On-the-fly Deformations for Keyword Spotting

George Retsinas, Giorgos Sfikas, Basilis Gatos and Christophoros Nikou
National Technical University of Athens, NCSR Demokritos & University of Ioannina
gretsinas@central.ntua.gr, sfikas@cs.uoi.gr, bgat@iit.demokritos.gr, cnikou@cs.uoi.gr

Motivation

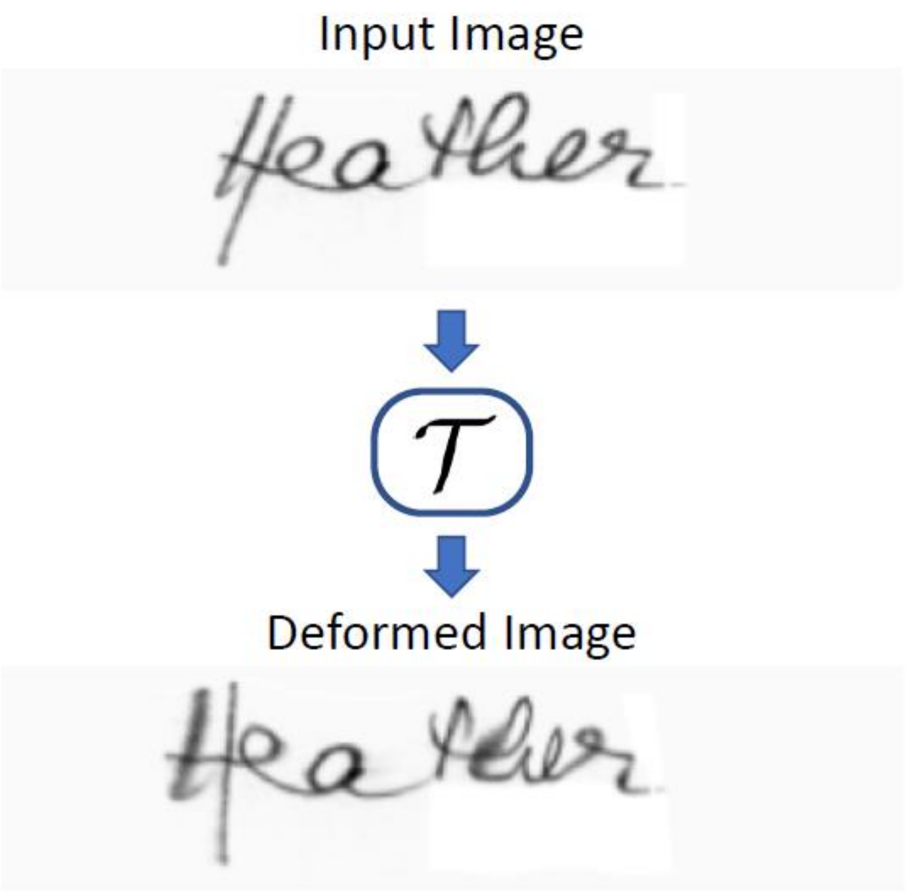
Task: Keyword Spotting on <u>segmented</u> word images

Motivation: Transform/deform images to be as close to query as possible

Motivation

Task: Keyword Spotting on segmented word images

Motivation: Transform/deform images to be as close to query as possible



Overview of the proposed method:

