technische universität
dortmund

Deep Learning for Word Spotting: Foundations and Current Developments

— SSDA 2023 Tutorial, Fribourg, Switzerland —

Gernot A. Fink

July 4, 2023

- Introduction
- Deep Learning Fundamentals
- Deep Learning for Word Spotting
- Semantic Word Spotting
- Self-Training for Word Spotting
- Summary

with contributions by Sebastian Sudholt, Oliver Tueselmann, and Fabian Wolf

Introduction: Automatic Reading Systems

State of Automatic Reading:

- One of the earliest application fields studied in computer science
- So-called OCR achieves high-quality results for machine-printed text in well-defined settings.
- Online handwriting recognition again gaining popularity
- Offline handwriting recognition: Remarkable results, but still an open research problem

General Methodology:

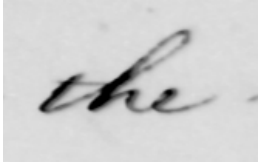
Statistical sequence models (BLSTMs, Transformer) that are trained from *extensive* amounts of example data

Introduction: Why Word Spotting?

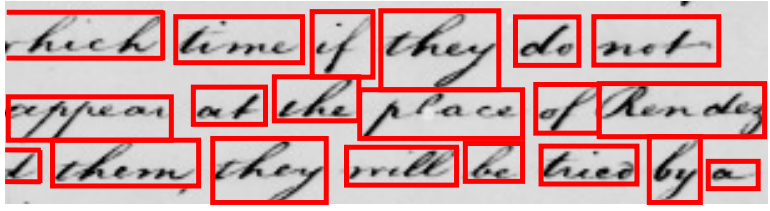
What if automatic transcription of handwriting is no longer feasible?

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

Query word image



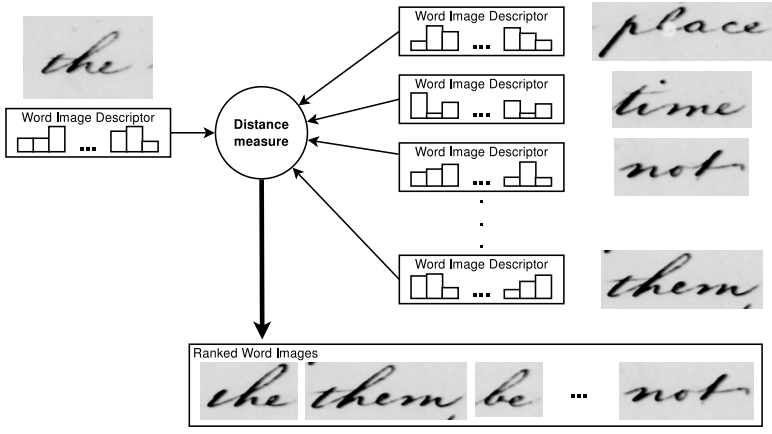
Document image



Introduction: Basic Methodology

Query-by-example word spotting

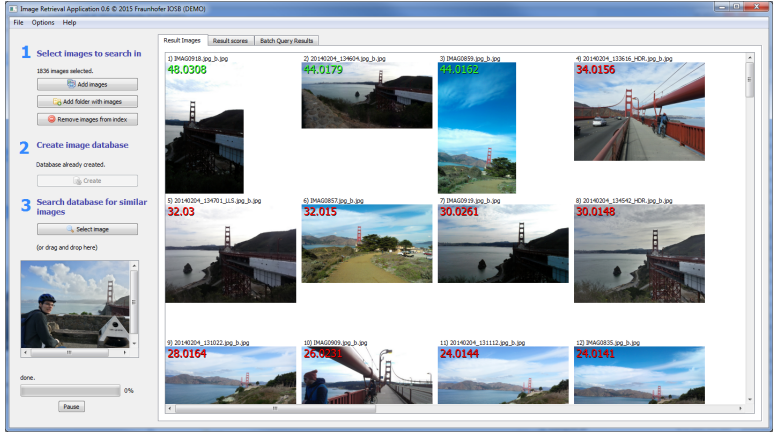
Does that seem familiar?



Based on [Rath & Manmatha, IJRAR'07]

Introduction: Basic Methodology II

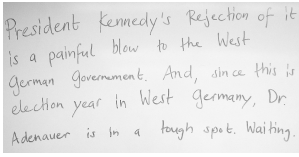
QbE word spotting ≈ special case of **content-based image retrieval**



(Source: Fraunhofer IOSB CBIR Demo)

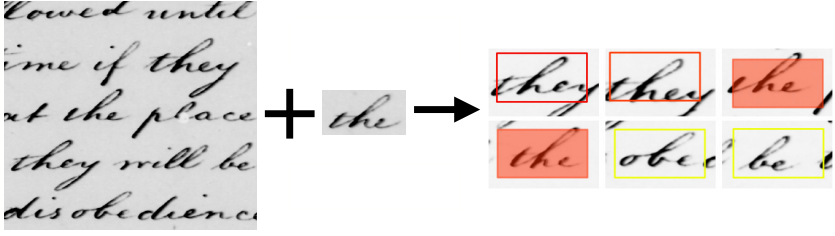
Tasks in Handwriting Recognition

Document Transcription (= "classical" recognition)



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")



Word Spotting: Fundamentals

Core Methodology:

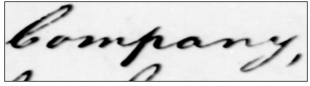
- Specialized image retrieval
- Important ingredient: Image matching procedure
- Frequently required: Pre-segmentation (words / lines)

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")

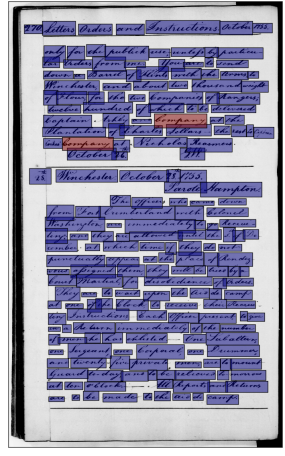
Word Spotting Tasks

Query by Example



Query by String

Company



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

Word-Spotting: Classical Milestones

Manmatha *et al.* 1996: First influential work

(Binarization, Alignment, XOR distance)

Rath & Manmatha 2003: DTW matching

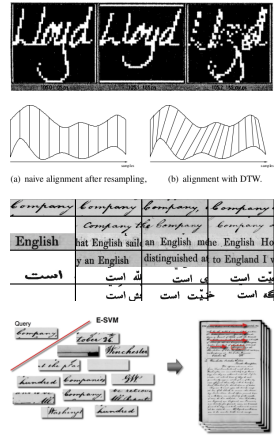
(Normalization, profile features)

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

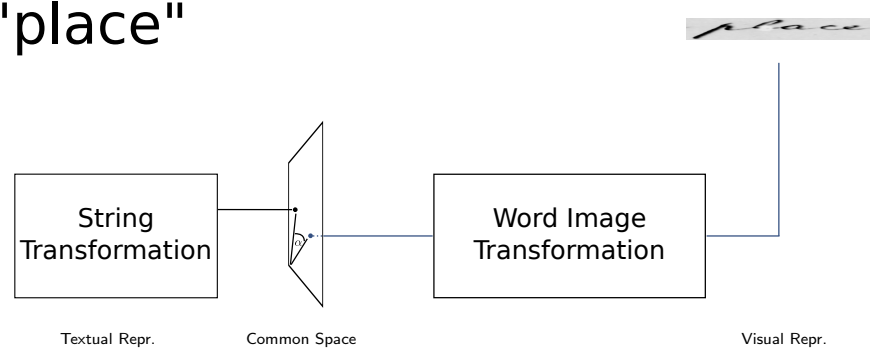
("Exemplar SVM", query expansion)



Important Breakthrough: Subspace Representations for Word Spotting

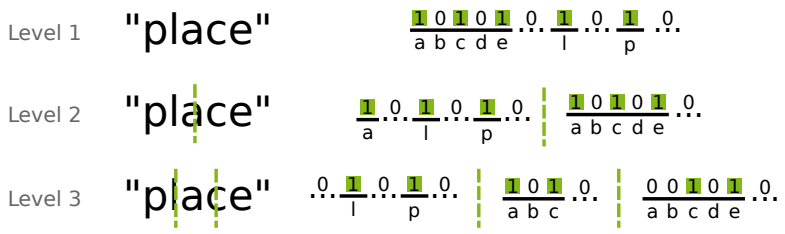
Idea: Project both textual and visual representation into a *common* space

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



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.

