

Word Spotting in the Era of Deep Learning

Gernot A. Fink

Paris, October 17, 2019

- ▶ Introduction
- ▶ Fundamentals: Bag-of-Features
- ▶ Learning Document Image Representations
- ▶ Learning Word Spotting Models
- ▶ Deep Learning Fundamentals
- ▶ Deep Learning for Word Spotting
- ▶ Outlook
- ▶ Summary

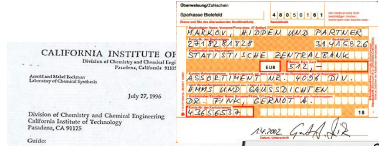
with contributions by

*Sebastian Sudholt,
Leonard Rothacker,
Eugen Rusakov,
Fabian Wolf,
Philip Oberdiek,
René Grzeszick*

Introduction: Automatic Reading Systems

State of Automatic Reading:

- ▶ One of the earliest applications studied in computer science
- ▶ OCR achieves high-quality results for machine-printed text.
- ▶ Online handwriting recognition again gaining popularity
- ▶ Offline handwriting recognition: Still an open research problem.



CALIFORNIA INSTITUTE OF TECHNOLOGY
Division of Chemistry and Chemical Engineering
Pasadena, California 91125

July 27, 1956

Division of Chemistry and Chemical Engineering
California Institute of Technology
Pasadena, CA 91125

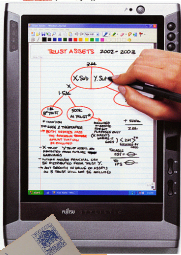
Guides:

I would like to provide for you in writing from what is expected from member of the research group. In addition to the usual work-day schedule, all of the members of the group to work evenings and weekends. You will find this in the memo here at Caltech. On occasion, I understand that personal will make demands on your time which will require you to be away from responsibilities to the laboratory. However, it is not acceptable to me, becomes a habit.

I have noticed that you have failed to come in to lab on several weeks and more recently have failed to show up in the evenings. Moreover, as said such times off, you recently requested some vacation. I have no problem vacation time that is well earned, but I do have a problem with continuous vacation time that interferes with the project. I find this very unwise disruptive to your science.

I expect you to correct your work-ethic immediately.

I have around the project that I can do.



General Methodology:

Transcription models are trained on massive amounts of **annotated** data.

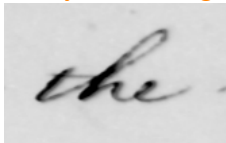


Introduction: Why Word Spotting?

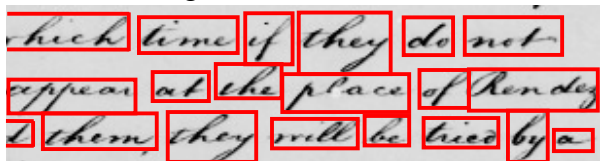
What if automatic transcription of handwriting is no longer feasible?

Alternative: Retrieval of individual words rather than transcription
("query-by-example")

Query word image



Document image

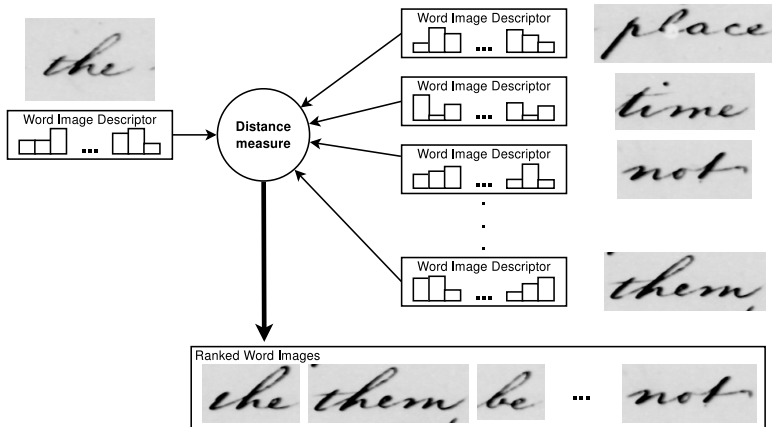


Images from *The George Washington Papers at the Library of Congress, 1741-1799*

Introduction: Basic Methodology

Query-by-example word spotting

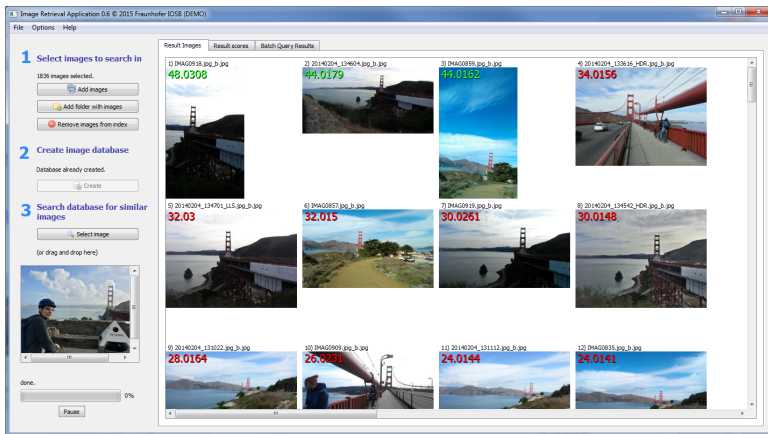
Does that seem familiar?



Based on [Rath & Manmatha, IJRR'07]

Introduction: Basic Methodology II

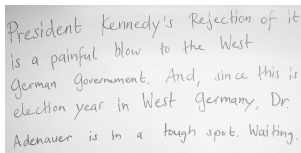
QbE word spotting \approx special case of content-based image retrieval



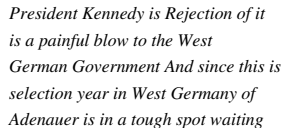
(Source: Fraunhofer IOSB CBIR Demo)

Tasks in Handwriting Recognition

Document Transcription (= "classical" recognition)

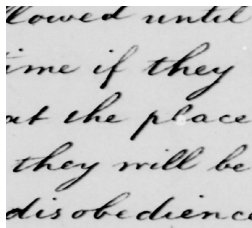


President Kennedy's Rejection of it
is a painful blow to the West
German Government. And, since this is
election year in West Germany, Dr.
Adenauer is in a tough spot. Waiting.



President Kennedy is Rejection of it
is a painful blow to the West
German Government And since this is
selection year in West Germany of
Adenauer is in a tough spot waiting

Document Retrieval (aka "Word Spotting")



lowed until
ime if they
at the place
they will be
disobedienc



the



Word Spotting: Fundamentals

Core Methodology:

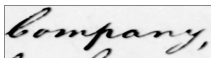
- ▶ Specialized image retrieval
- ▶ Important ingredient: Image matching procedure
- ▶ Frequently required: Pre-segmentation (words / lines)

Taxonomy:

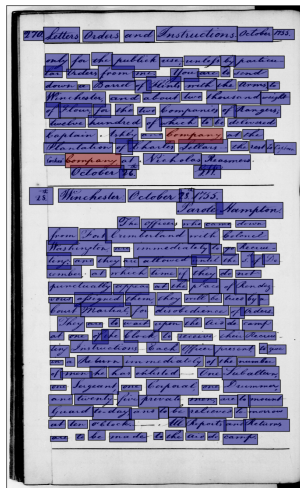
- ▶ *Segmentation-based*
- ▶ *Segmentation-free*, i.e., segmentation problem covered during retrieval
- ▶ *Query-by-Example*, i.e., word image directly used as query
- ▶ *Query-by-String*, i.e., query model derived from textual query (“string”)

Word Spotting Tasks

Query by Example



Query by String



Word-Spotting: Classical Milestones

Manmatha *et al.* 1996: First influential work

(Binarization, Alignment, XOR distance)

Rath & Manmatha 2003: DTW matching

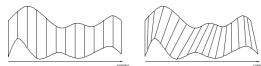
(Normalization, profile features)

Rusiñol *et al.* 2011: First influential work using BoF, first with Spatial Pyramid

(SIFT, BoF, Spatial Pyramid, LSI, segmentation-free decoding)

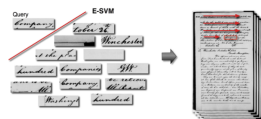
Almazan *et al.* 2014: HOG features

(“Exemplar SVM”, query expansion)



(a) naive alignment after resampling. (b) alignment with DTW.

company	company	company	company
English	hat English sail	an English men	English Ho
است	ی ان English	distinguished at	to England I
است	الله است	ی است	یت است
	است	ت است	ت است



Overview

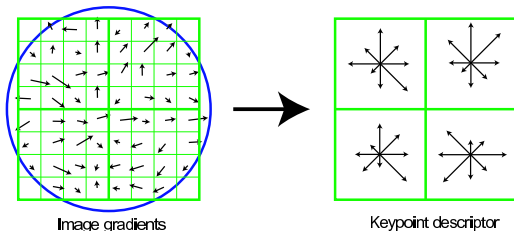
- ▶ Introduction
- ▶ **Fundamentals: Bag-of-Features**
- ▶ Learning Document Image Representations
- ▶ Learning Word Spotting Models
- ▶ Deep Learning Fundamentals
- ▶ Deep Learning for Word Spotting
- ▶ Outlook
- ▶ Summary

Local Image Descriptors

Fundamentals:

- ▶ Local gradient statistics
(i.e. implicit description of object contours)
- ▶ Grouping of multiple such local statistics in a certain neighborhood and *normalization*
(coarse description of structural properties)

Example: SIFT descriptor [Lowe, 2004]



(Source: [Lowe, IJCV 2004])

Note: Ex. several similar descriptors, e.g., HOG, SURF

Statistical Image Modeling

Representation: Bag-of-Features Models (BoF)

- ▶ Originally proposed as Bag-of-Words models for representing texts

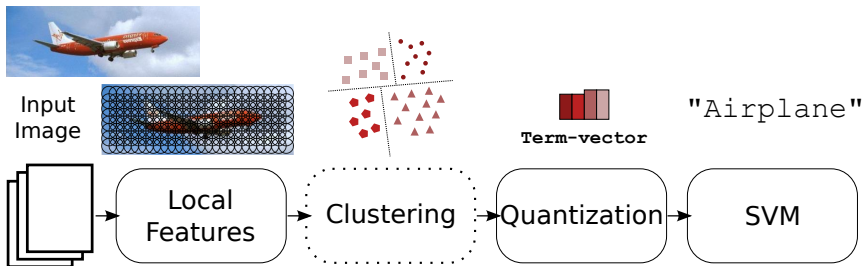
BoF-Approach: (cf. e.g. [O'Hara & Draper 2011])

0. Extract local image features (usually in a grid)
1. Compute *visual vocabulary* by quantizing local image features
2. For given image, compute histogram of quantized features (i.e. orderless “bag” of features)
3. Any classification / matching technique can be applied to BoF representations

S. O'Hara & B. A. Draper: [Introduction to the Bag of Features Paradigm for Image Classification and Retrieval](#), Computing Research Repository, arXiv:1101.3354v1, 2011.

Statistical Image Modeling II

BoF-Approach: Processing Pipeline

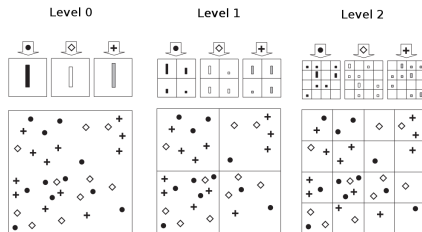


Main drawback: Spatial information is lost!

