<td>$X_2$: LOG</td>
<td>100</td>
</tr>
</tbody>
</table>
A. Experiments with USPS Character Recognition and Caltech101 Object Classification

In the first experiments, we empirically analyze the different effects of shallow “mainstream” classification and deep learning for digital character recognition on USPS dataset and image classification on Caltech101 dataset [19].

At first, for USPS dataset, we train a shallow SVM (from LIBSVM [18]) classifier (named as ShallowUSPS) with weighted direction code histogram features [20] and a convolutional neural networks classifier (named as DeepUSPS) respectively on the training set. The network contains two convolution layers, and two sub sampling layers. The kernel size of convolution layer is $5 \times 5$, and kernel size of sub sampling layer is $2 \times 2$. The number of features map for each convolution layer are 6 and 12 for layer one and two respectively.

For Caltech101 dataset, we use a shallow image classification system with multiple kernels [21] $^3$ (named as ShallowCaltech101) and a deep network classifier [22] $^4$ (named as DeepCaltech101). We select a part of these data sets each of which has more than 70 (fairly plenty) samples. We also remove 6 easy datasets (with 100% accuracy). As a result, there are 26 categories used in our experiments, i.e. airplanes, motorbikes, faces, faces_easy, watch, ketch, hawksbill, brain, butterfly, helicopter, menorah, kangaroo, starfish, ever; buddha, sunflower, scorpion, revolver, laptop, ibis, llama, umbrella, electric_guitar, crab, bonsai, and car_side.

Then, we manually divide the testing set into 10 subsets with roughly same number of examples ($\{S_1, S_2, ..., S_{10}\}$) by categorization with the image quality (roughly from high quality to low quality). Examples for USPS are shown in Figure 2 and Caltech101 datasets in Figure 3 and 4.

Next, we perform classification by 10 times with the shallow classifier and the deep learning respectively in the following way: first on $S_1$, then on $S_1 \cup S_2$, next on $S_1 \cup S_2 \cup S_3$, ..., and finally on all testing set ($S_1 \cup S_2 \cup ... \cup S_{10}$). We present the average recognition accuracy results of this series of classification in Figure 5 and 6 for USPS and Caltech101 tasks respectively.


From both experimental results (Figure 5 and 6), some conclusions can be drawn as follows. In the first several steps, shallow classification techniques have an impressive performance for fairly good quality image data. While, in the middle steps, deep learning methods have outperformed results with rather low-quality but plenty samples. Finally, in the last step, because there are very few samples for the very low-quality category, the classification accuracy of deep learning methods is not encouraged.

From the above experiments, we find that deep learning and shallow classification may have complementary performance in classification systems with mixed-quality, plenty but not enough training examples. Consequently, in order to improve the performance, we construct a hybrid recognition system. In these experiments, we compare three different classification methods, i.e., the shallow classification (ShallowUSPS and ShallowCaltech101), the deep learning (DeepUSPS and DeepCaltech101), and the hybrid recognition methods (in Section II, HybridUSPS and HybridCaltech101) on USPS and Caltech101 datasets. Experimental results are shown in Table II.

As shown in Table II, our hybrid recognition method has the best performance, which empirically verify that adaptively combining two kinds of shallow and deep classification systems is a good choice in real classification and learning systems.
Classification with shallow classifiers. In this conventional OCR flow, the detected text region is first binarized and segmented by text segmentation, and then the segmented text is fed into efficient text recognition with a conventional open source OCR engine (Tesseract-ocr (Version: 3.02.02) 6) for word recognition. In our system, we segment and extract texts by combining results from both pruned MSERs in the text detection stage and local thresholding with the Niblack’s method. This classification stage is effective and efficient for recognizing fairly large and clear scene texts.

Classification switching. After the fast classification with shallow classifiers, we estimate the classifier (OCR engine) and image quality, and switch the system to classification with deep learning if necessary. The classification switcher is learned on not only the classifier outputs 7 but also the low-level vision features of image quality (i.e., Laplacian of Gaussian, LOG). More details are shown in Section II.

Classification with deep learning. In this deep learning flow, the detected text region is directly fed into text recognition by convolutional neural networks without text segmentation. Actually, the character classifier with convolutional neural networks is directly trained on the grayscale character image without any feature extraction. This classification stage is more suitable for recognizing small and blurred scene texts. Here this stage mainly includes two steps: character recognition and word verification. First, in isolated character recognition, we use a two-layer convolutional neural networks as in [17]. In our experiments, we use the character classifier from [17] 9.

Second, in word verification, we use a dynamic programming strategy to link recognized characters and verify a valid word in a dictionary (from Tesseract-ocr), which is actually one of the word spotting techniques in [27], [17]. We run sliding window (by character recognition) over the grayscale word image from left to right. Then based on the classifier outputs’ confidences and the verification scores with words in the dictionary, we use Viterbi algorithm to select the most reliable path (with high scores), which corresponds to the final recognized valid word.

B. Experiments with ICDAR 2011 End-to-End Scene Text Recognition

1) Overview of End-to-End Recognition System: End-to-end scene text recognition is generally first to locate text regions in natural scene images (text detection), then to extract texts (text segmentation), and finally to recognize words in these texts (word recognition). In these scene text end-to-end recognition experiments, text detection is based on our previous state-of-the-art method, USTB_TexStar [23], [24]. This method first identifies MSERs and then prunes them using the strategy of minimizing regularized variations of ERs to extract character candidates. These character candidates are grouped into text candidates by adaptive single-link clustering in which similarity weights and a clustering threshold are learned by a self-training distance metric learning algorithm. Text candidates are evaluated using a character classifier and non-text regions are removed. Finally, each text region is divided into word regions by a word partition step [25].

