# Joint Training of Conditional Random Fields and Neural Networks for Stroke Classification in Online Handwritten Documents

Jun-Yu Ye, Yan-Ming Zhang, Cheng-Lin Liu
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, P.R. China
Email: {junyu.ye, ymzhang, liucl}@nlpr.ia.ac.cn

***Abstract*—The task of text/non-text stroke classification in online handwritten documents is an essential preprocessing step in document analysis. It is also a challenging problem since in many cases local features are not enough to generate high accuracy results and contextual information, such as temporal information and spatial information, must be carefully considered. In this paper, we propose a novel method, which jointly trains a combined model of conditional random fields and neural networks, to solve this problem. Both our unary and pairwise potentials are formulated as neural networks. The parameters of conditional random fields and neural networks are learned together during the training process. With much fewer parameters and faster speed, our method achieves impressive performance on the IAMonDo database, a publicly available database of freely handwritten documents.**

## I. Introduction

In recent years, with the emergence of digital pens, tablets, electronic whiteboards equipped with pen-based and touch-based handwriting interfaces, online handwritten documents have become more and more popular. However, since these documents always contain various type contents, e.g. text blocks, list, tables, diagrams, it is hard to design one general engine to deal with all different types. One natural solution is to first separate textual strokes from non-textual strokes and then process them with different engines. For these reasons, the task of text/non-text stroke classification is an essential preprocessing step in document analysis [1].

The task of text/non-text classification in online handwritten document is very challenging since text strokes and non-text strokes can be extremely similar. As shown in Fig. 1, a circle can be either a letter 'o' or a wheel of a car. Thus, the feature of an individual stroke is not enough to determine its label and the contextual information, such as spatial relationship and temporal relationship, plays a key role in the classification.

Conditional random fields (CRFs) is one of the most popular methods for exploiting the contextual information. Previous works, such as [3], [1], typically employ a two-step procedure. First, they train a local classifier. Then, they learn a CRF model by using this classifier as the unary potential and some simple function as the pairwise potential. Although this sequential training approach is efficient, it is suboptimal since the dependencies between CRFs and classifiers are ignored.

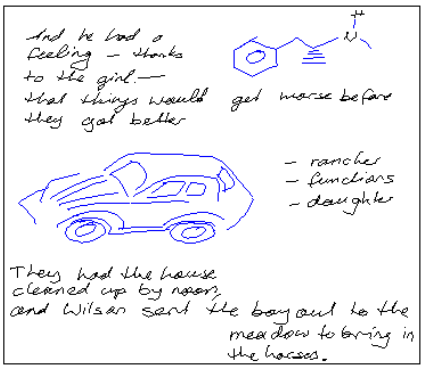



Fig. 1. A sample document from the IAMonDo database. Text strokes are shown in black, non-text strokes are shown in blue.

In this paper, we propose a novel method to combine conditional random fields and neural networks (NNs), and apply it to text/non-text classification in online handwritten documents. Specifically, our contribution is in two folds. First, we propose a chain CRF model that formulates the unary potentials and pairwise potentials by two NNs. Benefit from the two NNs, our model has more powerful representation ability than previous works and thus can better exploit the contextual information. Second, by taking advantage of the chain structure of CRF, we design an efficient algorithm that jointly learns the parameters of these networks. Experiment results demonstrate the effective and superiority of this combined model and our training approach. With much fewer parameters and faster speed, our method obtains the top result on the IAMonDo database.

The rest of this paper are organized as follows. In Section 2, we briefly review works that are related to this paper. In Section 3, we present our method for text/non-text stroke classification in online handwritten documents. In Section 4, the experiments and results are presented. Finally, we draw conclusion in Section 5.

