Parallel Sequence Classification using Recurrent Neural Networks and Alignment

Federico Raue*†, Wonmin Byeon*†, Thomas M. Breuel*, Marcus Liwicki*‡
{federico.raue, wonmin.byeon}@dfki.de, tmb@cs.uni-kl.de and marcus.liwicki@unifr.ch
*University of Kaiserslautern, Germany
†German Research Center for Artificial Intelligence (DFKI), Germany
‡University of Fribourg, Switzerland

***Abstract*—The aim of this work is to investigate Long Short-Term Memory (LSTM) for finding the semantic associations between two parallel text lines of different instances of the same class sequence. In this work, we propose a new model called *class-less classifier*, which is cognitive motivated by a simplified version of the infants learning. The presented model not only learns the semantic association but also learns the relation between the labels and the classes. In addition, our model uses two parallel class-less LSTM networks and the learning rule is based on the alignment of both networks. For testing purposes, a parallel sequence dataset is generated based on MNIST dataset, which is a standard dataset for handwritten digit recognition. The results of our model were similar to the standard LSTM.**

I. Introduction

One challenging area is to exploit the semantic relation in multimodal scenarios. Several examples showed how the performance was improved in different applications, such as, video scene classification [1], semantic indexing [2], multimedia retrieval [3]. Finding the semantic relation in those scenarios shares similar conditions to infants learning. Infants have two independent pathways (visual and auditory) for developing object recognition and vocabulary acquisition (respectively). Although both pathways are independent, several authors claimed there is a relation between the object recognition and vocabulary acquisition [4], [5], [6]. This common representation could be interpreted as the semantic relation in pattern recognition problems.

Consequently, we are defining a new problem based on how infants find semantic relations between two different inputs (and modalities). More formally, the goal is to find associations between two parallel text lines of different instances of the same class-sequence. In this case, the relation between the 'classes' and the 'labels' are also learned. We are referring 'labels' to the semantic meaning of the entities and 'classes' to the representations of the labels in the networks. In contrast to standard training rules, the classes are initially labeled and are fixed during the training.

Long Short-Term Memory(LSTM) has been applied successfully to OCR problems. *Graves and Schmidhuber* [7] applied a multidimensional LSTM to offline handwritten recognition problems. *Liwicki et al.* [8] improved their work in on-line handwriting recognition using LSTM against the previous model, which was developed based on Hidden Markov Models (HMM). *Breuel et al.* [9] found results comparable to other OCR systems in printed text and Fraktur using a combination of line normalization and LSTM.

In this work, a new model is proposed, called *class-less classifier*. The proposed model uses two parallel LSTM for finding the association between two text lines. *Mioulet et al.* [10] proposed a similar idea of combine two parallel LSTM for creating feature combinations. However, our model has several difference with their work. First, the training rule is different. Our model trains one network given the other network as target, and vice versa. In contrast, they do not modify the standard training. Second, their work assumed that the inputs have the same length; whereas, our model align the output of both networks because the lengths are different. Third, the training of our model is cognitive motivated, that involve to use unlabeled classes. This paper is organized as follows. Section II explains in more detail the *class-less classifier* and its relation to LSTM. Section III explains the setup of our experiment and the results that our model obtained. Section IV is the conclusions and future work.

II. Methodology

The goal of our model is to converge two sequences of text lines with the same ordered elements. The proposed model uses a modified learning rule, which is based on unlabeled classes and alignment of both networks. In other words, the class representations of the labels are not defined beforehand and they are learnt during the training. We introduce a statistical constraint that is applied for labeling the classes given the current output of the network. Also, we introduce a learning rule for converging both networks to the same structure between the labels and the classes. The learning rule uses one network as the target of the other network and vice versa. Figure 2 shows a general overview of the proposed model.

A. Long Short-Term Memory (LSTM)

LSTM is a recurrent neural network with memory cells and gates for avoiding the vanishing gradient problems[11], [12]. Recent results show the abilities of LSTM using an extra layer -Connectionist Temporal Classification (CTC)- for classifying unsegmented inputs, for example, speech recognition[13] and OCR[9]. CTC adds an extra class called *blank class (b)*, which its role is to tag the beginning and the ending of each class. For example, the target sequence is *"246"* and CTC converts the initial sequence to *"b2b4b6b"*. In addition, CTC is motivated by the forward-backward algorithm for training Hidden Markov Models(HMM)[14]. Algorithm 1 shows a summary in pseudocode of the standard training rule of LSTM.