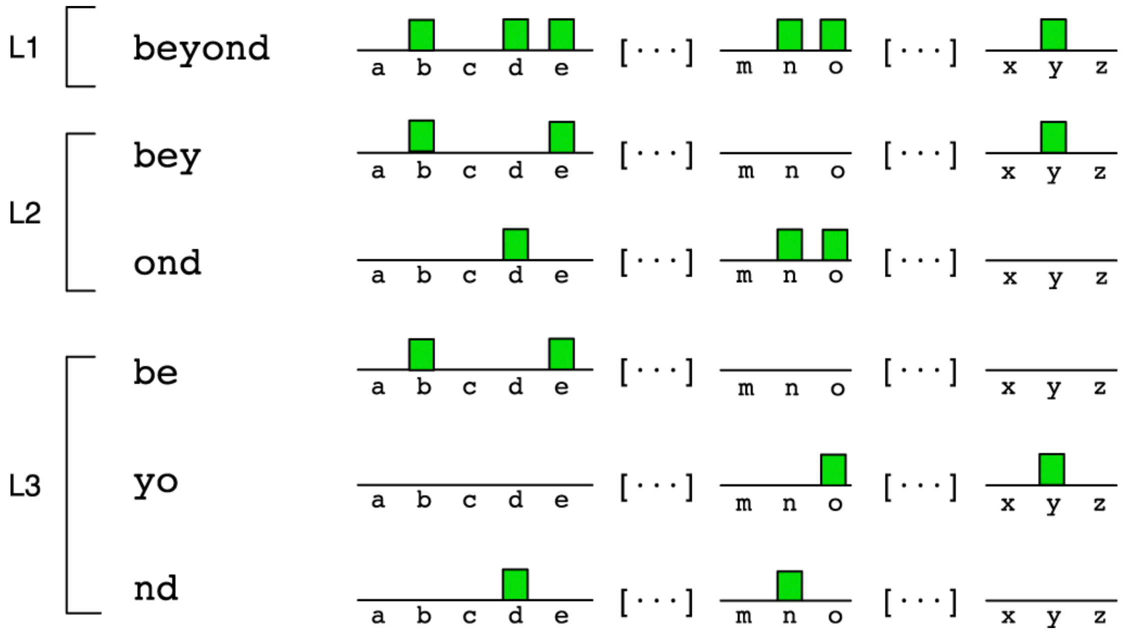
Iteratively deform an image in order to minimize its distance to a query image w.r.t. a feature space.

Features are extracted from a DNN!

Network Architecture

PHOCNet* alternative:

✓ Targets are PHOC (Pyramidal Histogram of Characters) representations



*S. Sudholt et al., “PHOCNet: A deep convolutional neural network for word spotting in handwritten documents”, ICFHR, 2016

Network Architecture

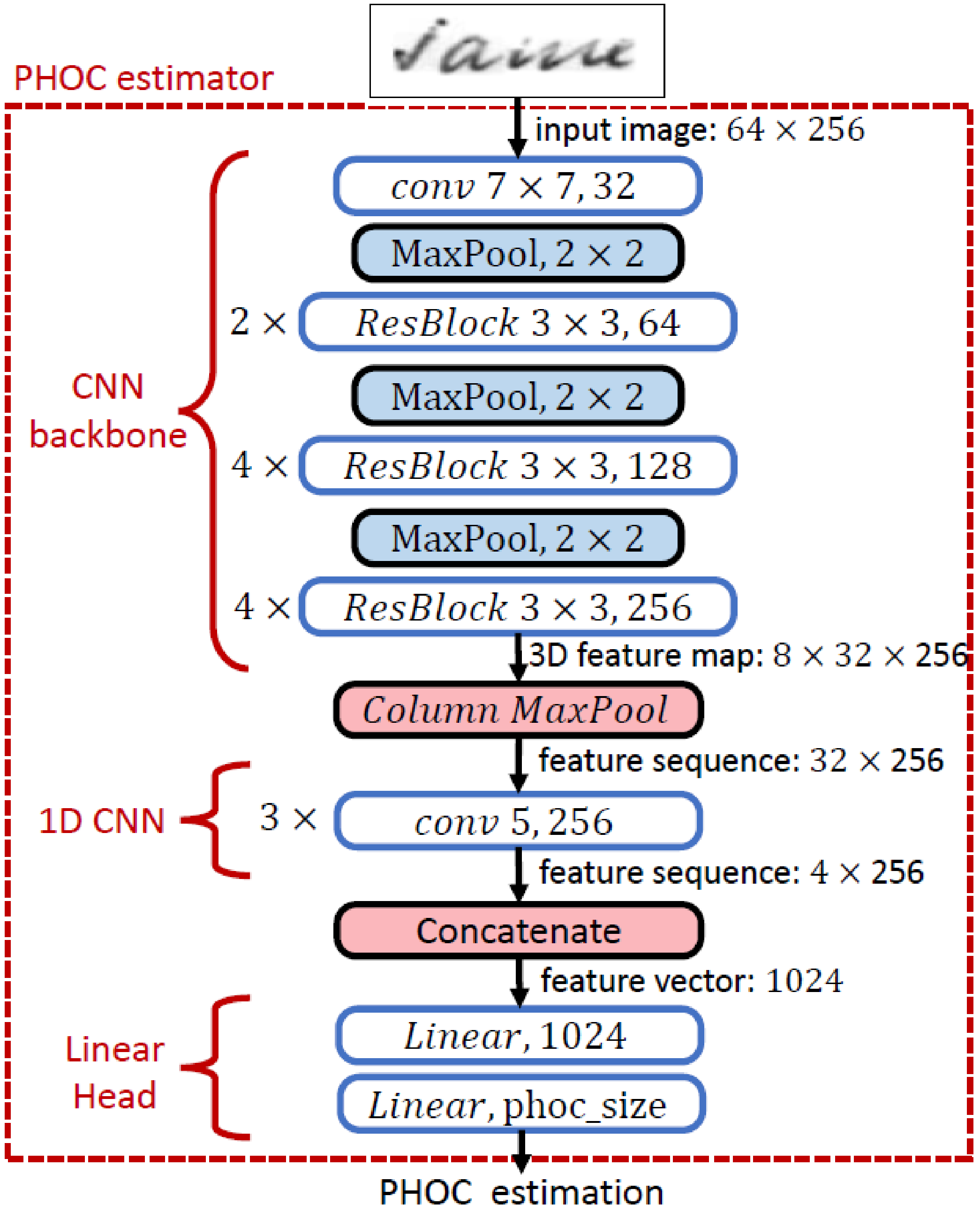
PHOCNet* alternative:

- ✓ Targets are PHOC (Pyramidal Histogram of Characters) representations
- ✓ Architecture:
  - ResNet-based CNN backbone
  - Column-wise max-pooling
  - 1D CNN for encoding temporal information
  - linear head for predicting PHOCs
- ✓ Compact architecture: ~8M parameters

Training Details:

- ✓ BCE loss
- ✓ Adam optimizer (lr=0.001) with multistep scheduler.

*S. Sudholt et al., "PHOCNet: A deep convolutional neural network for word spotting in handwritten documents", ICFHR, 2016



Deformations

Considered Deformations:

- Global Affine
  - *3×2 transformation matrix*
- Local Affine
  - *split image along x-axis*
  - *apply an affine transformation to each part*
  - *bilinear interpolation of local affine parameters for consistency*
- Local Deformation
  - *x,y translation vectors over 8×8 image patches*

Deformations

Considered Deformations:

- Global Affine
  - *3 × 2 transformation matrix*
- Local Affine
  - *split image along x-axis*
  - *apply an affine transformation to each part*
  - *bilinear interpolation of local affine parameters for consistency*
- Local Deformation
  - *x,y translation vectors over 8 × 8 image patches*

Deformation parameters were selected in order to clearly show the effect of the deformations without significantly distorting the image

Query-based Deformation

$$\mathbf{I}' = \mathcal{T}(\mathbf{I}; \mathbf{d})$$

Initial Image

Deformation Parameters

$$\{\mathbf{d}_{ga}, \mathbf{d}_{la}, \mathbf{d}_{ld}\}$$

Query-based Deformation

$$\mathbf{I}' = \mathcal{T}(\mathbf{I}; \mathbf{d})$$

Initial Image

Deformation Parameters

$$\{\mathbf{d}_{ga}, \mathbf{d}_{la}, \mathbf{d}_{ld}\}$$

Global Affine

Local Affine

Local Deformation

Query-based Deformation

$$\mathbf{I}' = \mathcal{T}(\mathbf{I}; \mathbf{d})$$

Initial Image

Transformation Function

STN (Spatial Transformer Network) formulation

Transformed image computed as a grid-based interpolation

Deformation Parameters

$$\{\mathbf{d}_{ga}, \mathbf{d}_{la}, \mathbf{d}_{ld}\}$$

Global Affine

Local Affine

Local Deformation

Query-based Deformation

$$\mathbf{I}' = \mathcal{T}(\mathbf{I}; \mathbf{d})$$

Transformed Image

Initial Image

Transformation Function

STN (Spatial Transformer Network) formulation
Transformed image computed as a grid-based interpolation