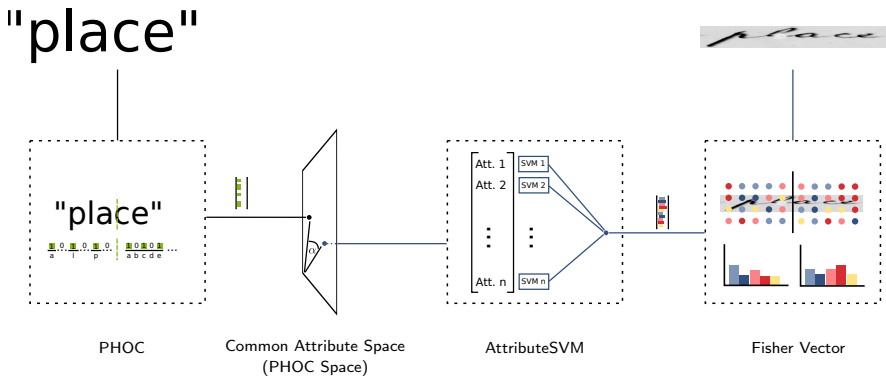
Pyramidal Histogram of Characters (PHOC)



- Concatenate histograms for all levels to form PHOC
- Levels used by Almazán *et al.*: 2,3,4 and 5
- 26 Characters + 10 Digits
- 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

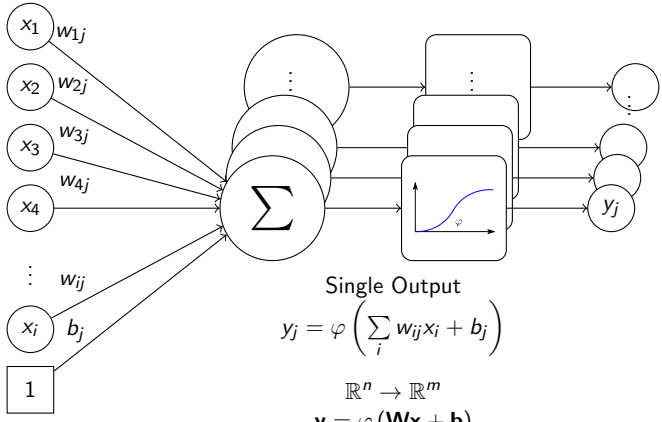


- AttributeSVM: ensemble of SVMs
- Each SVM predicts one attribute within the PHOC
- No end-to-end optimization possible!

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

The Perceptron



Single Output

$$y_j = \varphi\left(\sum_i w_{ij} x_i + b_j\right)$$

$$\mathbb{R}^n \to \mathbb{R}^m$$

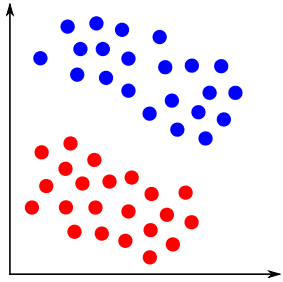
$$\mathbf{y} = \varphi\left(\mathbf{W}\mathbf{x} + \mathbf{b}\right)$$

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

Capabilities of the Perceptron

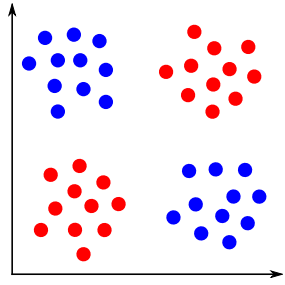
What a Perceptron can do:

Calssify two linearly separable classes



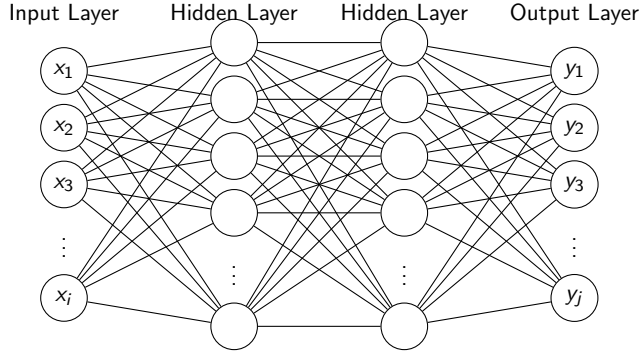
What a Perceptron can't do:

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



Solution: Stack layers of Perceptrons $\Rightarrow$ Multi Layer Perceptron

Multi Layer Perceptron (MLP)



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

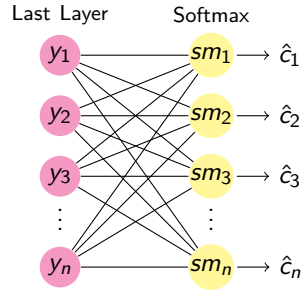
here: $\mathbf{y} = \mathbf{W}_{\text{out}} \cdot \varphi\left(\mathbf{W}_{h2} \cdot \varphi\left(\mathbf{W}_{h1}\mathbf{x} + b_{h1}\right) + b_{h2}\right) + b_{\text{out}}$

Classifying with MLPs

- For classification, the output of the MLP is *usually* forwarded through a Softmax Function:

$$sm_i(\mathbf{y}) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

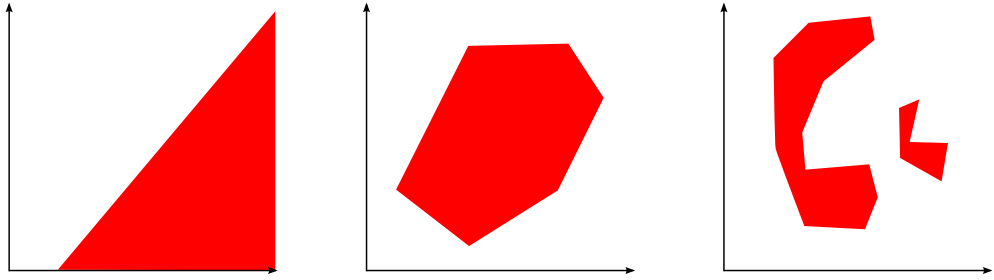
- Softmax can be seen as an additional layer
- sm_i is pseudo-probability for class c_i
- Predicted class: $\hat{c} = \max_i sm_i$



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):



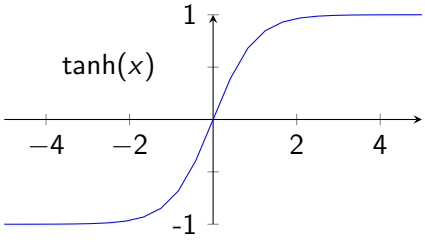
1. Layer: Halfspaces
2. Layer: Convex polyhedron
3. Layer: Multiple non-convex, non-connected polyhedra

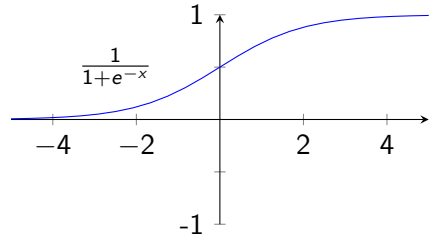
A Word on Activation Functions

- 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")





Training an MLP

How to determine weights such that desired functition is performed?

Basic Idea: Compare (computed) output of MLP

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

to *desired* output $\boldsymbol{y}$ and update weights such that $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$ become more similar.

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

- *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): $\epsilon_{\text{CE}} = -\sum_i y_i \log \hat{y}_i$ ‘

Method: Gradient Descent / Error Back-Propagation ... not in this tutorial ;-)

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 $\boldsymbol{\theta}$:

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

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

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

Minimum-Error and Maximum-Likelihood Training II

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

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

ML-Estimate is then given by:

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

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

$\Rightarrow$ 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 $N \to \infty$.
  (if true distribution lies within the model assumed)
- *Statistical Efficiency*: With increasing N, no other estimator produces lower MSE wrt. true θ.

$\Rightarrow$ ML Estimation often preferred in Machine Learning!

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

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

Example: RGB Image of 224×224 pixels (cf. ImageNET)

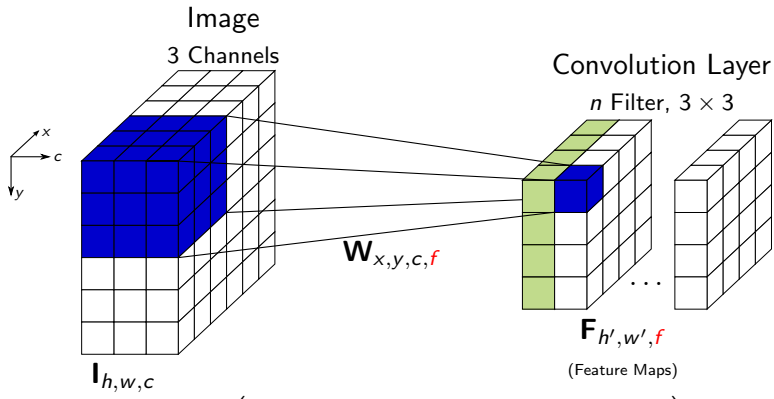
- e.g. 1024 neurons in first hidden MLP layer (small layer):

$\Rightarrow$ $224 \cdot 224 \cdot 3 \cdot 1024 \approx 1.5 \cdot 10^8$ weights in the first layer (approx. 1 GB)