Algorithm 1 Pseudocode of the standard training rule of LSTM for sequence classification using CTC

```
for $t = 1$ to $T$ do
    $z_t \leftarrow lstm.forward\_step(input_t)$
end for
$target_{1...T} \leftarrow forward\_backward(z_{1...T}, target\_sequence)$
$delta_{1...T} \leftarrow target_{1...T} - z_{1...T}$
for $t = T$ to $1$ do
    $lstm.backward\_step(delta_t)$
end for
$classes \leftarrow decoding(z_{1...T})$
```

The proposed model modified the standard training rule of LSTM in three aspects. First, the $forward\_backward$ algorithm in CTC required the labeled ground truth, which usually is fixed during the training. In contrast, our model learns the labeled ground truth during the training. With this in mind, the proposed model adds an extra component, which assigns a label to each class given the output of the network. Second, the $deltas$ in LSTM are calculated between the output of the network and the output of the $forward\_backward$ algorithm. In our model, one network uses the other network as a target. In other words, the $deltas$ are calculated between the output of one network and the output of the $forward\_backward$ algorithm from the other network, and vice versa. Third, both networks are aligned for finding the relation between the time steps using using Dynamic Time Warping (DTW). In this component, each network is transferring information to the other network.

B. Statistical Constraint

The goal of this component is to label the classes given the output of the network. As a result, we define a set of $weighted\ terms(\gamma_{label})$, which modify the distribution of the output in order to assign one label-class relation based on *winner-take-all* rule (see Figure 1).

For explanation purposes, the $weighted\ term(\gamma_{label})$ and the output of LSTM for each time step (z_t) are defined as the following vectors

$$\gamma_{label} = \begin{bmatrix} \gamma_{label,0} \\ \vdots \\ \gamma_{label,n-1} \end{bmatrix} ; z_t = \begin{bmatrix} z_{t,0} \\ \vdots \\ z_{t,n-1} \end{bmatrix}$$

where n is the number of classes and the second index represents the classes $c = 0, \ldots, n-1$. In addition, we defined the $weighted\ distribution$ as the average of the network output over the timesteps:

$$\hat{z}_{label} = \frac{1}{T} \sum_{t=1}^{T} (z_t)^{\gamma_{label}} \quad (1)$$

Hence, the relation between the label and the class representation is obtained by retrieving the max element of z_{label}:

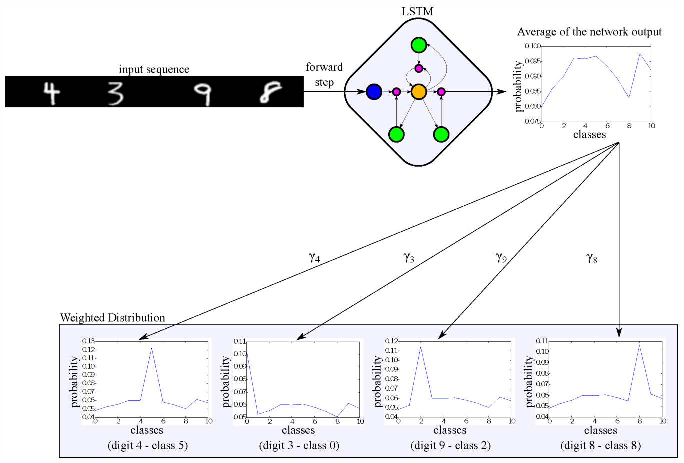

Fig. 1. Example of the statistical constraint. The output of the network is averaged over all the time steps. The $weighted\ terms(\gamma_4, \gamma_3, \gamma_9, \gamma_8)$ represent the input sequence. It is observed how the $weighted\ terms$ change the distribution (bottom) of average output (top right) and find the relation between the labels and the classes.

$$c^* = arg\ max_c\ \hat{z}_{label} \quad (2)$$

In order to update the $weighted\ term(\gamma_{label})$, a cost function is defined as

$$cost_{\gamma_{label}} = (\hat{z}_{label} - \phi)^2 \quad (3)$$

where ϕ represents the target distribution for each class. In addition, we assumed a uniform distribution among all the classes. For example in 4 classes, ϕ could be represented as

$$\begin{bmatrix} 0.25 \\ 0 \\ 0 \\ 0 \end{bmatrix} ; \begin{bmatrix} 0 \\ 0.25 \\ 0 \\ 0 \end{bmatrix} ; \begin{bmatrix} 0 \\ 0 \\ 0.25 \\ 0 \end{bmatrix} \ or \ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.25 \end{bmatrix}$$