## II. Related works

The task of text/non-text classification has been researched for a long period. The work of Jain et al. [7] proposed a method which extracted strictly local features, length and curvature, to distinguish between text and non-text strokes. Indermühle et al. [6] applied same features with a support vector machine for isolated stroke classification and achieved 91.3% accuracy on the IAMonDo database. Peterson et al. [14] proposed to extract features not only from the stroke to be classified but also from surrounding strokes to integrate contextual information. In order to exploit more contextual information, graphical models, like Markov random fields and conditional random fields, have been used to tackle this problem. Zhou et al. [17] and Delaye et al. [1] proposed methods based on Markov random fields and conditional random fields to better integrate contextual information, respectively. Delaye explored multiple interactions between strokes, like spatial system, temporal system, intersecting system, lateral system, stroke continuation system and discussed the influence of different systems on the classification accuracy. His fully system obtained 97.23% accuracy on the IAMonDo database. Besides, since the temporal information in online documents is crucial, tools for exploiting temporal relationship, like recurrent neural networks(RNNs), are also popular. Indermühle et al. [5] presented a method based on bidirectional long short-term memory(BLSTM) neural networks and achieved 97.01% accuracy on the IAMonDo datebase. Van et al. [15] proposed to use multiple BLSTM neural networks to capture global and local contexts and the ensemble classifiers achieved the state-of-art accuracy 98.30% on the IAMonDo database.

Recently, as convolutional neural networks(CNNs) become the preferred choice in image processing and computer vision, the deep structured model, combined with CNNs and CRFs, obtained state-of-art results in semantic segmentation. Zheng, et al. [16] implemented the mean-field inference procedure of CRFs as RNNs which enables the end-to-end joint training of CRFs and CNNs. Lin, et al. [11] formulated both unary potentials and pairwise potentials as CNNs and jointly trained the parameters of CNNs and CRFs with piecewise training.

## III. Method Description

In this work, we formulate the online stroke classification task as a particular structured prediction problem and solve it under the framework of CRFs. In the following, we first introduce our model which is a combination of CRFs and NNs, and then propose efficient learning and inference algorithms for the model.

### A. Problem definition

We are given a set of labeled online documents $S = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}), i = 1, \ldots, N\}$, in which each document $\mathbf{x}^{(i)}$ is represented by a sequence of strokes $\{\mathbf{x}_t^{(i)}, t = 1, \cdots, T_i\}$ and each stroke has one associated label $\mathbf{y}_t^{(i)} \in \{+1, -1\}$ for text and non-text classes. The task is to learn a model from the training set $S$ that can predict test documents with high accuracy.

### B. Model formulation

CRFs were first introduced by Lafferty et al. [10], as a type of discriminative undirected probabilistic graphical model. It is a powerful method for various structured prediction problems, such as image semantic segmentation [9], part-of-speech tagging [2], name identity recognition [13]. A CRF model can be defined as:

$$P(\mathbf{y}|\mathbf{x}; \mathbf{w}) = \frac{1}{Z(\mathbf{x}; \mathbf{w})} \exp\{F(\mathbf{x}, \mathbf{y}; \mathbf{w})\}, \tag{1}$$

where $\mathbf{w}$ is the learnable model parameters, $F(\mathbf{x}, \mathbf{y}; \mathbf{w})$ is the potential function and $Z(\mathbf{x}; \mathbf{w}) = \sum_{\mathbf{y}} \exp\{F(\mathbf{x}, \mathbf{y}; \mathbf{w})\}$ is the partition function.

The potential function is designed to capture the dependency between $\mathbf{y}$ given $\mathbf{x}$, and thus can be of arbitrary forms in general. However, for the online stroke classification problem considered in this work, the most important relationship within data is the temporal relationship. Therefore, we can just model the pairwise relationship between adjacent strokes. This results in a chain CRF model whose potential function can be formulated as:

$$F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \sum_t F_U(\mathbf{x}, \mathbf{y}_t; \mathbf{w}) + \sum_t F_P(\mathbf{x}, \mathbf{y}_t, \mathbf{y}_{t+1}; \mathbf{w}). \tag{2}$$

Here, $F_U$ is the unary potential function which works as a stroke classifier, while $F_P$ is the pairwise potential function which aims to capture the dependency between adjacent strokes. The key benefit of this simple structure is that it allows fast exact inference in testing and also brings much convenience in the training process.

Next we introduce how to formulate the unary potentials $F_U$ and pairwise potentials $F_P$ with NNs.

