Feature extraction with convolutional neural networks for handwritten word recognition

Théodore Bluche, Hermann Ney and Christopher Kermorvant

A2iA, Paris, LIMSI-CNRS, Orsay

Introduction

We explore different strategies for handwritten word recognition, including:

- **different segmentations** (implicit vs. explicit)
- **different features** (pixels vs. handcrafted vs. nnet)
- **different modelling** (GMM vs. nnet vs. tandem)

We show that **learnt features and neural networks perform better** than their GMM and handcrafted features counterparts, and that **explicit segmentation**, while performing worse, **yields much faster systems and helps in a combination of systems**.

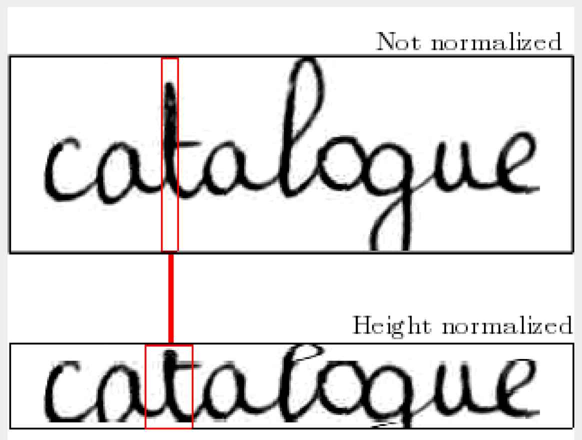
Preprocessing

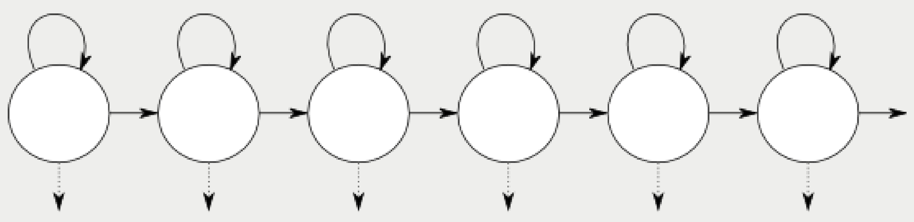
- Deslant
- Contrast enhancement
- Padding
- Height normalization

Sliding window models (*sw-*)

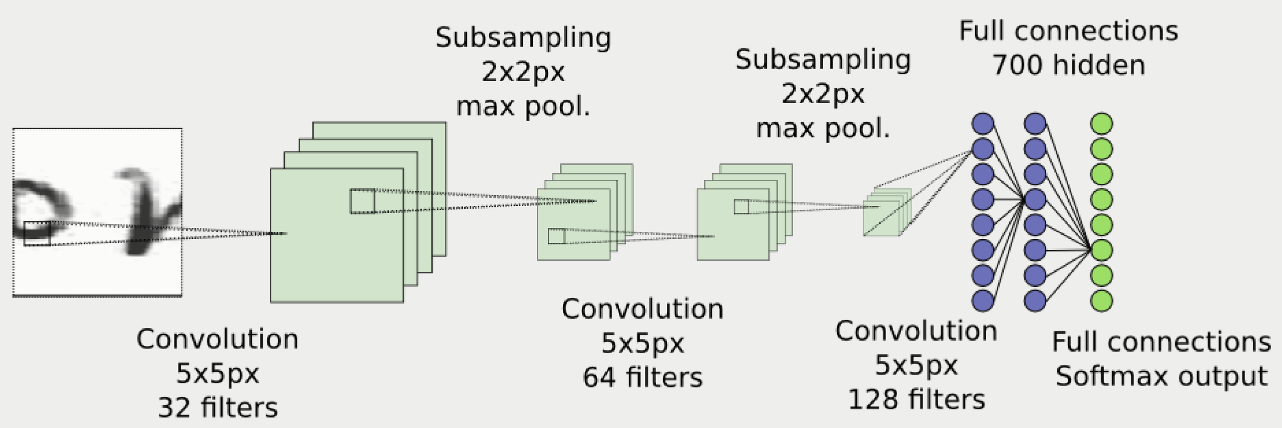
Features

- **Handcrafted** (*-feat*) : sliding win. 9px (shift 3px), 34 geometrical and statistical features + deltas (Bianne et al., 2011)
- **Pixels** (*-pix*) : sliding win. 39px (shift 3px) on height normalized image, rescaled to 32x32px, 30 dim. PCA on pixel intensities
- **Pixels for ConvNN** : same as *Pixels*, but no PCA





Convolutional Neural Networks



- *input*: 32x32px image
- *output*: probability of every HMM state

Motivations

- Take into account 2D structure of the image
- Invariance to small distortions (pooling operations)
- Learnt features extraction

Combination with HMM

- **Hybrid** (*-hyb*): pseudo-likelihoods computed by the neural network replace GMM likelihoods.
- **"Tandem"** (*-tan*): pseudo-likelihoods are decorrelated with PCA (50 dimensions) and are features for a standard GMM-HMM.