Statistical Image Modeling III

Goal: Compensate loss of spatial information

Method: Spatial-pyramid models



Lazebnik, S., Schmid, C., Ponce, J.: *Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories*, IEEE Conference on Computer Vision and Pattern Recognition, 2006.

Disadvantage: Considerably increased model complexity

Overview

- ▶ Introduction
- ▶ Fundamentals: Bag-of-Features
- ▶ **Learning Document Image Representations**
- ▶ Learning Word Spotting Models
- ▶ Deep Learning Fundamentals
- ▶ Deep Learning for Word Spotting
- ▶ Outlook
- ▶ Summary

Bag-of-Features Models for Word Spotting

Basic Methodology:

Image Features:

Gradient-based descriptors, e.g., SIFT, HOG

Feature Extraction:

Dense grid, i.e., no keypoint detection involved

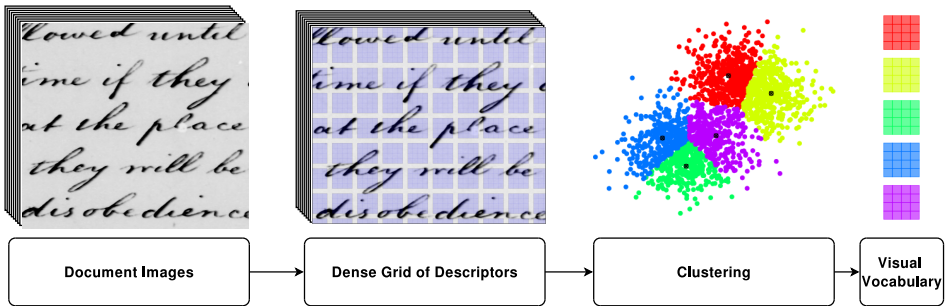
Visual Vocabulary:

- ▶ Quantization of descriptors (as “usual”)
- ▶ **Special:** Large vocabularies (i.e. 2K to 4K) in order to capture appearance of “individuals” precisely

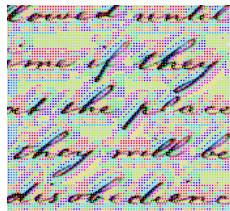
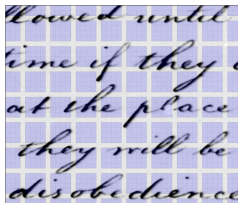
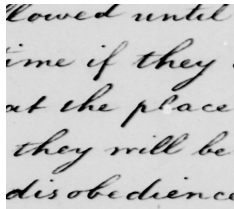
Query Model:

- ▶ Segmentation- or patch-based processing
- ▶ 1D spatial pyramid for improved spatial modeling

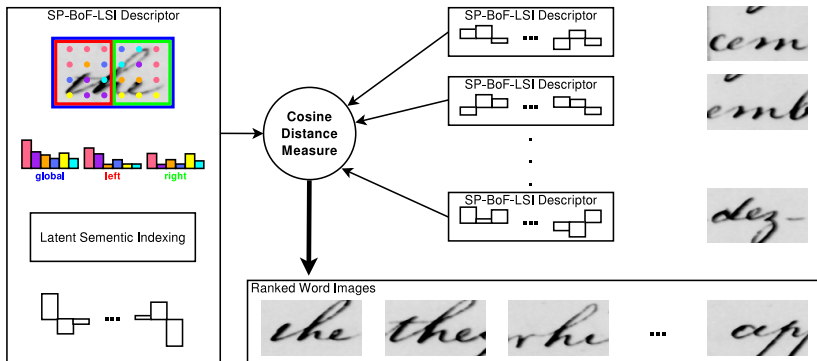
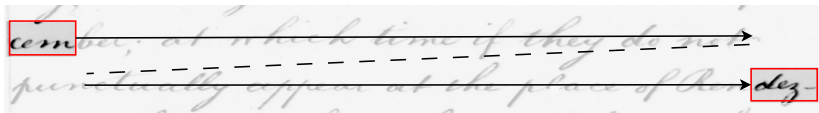
Bag-of-Features Models: Visual Vocabulary



Bag-of-Features Models: Term Vector

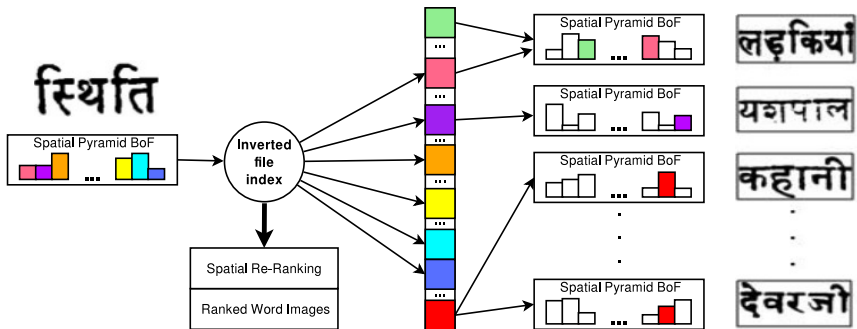


Segmentation-free Word Spotting with Bag-of-Features



Rusiñol, M., and Aldavert, D., and Toledo, R., and Lladós, J.: [Browsing heterogeneous document collections by a segmentation-free word spotting method](#), Int. Conf. on Document Analysis and Recognition, Beijing, pp. 63-67, 2011.

Segmentation-based Word Spotting with Bag-of-Features



Shekhar, R., and Jawahar, C.: [Word image retrieval using bag of visual words](#), Int. Workshop on Document Analysis Systems, pp. 297-301, 2012.

Overview

- ▶ Introduction
- ▶ Fundamentals: Bag-of-Features
- ▶ Learning Document Image Representations
- ▶ **Learning Word Spotting Models**
- ▶ Deep Learning Fundamentals
- ▶ Deep Learning for Word Spotting
- ▶ Outlook
- ▶ Summary

Learning BoF-Based Word Spotting Models

What we have so far:

- ▶ Expert-designed image descriptors
- ▶ Automatically learned document image features
(visual vocabulary → histograms of quantized descriptors)
- ▶ Matching: Nearest neighbor

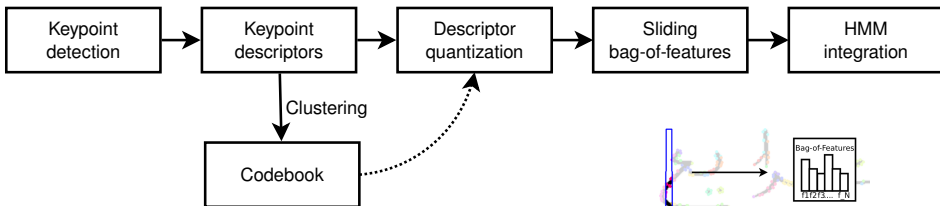
What we want in addition:

- ▶ More powerful image matching
- ▶ Categorization capabilities
(i.e. links between image appearance and textual representation)

What we need: Learned classification models

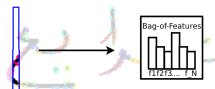
Bag-of-Features HMMs

- ▶ Extension of HMMs towards learned feature representation
- ▶ Extension of BoF models towards fine-grained script representation



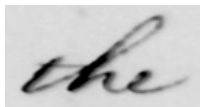
$$b_j(\mathbf{f}) = \sum_{k=1}^{|\mathcal{V}|} c_{jk} f_k$$

with \mathbf{f} : term vector
 \mathcal{V} : vis. voc.

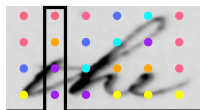


Rothacker, L., Vajda, S., Fink, G. A.: *Bag-of-Features Representations for Offline Handwriting Recognition Applied to Arabic Script*, In Proc. ICFHR, Bari, 2012.

Bag-of-Features HMMs for QbE Word Spotting



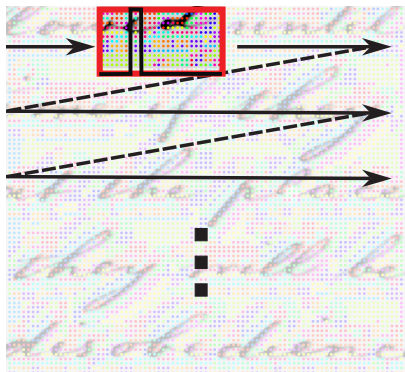
Query Word



Bag-of-Features Sequence
(sliding window)



Bag-of-Features
Hidden Markov Model



Approach can be sped-up by intelligent patch pre-filtering!

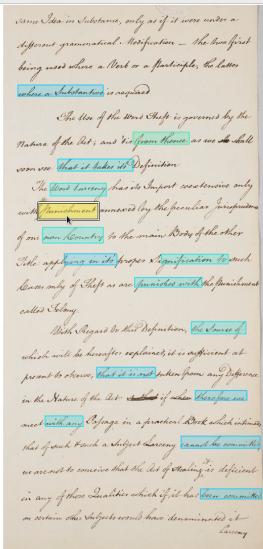
Rothacker, L., Rusinol, M., Fink, G. A.: *Bag-of-Features HMMs for Segmentation-Free Word Spotting in Handwritten Documents*, In Proc. Int. Conf. on Document Analysis and Recognition, Washington DC, USA, 2013.

Rothacker, L., Rusinol, M., Lladós, J., Fink, G. A.: *A Two-Stage Approach to Segmentation-Free Query-by-Example Word Spotting*, manuscript cultures, 1(7), pages 47-57, 2014.

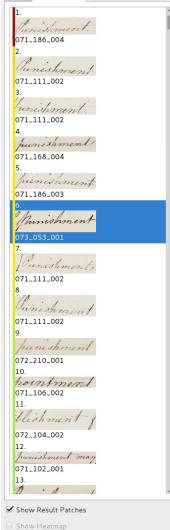
QbE Word Spotting: Example ("Bentham")

Wordspotting - Bentham - ICFHR14 KWS Competition

Projects Settings Help



Query Query Results



(Video created by Matthias Kasperidus)

Bag-of-Features HMMs for QbS Word Spotting

Training set

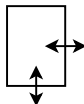
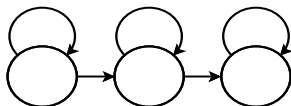


c a p t a i n

⋮



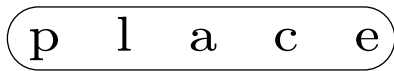
t w e l v e



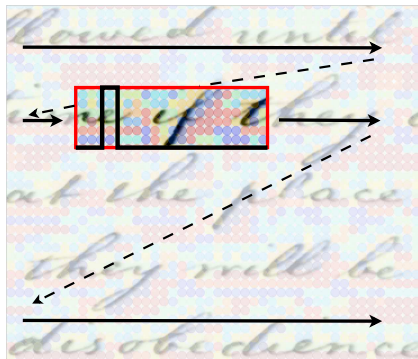
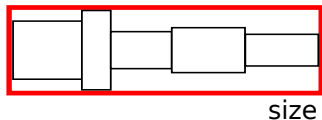
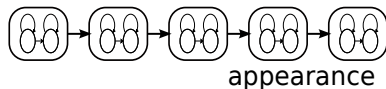
- ▶ Appearance model: Bag-of-Features HMMs (per character)
- ▶ Spatial size model: Width & height estimates

Rothacker, L., Fink, G. A.: *Segmentation-Free Query-by-String Word Spotting with Bag-of-Features HMMs*, Int. Conf. on Document Analysis and Recognition, Nancy, France, 2015.