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

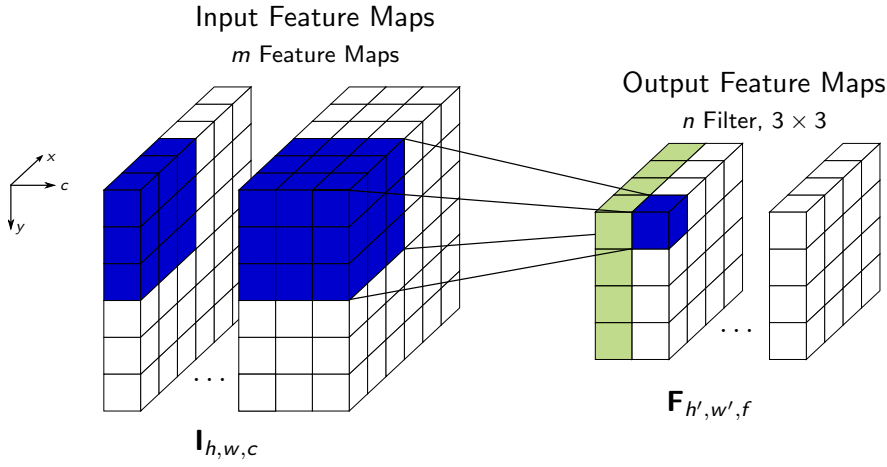
$\Rightarrow$ Convolutional Layer

Convolutional Layer

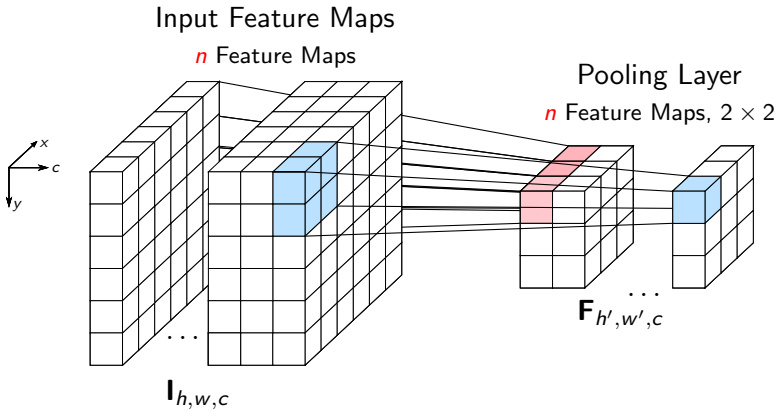


$$\mathbf{F}_{x,y,f} = \varphi\left(\sum_{c=1}^{K}\sum_{i=1}^{3}\sum_{j=1}^{3}\mathbf{W}_{i,j,c,f}\cdot\mathbf{I}_{x+i,y+j,c} + b_f\right)$$

Cascade of Convolutional Layers

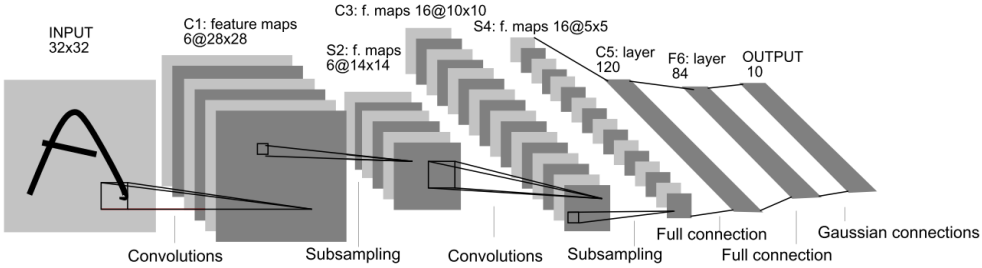


Pooling Layer



$$\mathbf{F}_{x,y,f} = \max_{i,j} \mathbf{I}_{x+i,y+j,f}$$

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)

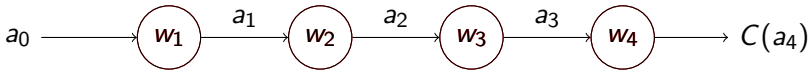
Technically: Deeper means more layers, not a deeper understanding

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

Why? Vanishing Gradient Problem

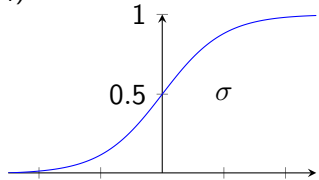
Vanishing Gradient Problem

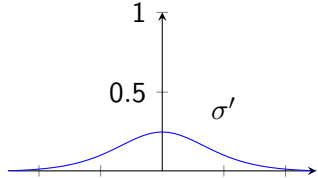
Four neuron network, 1D input, 1D output



$$z_i = w_i a_{i-1} + b_i \qquad a_i = \sigma(z_i)$$

$$\begin{aligned}
\frac{\partial C}{\partial w_1} &= \frac{\partial C}{\partial z_4}\frac{\partial z_4}{\partial a_3} \cdot \frac{\partial a_3}{\partial z_3}\frac{\partial z_3}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_2}\frac{\partial z_2}{\partial a_1} \cdot \frac{\partial a_1}{\partial z_1}\frac{\partial z_1}{\partial w_1} \\
&= \sigma'(z_4) w_4 \cdot \sigma'(z_3) w_3 \cdot \sigma'(z_2) w_2 \cdot \sigma'(z_1) a_0 \\
&= \sigma'(z_4)\sigma'(z_3)\sigma'(z_2)\sigma'(z_1) \cdot w_4 w_3 w_2 a_0 \\
&= \underbrace{\sigma'(z_4)\sigma'(z_3)\sigma'(z_2)\sigma'(z_1)}_{\leq \frac{1}{4}\cdot\frac{1}{4}\cdot\frac{1}{4}\cdot\frac{1}{4}} \cdot w_4 w_3 w_2 a_0 \\
&\leq \frac{1}{256} \cdot w_4 w_3 w_2 a_0
\end{aligned}$$



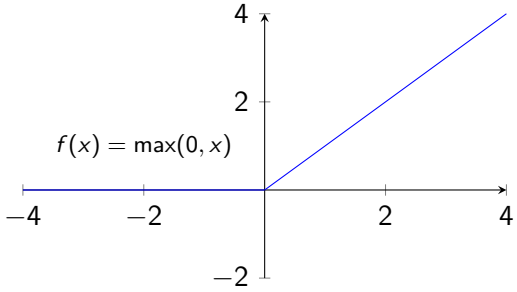


Vanishing Gradient Problem

- 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?

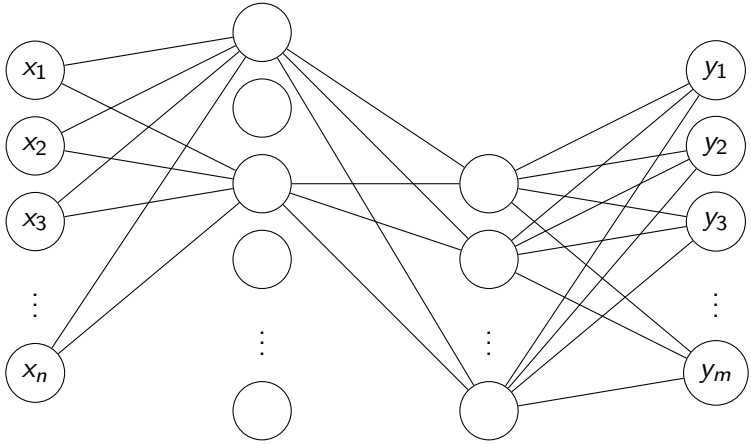
Problem: CNNs easily contain billions of parameters (weights)!
$\Rightarrow$ Could easily learn training samples "by heart".

Solution: Apply *Regularization* during training

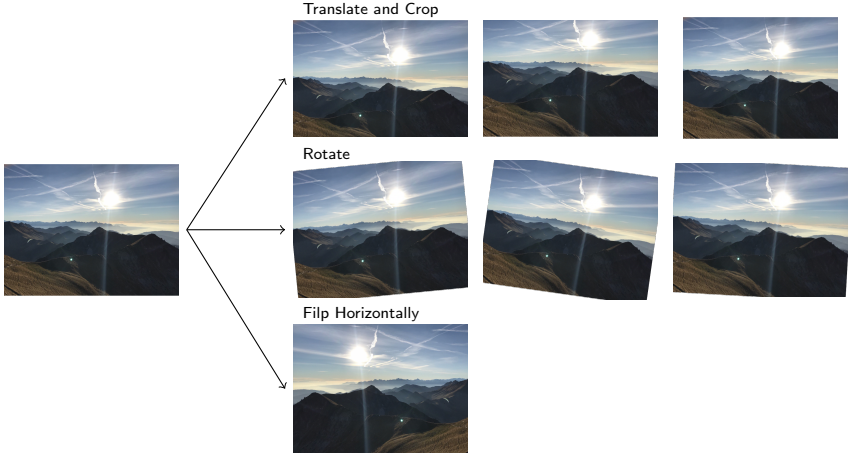
Fundamental Techniques:

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

Dropout

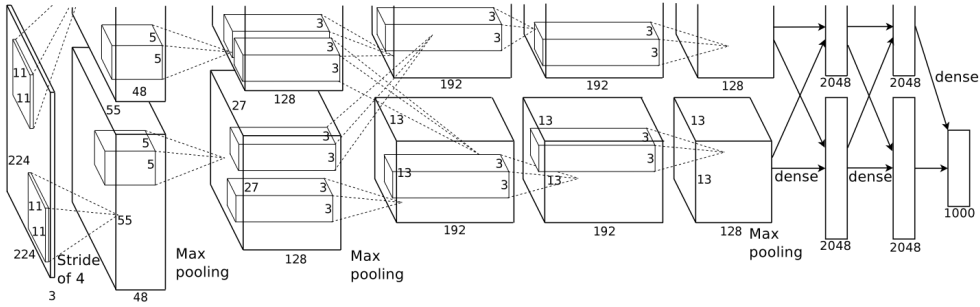


Data Augmentation



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

Well-known Deep Learning Architectures: AlexNet

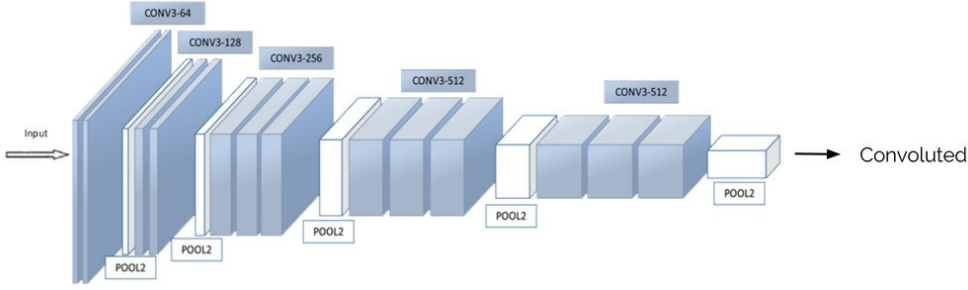


(Source: [Krizhevsky *et al.*, 2012])