*1) Unary potentials:* We formulate the unary potential function as follows:

$$F_U(\mathbf{x}, \mathbf{y}_t; \mathbf{w}) = \sum_{k=1}^{K} \delta(k = \mathbf{y}_t) \phi_k(\mathbf{x}_t; \mathbf{w}). \tag{3}$$

Here $\delta(\cdot)$ is the indicator function, $\phi_k$ is the unary network output value that corresponds to the $k$-th class, $K = 2$ is the number of classes. The output of unary network is equal to $K$. The input of unary network is the features extracted from a single stroke as presented in TABLE I. These features are expansion of [1] which have been showed effectiveness in this task.

*2) Pairwise potentials:* We formulate the pairwise potential function as follows:

$$F_P(\mathbf{x}, \mathbf{y}_t, \mathbf{y}_{t+1}; \mathbf{w}) = \sum_{p=1}^{K} \sum_{q=1}^{K} \delta(p = \mathbf{y}_t) \delta(q = \mathbf{y}_{t+1}) \phi_{p,q}(\mathbf{x}_t, \mathbf{x}_{t+1}; \mathbf{w}). \tag{4}$$

Here $\phi_{p,q}$ is the pairwise network output. It is the score value for the node pair $(t, t+1)$ when they are labeled with the class value $(p, q)$, which measures the compatibility of

TABLE I
23 DESCRIPTORS EXTRACTED FROM STROKE $x_k$ AND LOCAL CONTEXT.

| # | Description |
|---|---|
| 1 | Trajectory length of $x_k$ |
| 2 | Area of the convex hull of $x_k$ |
| 3 | Duration of the stroke |
| 4 | Ratio of the principal axis of $x_k$ |
| 5 | Rectangularity of the minimum area bounding rectangle of $x_k$ |
| 6 | Circular variance of points of $x_k$ around its centroid |
| 7 | Normalized centroid offset along the principal axis |
| 8 | Ratio between first-to-last point distance and trajectory length |
| 9 | Accumulated curvature |
| 10 | Accumulated squared perpendicularity |
| 11 | Accumulated signed perpendicularity |
| 12 | Width of $x_k$, normalized by the median stroke height in the document |
| 13 | Height of $x_k$, normalized by the median stroke height in the document |
| 14 | Number of temporal neighbours of $x_k$ |
| 15 | Number of spatial neighbours of $x_k$ |
| 16 | Average of the distances from $x_k$ to time neighbours |
| 17 | Standard deviation of the distances from $x_k$ to time neighbours |
| 18 | Average of lengths of time neighbours |
| 19 | Standard deviation of lengths of time neighbours |
| 20 | Average of the distances from $x_k$ to space neighbours |
| 21 | Standard deviation of the distances from $x_k$ to space neighbours |
| 22 | Average of lengths of space neighbours |
| 23 | Standard deviation of lengths of space neighbours |

TABLE II
19 DESCRIPTORS EXTRACTED FOR A PAIR OF STROKE $x_u, x_v$.

| # | Description |
|---|---|
| 1 | Minimum distance between 2 strokes |
| 2 | Minimum distance between the endpoints of 2 strokes |
| 3 | Maximum distance between the endpoints of 2 strokes |
| 4 | Distance between the centers of the 2 bounding boxes of 2 strokes |
| 5 | Horizontal distances between the centroids of 2 strokes |
| 6 | Vertical distances between the centroids of 2 strokes |
| 7 | Off-stroke distance between 2 strokes |
| 8 | Off-stroke distance projected on $X$ and $Y$ axes |
| 9 | Temporal distance between 2 strokes |
| 10 | Ratio of off-stroke distance to temporal distance |
| 11 | Ratio of off-stroke distance to projected on $X, Y$ axes to temporal distance |
| 12 | Ratio of area of the largest bounding box of 2 strokes to that of their union |
| 13 | Ratio of widths of the bounding boxes of 2 strokes |
| 14 | Ratio of heights of the bounding boxes of 2 strokes |
| 15 | Ratio of diagonals of the bounding boxes of 2 strokes |
| 16 | Ratio of areas of the bounding boxes of 2 strokes |
| 17 | Ratio of lengths of 2 strokes |
| 18 | Ratio of durations of 2 strokes |
| 19 | Ratio of curvatures of 2 strokes |

the label pair $(\mathbf{y}_i, \mathbf{y}_j)$ given the input strokes. The output of pairwise network is equal to $K^2$. The input of pairwise network is the pairwise features extracted from a pair of strokes as presented in TABLE II, the same as in [15].