Bag-of-Features HMMs for QbS Word Spotting II

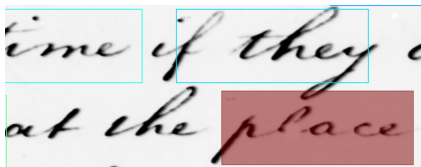
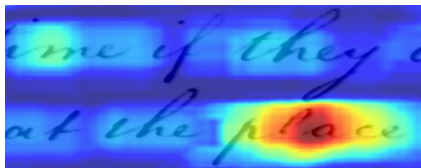
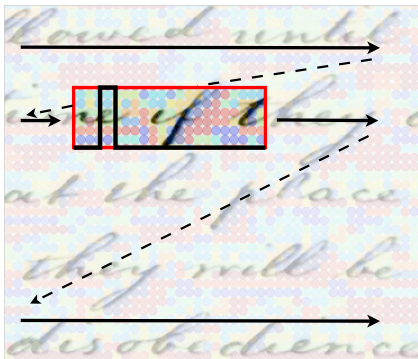


 query



Rothacker, L., Fink, G. A.: *Segmentation-Free Query-by-String Word Spotting with Bag-of-Features HMMs*, Int. Conf. on Document Analysis and Recognition, Nancy, France, 2015.

Bag-of-Features HMMs for QbS Word Spotting II



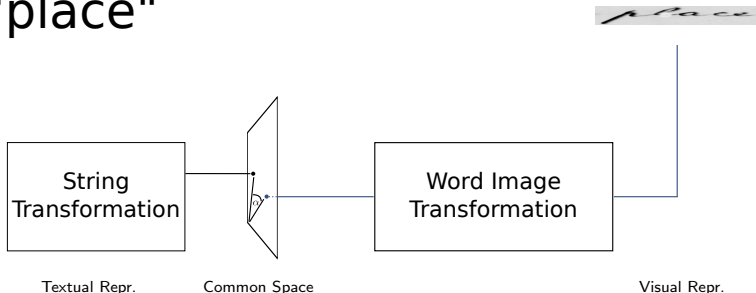
Rothacker, L., Fink, G. A.: *Segmentation-Free Query-by-String Word Spotting with Bag-of-Features HMMs*, Int. Conf. on Document Analysis and Recognition, Nancy, France, 2015.

Subspace Representations for Word Spotting

Idea: Project both textual and visual representation into a *common space*

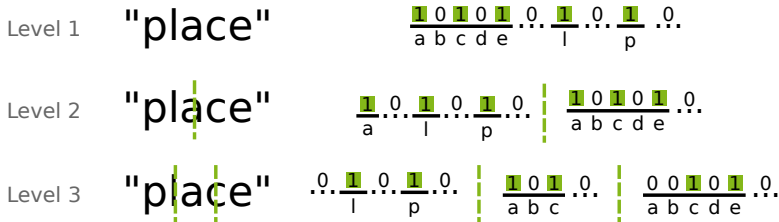
Benefits: QbE and QbS are now a simple nearest neighbor search

"place"



J. Almazán, A. Gordo, A. Fornés and E. Valveny: [Word Spotting and Recognition with Embedded Attributes](#), IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 36, no. 12, pp. 2552-2566, 2014.

Pyramidal Histogram of Characters (PHOC)

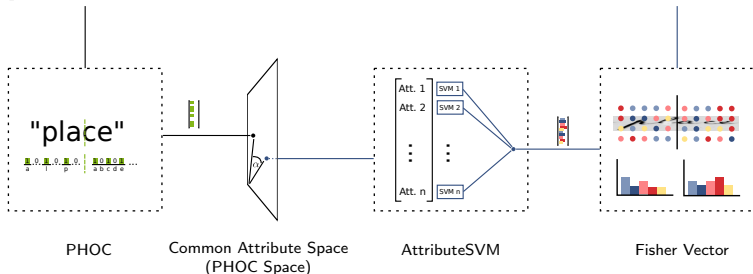


- ▶ Concatenate histograms for all levels to form PHOC
- ▶ Levels used by Almazán *et al.*: 2,3,4 and 5
- ▶ 26 Characters + 10 Digits
- ▶ $\text{PHOC} \in \{0, 1\}^{604}$

J. Almazán, A. Gordo, A. Fornés and E. Valveny: [Word Spotting and Recognition with Embedded Attributes](#), IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 36, no. 12, pp. 2552-2566, 2014.

Learning the PHOC representation

"place"



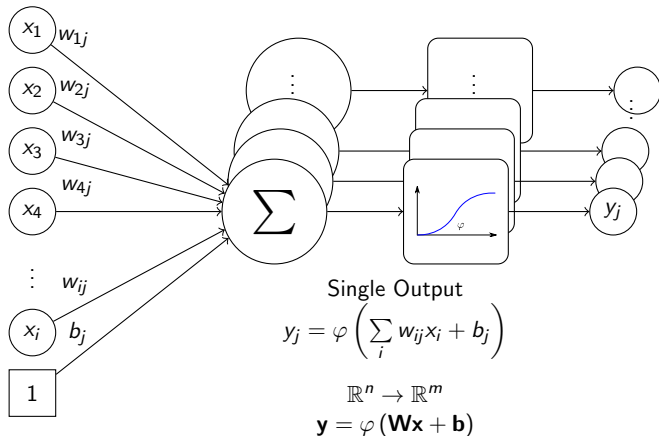
- ▶ AttributeSVM: ensemble of SVMs
- ▶ each SVM predicts one attribute within the PHOC

J. Almazán, A. Gordo, A. Fornés and E. Valveny: [Word Spotting and Recognition with Embedded Attributes](#), IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 36, no. 12, pp. 2552-2566, 2014.

Overview

- ▶ Introduction
- ▶ Fundamentals: Bag-of-Features
- ▶ Learning Document Image Representations
- ▶ Learning Word Spotting Models
- ▶ **Deep Learning Fundamentals**
- ▶ Deep Learning for Word Spotting
- ▶ Outlook
- ▶ Summary

The Perceptron

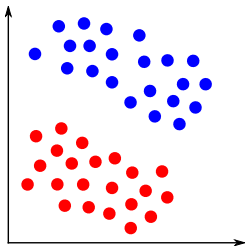


F. Rosenblatt: [The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain](#), Psychological Review, 65(6), 1958.

Capabilities of the Perceptron

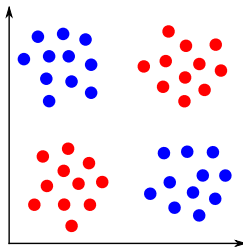
What a Perceptron can do:

Classify two linearly separable classes



What a Perceptron can't do:

Classify two non-linearly separable classes (e.g. XOR-Problem)

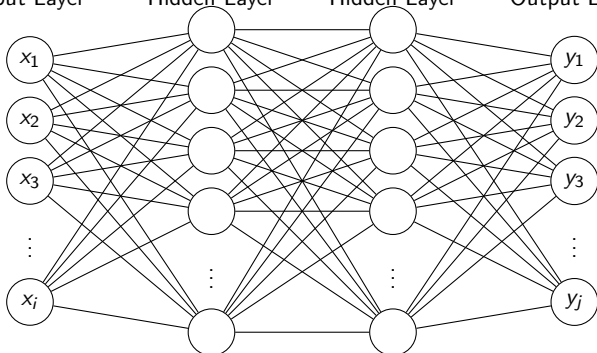


Solution: Stack layers of Perceptrons

⇒ Multi Layer Perceptron

Multi Layer Perceptron (MLP)

Input Layer Hidden Layer Hidden Layer Output Layer



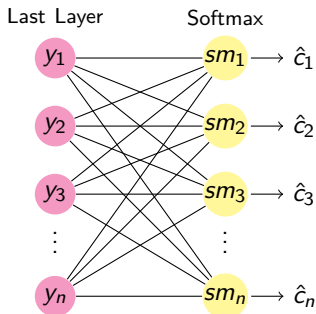
$$\mathbf{y} = \mathbf{f}^L \left(\mathbf{f}^{L-1} (\dots \mathbf{f}^2 (\mathbf{f}^1 (\mathbf{x})) \dots) \right)$$

here: $\mathbf{y} = \mathbf{W}_{\text{out}} \cdot \varphi (\mathbf{W}_{h2} \cdot \varphi (\mathbf{W}_{h1} \mathbf{x} + b_{h1}) + b_{h2}) + b_{\text{out}}$

Note: Activation function is typically left out for last layer

Classifying with MLPs

- ▶ For classification, the output of the MLP is *usually* forwarded through a Softmax Function: $sm_i(\mathbf{y}) = \frac{e^{y_i}}{\sum_j e^{y_j}}$
- ▶ Softmax can be seen as an additional layer in the MLP
- ▶ sm_i is pseudo-probability for class c_i
- ▶ Predicted class: $\hat{c} = \max_i sm_i$

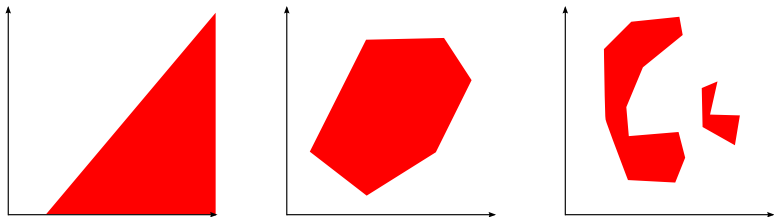


What an MLP Can Do!

... approximate any function (even with only 2 layers!)

[Hornik *et al.* 1989]

Interpretation with 3 layers (2 hidden, 1 output):



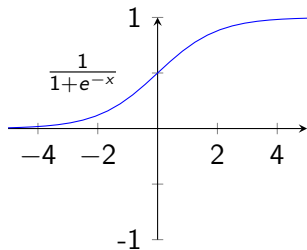
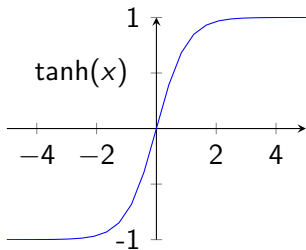
1. Layer: Halfspaces
2. Layer: Convex polyhedron
3. Layer: Multiple non-convex, non-connected polyhedra

A Word on Activation Functions

- ▶ Activation functions are crucial for MLPs
- ▶ Without non-linearities, an MLP implements a linear transform:

$$\mathbf{y} = \mathbf{W}_L \mathbf{W}_{L-1} \dots \mathbf{W}_2 \mathbf{W}_1 \mathbf{x} = \mathbf{W}' \mathbf{x}$$

Classic Activation Functions: sigmoidal shape (“threshold-like”)



Note: With $\phi_s(x) = \frac{1}{1+e^{(-x)}}$ we have $\frac{\partial \phi_s(x)}{\partial x} = \phi_s(x) (1 - \phi_s(x))$.

Training an MLP

How to determine weights such that desired function is performed?

Basic Idea: Compare (computed) output of MLP

$$\hat{\mathbf{y}} = \mathbf{f}^L \left(\mathbf{f}^{L-1} (\dots \mathbf{f}^2 (\mathbf{f}^1 (\mathbf{x})) \dots) \right)$$

to *desired* (ideal) output \mathbf{y} and update weights such that $\hat{\mathbf{y}}$ and \mathbf{y} become more similar.

Comparison requires *loss function* that evaluates similarity of $\hat{\mathbf{y}}$ and \mathbf{y} :

- Mean Square Error (MSE):

$$\epsilon_{\text{MSE}} = \frac{1}{2} \cdot \sum_i (y_i - \hat{y}_i)^2$$

- Cross-Entropy (in comb. w. Softmax):

$$\epsilon_{\text{CE}} = - \sum_i y_i \log \hat{y}_i$$

Training an MLP II

How to update weights in order to reduce loss?