- 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



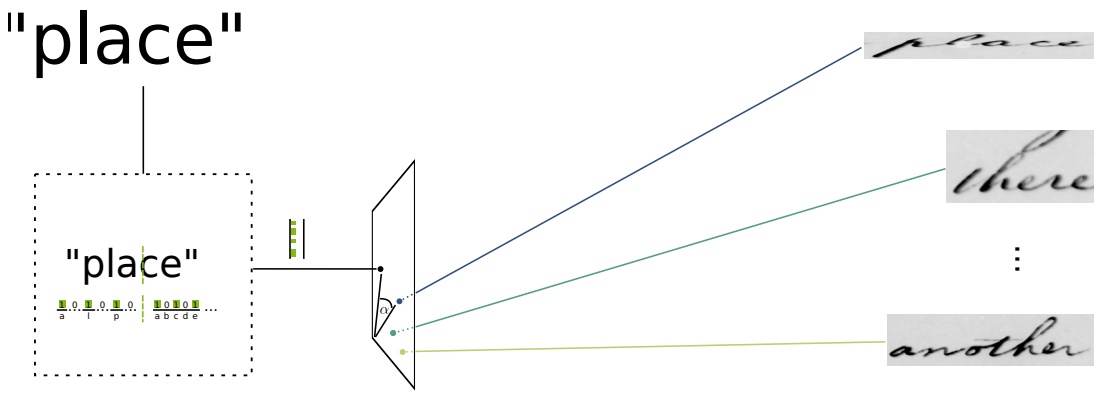
(Source: http://html.scirp.org/)

- 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

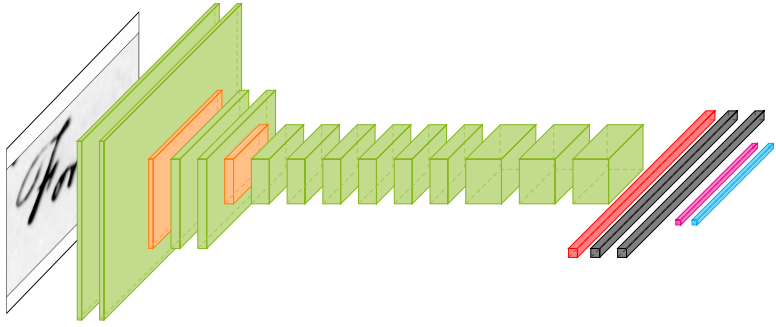
K. Simonyan, A. Zisserman: Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv, 2014.

Overview

Reminder: General Framework



PHOCNet

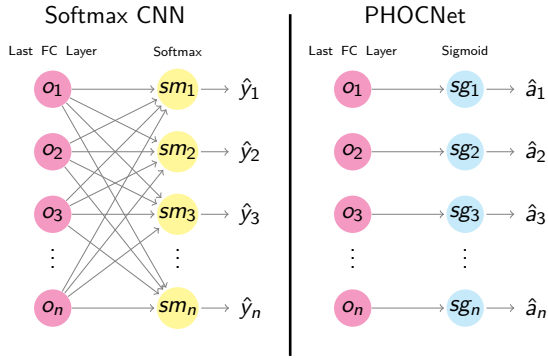


- 3×3 Convolutional Layer + ReLU
- 2×2 Max Pooling Layer
- 3-level Spatial Pyramid Max Pooling Layer
- Fully Connected Layer + ReLU and Dropout
- Fully Connected Layer + Linear Activation
- Sigmoid Activation

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

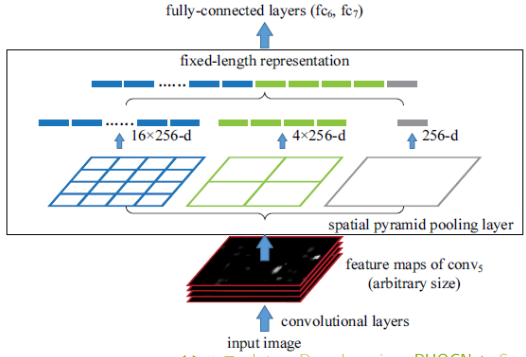
- In order to classify attributes, replace softmax with a sigmoid activation
- Each output neuron predicts one attribute



Spatial Pyramid Pooling Layer

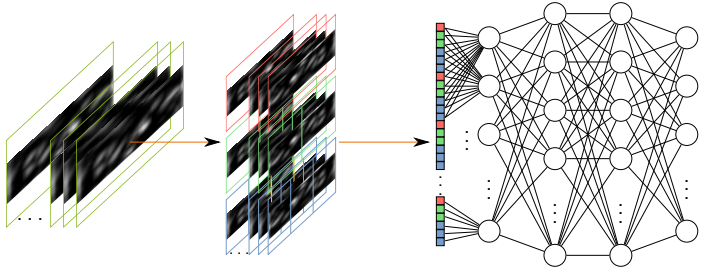
- Convolutional layers can already deal with arbitrary image sizes
- Only MLP part has a problem with changing image sizes

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



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

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

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

  (here: p corresponds to "success" probability, $k = 1$)
- Attributes exhibit dependencies but modeling these is prohibitive!
  $\Rightarrow$ 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:

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

Fink 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 dimensionalty, i.e., Euclidean distance becomes meaningless in high-dimensional spaces!

How can pair-wise independence assumtion be avoided?

$$\Rightarrow \text{Consider binary vectors as a whole!}$$

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

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

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

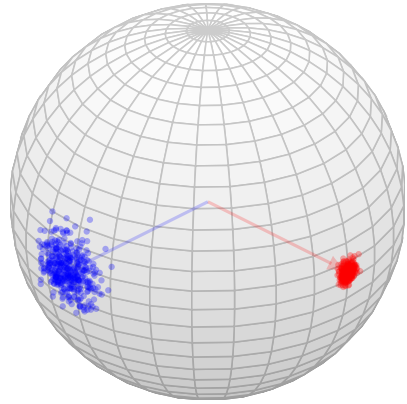
Training the PHOCNet: Loss Functions IV

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)

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

with $||\boldsymbol{\mu}|| = ||\boldsymbol{x}|| = 1$
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):

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

Advantage: Can cope with non-binary encodings (DCToW, SPOC)

Word Spotting: Impact of Deep Learning

Segmentation-based Word Spotting Performance in mAP [%]

Method	GW	IAM	Esposalles	IFN/ENIT	
TPP-PHOCNet	**97.92**	**93.42**	94.32	94.53	Deep Learning
PHOCNet	97.44	**91.12**	**94.89**	93.87	
Attribute SVM + FV [4]	91.29	73.72	–	–	Traditional Machine Learning
LSA Embedding [1]	56.54	–	–	–	
SC-HMM [33]	53.10	–	–	41.60	

Sudholt, S., Fink, G. A.: Attribute CNNs for Word Spotting in Handwritten Documents, Int. Journal on Document Analysis and Recognition, 2018.

Word Spotting: Recent Developments

Rothacker *et al.* 2017:: Word Hypotheses

(Segmentation-free, PHOCNet)

Wilkinson *et al.* 2017:: Neural CTRL-F

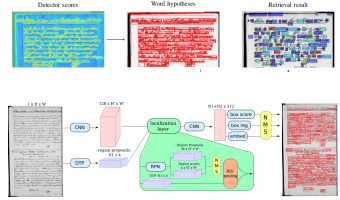
(Segmentation-free, Region Proposal Network)

Sánchez, Vidal *et al.* 2019:: Probabilistic Indexing

(CRNN, Large Scale Application)

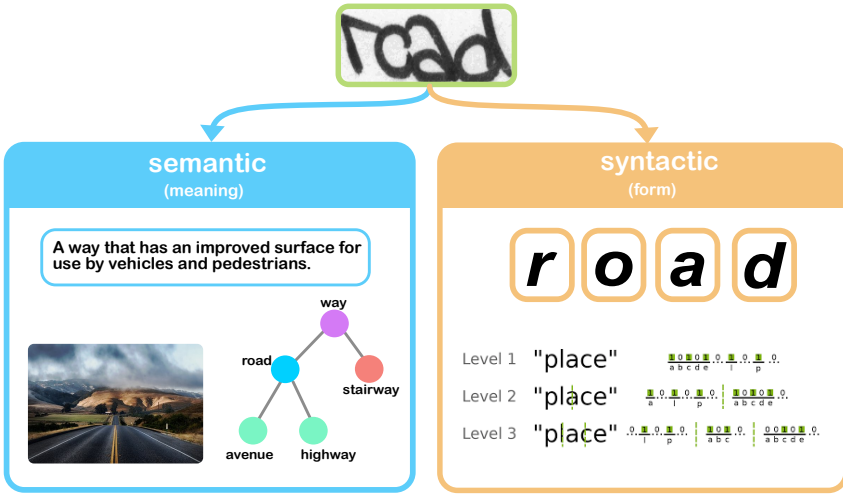
Krishnan *et al.* 2023: HWNet v3

(Word Embedding for retrieval and recognition, Synthetic data)