### C. Inference

We adopt the maximum a posteriori(MAP) strategy to predict the labels of strokes in a new document. This leads to the following problem:

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}). \tag{5}$$

Since the structure of our CRF is a chain, we can solve this problem by applying exact inference algorithms, such as max-sum algorithm [8]. Max-sum algorithm is a classic message passing algorithm for performing inference on graphical models. When applied to tree or chain structures, its computational complexity is linear in the number of nodes.

### D. Learning

Given the training data set $S = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}), i = 1, \ldots, N\}$, we train our model $P(\mathbf{y}|\mathbf{x}; \mathbf{w})$ by maximizing its likelihood on $S$. This is equivalent to solving the following problem with respect to the parameters $\mathbf{w}$:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{i=1}^{N} -\ln P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}; \mathbf{w}). \tag{6}$$

We solve this problem by the limited-memory BFGS algorithm. The gradient of the negative log likelihood for a particular sample $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ is

$$\begin{aligned}
&-\frac{\partial}{\partial \mathbf{w}} \ln P(\hat{\mathbf{y}}|\hat{\mathbf{x}}; \mathbf{w}) \\
&= \frac{\partial \ln Z(\hat{\mathbf{x}}; \mathbf{w})}{\partial \mathbf{w}} - \frac{\partial F(\hat{\mathbf{x}}, \hat{\mathbf{y}}; \mathbf{w})}{\partial \mathbf{w}} \\
&= \mathbb{E}_{P(\mathbf{y}|\hat{\mathbf{x}}; \mathbf{w})}[\frac{\partial F(\hat{\mathbf{x}}, \mathbf{y}; \mathbf{w})}{\partial \mathbf{w}}] - \frac{\partial F(\hat{\mathbf{x}}, \hat{\mathbf{y}}; \mathbf{w})}{\partial \mathbf{w}} \\
&= \mathbb{E}_{P(\mathbf{y}|\hat{\mathbf{x}}; \mathbf{w})}[\sum_t \frac{\partial F_U(\hat{\mathbf{x}}, \mathbf{y}_t; \mathbf{w})}{\partial \mathbf{w}} + \sum_t \frac{\partial F_P(\hat{\mathbf{x}}, \mathbf{y}_t, \mathbf{y}_{t+1}; \mathbf{w})}{\partial \mathbf{w}}] \\
&\quad - \left( \sum_t \frac{\partial F_U(\hat{\mathbf{x}}, \hat{\mathbf{y}}_t; \mathbf{w})}{\partial \mathbf{w}} + \sum_t \frac{\partial F_P(\hat{\mathbf{x}}, \hat{\mathbf{y}}_t, \hat{\mathbf{y}}_{t+1}; \mathbf{w})}{\partial \mathbf{w}} \right).
\end{aligned} \tag{7}$$

Thus, it remains to explain how to calculate the gradient of the unary network and the pairwise network with respect to $\mathbf{w}$. We first consider the gradient of the unary network:

$$\begin{aligned}
&\mathbb{E}_{P(\mathbf{y}|\hat{\mathbf{x}}; \mathbf{w})}[\frac{\partial F_U(\hat{\mathbf{x}}, \mathbf{y}_t; \mathbf{w})}{\partial \mathbf{w}}] - \frac{\partial F_U(\hat{\mathbf{x}}, \hat{\mathbf{y}}_t; \mathbf{w})}{\partial \mathbf{w}} \\
&= \left( \sum_{\mathbf{y}} P(\mathbf{y}|\hat{\mathbf{x}}; \mathbf{w}) \frac{\partial F_U(\hat{\mathbf{x}}, \mathbf{y}_t; \mathbf{w})}{\partial \mathbf{w}} \right) - \frac{\partial F_U(\hat{\mathbf{x}}, \hat{\mathbf{y}}_t; \mathbf{w})}{\partial \mathbf{w}} \\
&= \left( \sum_{\mathbf{y}_t} P(\mathbf{y}_t|\hat{\mathbf{x}}; \mathbf{w}) \frac{\partial F_U(\hat{\mathbf{x}}, \mathbf{y}_t; \mathbf{w})}{\partial \mathbf{w}} \right) - \frac{\partial F_U(\hat{\mathbf{x}}, \hat{\mathbf{y}}_t; \mathbf{w})}{\partial \mathbf{w}} \\
&= \sum_{\mathbf{y}_t} [P(\mathbf{y}_t|\hat{\mathbf{x}}; \mathbf{w}) - \delta(\mathbf{y}_t = \hat{\mathbf{y}}_t)] \frac{\partial F_U(\hat{\mathbf{x}}, \mathbf{y}_t; \mathbf{w})}{\partial \mathbf{w}}.
\end{aligned} \tag{8}$$