Each label of the input sequence is assigned to one of this target distribution vector based on current state of $\hat{z}_{label}$ (Eq. 2). The $weighted\ terms$ are updated using gradient descent and it is defined by

$$\gamma_{label} = \gamma_{label} - \alpha * \nabla cost_{\gamma_{label}} \quad (4)$$

where α is the learning rate of the gradient descent and $\nabla cost_{\gamma_{label}}$ is the derivatives of the cost function with respect to γ_{label}.

For labeling the classes, we follow a similar mechanism of the standard LSTM. The output of the network is decoded for finding the classes. Afterwards, the classes are labeled based on the following equation

$$label^* = arg\ max_{label}\ \hat{z}_{0,c^*}, \ldots, \hat{z}_{label,c^*} \quad (5)$$

where c^* is the decoded class given the output of the network.

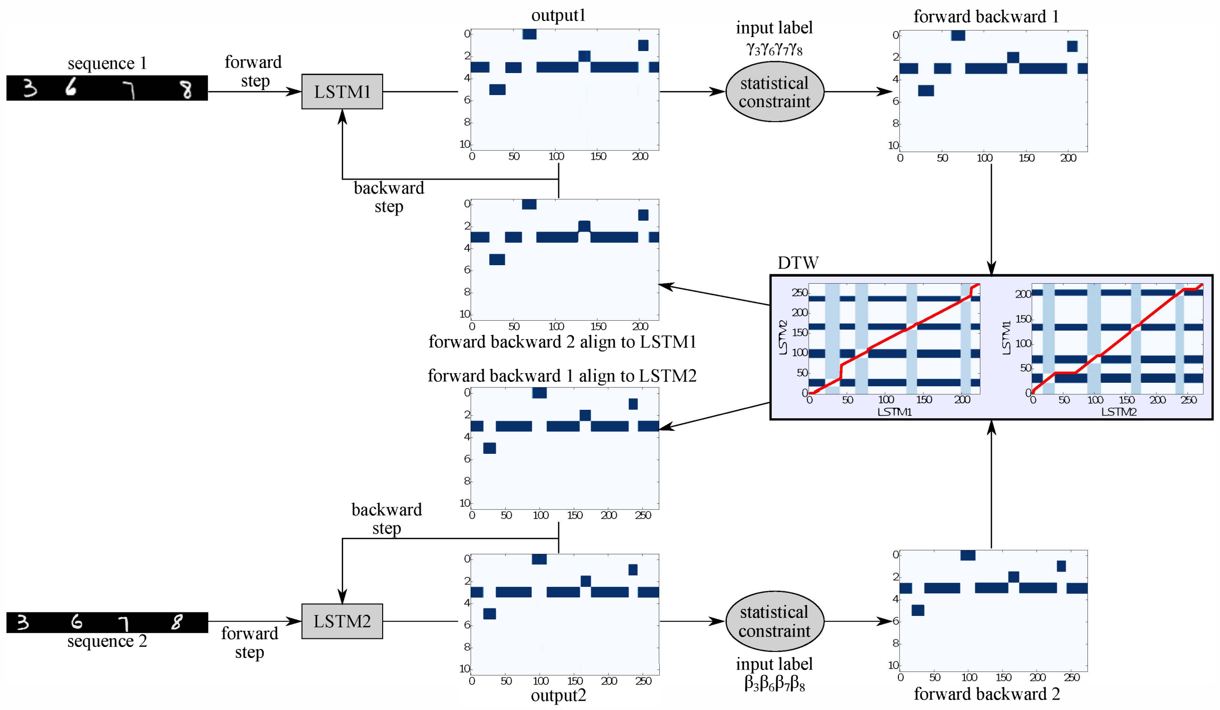

Fig. 2. Overview of the Class-less Classifier. First, both inputs are forwarded to each network. Second, the classes are labeled based on the output of the network and the statistical constraints. Third, the forward-backward algorithm is applied to both outputs. Fourth, both networks are aligned by Dynamic Time Warping (DTW). Fifth, the deltas are calculated between the output of one of the network and the aligned forward-backward of the other network. Sixth, the deltas are backpropated to each network and the weights are updated.