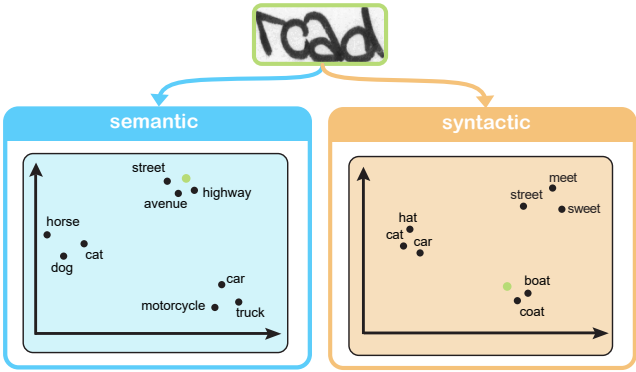


Overview

Semantic Word Spotting - Motivation

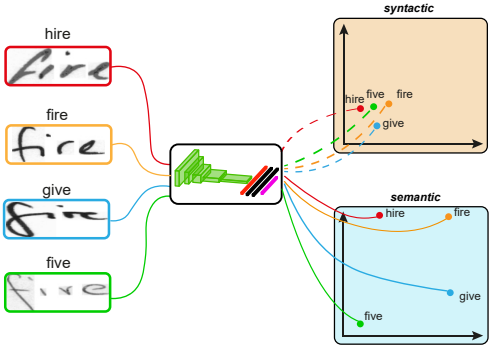


Semantic Word Spotting



- Methodology of semantic word spotting approaches is analogous to syntactic spotting methods
- Learning a mapping of word images to a textually pre-trained semantic word embedding space

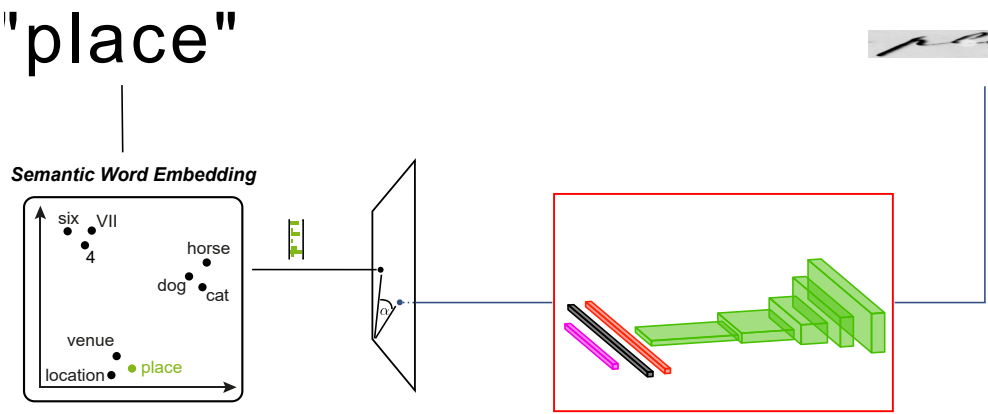
Key Challenges of Semantic Word Spotting



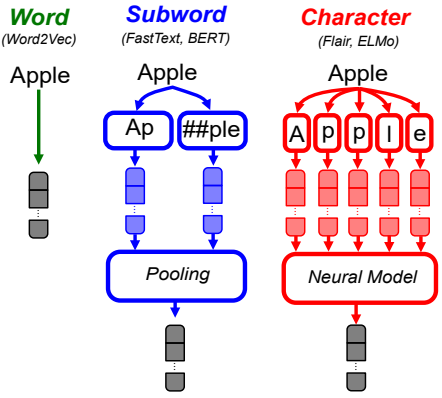
- Weak or no connection between form and semantics of words
- Visually similar word images have to be projected to:
  - different areas in semantic space
  - similar areas in syntactic space
- Hard to predict semantic representations for word images whose transcriptions were not part of training

O. Tueselmann, F. Wolf, G. A. Fink: Identifying and Tackling Key Challenges in Semantic Word Spotting, Proc. Int. Conf. on Frontiers in Handwriting Recognition, Dortmund, Germany, 2020, pp.55-60.

Semantic Word Spotting - Approach

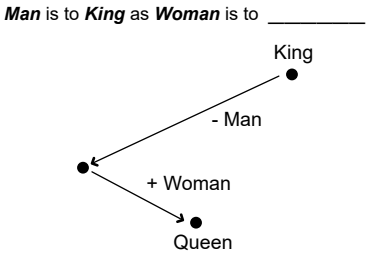


Evaluation of Semantic Word Embeddings



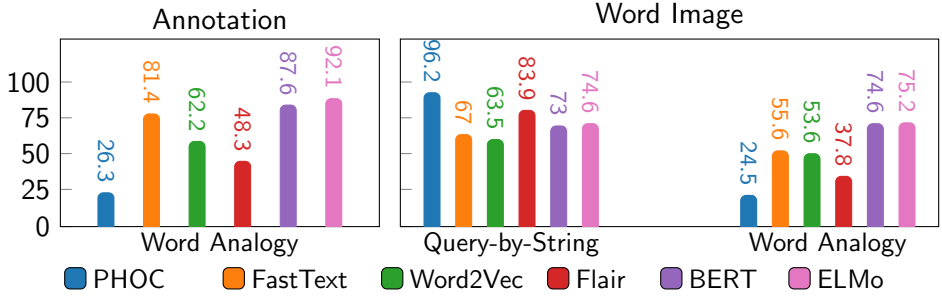
- Characteristics and concepts differ considerably between embedding approaches
- Question: Is there an approach that works best for handwritten word images?

Word Analogy



- Standard metric in NLP domain
- Predefined set of human-generated analogies: A is to B as C is to D
- Approach:
  - Calculate target position $\vec{d} = \vec{b} - \vec{a} + \vec{c}$
  - Determine nearest neighbor $\hat{D}$ for $\vec{d}$ w.r.t. all test word images
  - Analogy solved if annotation of word image $\hat{D}$ is equal to D

Semantic Word Embeddings

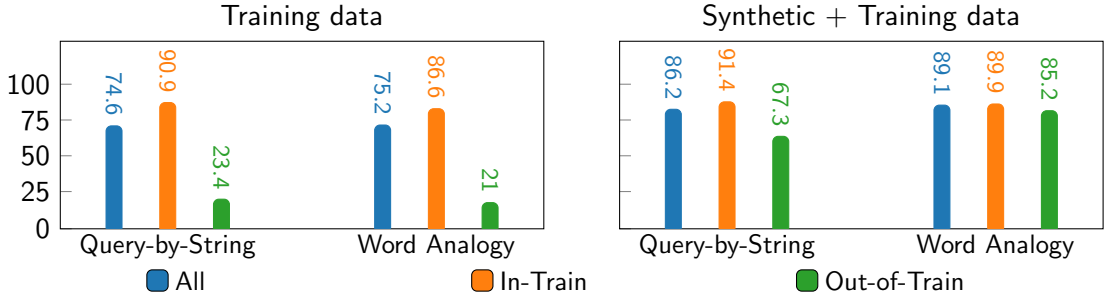


- Embedding selection is fundamental
- Static word embeddings from ELMo and BERT models provide best results

* Results for IAM-DB Benchmark

O. Tueselmann, G. A. Fink: Exploring Semantic Word Representations for Recognition-free NLP on Handwritten Document Images, Proc. Int. Conf. on Document Analysis and Recognition, San Jose, CA, USA, 2023, to appear.

Synthetic Word Images



- Word images with transcriptions that were not part of the training cause low performance
- Alleviation of unseen words by including synthetic word images during training

* Results for IAM-DB Benchmark

Weighted Combination of Semantic and Syntactic Representations

O. Tueselmann, K. Brandenbusch, M. Chen, G. A. Fink: A Weighted Combination of Semantic and Syntactic Word Image Representations, Proc. Int. Conf. on Frontiers in Handwriting Recognition, Hyderabad, India, 2022, pp.285-299.

Overview

What about Word Spotting today?

Question: How many publications on word spotting at ICDAR 2023?

Answer: *A single one!* $\Rightarrow$ Problem solved?

Application Area: Exploration of a large scale unknown document collection.

Problem :

- ✓ Well performing deep learning models
- ↯ Manually labeled data required

Taxonomy II

Reminder:

Query-by-example ⇔ Query-by-string

Segmentation-based ⇔ Segmentation-free

Extension:

(Learning-) Training-free ⇔ (Learning-) Training-based

- Is training a model really the limiting factor for an application?