Thanks to the chain structure, the marginal distribution $P(\mathbf{y}_t|\hat{\mathbf{x}}; \mathbf{w})$ can be calculated by the sum-product algorithm [8] in linear time. Furthermore, (8) can be computed efficiently by one-pass backward propagation when $P(\mathbf{y}_t|\hat{\mathbf{x}}; \mathbf{w})$ is known.

In a very similar way, the gradient of the pairwise network can be calculated as follows,

$$\begin{aligned}&\mathbb{E}_{P(\mathbf{y}|\hat{\mathbf{x}};\mathbf{w})}[\frac{\partial F_P(\hat{\mathbf{x}},\mathbf{y}_t,\mathbf{y}_{t+1};\mathbf{w})}{\partial \mathbf{w}}] - \frac{\partial F_P(\hat{\mathbf{x}},\hat{\mathbf{y}}_t,\hat{\mathbf{y}}_{t+1};\mathbf{w})}{\partial \mathbf{w}}\\ &= \left(\sum_{\mathbf{y}_t,\mathbf{y}_{t+1}} P(\mathbf{y}_t,\mathbf{y}_{t+1}|\hat{\mathbf{x}};\mathbf{w})\frac{\partial F_P(\hat{\mathbf{x}},\mathbf{y}_t,\mathbf{y}_{t+1};\mathbf{w})}{\partial \mathbf{w}}\right)\\ &\quad - \frac{\partial F_P(\hat{\mathbf{x}},\hat{\mathbf{y}}_t,\hat{\mathbf{y}}_{t+1};\mathbf{w})}{\partial \mathbf{w}}.\end{aligned} \tag{9}$$

Similarly, the marginal distribution $P(\mathbf{y}_t,\mathbf{y}_{t+1}|\hat{\mathbf{x}};\mathbf{w})$ can be calculated by the sum-product algorithm and (9) can be calculated with one-pass backward propagation through the pairwise network. Algorithm 1 summarizes the procedure of calculating the gradient.

## IV. Experiment and Results

### A. IAMonDo database

We conduct our experiments on the IAMonDo database [6], a publicly available collection of freely handwritten online documents with full ground truth content annotation and transcription. The database consists of about 1000 documents written by 200 writers, mixing handwritten text, drawings, diagrams, formulas, tables, lists and marking elements arranged in an unconstrained way. The database is split into five disjoint sets, each containing roughly 200 documents. For our experiments, we use 403 documents from set 0 and 1 for training, 200 documents from set 2 for validation and 203 documents from set 3 for testing. In order to measure the method's ability to distinguish text strokes from non-text strokes, we derive the corresponding ground truth of context categories to text or non-text as suggested by the database authors [4].

### B. Feature normalization

In order to make the feature density function closer to a Gaussian distribution, we preprocess features using power transform with the coefficient set to 0.5. Thus,

$$f' = \text{sgn}(f)\sqrt{|f|}. \tag{10}$$

Further more, we normalize the values of each feature based on the mean $\mu$ and standard deviation $\sigma$ in order to standardize the feature values into the same scale. The normalized feature value is calculated as

$$f'' = (f' - \mu)/\sigma. \tag{11}$$

### C. Hyperparameters and experimental setting

In the feature extraction of single strokes and pairs of strokes, we follow the strategy of [15] that two strokes are considered as temporal and spatial neighbours if the temporal and spatial distances between them are less than thresholds 3.5s and 4 pixels. The unary and pairwise networks are networks with one hidden layer. The number of nodes in the hidden layers of unary and pairwise networks are 10 and 5, respectively. The activation function of the hidden layer is sigmoid function and the activation function of output layer is identity function. In the training process, a second order gradient descent algorithm, limited-memory BFGS [12] is used to minimize the loss. We adopt the libLBFGS [1] library for implementation.