Compute **gradient** of loss wrt. the weights:

$$\frac{\partial \epsilon}{\partial w_{ij}^l}$$

Update weights in the negative direction of the gradient:

$$w_{ij}^l \leftarrow w_{ij}^l - \beta \cdot \frac{\partial \epsilon}{\partial w_{ij}^l}$$

⇒ Training the network $\hat{=}$ Gradient Descent

Note: *Learning rate* β controls step size!

Training an MLP: Error Backpropagation

How to compute gradient of loss wrt. network weights?

Define computation of outputs of neurons per layer¹:

$$f_j^l = \phi(g_j^l) = \phi\left(\sum_i w_{ij}^l f_i^{l-1}\right)$$

Rewrite gradient applying chain rule:

$$\frac{\partial \epsilon}{\partial w_{ij}^l} = \frac{\partial \epsilon}{\partial f_j^l} \cdot \frac{\partial f_j^l}{\partial g_j^l} \cdot \frac{\partial g_j^l}{\partial w_{ij}^l}$$

⇒ Evaluation *straight forward* for final layer (using ϵ_{MSE} and ϕ_s):

$$\frac{\partial \epsilon}{\partial w_{ij}^L} = -(y_j - f_j^L) \cdot f_j^L (1 - f_j^L) \cdot f_i^{L-1}$$

¹ignoring bias for simplicity

Training an MLP: Error Backpropagation II

How to evaluate gradient for inner weights?

Hidden layers: expand gradient using chain rule

$$\frac{\partial \epsilon}{\partial w_{ij}^l} = \frac{\partial \epsilon}{\partial \mathbf{f}^L} \cdot \frac{\partial \mathbf{f}^L}{\partial \mathbf{f}^{L-1}} \cdot \dots \cdot \frac{\partial \mathbf{f}^{l+1}}{\partial f_j^l} \cdot \frac{\partial f_j^l}{\partial g_j^l} \cdot \frac{\partial g_j^l}{\partial w_{ij}^l} = \delta_j^l \cdot \frac{\partial g_j^l}{\partial w_{ij}^l}$$

Exploit *generalized chain rule*: For the gradient we obtain:

$$\frac{\partial \epsilon}{\partial f_j^l} = \sum_{k=1} \underbrace{\frac{\partial \epsilon}{\partial f_k^{l+1}} \cdot \frac{\partial f_k^{l+1}}{\partial g_k^{l+1}}}_{\delta_k^{l+1}} \cdot \underbrace{\frac{\partial g_k^{l+1}}{\partial f_j^l}}_{w_{jk}^{l+1}} = \sum_{k=1} \delta_k^{l+1} w_{jk}^{l+1}$$

$$\Rightarrow \delta_j^l = \frac{\partial \epsilon}{\partial f_j^l} \frac{\partial f_j^l}{\partial g_j^l} = \left\{ \sum_{k=1} w_{jk}^{l+1} \delta_k^{l+1} \right\} \cdot \frac{\partial f_j^l}{\partial g_j^l}$$

Local error δ_j^l can be computed *backward* through the network

⇒ Error Backpropagation

Minimum-Error and Maximum-Likelihood Training

... can be shown to be equivalent assuming certain distributions of the desired outputs of the network!
 [Goodfellow et al. 2016, Sec. 5.5]

ML-Estimation of (network) parameters θ :

$$\theta_{\text{ML}} = \underset{\theta}{\operatorname{argmax}} p(\mathbf{y}|\mathbf{X}, \theta)$$

Assuming i.i.d. samples $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$:

$$\begin{aligned}
 \theta_{\text{ML}} &= \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^N p(y_i|\mathbf{x}_i, \theta) \\
 &= \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^N \log p(y_i|\mathbf{x}_i, \theta)
 \end{aligned}$$

Minimum-Error and Maximum-Likelihood Training II

Assuming that network outputs \mathbf{y} follow a Gaussian distribution:

$$p(y_i | \mathbf{x}_i, \boldsymbol{\theta}) = \mathcal{N}(y_i | \hat{y}_i(\mathbf{x}_i | \boldsymbol{\theta}), \sigma^2)$$

ML-Estimate is then given by:

$$\begin{aligned}
 \theta_{\text{ML}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^N \log p(y_i | \mathbf{x}_i, \boldsymbol{\theta}) \\
 &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^N \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(y_i - \hat{y}_i)^2}{2\sigma^2} \right\} \\
 &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^N \left\{ -\frac{(y_i - \hat{y}_i)^2}{2\sigma^2} \right\} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^N (y_i - \hat{y}_i)^2 = \theta_{\text{MSE}}
 \end{aligned}$$

Note: Assumed distribution needs to be compatible with activation function in *final* layer!

⇒ here assumed *linear*, i.e., NN performs regression

Minimum-Error and Maximum-Likelihood Training III

Why cast minimum-error optimization into ML framework?

Properties of ML Estimation

- ▶ *Consistency*: ML estimate will converge to true θ as $N \rightarrow \infty$.
(if true distribution lies within the model assumed)
- ▶ *Statistical Efficiency*: With increasing N , no other estimator produces lower MSE wrt. true θ .

⇒ ML Estimation often preferred in Machine Learning!

Reminder: For NN training, ML estimation is performed via gradient descent!

Stochastic Gradient Descent (SGD)

Gradient can be computed ...

- (1) for complete training data (avg.)
- (2) “online” for every new sample.

But *neither* works well!

- ⇒ Training takes forever!
- ⇒ Gradient is extremely noisy!

Solution:

Compute estimate of gradient for a small sub-set of the training data (so-called “mini batch”)

Note: Samples are drawn *randomly* from the training set!

⇒ Gradient estimate is *stochastic*!

Stochastic Gradient Descent (SGD) II

Disadvantage of mini-batch processing:
 Gradient is still somewhat noisy.

Improved Solution:

Introduce so-called *momentum* η ($0 \ll \eta < 1$):

$$w_{ij}^{(t+1)} \leftarrow w_{ij}^{(t)} + \beta \left[(1 - \eta) \Delta w_{ij}^{(t+1)} + \eta \Delta w_{ij}^{(t)} \right]$$

where $\Delta w_{ij}^{(t)} = -\frac{\partial \epsilon^{(t)}}{\partial w_{ij}^t}$

Note: Computes smoothed sequence of gradient estimates
 (i.e. sliding average with exponential decay).

Classifying Images with Neural Networks

Problem:

- ▶ Using MLPs for image classification is only possible for very small images (e.g. 28×28 pixels)
- ▶ Number of weights would explode for bigger images

Examples:

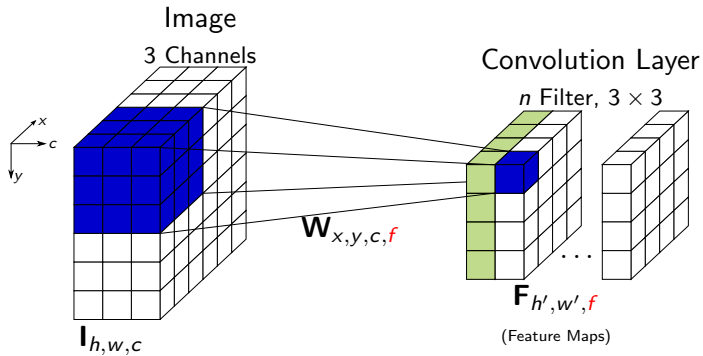
- ▶ 224×224 RGB image², 1000 neurons in 1st layer
 $\approx 1.5 \cdot 10^8$ weights / 575 MB
- ▶ 300 dpi A4 scan, 1000 neurons in 1st layer
 $\approx 2.5 \cdot 10^{10}$ weights / 95 GB

Solution: Don't use fully connected layer but rather apply a small number of weights at all possible locations in the image

⇒ Convolutional Layer

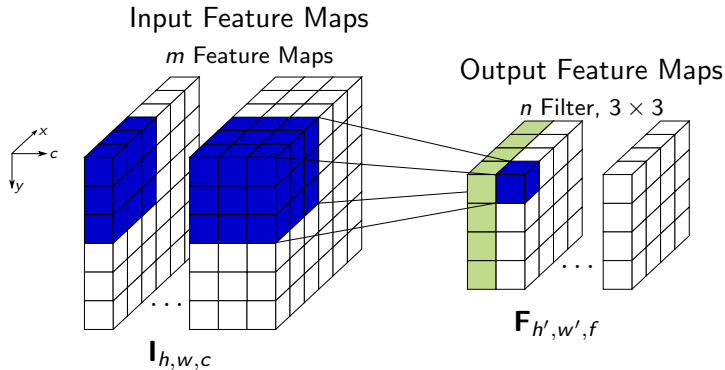
²Size used in ImageNet challenge

Convolutional Layer

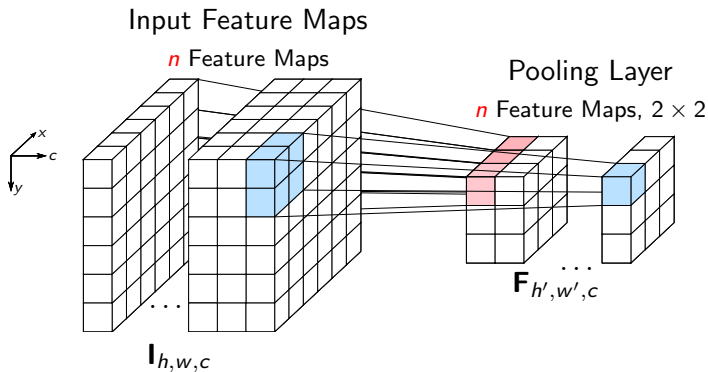


$$F_{x,y,f} = \varphi \left(\sum_{c=1}^K \sum_{i=1}^3 \sum_{j=1}^3 W_{i,j,c,f} \cdot I_{x+i,y+j,c} + b_f \right)$$

Cascade of Convolutional Layers

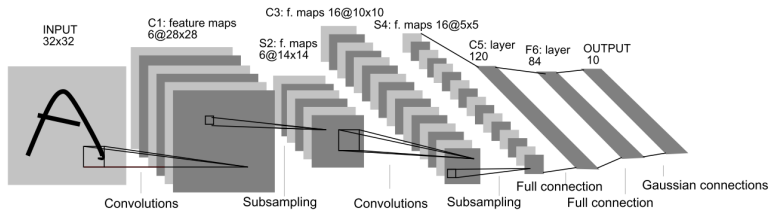


Pooling Layer



$$F_{x,y,f} = \max_{i,j} I_{x+i,y+j,f}$$

LeNet



(Source: [LeCun et al., 1990])

- ▶ LeNet predicts one of 10 character classes for a given input image
- ▶ Subsampling = Pooling Layer
- ▶ Gaussian Connections = FC Layer + Euclidean Loss

Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L.D. Jackel: [Handwritten Digit Recognition with a Back-Propagation Network](#), Neural Information Processing Systems, pp. 396–404, 1990.

Deep Learning

In general: Deeper network architectures perform better than shallower ones for vision tasks

Important: Only empirical evidence (no theoretical proofs)

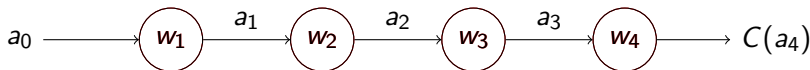
Technically: Deeper means more layers, not a deeper understanding

Even with high computation power and large datasets,
Deep Learning did not really pick up until 2012!