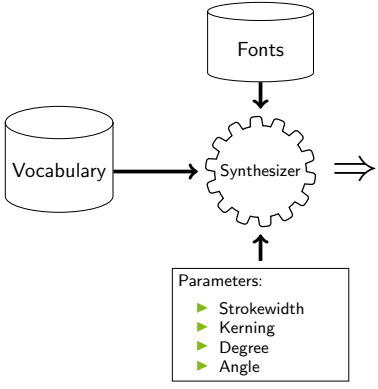
Annotation-free ⇔ Annotation-based

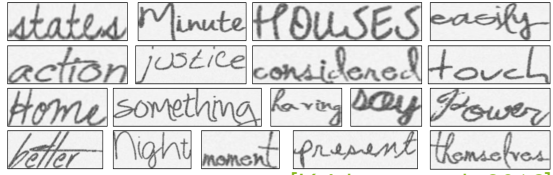
Leonard Rothacker, Fabian Wolf and Gernot A. Fink Annotation-free Word Spotting with Bag-of-Features HMMs, Int. Journal of Pattern Recognition and Artificial Intelligence, pp. 2153001, 2020.

Angelos P. Giotis, Giorgos Sfikas, Basilis Gatos and Christophoros Nikou: A survey of document image word spotting techniques, Pattern Recognition, pp. 310–332, 2017.

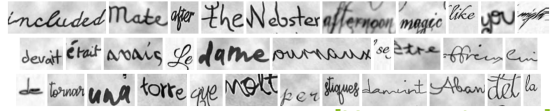
How to reduce annotation effort: Make your own data!

- Text corpora easily available
- Large variety of fonts to render images





[Krishnan et al. 2016]

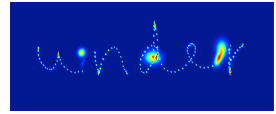


[Kang et al. 2020]

Synthetic Data for Handwritten Documents

Graves *et al.* 2013: Handwriting Synthesis

(Online data, Autoregressive model, RNN)

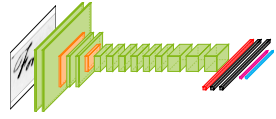


Krishnan *et al.* 2016, 2019, 2023: IIIT-HWS Dataset

(Word Images, True Type Fonts)

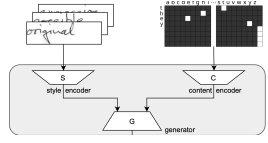
Gurjar *et al.* 2018: PHOCNet on Synthetic Data

(Pretrain on IIIT-HWS, Finetuneing, Drastic reduction of training data)

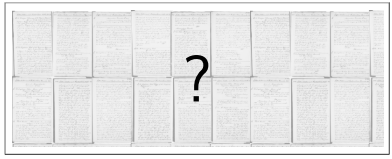


Kang *et al.* 2020, Mattick 2021 *et al.*, *and more...* GANs

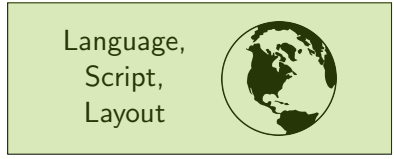
(Often need training data, Limited performance gains)



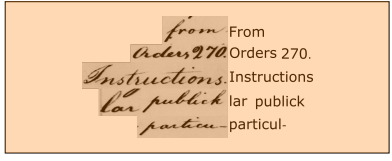
Data Situation



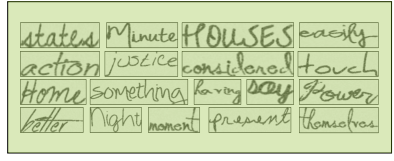
Unlabeled Document Collection



Domain Knowledge



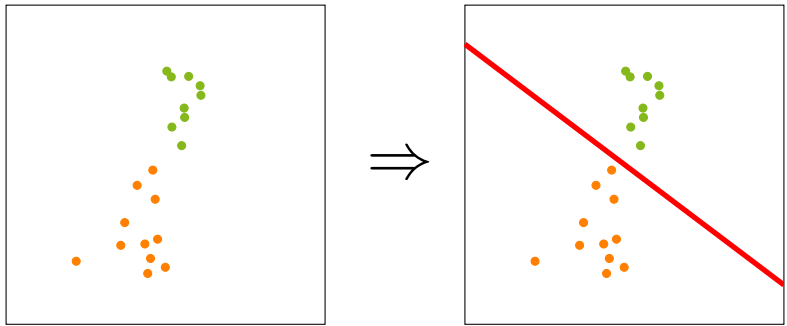
Labeled Data



Synthetic Data

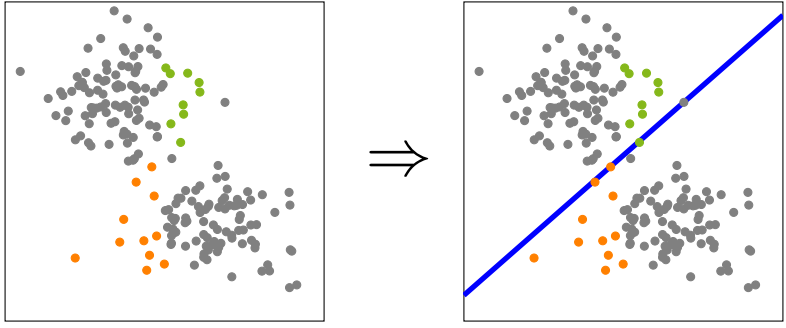
Why should learning from unlabeled data help?

Question: Where would you put a decision boundary?



Why should learning from unlabeled data help?

Question: Is it still the same?



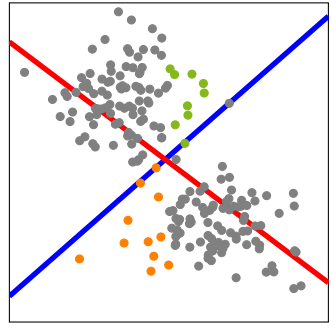
Why should learning from unlabeled data help?

Question: Is the second decision boundary better?

Answer: Yes, but only if the following holds true

- *Cluster Assumption*: Feature vectors belonging to the same class form clusters in the feature space. If two points $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ fall in the same cluster they share the same class label.
- *Smoothness Assumption*: If two points $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ are close to each other in the feature space, then the outputs should be also close or, in case of classification, the label should be the same.

$\Rightarrow$ Low density seperation



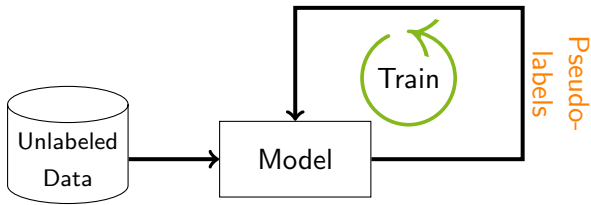
Entropy Regularization

Idea: Enforce low entropy $\Rightarrow$ *Cluster Assumption*

Pseudo-Labeling: Iteratively, use model to label data points for training.

Theory: Training on pseudo-labels constitutes a form of entropy regularization.

[Chapelle *et al.*, 2006]



A brief history of Self-Training

Self-Training: Self-learning, self-labeling, pseudo-labeling, decision directed learning

Self-Supervision: Representation learning without any annotated-labels

Scudder *et al.* 1965, Fralick *et al.* 1965: Old Concept

(Empirical results for probably every ML model, diverse applications)

Lee *et al.* 2013: First application to neural network

(MNIST, NN with one hidden layer)

Berthelot, Sohn, *et al.* 2019,2020: Mix-, ReMix-, FixMatch

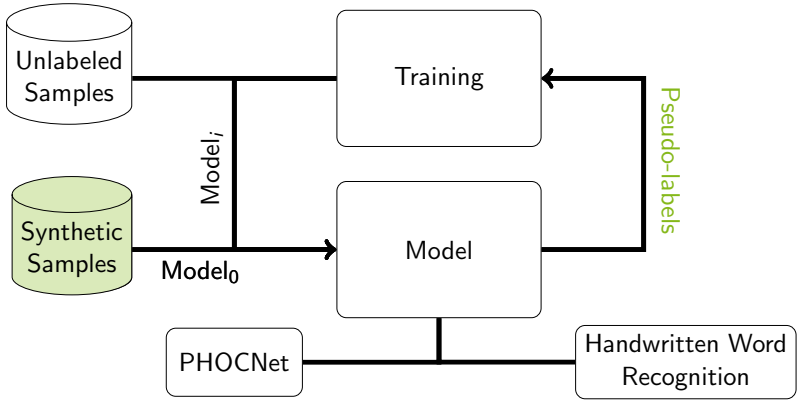
(Object classification, Augmentations, Regularization)

H. J. Scudder III. Probability of error of some adaptive pattern-recognition machines, IEEE Trans. Inf. Theory, pp. 363–371, 1965.

D. Lee. Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks, ICML Workshop on Challenges in Representation Learning, 2013.

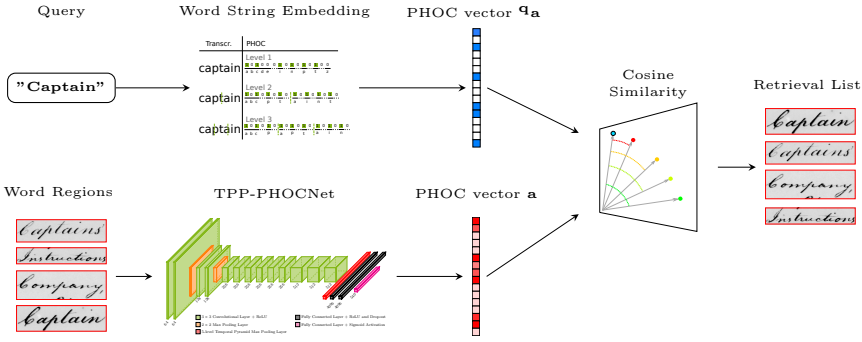
David Berthelot and Nicholas Carlini and Ian J. Goodfellow and Nicolas Papernot and Avital Oliver and Colin Raffel MixMatch: A Holistic Approach to Semi-Supervised Learning, NIPS, pp. 5050–5060, 2019.