C. *Dynamic Time Warping (DTW)*

The goal of this step is to converge both network based on transferring information from one network to the other network. In this case, the number of time steps that each LSTM requires are different. Thus, both outputs of LSTM must be aligned between each other. *Dynamic Time Warping (DTW)* [15] is a well-known algorithm for alignment two sequences. DTW requires a distance function; which in our case, it is defined by the distance between each timestep of the forward-backward algorithm from both networks. Equation 6 shows the constrains of the DTW path.

$$DTW[i,j] = dist[i,j] + min \begin{cases} DTW[i-1,j-1] \\ DTW[i-1,j] \\ DTW[i,j-1] \end{cases} \quad (6)$$

where $dist[i,j]$ is the distance between the time step i of LSTM1 and the time step j of LSTM2.

D. *Overview of the Class-less Classifier*

The goal of our model is to converge two different instances of the same sequences using unlabeled classes. In other words, both networks need to find a relation between the labels and the class representations. In this work, we introduce a novel model called *class-less classifier*. We are referring *class-less* to the condition that our model does not use a fixed set of labeled classes for training. In contrast, our model slowly converges to a set of common class representations for each LSTM.

The training rule of our model is defined by the following steps. First, each input sequence is forwarded to each LSTM. Second, the $weighted\ terms$ assign labels to each class given current state of the output of the network and $weighted\ terms$. Third, the $forward\_backward$ algorithm is applied to both outputs of LSTM. Fourth, the outputs of the $forward\_backward$ are aligned using DTW. Fifth, the $deltas$ are calculated between the output of one networks against the aligned output of the $forward\_backward$ algorithm from the other network, and vice versa. Sixth, the $deltas$ are backpropaged to each LSTM. A summary of the algorithm in pseudocode is in Algorithm 2.

III. Results and Discussion

A. *Dataset*

In this paper, a set of parallel text lines of digits were generating based on MNIST dataset[16], which is a standard dataset for handwritten digit recognition. Two different instances of the same sequence were generated for creating the dataset with the following procedure. Each selected digit from MNIST was attached a random black background (between 3 and 10 columns), which was located before and after the digit. Afterwards, everything was horizontally stacked. Figure 3 shows several examples of the generated dataset. MNIST has already defined a training set with 60,000 examples and a testing set with 10,000 examples. We wanted to keep the same division; and, used only the example of the training set for creating our training dataset. We followed the same procedure for the testing dataset. The size of training set and the testing set are 50,000 sequences and 15,000 sequences, respectively.

Algorithm 2 Pseudocode of the training rule for the class-less classifier

```
$z_1 \leftarrow lstm_1.forward\_step(input_1)$
$z_2 \leftarrow lstm_2.forward\_step(input_2)$

$representation_1 \leftarrow find\_classes(labels, z_1, \gamma)$
$representation_2 \leftarrow find\_classes(labels, z_2, \beta)$

$fb_1 \leftarrow forward\_backward(representation_1, z_1)$
$fb_2 \leftarrow forward\_backward(representation_2, z_2)$

$path_{1\ to\ 2} \leftarrow DTW(fb_1, fb_2)$
$path_{2\ to\ 1} \leftarrow DTW(fb_2, fb_1)$

$delta_1 \leftarrow z_2[path_{2\ to\ 1}] - z_1$
$delta_2 \leftarrow z_1[path_{1\ to\ 2}] - z_2$

$lstm_1.backward\_step(delta_1)$
$lstm_2.backward\_step(delta_2)$

$update(\gamma, \beta)$
```

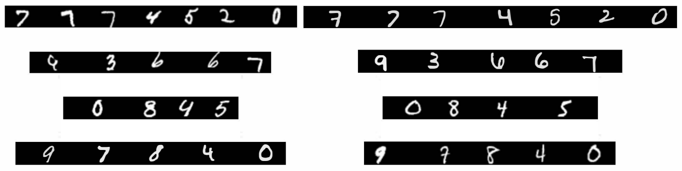

Fig. 3. Several examples of the dataset. It is observed that each row represents the same semantic sequence with different instances.