Why? Vanishing Gradient Problem

Vanishing Gradient Problem

Four neuron network, 1D input, 1D output



$$z_i = w_i a_{i-1} + b_i$$

$$a_i = \sigma(z_i)$$

$$\frac{\partial C}{\partial w_1} = \frac{\partial C}{\partial z_4} \frac{\partial z_4}{\partial a_3} \cdot \frac{\partial a_3}{\partial z_3} \frac{\partial z_3}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_2} \frac{\partial z_2}{\partial a_1} \cdot \frac{\partial a_1}{\partial z_1} \frac{\partial z_1}{\partial w_1}$$

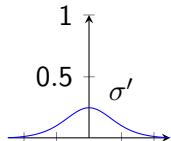
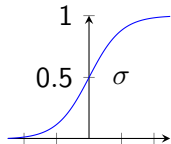
$$= \sigma'(z_4) w_4 \cdot \sigma'(z_3) w_3 \cdot \sigma'(z_2) w_2 \cdot \sigma'(z_1) a_0$$

$$= \sigma'(z_4) \sigma'(z_3) \sigma'(z_2) \sigma'(z_1) \cdot w_4 w_3 w_2 a_0$$

$$= \underbrace{\sigma'(z_4) \sigma'(z_3) \sigma'(z_2) \sigma'(z_1)}_{\leq \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4}} \cdot w_4 w_3 w_2 a_0$$

$$\leq \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{4}$$

$$< \frac{1}{256} \cdot w_4 w_3 w_2 a_0$$

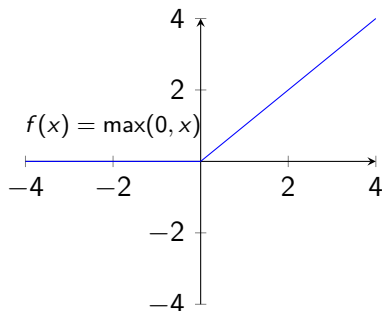


Vanishing Gradient Problem

- ▶ Derivative of sigmoidal activation functions < 1
- ▶ Exponential decay of gradient magnitude

Desirable: Activation function with derivative = 1 but non-linear
(> 1 = exploding gradient)

Solution: Rectified Linear Unit (ReLU) [Glorot & Bengio 2010]



How to Get Along With Limited Training Data?

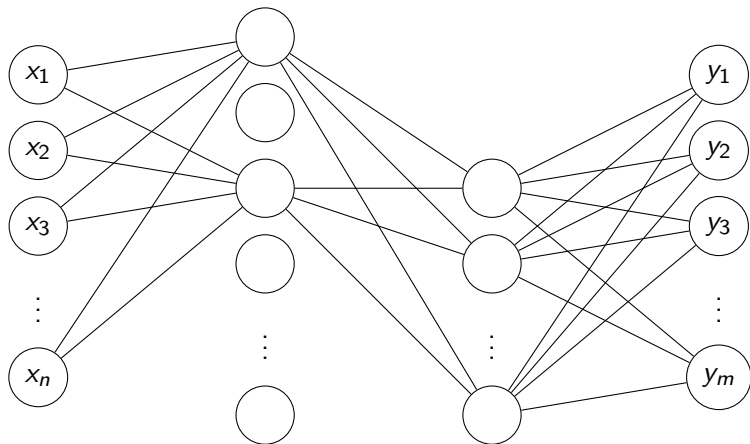
Problem: CNNs easily contain billions of parameters (weights)!
⇒ Could easily learn training samples “by heart”.

Solution: Apply *Regularization* during training

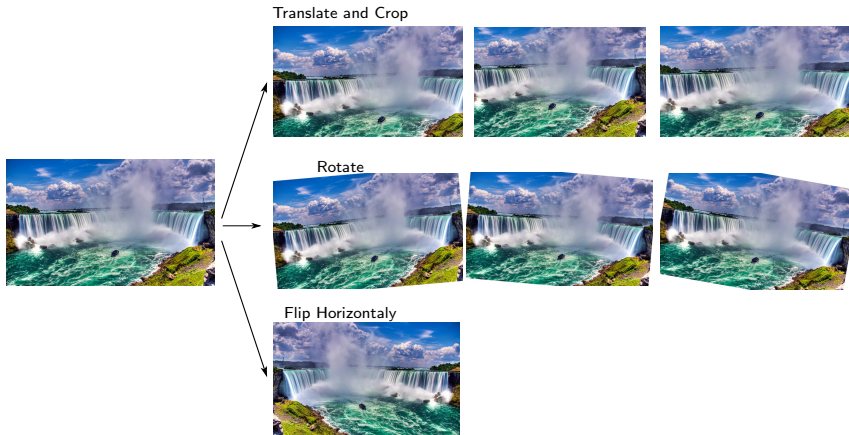
Fundamental Techniques:

- ▶ *Convolutional layers*
- ▶ *Dropout*
Randomly set outputs of neurons to zero
(usually 50% of fully connected layers)
- ▶ *Data Augmentation:*
Generate new, slightly different training samples from existing ones by certain transforms
(e.g. slight translations, rotations, ...)

Dropout

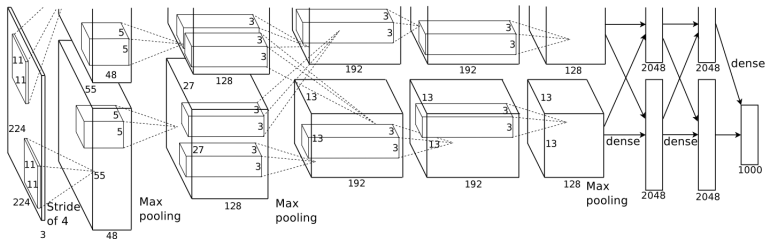


Data Augmentation



Note: Usually different augmentation techniques are mixed to create a single augmented image

Well-known Deep Learning Architectures: AlexNet

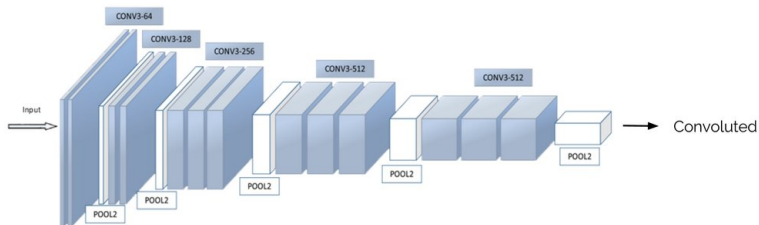


(Source: [Krizhevsky et al., 2012])

- ▶ CNN which kicked off the current Deep Learning hype
- ▶ Architecture similar to LeNet but more layers/parameters
- ▶ Trained on two graphic cards for over a week on ImageNet

A. Krizhevsky, I. Sutskever, G. E. Hinton: [ImageNet Classification with Deep Convolutional Neural Networks](#), Neural Information Processing Systems, pp. 1097–1105, 2012.

Well-known Deep Learning Architectures: VGGNet



(Source: <http://html.scrip.org/>)

- ▶ First CNN to use only 3×3 convolutions (standard for current CNNs)
- ▶ Low number of filters in the early layers, high number of filters in the later layers
- ▶ Anytime pooling is applied, the number of filters is doubled

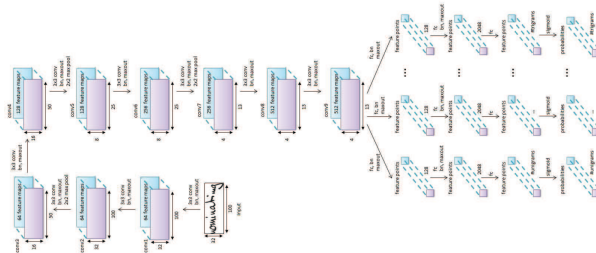
K. Simonyan, A. Zisserman: [Very Deep Convolutional Networks for Large-Scale Image Recognition](#), arXiv, 2014.

Overview

- ▶ Introduction
- ▶ Fundamentals: Bag-of-Features
- ▶ Learning Document Image Representations
- ▶ Learning Word Spotting Models
- ▶ Deep Learning Fundamentals
- ▶ **Deep Learning for Word Spotting**
- ▶ Outlook
- ▶ Summary

Related Work on Deep Learning

CNN-N-Gram [Poznanski *et al.*, 2016]



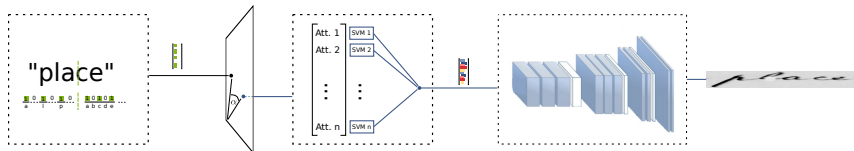
- ▶ Considers isolated word *recognition*
- ▶ Uses PHOC + 1 level of trigrams
- ▶ Attributes are predicted directly with *separate* MLPs

Poznanski, A., Wolf, L.: [CNN-N-Gram for Handwriting Word Recognition](#), IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, pp. 2305–2314, Las Vegas, USA, 2016.

Related Work on Deep Learning for Word Spotting

Deep Feature Embedding

[Krishnan *et al.*, 2016]



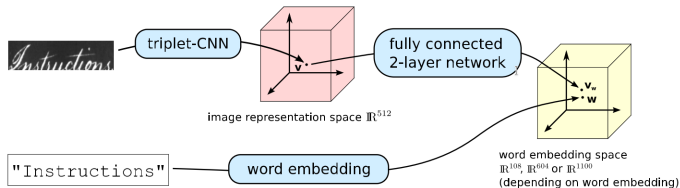
- ▶ Uses CNN to produce feature representation of images
- ▶ CNN is pre-trained on synthetically generated data
- ▶ SVMs predict attributes of PHOC representation

Krishnan, P., Dutta, K., Jawahar, C. V.: [Deep Feature Embedding for Accurate Recognition and Retrieval of Handwritten Text](#), Int. Conf. on Frontiers in Handwriting Recognition, Shenzhen, China, pp. 289–294, 2016.

Related Work on Deep Learning for Word Spotting II

Triplet-CNN

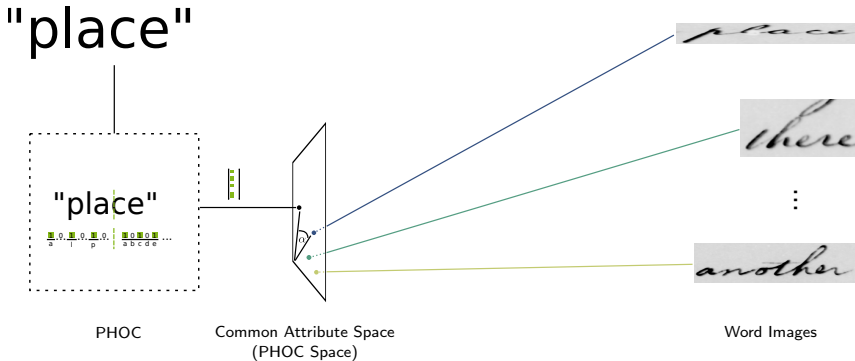
[Wilkinson & Brun, 2016]



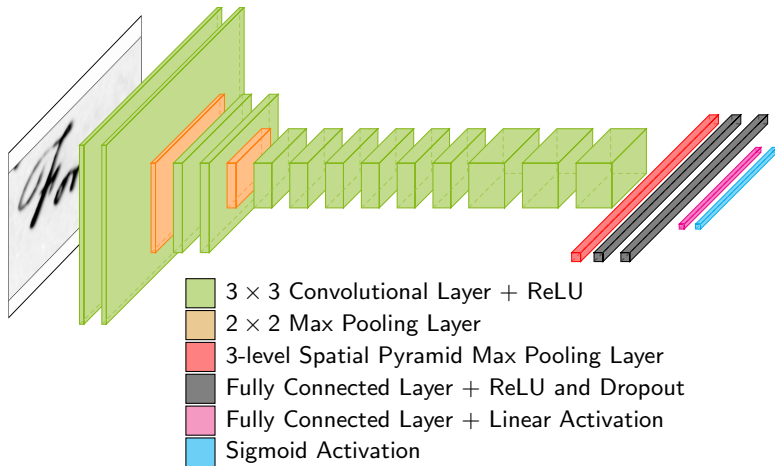
- ▶ Residual network learns word descriptors using triplet loss
- ▶ Separate MLP predicts attribute representation
- ▶ Proposed *DCT of Words* embedding

Wilkinson, T., Brun, A.: [Semantic and verbatim word spotting using deep neural networks](#). Int. Conf. on Frontiers in Handwriting Recognition, Shenzhen, China, pp. 307–312, 2016.