Concept Self-Training



Self-Training the PHOCNet

Reminder:



Question: How to derive a pseudo-label? Predicted PHOC vector?

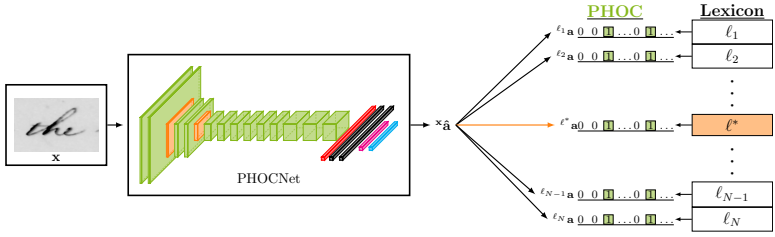
Pseudo-labeling: Exploiting Domain Knowledge

Word Recognition: Lexicon as domain knowledge

- Predict PHOC vector for input image $\boldsymbol{x}$: $\text{PHOCNet}(\boldsymbol{x}) = {}^{\mathsf{x}}\hat{\mathbf{a}}$
- Derive PHOCs for lexicon: ${}^{\ell_i}\mathbf{a}$
- Recognition result:

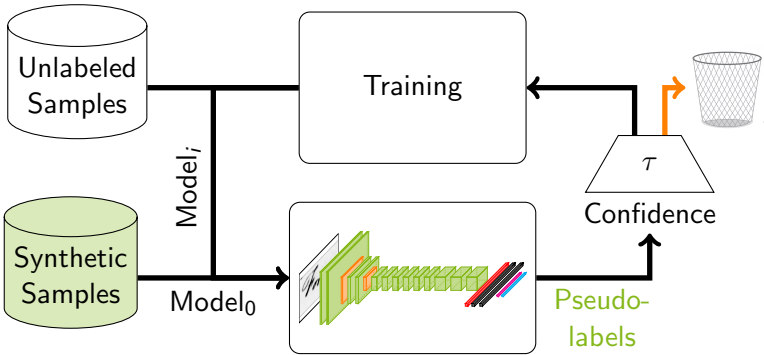
$$\ell^* = \underset{\ell \in \mathbb{L}}{\operatorname{argmin}}\, d_{cos}\left({}^{\ell}\mathbf{a}, {}^{\mathsf{x}}\hat{\mathbf{a}}\right).$$

Lexicon: No need to be perfect! Only for pseudo-label generation



Fabian Wolf and Gernot A. Fink Annotation-free Learning of Deep Representations for Word Spotting using Synthetic Data and Self Labeling, Proc. of the Int. Workshop on Document Analysis Systems (DAS) vol. 12116, pp. 293–308, 2020.

Self-Training the PHOCNet II



- Pseudo-labels are error prone!
- Initital model has only seen synthetic data
- Remove bad predictions from training?

$\Rightarrow$ Apply suitable confidence measure!

Quantitative Results

Method	Ann.-Free	GW	IAM	BT14	BT15
Synthetic Training	✓	74.7	57.8	79.8	70.1
Self-Training	✓	93.8	80.6	**96.4**	77.6
Self-Training (Confidence)	✓	**93.9**	**81.2**	95.2	**88.8**
Fisher Vectors [3]	✓	62.7	15.6	–	–
Zagoris et al. [50]	✓	–	–	60.0	50.1
Retsinas et al. [31]	✓	77.1	28.1	71.1	58.4
TPP-PHOCNet [42]		97.9	84.8	–	–
HWNet v3 [19]		99.5	93.2	–	–
Retsinas et al. [32]		97.9	92.0	–	–

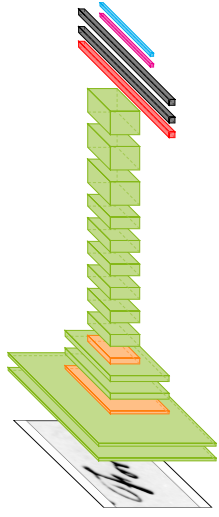
Evaluation results for query-by-example given in mAP

Note: *Our* annotation-free method can perform query-by-string, too!

Overview

Summary

- Deep Learning has become the dominant methodology in Computer Vision *and* Document Image Analsis
- CNNs are (still) the most popular deep networks
  - ⇒ *Especially suitable for (document) images!*
- The most successful Word Spotting methods are *learning-based*, i.e., use Deep Neural Networks
  - Specialized word representations allow to incorporate semantic information
  - Annotation requirements can be reduced *substantially* by using data synthesis and self-training
- More information: *Ask the experts in the audience ;-)*



References I

[1] David Aldavert, Marcal Rusinol, Ricardo Toledo, and Josep Llados.
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] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel.
MixMatch: A holistic approach to semi-supervised learning.
In *NIPS*, pages 5050–5060, Vancouver, BC, Canada, 2019.

[6] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors.
Semi-Supervised Learning.
The MIT Press, 2006.

References III

[7] Angelos P. Giotis, Giorgos Sfikas, Basilis Gatos, and Christophoros Nikou.
A survey of document image word spotting techniques.
Pattern Recognition, 68:310 – 332, 2017.

[8] Xavier Glorot and Yoshua Bengio.
Understanding the difficulty of training deep feedforward neural networks.
AISTATS, 9:249–256, 2010.

[9] Phillip Good.
Permutation Tests - A Practical Guide to Resampling Methods for Testing Hypothesis.
Springer, 2 edition, 2000.

References IV

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
Deep Learning.
The MIT Press, 2016.

[11] Alex Graves.
Generating sequences with recurrent neural networks.
CoRR, abs/1308.0850, 2013.

[12] 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.

References V

[13] 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.

[14] Kurt Hornik, Maxwell Stinchcombe, and Halbert White.
Multilayer feedforward networks are universal approximators.
Neural Networks, 2(5):359–366, 1989.

[15] H. J. Scudder III.
Probability of error of some adaptive pattern-recognition machines.
IEEE Trans. Inf. Theory, 11(3):363–371, 1965.

References VI

[16] Lei Kang, Pau Riba, Marçal Rusiñol, Alicia Fornés, and Mauricio Villegas.
Distilling content from style for handwritten word recognition.
In *Proc. Int. Conf. on Frontiers in Handwriting Recognition*, pages 139–144, Dortmund, Germany (virtual), 2020.

[17] Lei Kang, Marçal Rusinol, Alicia Fornés, Pau Riba, and Mauricio Villegas.
Unsupervised writer adaptation for synthetic-to-real handwritten word recognition.
In *WACV*, pages 3502–3511, Snow Mass Village, CO, USA, 2020.

[18] 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.

References VII

[19] Praveen Krishnan, Kartik Dutta, and CV Jawahar.
Hwnet v3: a joint embedding framework for recognition and retrieval of handwritten text.
Int. Journal on Document Analysis and Recognition, pages 1–17, 2023.

[20] Praveen Krishnan and C. V. Jawahar.
Generating synthetic data for text recognition.
CoRR, abs/1608.04224, 2016.

[21] Praveen Krishnan and C. V. Jawahar.
Hwnet v2: an efficient word image representation for handwritten documents.
Int. Journal on Document Analysis and Recognition, 22(4):387–405, 2019.

References VIII

[22] 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.

[23] 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 IX

[24] D. Lee.
Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks.
In *ICML Workshop on Challenges in Representation Learning*, Atlanta, GA, USA, 2013.

[25] 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.

References X

[26] 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.

[27] 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.

[28] 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'Ecrit et le Document*, Hammamet, Tunisia, October 2002.

References XI

[29] 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.

[30] 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.

[31] G. Retsinas, G. Louloudis, N. Stamatopoulos, and B. Gatos.
Efficient learning-free keyword spotting.
IEEE Trans. on Pattern Analysis and Machine Intelligence, 41(7):1587–1600, 2019.

References XII

[32] George Retsinas, Giorgos Sfikas, Christophoros Nikou, and Petros Maragos.
From seq2seq recognition to handwritten word embeddings.
In *British Machine Vision Conference*, page 98, Virtual, 2021.

[33] José A. Rodríguez-Serrano and Florent Perronnin.
A model-based sequence similarity with application to handwritten word spotting.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(11):2108–2120, 2012.

[34] 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 XIII

[35] F. Rosenblatt.
The perceptron: A probabilistic model for information storage and organization in the brain.
Psychological Review, 65(6):386–408, 1958.

[36] Leonard Rothacker, Sebastian Sudholt, Eugen Rusakov, Matthias Kasperidus, and Gernot A. Fink.
Word hypotheses for segmentation-free word spotting in historic document images.
In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 1174–1179, Kyoto, Japan, 2017.

[37] Leonard Rothacker, Fabian Wolf, and Gernot A. Fink.
Annotation-free word spotting with bag-of-features hmms.
Int. Journal of Pattern Recognition and Artificial Intelligence, 35(4):2153001, 2020.