Deformation Parameters

$$\{\mathbf{d}_{ga}, \mathbf{d}_{la}, \mathbf{d}_{ld}\}$$

Global Affine

Local Affine

Local Deformation

Query-based Deformation

Our problem is then formulated as the maximization of:

$$S_C(f(\mathcal{T}(\mathbf{I}_w; \mathbf{d})), f(\mathbf{I}_q))$$

Cosine Similarity

Word Image

Query Image
Template

✓ Compare the features of the ***transformed word image*** and the **template query image**

Query-based Deformation

Our problem is then formulated as the maximization of:

$$S_C(f(\mathcal{T}(\mathbf{I}_w;\mathbf{d})), f(\mathbf{I}_q))$$

Cosine Similarity

Word Image

Feature Extractor

Deep Features

Query Image

Template

✓ Compare the features of the ***transformed word image*** and the **template query image**

Query-based Deformation

Our problem is then formulated as the maximization of:

$$S_C(f(\mathcal{T}(\mathbf{I}_w;\mathbf{d})), f(\mathbf{I}_q))$$

Cosine Similarity

Word Image

Feature Extractor
Deep Features

Query Image
Template

- ✓ Compare the features of the ***transformed word image*** and the **template query image**
- ✓ Optimize w.r.t. **deformation parameters d**

Query-based Deformation

Our problem is then formulated as the maximization of:

$$S_C(f(\mathcal{T}(\mathbf{I}_w;\mathbf{d})), f(\mathbf{I}_q))$$

Cosine Similarity

Word Image

Feature Extractor
Deep Features

Query Image
Template

✓ Compare the features of the ***transformed word image*** and the **template query image**
✓ Optimize w.r.t. **deformation parameters d**

HOW?

- Deep features are extracted from the output of the 1D CNN component
- Optimize via gradient descent (Adam optimizer)
- NN weights are kept fixed
- Update only deformation parameters
- CONSTRAINTS NEEDED! *(unconstrained optimization may considerably distort images)*

Proposed Loss (to be minimized):

$$\mathcal{L}(\mathbf{d}) = 1 - \overbrace{\boxed{S_C(f(\mathcal{T}(\mathbf{I}_w;\mathbf{d})), f(\mathbf{I}_q))}}^{\text{Feature Similarity Loss}} + \underbrace{\boxed{a\|\mathcal{T}(\mathbf{I}_w;\mathbf{d}) - \mathbf{I}_q\|_2 + b\|\mathbf{d}\|_2}}_{\text{extra constraint terms}}$$

a, b : *user-defined hyper-parameters*

Empirically set to $\mathrm{a} = 10, \mathrm{b} = 1$

Proposed Loss (to be minimized):

Feature Similarity Loss

$$\mathcal{L}(\mathbf{d}) = 1 - \boxed{S_C(f(\mathcal{T}(\mathbf{I}_w;\mathbf{d})), f(\mathbf{I}_q))} + a\boxed{\|\mathcal{T}(\mathbf{I}_w;\mathbf{d}) - \mathbf{I}_q\|_2} + b\boxed{\|\mathbf{d}\|_2}$$

Visual Similarity Loss

Bring transformed image close to the initial one

Regularization Loss

Keep the deformation parameters close to zero

a, b : *user-defined hyper-parameters*

Empirically set to $\mathrm{a} = 10, \mathrm{b} = 1$

Query-based Deformation

Algorithm Overview: Iterate over the proposed loss

Algorithm : On-the-fly Deformation

Input: Adam hyperparameters, number of iterations K, initial deformation $\mathbf{d}_0$, loss hyperparameters a, b

Output: optimized deformation parameters $\mathbf{d}_K$

```
Initialize d as d_0
for i = 0 to K − 1 do
    Forward Pass: Compute L(d_i) according to Eq. 2
    Backward Pass: Compute ∇L(d_i)
    Adam Update: d_{i+1}
end for
```

Implementation Aspects

Complexity Issue: perform gradient descent for each pair (word, query)

Linear dependence to both the number of iterations and the number of words in the dataset

Complexity Issues

Number of words N_w

Number of iterations K

Complexity Issues

Number of words N_w

Assume that we have a <u>well-performing feature extractor</u>

"fine-tune" matching score with the proposed method for a limited subset of the N_w most relevant words!