Reminder: General Framework



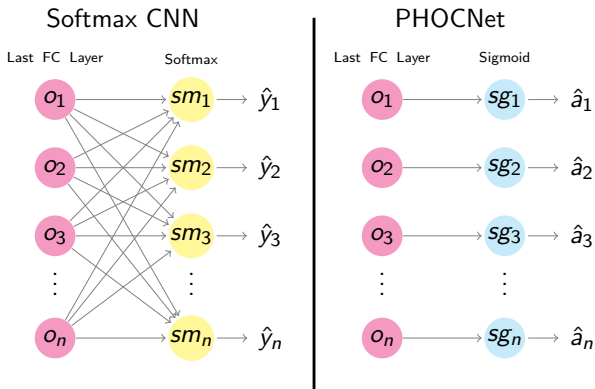
PHOCNet



S. Sudholt, G. A. Fink: **PHOCNet: A Deep Convolutional Neural Network for Word Spotting in Handwritten Documents**, Proc. Int. Conf. on Frontiers in Handwriting Recognition, Shenzhen, China, 2016.

Softmax CNN vs. PHOCNet

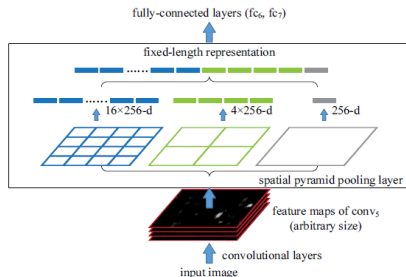
- ▶ In order to classify attributes, replace softmax with a sigmoid activation
- ▶ Each output neuron predicts one attribute



Spatial Pyramid Pooling Layer

- ▶ Convolutional layers can already deal with arbitrary image sizes
- ▶ Only MLP part has a problem with changing image sizes

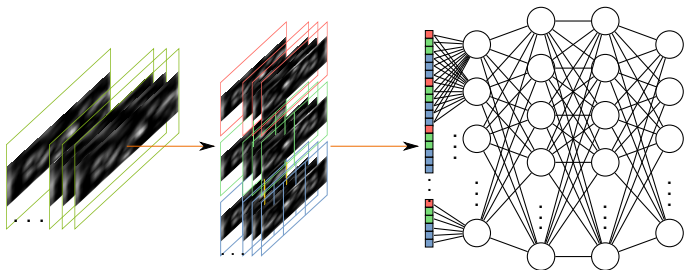
Solution: apply spatial pyramid concept to the last convolutional output to generate fixed-size representation



K. He, X. Zhang, S. Ren, J. Sun: [Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition](#), Proc. European Conference on Computer Vision, pp. 346–361, 2014.

Temporal Pyramid Pooling Layer

- Pooling focusses on *horizontal* axis, i.e., writing direction, only!



- Input: All k feature maps from last convolutional layer
- Performs pyramidal pooling along horizontal axis for each feature map and with different splits (cf. PHOC)
- Produces fixed size input for MLP

Sudholt, S., Fink, G. A.: [Evaluating Word String Embeddings and Loss Functions for CNN-based Word Spotting](#), Proc. Int. Conf. on Document Analysis and Recognition, Kyoto, Japan, 2017.

Training the PHOCNet: Loss Functions

Reminder: Casting NN optimization into ML framework allows to derive loss functions.

Procedure:

1. Define (assumed) distribution of label data
2. Set up (negative) log-likelihood function and derive loss

What are appropriate distribution assumptions for PHOC vectors?

- ▶ Every attribute can be considered a binary variable, i.e., with Bernoulli distribution

$$p_B(k|p) = p^k(1-p)^{(1-k)} \quad \text{for } k \in \{0, 1\}$$

(here: p corresponds to “success” probability, $k = 1$)

- ▶ Attributes exhibit dependencies but modeling these is prohibitive!
⇒ assume pair-wise independent PHOC attributes

Training the PHOCNet: Loss Functions II

Minimize negative log-likelihood for vector of D pair-wise independent Bernoulli-distributed attributes:

$$\begin{aligned}
 \theta_{\text{ML}} &= \underset{\theta}{\operatorname{argmin}} - \log \prod_{i=1}^N \prod_{d=1}^D p_{\mathcal{B}}(y_i^{(d)} | \hat{y}_i^{(d)}(\mathbf{x}_i | \theta)) \\
 &= \underset{\theta}{\operatorname{argmin}} - \sum_{i=1}^N \sum_{d=1}^D \log p_{\mathcal{B}}(y_i^{(d)} | \hat{y}_i^{(d)}) \\
 &= \underset{\theta}{\operatorname{argmin}} - \sum_{i=1}^N \sum_{d=1}^D \log \left\{ (\hat{y}_i^{(d)})^{y_i^{(d)}} \cdot (1 - \hat{y}_i^{(d)})^{(1-y_i^{(d)})} \right\} \\
 &= \underset{\theta}{\operatorname{argmin}} - \sum_{i=1}^N \sum_{d=1}^D y_i^{(d)} \log \hat{y}_i^{(d)} + (1 - y_i^{(d)}) \log(1 - \hat{y}_i^{(d)})
 \end{aligned}$$

Note: Usually referred to as *Binary Cross Entropy Loss* or *Sigmoid Cross Entropy Loss*

Training the PHOCNet: Loss Functions III

Alternative view on a loss for PHOC representations:

🛑 Euclidean distance / MSE loss surely not suitable!

Reason: Curse of dimensionality, i.e., Euclidean distance becomes meaningless in high-dimensional spaces!

How can pair-wise independence assumption be avoided?

⇒ Consider binary vectors as a whole!

Observation: Cosine dissimilarity works well for *directional* data and likewise for *histogram-like* data!

$$d_{\cos}(\mathbf{a}, \mathbf{b}) = 1 - \frac{\mathbf{a}^T \mathbf{b}}{\|\mathbf{a}\| \cdot \|\mathbf{b}\|}$$

Reason: Not the length of the vector but the direction matters!
(direction \approx shape of the histogram)

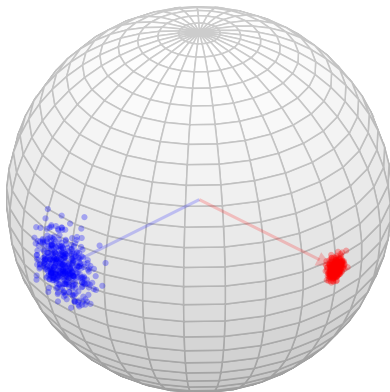
Training the PHOCNet: Loss Functions IV

Distribution for directional data: Von Mises-Fisher distribution

- ▶ Normal distribution on the unit hypersphere
- ▶ Parameters: *mean direction* $\boldsymbol{\mu}$ and *concentration* κ (\approx inverse variance)

$$p_{\mathcal{MF}}(\mathbf{x}|\boldsymbol{\mu}, \kappa) = C_D(\kappa) \exp(\kappa \boldsymbol{\mu}^T \mathbf{x})$$

with $\|\boldsymbol{\mu}\| = \|\mathbf{x}\| = 1$
 and $C_D(\kappa)$ a normalization factor



Training the PHOCNet: Loss Functions IV

Minimize negative log-likelihood for vectors following von-Mises-Fisher distributions (with identical κ):

$$\begin{aligned}
 \theta_{\text{ML}} &= \underset{\theta}{\operatorname{argmin}} - \log \prod_{i=1}^N p_{\mathcal{MF}}(\mathbf{y}_i | \hat{\mathbf{y}}_i(\mathbf{x}_i | \theta), \kappa) \\
 &= \underset{\theta}{\operatorname{argmin}} - \sum_{i=1}^N \log p_{\mathcal{MF}}(\mathbf{y}_i | \hat{\mathbf{y}}_i) \\
 &= \underset{\theta}{\operatorname{argmin}} - \sum_{i=1}^N \log C_D(\kappa) \exp(\kappa \hat{\mathbf{y}}_i^T \mathbf{x}_i) \\
 &= \underset{\theta}{\operatorname{argmin}} - \sum_{i=1}^N \kappa \hat{\mathbf{y}}_i^T \mathbf{x}_i = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^N 1 - \hat{\mathbf{y}}_i^T \mathbf{x}_i \\
 &= \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^N 1 - \frac{\hat{\mathbf{y}}_i^T \mathbf{x}_i}{\|\hat{\mathbf{y}}_i\| \cdot \|\mathbf{x}_i\|} = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^N d_{\cos}(\hat{\mathbf{y}}_i, \mathbf{x}_i)
 \end{aligned}$$

Training the PHOCNet: Weight Initialization

Problem: SGD optimization of network parameters needs initial parameter set θ_0 to start from!

Observation: Training of deep networks is quite sensitive to the choice of this initialization!

Procedure for the PHOCnet: [He *et al.* 2015]

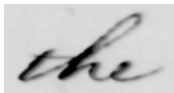
$$w_{ij} \sim \mathcal{N}\left(0, \frac{2}{n_j}\right) \quad \text{and} \quad b_j \leftarrow 0$$

(Here n_j is equal to the number of inputs per neuron.)

Note: Referred to as 'MSRA' (Microsoft Research Asia) in *Caffe* and `kaiming_normal_(...)` in *Pytorch*.

Word Spotting: Evaluation

Query word image



or string ("the")

Retrieved patches sorted by score



- ▶ *Precision*: How relevant is the list?
- ▶ *Recall*: How complete is the list w.r.t. relevant items?
- ▶ *Average Precision*: How well is the retrieval list sorted?
(Implicitly takes Recall into account!)

Notes:

- ▶ Usually mean values over many queries are reported (mAP and mR).
- ▶ Patch overlap threshold required for segmentation-free case.

Word Spotting: Evaluation II

Average Precision: How well is the retrieval list sorted?

- ▶ Let's make the example a little more complex:

[1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1]

Total number of relevant items in dataset: 10

- ▶ Precision: $\frac{8}{15} \approx 0.53$, Recall: $\frac{8}{10} = 0.8$
 No information about the list's order!
- ▶ Average Precision: Precision averaged at different recall levels
 (cf. e.g. [Lladós *et al.* 2012]):

$$\frac{\sum_{k=1}^n \text{Precision}_k \times \text{rel}(k)}{\# \text{Relevant Items in Dataset}}$$

rel(k): Relevancy of item k , Precision $_k$: Precision at cut-off k

Accumulate Precision whenever the Recall changes and normalize.