B. Experiments

Ten times were repeated the following procedure. 10,000 pairs of sequences were randomly selected from the training dataset; and, 3,000 pairs of sequences were also randomly selected from the testing dataset. The *Label Error Rate* (LER) is reported in this work, and it is defined by

$$LER = \frac{1}{|Z|} \sum_{(x,y)\in Z} \frac{ED(x,y)}{|y|} \qquad (7)$$

where ED is the edit distance between the classification of the network x and the ground-truth y and Z is the size of the dataset.

Our model is compared against LSTM with CTC layer. The following parameters were selected for both networks. The hidden size of each LSTM is 20, the learning rate is 1e-5 and the momentum is 0.9. The learning rate for updating the $weighted\ terms$ for both networks is 0.01. The $weighted\ terms$ were were initialized with 1.0. A python version of LSTM provided in OCRopus[1] was used.

C. Analysis

We found that our model reaches similar results than standard LSTM (see Table I). In this scenario, DTW in combination with both networks is able to learn classes similar to the standard LSTM. The effect of this combination is a modified version of the $forward\_backward$ algorithm. This modification made the training more flexible because the constraints of the path (see Equation 6) add more information to the probability propagation. Besides to the alignment, the statistical constraints guide each network for converging the underlying structure between the labels and their class representations. We noticed that the *blank class* is the first class on which both networks converge. This behaviour is similar to the standard LSTM where the first learned class is also the *blank class*. After both networks converge, the rest of the labels are learnt and they slowly converge to the same structure. Figure 4 shows an example of an structure between the classes and the labels. We want to point out that both networks converge to the same relation. For example, the *label '7'* is represented by the *class '2'*. This is observed at the first, third and fifth row. Furthermore, LSTM is able to learn sequences with information that is provided from a different source, in this case, the output of another network in combination with DTW.

[1]OCRopus - Open Source Document Analysis and OCR System [Online]:https://github.com/tmbdev/ocropy

TABLE I. LABEL ERROR RATE (%) BETWEEN LSTM AND THE CLASS-LESS CLASSIFIER

METHOD		MNIST
STANDARD LSTM	SEQUENCE1	3.47 ± 0.99
	SEQUENCE2	3.52 ± 0.80
OUR MODEL	SEQUENCE1	2.29 ± 0.27
	SEQUENCE2	2.21 ± 0.17

IV. CONCLUSIONS AND FUTURE WORK

In this work, the association problem between text lines was studied. We introduced a novel model, called *"class-less classifier"*, which modifies the standard training rule of LSTM. In addition, our model includes an extra constraint that uses unlabeled classes for the training. The results of the proposed model reaches similar results than LSTM. We will explore in more detail the relation between printed texts and handwritten texts. Moreover, we are interested to evaluate different associations of the output of both networks, such as, max-operator and average-operator. Finally, we will examine our model in more complex scenarios, such as, handling missing class occurrences between the sequences. For example, one sequence is *"1 5 7 8"* and the other sequence is *"1 5 8"*.

ACKNOWLEDGMENT

This work was partially funded by the BMBF (German Federal Ministry of Education and Research), project Kallimachos (01UG1415C).

REFERENCES

[1] J. Huang, Z. Liu, Y. Wang, Y. Chen, and E. K. Wong, "Integration of multimodal features for video scene classification based on hmm," in *Multimedia Signal Processing, 1999 IEEE 3rd Workshop on*. IEEE, 1999, pp. 53–58.

[2] R. Leonardi, P. Migliorati, and M. Prandini, "Semantic indexing of soccer audio-visual sequences: a multimodal approach based on controlled markov chains," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 5, pp. 634–643, 2004.