System combination

- For each system, we extracted ***N***-best lists ($N = 10$),
- **Sum** = sum up the scores of all considered systems
- **Wei.Sum** = weighted sum of the scores, where the weights are heuristically chosen to be proportional to the accuracy of each system
- Then, we pick the word with highest score

Speed comparison

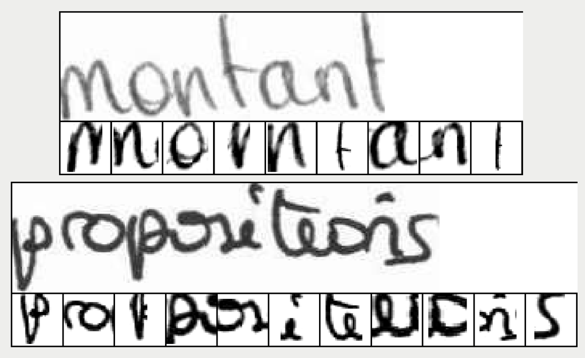
	num.observation	decoding time
Sliding Window	551,959	719 ms/word
Grapheme	52,606	46 ms/word

Table : Grapheme vs sliding window on devset

Grapheme models (*gpm-*)

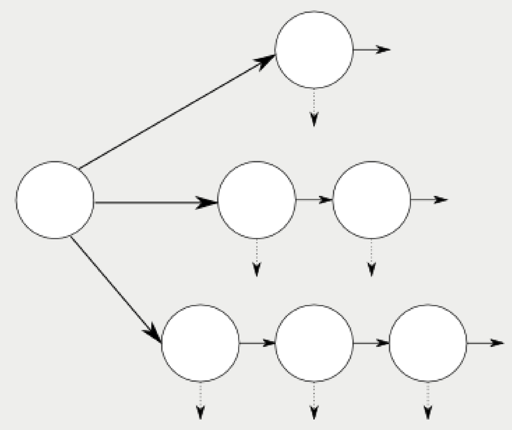
Grapheme segmentation

The grapheme segmentation is a **heuristic over-segmentation** algorithm which breaks the ink (the set of black pixels) into several parts corresponding to characters or parts of characters.



Features

- **Handcrafted** (*-feat*) : 74 geometrical and statistical features extracted from grapheme segmentation (Bianne et al., 2011)
- **Pixels** (*-pix*) : grapheme images rescaled to 32px along the largest dimension, padded to obtain 32x32px images, + 30 dim. PCA
- **Pixels for ConvNN** : same as *Pixels*, but no PCA



Results of individual systems on Rimes testset

Model	Rimes-WR2	Rimes-WR3
gpmpix	39.0%	41.8%
gpmfeat	28.0%	30.6%
gpmhyb	16.8%	18.9%
swpix	19.8%	21.4%
swfeat	14.6%	16.4%
swhyb	10.0%	11.7%
swtan	8.5%	9.9%
swtan + CD + MMI	7.9%	9.2%
7 RNN + HMM (Menasi, 2012)	-	**4.8%**
RNN (Graves, 2009)	**6.8%**	9.0%
Tandem LSTM-HMM (Doetsch, 2011)	-	9.7%

Table : Word error rate on the test set for the different systems on the ICDAR-2009 evaluation set for two different vocabulary sizes (WR2 and WR3).

Results of combined systems on Rimes devset

Models	Sum	Wei.Sum
gpmpix	41.2%	41.2%
gpmfeat	30.5%	30.5%
gpmhyb	17.7%	17.7%
swfeat	14.7%	14.7%
swhyb	10.9%	10.9%
swtan	**8.7%**	**8.7%**
swtan, swhyb	8.6%	8.5%
swhyb, gpmhyb	8.5%	8.2%
swtan, gpmhyb	7.9%	7.7%
swtan, swhyb, gpmhyb	7.3%	7.3%
swtan, swhyb, swfeat, gpmhyb	**7.1%**	7.1%
swtan, swhyb, swfeat, gpmhyb, gpmfeat	7.2%	**6.9%**

Table : Combination results on Rimes 2009 validation set

{tb,ck}@a2ia.com, ney@informatik.rwth-aachen.de