Word Spotting: Evaluation III

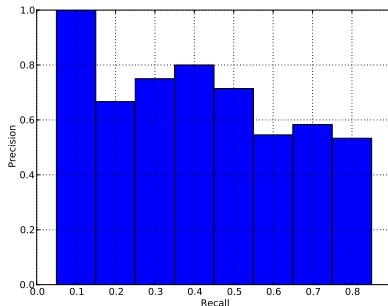
[1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1]

Average Precision:

$$\frac{\sum_{k=1}^n \text{Precision}_k \times \text{rel}(k)}{\# \text{Relevant Items in Dataset}}$$

$$\frac{\frac{1}{1} + \frac{2}{3} + \frac{3}{4} + \frac{4}{5} + \frac{5}{7} + \frac{6}{11} + \frac{7}{12} + \frac{8}{15}}{10}$$

≈ 0.56

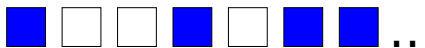


Word Spotting: Evaluation IV


Mean Average Precision: Average of AP for different queries

Example (with 4 relevant items to retrieve):


 AP: 1.00


 AP: 0.64


 AP: 0.48


 AP: 0.43

mAP: 0.63

Evaluation: Significance Testing

Question: Is a mAP of 0.93 saying that a method is better than another with a mAP of 0.91/0.77/0.3?

Problem: *You never know! Could be random effects!*

Solution: Test difference of evaluation results for *statistical significance!*

Method of choice: *Permutation Test*
(requires no assumptions about distribution of test statistic)

Evaluation: Permutation Test

... also known as randomization test (cf. Good 2000)

Null Hypothesis: Samples A and B obtained from two different sources (here: results of word spotting methods) follow the same underlying distribution, (i.e., systems perform the same).

Basic idea: Reject null hypothesis if difference T_{obs} between sample means μ_A and μ_B (here: mAP) is large enough.

Procedure: Generate all possible N permutations of assigning samples to sets A and B and compute difference T_n of means.

Result: Proportion of differences $T_n \geq T_{\text{obs}}$ is Probability (p -value) of accepting the null hypothesis.

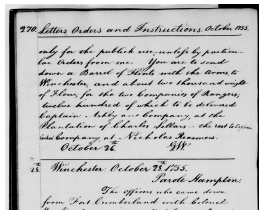
⇒ For small p it is concluded that sets A and B do not follow the same distribution (systems perform differently)!

Evaluation: Data Sets I

George Washington Benchmark:

Database of handwritten letters

- ▶ Likely single-writer documents
- ▶ 20 pages, 4860 words
- ▶ 4-fold cross validation



Esposalles Benchmark: Database of marriage license books

- ▶ High script variability
- ▶ Degradation (e.g. bleed through)
- ▶ 32k training / 13k test word images

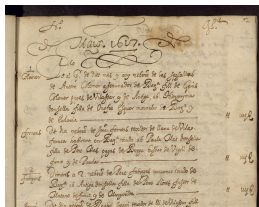


Image sources:

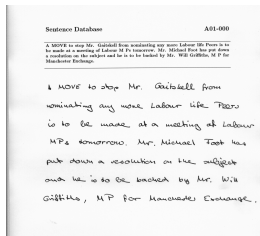
The George Washington Papers at the Library of Congress, 1741-1799

V. Romero et al.: *The ESPOSALLES Database: An Ancient Marriage License Corpus for Off-line Handwriting Recognition*, Pattern Recognition, Volume 46(6), pp. 1658-1669, 2013.

Evaluation: Data Sets II

IAM Database: Handwritten paragraphs (elicited)

- ▶ more than 600 writers
- ▶ more than 13k lines / 115k words
- ▶ 6k/1.8k/1.8k train/val/test lines



IFN/ENIT Database: Handwritten Tunesian city names (elicited, Arabic)

- ▶ more than 400 writers / 26k words
- ▶ training (A, B, C); test (D)
- ▶ reduced Arabic character set (50)

row#	word	row#	word
914c	فغان الشوق	3048	نارورد
3124	فقال	3024	نار
9112	القاضي	9112	نار
3265	تلاوتين 2 نوفمبر	3265	نار 7 نوفمبر
6016	عزل	6016	نار
3491	تل العزات	7141	نار
8194	سينويته الضالينه	8189	نار
4174	سائن الكبرى	4174	نار
3169	مركز الناصي	3087	نار

Image sources:

U. V. Marti, H. Bunke: *The IAM-Database: An English Sentence Database for Offline Handwriting Recognition*, IJDAR, vol. 5, pp. 39-46, 2002.

M. Pechwitz, S. Snoussi Maddouri, V. Märgner, N. Ellouze, H. Amiri: *IFN/ENIT-Database of handwritten Arabic words*, Proc. 7th Colloque International Francophone sur l'Ecrit et le Document, Hammamet, Tunisia, 2002.

Evaluation: Procedure & Protocol

- ▶ Consider only Query-by-String (QbS) scenario here
- ▶ Follow “Almazan” protocol ([Almazan et al. 2014](#))
 - ▶ Every *unique transcription* in the test set is used as query
 - ▶ For experiments on IAM-DB: discard stop words
- ▶ Query PHOC \mathbf{a}^q can be given directly
- ▶ PHOCNet predicts PHOC $\hat{\mathbf{a}}$ for each word image in the test set
- ▶ Test word images are ranked according to cosine dissimilarity to query

$$d_{\cos}(\hat{\mathbf{a}}, \mathbf{a}^q) = 1 - \frac{\hat{\mathbf{a}}^T \mathbf{a}^q}{\|\hat{\mathbf{a}}\| \cdot \|\mathbf{a}^q\|}$$

Evaluation Results

Segmentation-based Word Spotting Performance in mAP [%]

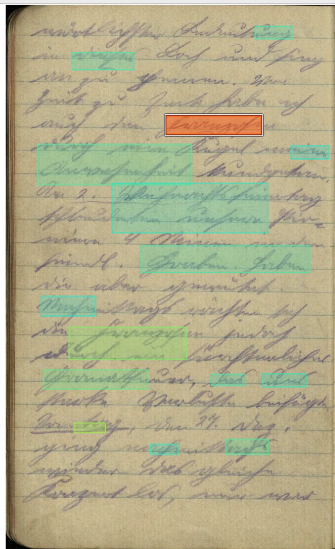
Method	GW	IAM	Esposalles	IFN/ENIT
TPP-PHOCNet	97.92	93.42	94.32	94.53
PHOCNet	97.44	91.12	94.89	93.87
Deep Feature Embedding [13]	92.84	91.58	—	—
Attribute SVM + FV [4]	91.29	73.72	—	—
LSA Embedding [1]	56.54	—	—	—
Triplet-CNN (*) [38]	93.69	89.49	—	—
BLSTM (*) [5]	84.00	78.00	—	—

Sudholt, S., Fink, G. A.: [Attribute CNNs for Word Spotting in Handwritten Documents](#), Int. Journal on Document Analysis and Recognition, 21(3), pp. 159–160, 2018.

QbS-Wordspotting: "War Diary WW I", Collection Dr. Bley

Wordspotting - Kriegstagebuch (Sammlung B. Bley)

Projects Settings Help



Query Query Results

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.
- 11.
- 12.

Show Result Patches

Show Heatmap

(Video created by Matthias Kasperidus)

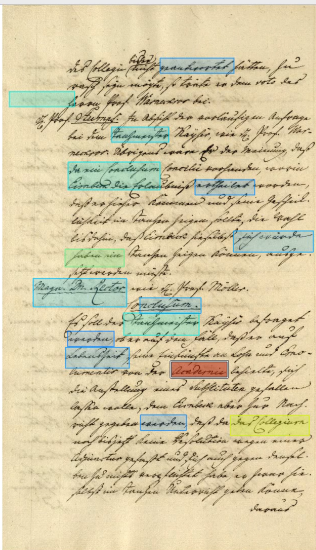
QbS Wordspotting: "Konzilsprotokolle", ICFHR 2014 KWS Comp.

Wordspotting - Konzilsprotokolle (ICFHR 2016 Competition)

Projects Settings Help



A vertical sidebar on the left side of the interface displaying a list of project thumbnails. Each thumbnail is labeled with a project ID: K0094, K0095, K0096, K0097, K0098, K0099, and K0100. The thumbnails show snippets of handwritten text from the documents.



Query Query Results



A sidebar on the right side of the interface showing search results. It lists 11 items, each with a number and a snippet of handwritten text. The snippets are: 1. Academie, 2. Academie, 3. Academie, 4. Academie, 5. Academie, 6. Kongregation, 7. Kongregation, 8. Kongregation, 9. Kongregation, 10. Kongregation, 11. Kongregation. The third result is highlighted with a blue background. At the bottom, there are checkboxes for 'Show Result Patches' (checked) and 'Show Heatmap' (unchecked).

(Video created by Matthias Kasperidus)

Overview

- ▶ Introduction
- ▶ Fundamentals: Bag-of-Features
- ▶ Learning Document Image Representations
- ▶ Learning Word Spotting Models
- ▶ Deep Learning Fundamentals
- ▶ Deep Learning for Word Spotting
- ▶ **Outlook**
- ▶ Summary

Deep Learning vs. BoF-HMMs

BoF HMMs

- ✓ No annotated training data required (for QbE)
- ⚡ Limited variability in writing style
- ✓ Instantaneous “training”
- ⚡ Relatively expensive during retrieval (sequence model)

⇒ Suitable for *completely new* document collections!

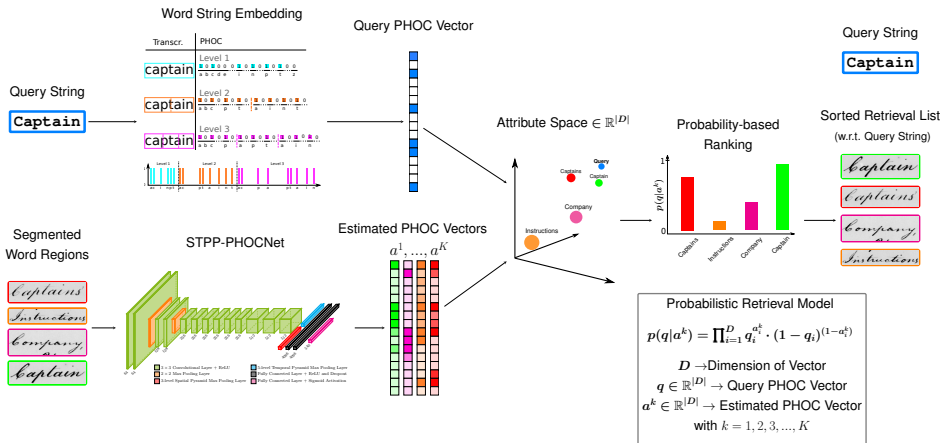
Deep Networks

- ⚡ Annotated training data required (more is better!)
- ✓ Can cope with large variability in writing style
- ⚡ Training computationally expensive
- ✓ Fast retrieval (representations can be pre-computed)

⇒ Suitable for *partly developed* document collections

Matching Revisited

Matching process can be realized in *probabilistic* framework:



E. Rusakov, L. Rothacker, H. Mo, G. A. Fink: **A probabilistic retrieval model for word spotting based on direct attribute prediction**, In *Proc. Int. Conf. on Frontiers in Handwriting Recognition*, Niagara Falls, USA, 2018.