Since the initial parameters of NNs affect the final classification accuracy, we randomly initialize these parameters from $U(-0.01, 0.01)$ to train the model and repeat each experiment for 20 times. Our method is implemented by C++ and all experiments are performed on a computer with an Intel Core I7-4790 CPU(3.60GHz).

### D. Results

The experiment results are shown in TABLE III. The column of 'mean rate' shows the average and standard deviation of the classification accuracy in 20 experiments. The column 'minimal rate' and 'maximal rate' shows the minimal and maximal accuracy achieved in 20 experiments. The column of 'time' shows the computation time for classification on the entire test set and does not include the time taken for feature extraction.

As far as we know, this is the best accuracy result achieved by a single model. Compared with [15], the mean accuracy of our model 97.80% is higher than the mean accuracy 97.56% achieved by its best global context based classifier GSC26_LSTM. The maximal and minimal accuracy 97.72% and 97.96% are also higher than its 97.30% and 97.81%. In addition, the classification time of our model 0.69s is also faster than GSC26_LSTM with 1.62s. Compared with its combined classifiers integrating global context and local context, the maximal accuracy of our model 97.96% is equal to the accuracy achieved by its best local context integrated classifier PCC combined with GSC26_LSTM and GPC19Q_LSTM. Since we don't ensemble classifiers to improve the accuracy, we don't compare our result with its ensemble accuracy 98.30%.

The another advantage of our model is the number of parameters. The unary network contains 23 input nodes, 10 hidden nodes and 2 output nodes; while the pairwise network contains 19 input nodes, 5 hidden nodes and 4 output nodes. Thus, the total number of parameters of our model is 386. In contrast, van, et al. [15] used four-layered networks with two fully connected recurrent hidden layers and the number of units in these two layers are 10 and 30, whose parameters are much more than ours.

As we have seen, our method achieves higher accuracy with fewer parameters on this task. It validates the effectiveness of this combined model and the joint training approach to learn the parameters. Fig. 2 shows two documents with successfully classified strokes.

### E. Evaluation of errors

Fig. 3 shows examples with misclassified strokes. In Fig. 3 (a), a formula with big size is misclassified as non-text since its

[1] http://www.chokkan.org/sofminimal and tware/liblbfgs/

**Algorithm 1** Computation of the gradient for one document

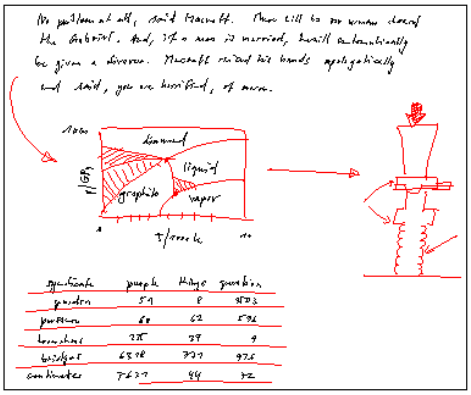
**Input:** Training example $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$

**Output:** Gradients $\nabla_{\mathbf{w}_U}$ and $\nabla_{\mathbf{w}_P}$ of the negative conditional log likelihood with respect to the unary network and pairwise network