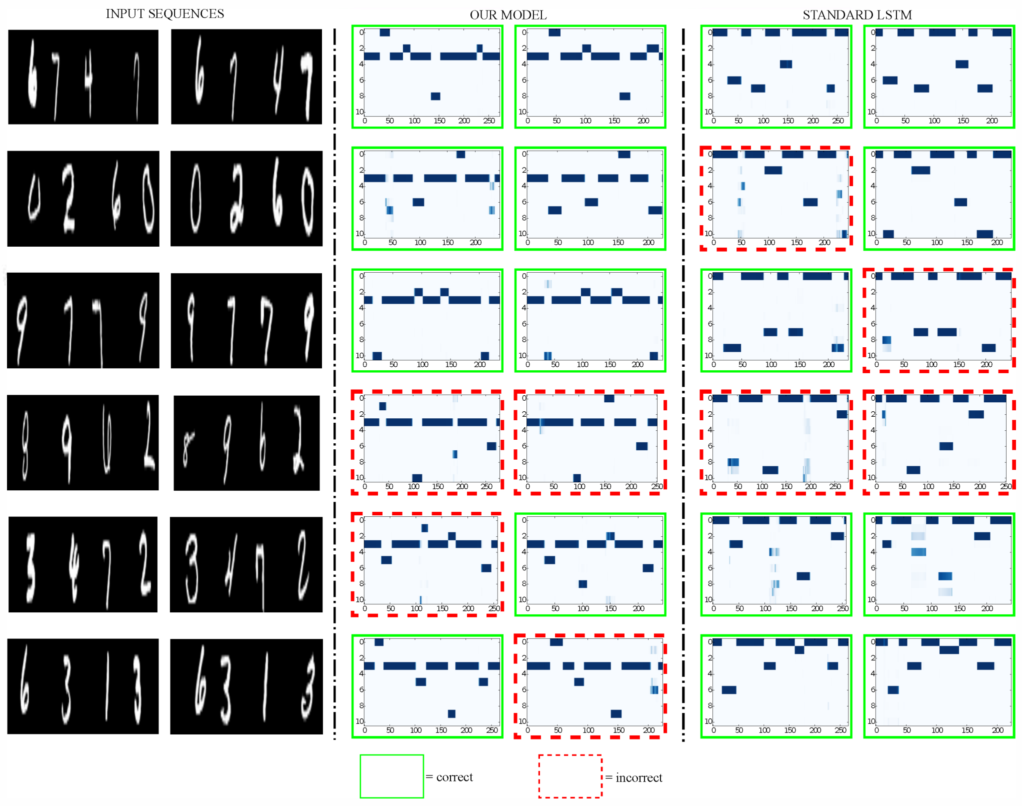

Fig. 4. Several examples of our model and standard LSTM. In this example, the relation between the classes and the labels is defined by $\{(class, label) : (0,6),(1,8),(2,7),(3,b),(4,5),(5,3),(6,2),(7,0),(8,4),(9,1),(10,9)\}$. It is observed that some cases our model classifies correctly the input sequence, even if the standard LSTM classifies incorrectly.

[3] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. Lanckriet, R. Levy, and N. Vasconcelos, "A New Approach to Cross-Modal Multimedia Retrieval," in *ACM International Conference on Multimedia*, 2010, pp. 251–260.

[4] M. T. Balaban and S. R. Waxman, "Do words facilitate object categorization in 9-month-old infants?" *Journal of experimental child psychology*, vol. 64, no. 1, pp. 3–26, Jan. 1997.

[5] L. Gershkoff-Stowe and L. B. Smith, "Shape and the first hundred nouns." *Child development*, vol. 75, no. 4, pp. 1098–114, 2004.

[6] A. F. Pereira and L. B. Smith, "Developmental changes in visual object recognition between 18 and 24 months of age." *Developmental science*, vol. 12, no. 1, pp. 67–80, Jan. 2009.

[7] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2009, pp. 545–552.

[8] M. Liwicki, A. Graves, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proc. 9th Int. Conf. on Document Analysis and Recognition*, vol. 1, 2007, pp. 367–371.

[9] T. Breuel, A. Ul-Hasan, M. Al-Azawi, and F. Shafait, "High-performance ocr for printed english and fraktur using lstm networks," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, Aug 2013, pp. 683–687.

[10] L. Mioulet, G. Bideault, C. Chatelain, T. Paquet, and S. Brunessaux, "Exploring multiple feature combination strategies with a recurrent neural network architecture for off-line handwriting recognition," *databases*, vol. 12, p. 15.

[11] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735—-1780, 1997.

[12] S. Hochreiter, "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 06, no. 02, pp. 107–116, Apr. 1998.

[13] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification," in *Proceedings of the 23rd international conference on Machine learning - ICML '06*. New York, New York, USA: ACM Press, 2006, pp. 369–376.

[14] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[15] D. J. Berndt and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series," pp. 359–370, 1994.

[16] Y. Lecun and C. Cortes, "The MNIST database of handwritten digits."