A Word on Segmentation

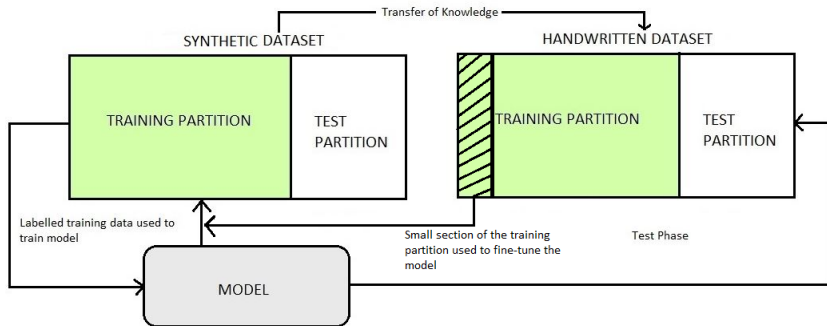
Note: PHOCNet model itself requires segmentation

Solutions:

- ▶ Largely heuristic (based on binarization and/or connected component analysis)
- ▶ Some Region-Proposal Networks (and variants thereof) have been proposed
- ⚡ No satisfactory solution available
- ✓ Performance of heuristic methods acceptable for most data sets

Annotated Training Data?

Potential Solution: Adaptation on in-domain material

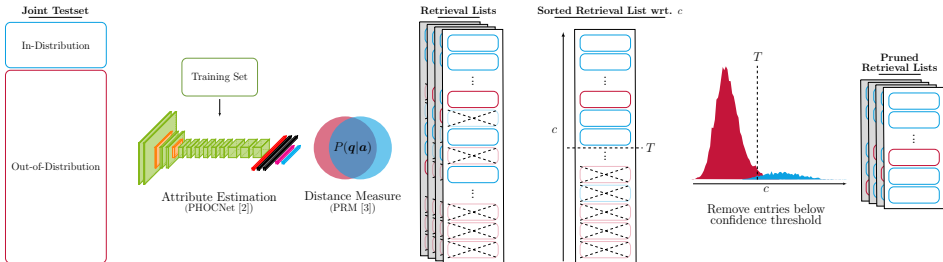


⇒ Competitive performance on homogeneous collections

N. Gurjar, S. Sudholt, and G. A. Fink: [Learning deep representations for word spotting under weak supervision](#), *Proc. Int. Workshop on Document Analysis Systems*, Vienna, Austria, 2018.

Knowing the Limits

What happens on unknown / out-of-domain material?



Approach: Use confidence measure to select reliably ranked hypotheses

F. Wolf, P. Oberdiek, G. A. Fink: [Exploring Confidence Measures for Word Spotting in Heterogeneous Datasets](#), In *Proc. Int. Conf. on Document Analysis and Recognition*, Sydney, Australia, 2019.

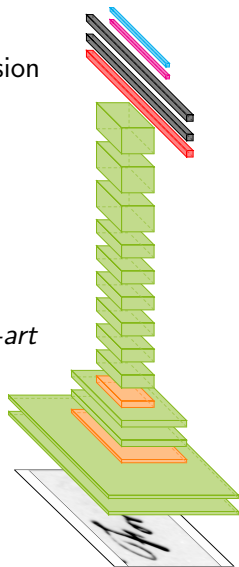
Overview

- ▶ Introduction
- ▶ Fundamentals: Bag-of-Features
- ▶ Learning Document Image Representations
- ▶ Learning Word Spotting Models
- ▶ Deep Learning Fundamentals
- ▶ Deep Learning for Word Spotting
- ▶ Outlook
- ▶ **Summary**

Summary

- ▶ Deep Learning is a hot topic in Computer Vision *and* Document Image Analysis
 - ⇒ *You should know about it!*
- ▶ CNNs are the most popular deep networks
 - ⇒ *Especially suitable for (document) images!*
- ▶ Deep Networks are currently the *state-of-the-art* methods for word spotting
 - ⇒ *Especially the PHOCNet :-)*

But for some scenarios alternative methods might be advantageous!



Demos Available ;-)

References I

- [1] David Aldavert, Marçal Rusinol, Ricardo Toledo, and Josep Lladós.
Integrating visual and textual cues for query-by-string word spotting.
In International Conference on Document Analysis and Recognition, pages 511–515, 2013.
- [2] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny.
Segmentation-free word spotting with exemplar svms.
Pattern Recognition, 47(12):3967 – 3978, 2014.
- [3] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny.
Word spotting and recognition with embedded attributes.
IEEE Trans. on Pattern Analysis and Machine Intelligence, 36(12):2552–2566, 2014.

References II

- [4] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny. Word Spotting and Recognition with Embedded Attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(12):2552–2566, 2014.
- [5] Volkmar Frinken, Andreas Fischer, R. Manmatha, and Horst Bunke. A novel word spotting method based on recurrent neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:211–224, 2012.

References III

- [6] Xavier Glorot and Yoshua Bengio.
Understanding the difficulty of training deep feedforward neural networks.
AISTATS, 9:249–256, 2010.
- [7] Phillip Good.
Permutation Tests - A Practical Guide to Resampling Methods for Testing Hypothesis.
Springer, 2 edition, 2000.
- [8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
Deep Learning.
The MIT Press, 2016.

References IV

- [9] Neha Gurjar, Sebastian Sudholt, and Gernot A. Fink.
Learning deep representations for word spotting under weak supervision.
In Proc. Int. Workshop on Document Analysis Systems, Vienna, Austria, 2018.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Spatial pyramid pooling in deep convolutional networks for visual recognition.
European Conference on Computer Vision, pages 346–361, 2014.

References V

- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, pages 1026–1034, Washington, DC, USA, 2015. initialization of deep neural networks termed 'MSRA' (Microsoft Research Asia) in Caffe.
- [12] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

References VI

- [13] Praveen Krishnan, Kartik Dutta, and C.V. Jawahar.
Deep feature embedding for accurate recognition and retrieval of handwritten text.
In International Conference on Frontiers in Handwriting Recognition, pages 289–294, 2016.
- [14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton.
Imagenet classification with deep convolutional neural networks.
In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Neural Information Processing Systems*, pages 1097–1105, 2012.

References VII

- [15] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce.
Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories.
In Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, volume 2, pages 2169–2178, 2006.
- [16] Yann LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel.
Handwritten digit recognition with a back-propagation network.
Neural Information Processing Systems, pages 396–404, 1990.

References VIII

- [17] Josep Lladós, Marçal Rusiñol, Alicia Fornés, David Fernández, and Anjan Dutta.
On the influence of word representations for handwritten word spotting in historical documents.
Int. J. Pattern Recognition and Artificial Intelligence, 26(5), 2012.
- [18] David Lowe.
Distinctive image features from scale-invariant keypoints.
Int. J. of Computer Vision, 60(2):91–110, 2004.
- [19] R. Manmatha, Chengfeng Han, E. M. Riseman, and W. B. Croft.
Indexing handwriting using word matching.
In *Proc. of the First ACM Int. Conf. on Digital Libraries*, DL '96, pages 151–159, New York, NY, USA, 1996. ACM.

References IX

[20] U.-V. Marti and H. Bunke.

The IAM-database: An English sentence database for offline handwriting recognition.

Int. Journal on Document Analysis and Recognition, 5:39–46, 2002.

[21] Stephen O'Hara and Bruce A. Draper.

Introduction to the bag of features paradigm for image classification and retrieval.

Computing Research Repository, arXiv:1101.3354v1, 2011.

References X

- [22] M. Pechwitz, S. Snoussi Maddouri, V. Märgner, N. Ellouze, and H. Amiri.
IFN/ENIT-database of handwritten Arabic words.
In Proc. 7th Colloque International Francophone sur l'Écrit et le Document, Hammamet, Tunisia, October 2002.
- [23] Arik Poznanski and Lior Wolf.
Cnn-n-gram for handwriting word recognition.
In Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, pages 2305–2314, Las Vegas, USA, 2016.

References XI

- [24] Toni M. Rath and R. Manmatha.
Word image matching using dynamic time warping.
In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–521–II–527 vol.2, June 2003.
- [25] Verónica Romero, Alicia Fornès, Nicolás Serrano, Joan Andreu Sánchez, Alejandro H. Toselli, Volkmar Frinken, Enrique Vidal, and Josep Llad'os.
The esposalles database: An ancient marriage license corpus for off-line handwriting recognition.
Pattern Recognition, 46(6):1658 – 1669, 2013.

References XII

[26] F. Rosenblatt.

The perceptron: A probabilistic model for information storage and organization in the brain.

Psychological Review, 65(6):386–408, 1958.

[27] Leonard Rothacker and Gernot A. Fink.

Segmentation-free query-by-string word spotting with bag-of-features HMMs.

In *Proc. Int. Conf. on Document Analysis and Recognition*, Nancy, France, 2015.

References XIII

- [28] Leonard Rothacker, Marçal Rusinol, and Gernot A. Fink.
Bag-of-features HMMs for segmentation-free word spotting in
handwritten documents.
In Proc. Int. Conf. on Document Analysis and Recognition,
Washington DC, USA, 2013.
- [29] Leonard Rothacker, Marçal Rusinol, Josep Lladós, and Gernot A.
Fink.
A Two-Stage Approach to Segmentation-Free Query-by-Example
Word Spotting.
manuscript cultures, 1(7):47–57, 2014.

References XIV

- [30] Leonard Rothacker, Szilard Vajda, and Gernot A. Fink.
Bag-of-features representations for offline handwriting
recognition applied to Arabic script.
*In Proc. Int. Conf. on Frontiers in Handwriting Recognition, Bari,
Italy, 2012.*
- [31] Eugen Rusakov, Leonard Rothacker, Hynuh Mo, and Gernot A.
Fink.
A probabilistic retrieval model for word spotting based on direct
attribute prediction.
*In Proc. Int. Conf. on Frontiers in Handwriting Recognition,
Niagara Falls, USA, 2018.*
Winner of the IAPR Best Student Paper Award.

References XV

- [32] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós.
Browsing heterogeneous document collections by a
segmentation-free word spotting method.
In Proc. Int. Conf. on Document Analysis and Recognition, pages
63 –67, Beijing, China, 2011.
- [33] R. Shekhar and C.V. Jawahar.
Word image retrieval using bag of visual words.
*In Document Analysis Systems (DAS), 2012 10th IAPR
International Workshop on*, pages 297–301, March 2012.
- [34] Karen Simonyan and Andrew Zisserman.
Very deep convolutional networks for large-scale image
recognition.
arXiv, pages 1–13, 2014.

References XVI

- [35] Sebastian Sudholt and Gernot A. Fink.
PHOCNet: A deep convolutional neural network for word spotting in handwritten documents.
In Proc. Int. Conf. on Frontiers in Handwriting Recognition, Shenzhen, China, 2016.
Winner of the IAPR Best Paper Award.
- [36] Sebastian Sudholt and Gernot A. Fink.
Evaluating word string embeddings and loss functions for cnn-based word spotting.
In Proc. Int. Conf. on Document Analysis and Recognition, Kyoto, Japan, 2017.

References XVII

- [37] Sebastian Sudholt and Gernot A. Fink.
Attribute cnns for word spotting in handwritten documents.
Int. Journal on Document Analysis and Recognition,
21(3):159–160, 2018.
- [38] Tomas Wilkinson and Anders Brun.
Semantic and verbatim word spotting using deep neural
networks.
*In International Conference on Frontiers in Handwriting
Recognition*, pages 307–312, 2016.

References XVIII

- [39] Fabian Wolf, Philipp Oberdiek, and Gernot A. Fink.
Exploring Confidence Measures for Word Spotting in
Heterogeneous Datasets.
In Proc. Int. Conf. on Document Analysis and Recognition,
Sydney, Australia, 2019.
To appear.