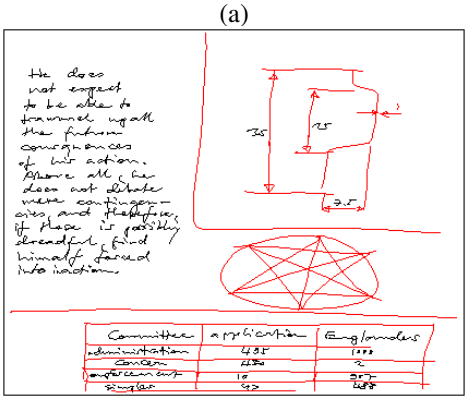
1: Forward pass to compute $F_U(\hat{\mathbf{x}}, \mathbf{y}_t; \mathbf{w})$ and $F_P(\hat{\mathbf{x}}, \mathbf{y}_t, \mathbf{y}_{t+1}; \mathbf{w}), \forall t$.
2: Compute marginal probabilities $P(\mathbf{y}_t|\hat{\mathbf{x}}; \mathbf{w})$ and $P(\mathbf{y}_t, \mathbf{y}_{t+1}|\hat{\mathbf{x}}; \mathbf{w})$ by sum-product algorithm.
3: Set the errors $[P(\mathbf{y}_t|\hat{\mathbf{x}}; \mathbf{w}) - \delta(\mathbf{y}_t = \hat{\mathbf{y}}_t)]$ of the unary network and use backward propagation to calculate the gradient $\nabla_{\mathbf{w}_U}(t), \forall t$. Set the errors $[P(\mathbf{y}_t, \mathbf{y}_{t+1}|\hat{\mathbf{x}}; \mathbf{w}) - \delta(\mathbf{y}_t = \hat{\mathbf{y}}_t)\delta(\mathbf{y}_{t+1} = \hat{\mathbf{y}}_{t+1})]$ of the pairwise network and use backward propagation to calculate the gradient $\nabla_{\mathbf{w}_P}(t, t+1), \forall t$.
4: The gradient of the unary network is $\nabla_{\mathbf{w}_U} = \sum_t \nabla_{\mathbf{w}_U}(t)$.
5: The gradient of the pairwise network is $\nabla_{\mathbf{w}_P} = \sum_t \nabla_{\mathbf{w}_P}(t, t+1)$.

TABLE III
RECOGNITION RATE FOR TEXT/NON-TEXT STROKE CLASSIFICATION BY DIFFERENT METHODS

| method | mean rate(%) | minimal rate(%) | maximal rate(%) | time(s) |
|---|---|---|---|---|
| Indermühle [5] | - | - | 97.01 | - |
| Delaye [1] | - | - | 97.23 | 190.00 |
| GSC26_LSTM [15] | 97.56 | 97.30 | 97.81 | 1.62 |
| PCC [15] | - | - | **97.96** | 3.23 |
| our model with two-step training | $96.56 \pm 0.10$ | 96.38 | 96.73 | 1.68 |
| our model with joint training | **97.80** $\pm 0.08$ | **97.72** | **97.96** | **0.69** |



(a)



(b)

Fig. 2. Examples with successfully classified strokes. Test strokes are shown in black, non-text strokes are shown in red.

size is much bigger than the usual size of texts in the database. In order to correct errors of this type, more scale invariant features should be extracted. In Fig. 3 (b), multiple strokes are misclassified because of the lack of modeling spatial relationship. It seems that integrating spatial information may help to solve this problem.

### *F. Compared to two-step training*

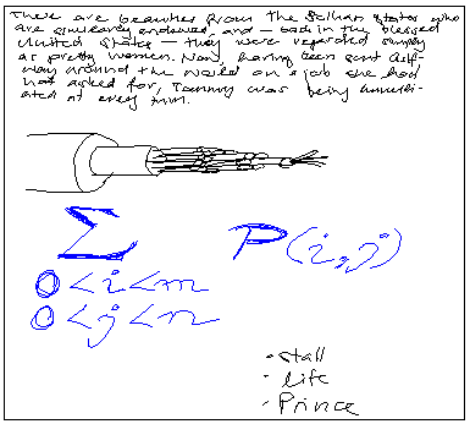
This part we evaluate the difference between joint training and two-step training. We define our compared model as follows:

$$P(\mathbf{y}|\mathbf{x}; \mathbf{w}, \boldsymbol{\theta}) = \frac{1}{Z(\mathbf{x}; \mathbf{w})} \exp\{\sum_t \sum_{\mathbf{y}_t} \theta_{\mathbf{y}_t} F_U(\mathbf{x}, \mathbf{y}_t; \mathbf{w}) + \sum_t \sum_{\mathbf{y}_t, \mathbf{y}_{t+1}} \theta_{\mathbf{y}_t, \mathbf{y}_{t+1}} F_P(\mathbf{x}, \mathbf{y}_t, \mathbf{y}_{t+1}; \mathbf{w})\}. \tag{12}$$