References XIV

[38] M. Rusiñol, D. Aldavert, R. Toledo, and J. Llados.
Browsing heterogeneous document collections by a segmentation-free word spotting method.
In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 63 –67, Bejing, China, 2011.

[39] Karen Simonyan and Andrew Zisserman.
Very deep convolutional networks for large-scale image recognition.
arXiv, pages 1–13, 2014.

References XV

[40] 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.

[41] 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.

[42] 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.

References XVI

[43] Alejandro Héctor Toselli, Verónica Romero-Gomez, Joan-Andreu Sánchez, and Enrique Vidal-Ruiz.
Making two vast historical manuscript collections searchable and extracting meaningful textual features through large-scale probabilistic indexing.
In *Proc. Int. Conf. on Document Analysis and Recognition*, pages 108–113, Sydney, NSW, Australia, 2019.

[44] Oliver Tueselmann, Kai Brandenbusch, Miao Chen, and Gernot A. Fink.
A Weighted Combination of Semantic and Syntactic Word Image Representations.
In *Proc. Int. Conf. on Frontiers in Handwriting Recognition*, pages 285–299, Hyderabad, India, 2022.

References XVII

[45] Oliver Tueselmann, Fabian Wolf, and Gernot A. Fink.
Identifying and tackling key challenges in semantic word spotting.
In *Proc. Int. Conf. on Frontiers in Handwriting Recognition*, pages 55–60, Dortmund, Germany (virtual), 2020. IEEE.

[46] Oliver Tüselmann and Gernot A. Fink.
Exploring semantic word representations for recognition-free nlp on handwritten document images.
In *Proc. Int. Conf. on Document Analysis and Recognition*, San Jose, CA, USA, 2023.

References XVIII

[47] 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.

[48] Tomas Wilkinson, Jonas Lindström, and Anders Brun.
Neural ctrl-f: Segmentation-free query-by-string word spotting in handwritten manuscript collections.
In *Proc. Int. Conf. on Computer Vision*, pages 4443–4452, Venice, Italy, 2017.

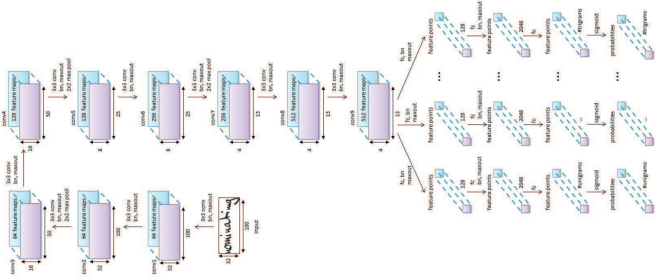
References XIX

[49] Fabian Wolf and Gernot A. Fink.
Annotation-free learning of deep representations for word spotting using synthetic data and self labeling.
In *Proc. Int. Workshop on Document Analysis Systems*, volume 12116 of *Lecture Notes in Computer Science*, pages 293–308, Wuhan, China (virtual), July 2020.
Winner of the Nakano Best Paper Award.

[50] K. Zagoris, I. Pratikakis, and B. Gatos.
Unsupervised word spotting in historical handwritten document images using document-oriented local features.
IEEE Trans. on Pattern Analysis and Machine Intelligence, 26(8):4032–4041, 2017.

Related Work on Deep Learning

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



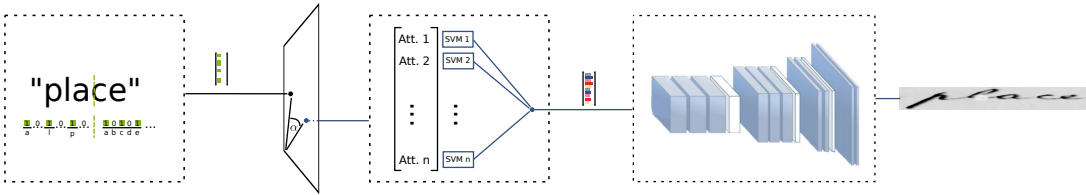
- Considers isolated word *recognition*
- Uses PHOC + 1 level of trigrams
- Attributes are predicted directly with *separate* MLPs

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]



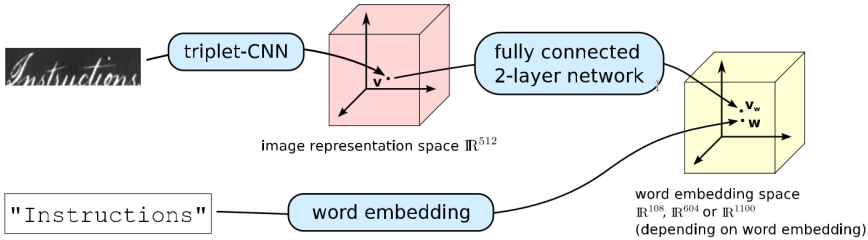
- Uses CNN to produce feature representation of images
- CNN is pre-trained on synthetically generated data
- SVMs predict attributes of PHOC representation

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]

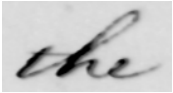


- Residual network learns word descriptors using triplet loss
- Separate MLP predicts attribute representation
- Proposed *DCT of Words* embedding

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

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.
- Transform result into relevant / non-relevant list: [0, 0, 1, 1, 0, 0]

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]):

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

  $\text{rel}(k)$: Relvance of item k, Precision_k: Precision at cut-off k

  Accumulate Precision whenever the Recall changes and normalize.

Word Spotting: Evaluation III

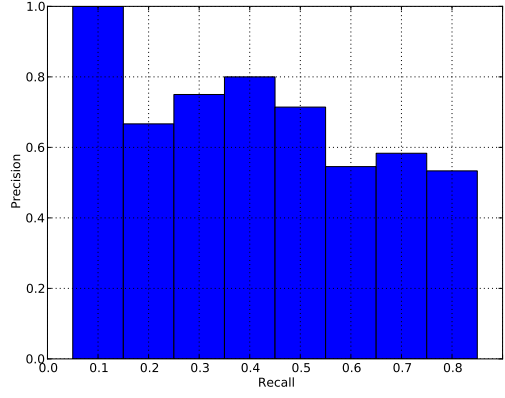
$$[1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1]$$

Average Precision:

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

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

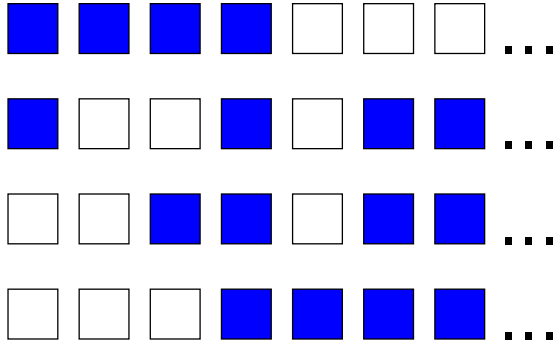
$$\approx 0.56$$



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.

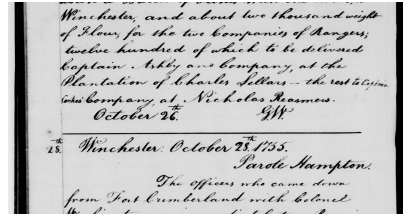
Result: Proportion of differences $T_n \geq T_{\text{obs}}$ is Probability (p-value) of accepting the null hypothesis.

$\Rightarrow$ 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

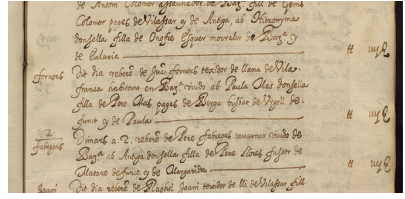
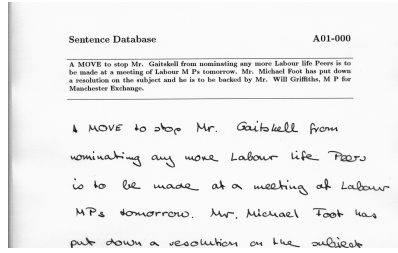


Image sources:
The George Washington Papers at the Library of Congress, 1741-1799
V. Romero *et al.*: *The ESPOSALLES Database: An Ancient Marriage License Corpus for Off-line Handwriting Recognition*, Pattern Recognition, Volume 46(6), pp. 1658-1669, 2013.

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)

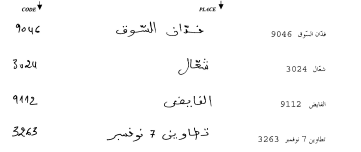


Image sources:
U. V. Marti, H. Bunke: *The IAM-Database: An English Sentence Database for Offline Handwriting Recognition*, IJDAR, vol. 5, pp. 39-46, 2002.
M. Pechwitz, S. Snoussi Maddouri, V. Märgner, N. Ellouze, H. Amiri: *IFN/ENIT-Database of handwritten Arabic words*, Proc. 7th Colloque International Francophone sur l'Ecrit et le Document, Hammamet, Tunisia, 2002.

Evaluation: Procedure & Protocol

- Consider only Qery-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 $\boldsymbol{a}^q$ can be given directly
- PHOCNet predicts PHOC $\hat{\boldsymbol{a}}$ for each word image in the test set
- Test word images are ranked according to cosine dissimilarity to query

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