Specifically, followed the flowchart in Figure 1, our hybrid recognition strategy for end-to-end scene text recognition with open-vocabulary is composed of the following stages 5:

1) Classification with shallow classifiers. In this conventional OCR flow, the detected text region is first binarized and segmented by text segmentation, and then the segmented text is fed into efficient text recognition with a conventional open source OCR engine (Tesseract-ocr (Version: 3.02.02) 6) for word recognition. In our system, we segment and extract texts by combining results from both pruned MSERs in the text detection stage and local thresholding with the Niblack’s method. This classification stage is effective and efficient for recognizing fairly large and clear scene texts.

2) Classification switching. After the fast classification with shallow classifiers, we estimate the classifier (OCR engine) and image quality, and switch the system to classification with deep learning if necessary. The classification switcher is learned on not only the classifier outputs 7 but also the low-level vision features of image quality (i.e., Laplacian of Gaussian, LOG). More details are shown in Section II.

3) Classification with deep learning. In this deep learning flow, the detected text region is directly fed into text recognition by convolutional neural networks without text segmentation. Actually, the character classifier with convolutional neural networks is directly trained on the grayscale character image without any feature extraction. This classification stage is more suitable for recognizing small and blurred scene texts. Here this stage mainly includes two steps: character recognition and word verification. First, in isolated character recognition, we use a two-layer convolutional neural networks as in [17]. In our experiments, we use the character classifier from [17] 9. Second, in word verification, we use a dynamic programming strategy to link recognized characters and verify a valid word in a dictionary (from Tesseract-ocr), which is actually one of the word spotting techniques in [27], [17]. We run sliding window (by character recognition) over the grayscale word image from left to right. Then based on the classifier outputs’ confidences and the verification scores with words in the dictionary, we use Viterbi algorithm to select the most reliable path (with high scores), which corresponds to the final recognized valid word.

2) Experimental Results: In these experiments with the ICDAR 2011 Robust Reading Competition Challenge 2 testing set, we compare our hybrid end-to-end scene text recognition to several state-of-the-art methods 9: NMICVPR2012-Neumann and Matas’s method in [29]; NMICV2012-Neumann and Matas in [30]; NMICV2013, the latest system of Neumann and Matas in [31]; and Weinman et al.’s system in [32]. Experimental results are shown in Table III with the case-sensitive performance for all methods.

![Fig. 5. The series results of the average recognition accuracy in the USPS experiment.](image1.png)

![Fig. 6. The series results of the average classification accuracy in the Caltech101 experiment.](image2.png)

More technical details of our end-to-end scene text recognition system are described in [6].


Here, as our task is word recognition, we use not only the classifier’s confidence but also the existence of the output word in the dictionary from Tesseract-ocr.

The Matlab code for this character classifier is available at http://ai.stanford.edu/~twangcat/ICPR2012_code/SceneTextCNN_demo.tar.

There is also a method for this task by Milyaev et al. [28] that uses a commercial OCR system. We put this method in a different category because the details of the recognition component are not described or published, so it cannot be implemented and extended. This method has shown the fairly good precision, recall and f measure for this task with a precision of 66%, recall of 46% and an f measure of 54%.
As can be seen from Table III, our method produces much better precision and f-measure over other methods on this database. Specifically, the f-measure of our method with case-sensitive performance is 54.5%, much better than the latest 45.2% of NM_{ICCV2013} [31]. Some recognition samples are shown in Figure 7. Note that we use the rejection strategy in the classification with deep learning stage, i.e., if the word recognition confidence from CNN is low, we will reject this result. Consequently, the precision is fairly high in our systems, while compared to NM_{ICCV2013}, our recall is slightly low.

At the same time, the f-measure of our system with “classification only with shallow classifiers” and “classification only with deep learning” is 47.9% and 33.3% respectively, while the f-measure of our hybrid recognition system is impressively 54.5% (see Table III). In other words, the hybrid recognition system can take advantages of both shallow classification and deep learning techniques, and obtain the best performance compared to state-of-the-art end-to-end scene text recognition methods.

IV. CONCLUSION

In this paper, we first argue that in common classification cases with plenty but not enough training examples, mixed-quality examples for dozens of categories, deep and shallow learning may have complementary performance. In real learning and classification systems, one more rational way is to adaptively combine these two kinds of classifiers or classification systems. Then, we propose a hybrid recognition system and try to adaptively combine shallow classification and deep learning. Finally, we present an experimental study about this point in detail. In USPS character recognition, Caltech101 object classification and ICDAR 2011 end-to-end scene text recognition tasks, we empirically verify that shallow classifiers are robust for image examples with a fairly good quality, while deep learning has an encouraged performance on examples with a rather low quality. And our hybrid system by switching these two classification techniques has an impressive performance on all these experiments. We also generally argue that for challenging document analysis and recognition problems in real systems, we should seek solutions from both conventional classification techniques and novel learning methods.

Note that how to combine shallow classification with deep learning with theoretical analysis and application verification in real learning systems is still an interesting and also open issue.

ACKNOWLEDGMENT

The research is partly supported by National Natural Science Foundation of China (61105018, 61175020).

REFERENCES