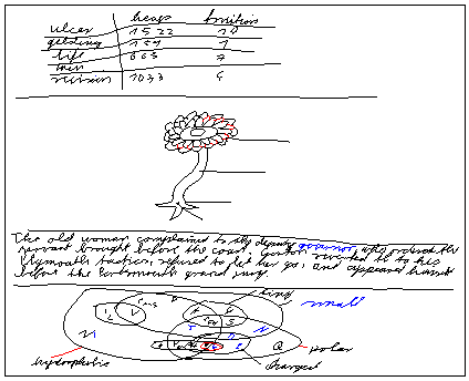
We first train the unary classifier and pairwise classifier with softmax loss and fix these classifiers to learn the parameters $\boldsymbol{\theta}$ of the CRF model. We also repeat the experiment for 20 times with different initialization for comparison. The compared results are shown in TABLE III. As we can see, the mean classification accuracy achieved by two-step training model is 1.24% lower than the joint training model, which shows the superiority of joint training approach.

## V. CONCLUSION

In this paper we present a combined model of CRFs and NNs for text/non-text stroke classification in online handwritten documents. The unary potentials and pairwise potentials in CRFs are both formulated by NNs to build more powerful representation. We discuss how to compute the gradient of




(a)



(b)

Fig. 3. Examples with misclassified strokes. Correct strokes are shown in black. Text strokes misclassified as non-text strokes are shown in blue. Non-text strokes misclassified as text strokes are shown in red.

this combined model with exact inference algorithm and back propagation. We achieve impressive performance on the IAMonDo database with fewer parameters and faster speed, which demonstrates the effectiveness and superiority of this combined model. We also compare the difference between the two-step training approach and joint training approach, and analyze the common misclassifications of our model.

One future direction is to consider integrating the spatial information in CRFs, e.g. model the spatial relationship by another NNs, and jointly train the combined model. Since the CRF structure will be more complex after including the spatial relationship, the exact computation of marginal probability will be impossible and some kind of approximation is needed. It remains to be a problem to derive an effective approach to jointly train the combined model of NNs and CRFs with complex structure.

## REFERENCES


[1] Adrien Delaye and Cheng-Lin Liu. Contextual text/non-text stroke classification in online handwritten notes with conditional random fields. *Pattern Recognition*, 47(3):959–968, 2014.

[2] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers*, volume 2, pages 42–47, 2011.

[3] Xuming He, Richard S Zemel, and Miguel Á Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–695, 2004.

[4] Emanuel Indermühle. *Analysis of Digital Ink in Electronic Documents*. PhD thesis, University of Bern, 2012.

[5] Emanuel Indermühle, Volkmar Frinken, and Horst Bunke. Mode detection in online handwritten documents using blstm neural networks. In *Proceedings of the 13th documents Intenetional Conference on Frontiers in Handwriting Recognition*, pages 302–307, 2012.

[6] Emanuel Indermühle, Marcus Liwicki, and Horst Bunke. Iamondo-database: an online handwritten document database with non-uniform contents. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 97–104, 2010.

[7] Anil K Jain, Anoop M Namboodiri, and Jayashree Subrahmonia. Structure in on-line documents. In *Proceedings of the 6th International Conference on Document Analysis and Recognition*, pages 844–848, 2001.

[8] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[9] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems*, pages 109–117, 2011.

[10] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 2001.

[11] Guosheng Lin, Chunhua Shen, Ian Reid, et al. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv preprint arXiv:1504.01013*, 2015.

[12] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[13] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the 7th Conference on Natural language learning at HLT-NAACL*, volume 4, pages 188–191, 2003.

[14] Eric Jeffrey Peterson, Thomas F Stahovich, Eric Doi, and Christine Alvarado. Grouping strokes into shapes in hand-drawn diagrams. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, volume 10, page 14, 2010.

[15] Truyen Van Phan and Masaki Nakagawa. Combination of global and local contexts for text/non-text classification in heterogeneous online handwritten documents. *Pattern Recognition*, 51:112–124, 2016.

[16] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[17] Xiang-Dong Zhou and Cheng-Lin Liu. Text/non-text ink stroke classification in japanese handwriting based on markov random fields. In *Proceedings of the 9th International Conference on Document Analysis and Recognition*, volume 1, pages 377–381, 2007.