Number of iterations K

Complexity Issues

Number of words N_w

Assume that we have a well-performing feature extractor

"fine-tune" matching score with the proposed method for a limited subset of the N_w *most relevant words!*

Number of iterations K

Treat our concept as a "counter-adversarial" example

Assume that minor changes in deformation parameters can affect performance

Perform the proposed method for a **small** number of iterations K

Complexity Issues

Number of words N_w

Assume that we have a <u>well-performing feature extractor</u>

"fine-tune" matching score with the proposed method for a limited subset of the N_w *most relevant words!*

Number of iterations K

Treat our concept as a "counter-adversarial" example

Assume that minor changes in deformation parameters can affect performance

↓

Perform the proposed method for a **small** number of iterations K

image deformations of large-magnitude should cautiously
perform many steps of the proposed algorithm with small lr

Complexity Issues

Number of words N_w

Assume that we have a well-performing feature extractor

"fine-tune" matching score with the proposed method for a limited subset of the N_w most relevant words!

Number of iterations K

Treat our concept as a "counter-adversarial" example

Assume that minor changes in deformation parameters can affect performance

Perform the proposed method for a **small** number of iterations K

image deformations of large-magnitude should cautiously
perform many steps of the proposed algorithm with small lr

Proof-of-concept Setting: applying random transformations of negligible magnitude

mean absolute difference in AP for all considered queries is ~ **1.5%**

Ablation

Ablation setting: KWS performance on IAM validation set

$lr = 0.01, K = 3, N_w = 50$

deformation	MAP (%)
reference	95.59
gaffine	95.91
laffine	96.22
ldeform	96.19
gaffine + laffine	96.12
gaffine + ldeform	96.14
laffine + ldeform	96.32
gaffine + laffine + ldeform	96.40

✓ Increased performance when using all possible deformations

Ablation

base parameters: $lr = 0.01, K = 3, N_w = 50$

K	MAP (%)	time (sec/query)
reference	95.59	-
1	95.97	0.24
2	96.34	0.40
3	96.40	0.57
4	96.26	0.74
5	96.23	0.91
10	96.11	1.75
15	96.07	2.61
20	95.98	3.45

N_w	MAP (%)	time (sec/query)
reference	95.59	-
10	96.21	0.14
25	96.33	0.30
50	96.40	0.57
75	96.39	0.83

- ✓ Time requirements increase linearly with K, N_w
- ✓ Iterating the approach multiple times may falsely match images to the query : *constraints are very important!*
- ✓ Increasing N_w over a specific threshold does not help

Letting an image to be significantly transformed may falsely bring not relevant words close to the query

Qualitative Examples

QUERY: 

Feature-based retrieval list:

steve	steve	here	there	steve	here	were	steve
score:	*0.154*	*0.210*	*0.211*	*0.221*	*0.225*	*0.227*	*0.235*

Qualitative Examples

QUERY:

Feature-based retrieval list (64.26% AP):

steve	steve	here	there	steve	here	were	steve
score:	*0.154*	*0.210*	*0.211*	*0.221*	*0.225*	*0.227*	*0.235*

differences are not visible!

Proposed updated retrieval list (91.66% AP):

steve	steve	steve	here	steve	there	were	here
score:	0.154	0.172	0.191	0.208	0.211	0.219	0.223

Qualitative Examples

QUERY: 

Feature-based retrieval list (64.26% AP):

steve	steve	here	there	steve	here	were	steve
score:	*0.154*	*0.210*	*0.211*	*0.221*	*0.225*	*0.227*	*0.235*

Proposed updated retrieval list (91.66% AP):

steve	steve	steve	here	steve	there	were	here
score:	0.154	0.172	0.191	0.208	0.211	0.219	0.223

Comparison to SOTA: IAM dataset

Method	MAP (%)
PHOCNet	72.51
HWNet	80.61
Triplet-CNN	81.58
PHOCNet-TPP	82.74
DeepEmbed	84.25
Deep Descriptors	84.68
Zoning Ensemble PHOCNet	87.48
End2End Embed	89.07
DeepEmbed	90.38
HWNetV2	92.41
NormSpot	92.54
Seq2Emb	92.04
Proposed Systems	
reference system	91.88
on-the-fly deformations	93.07

Comparison to SOTA: IAM dataset

Method	MAP (%)
PHOCNet	72.51
HWNet	80.61
Triplet-CNN	81.58
PHOCNet-TPP	82.74
DeepEmbed	84.25
Deep Descriptors	84.68
Zoning Ensemble PHOCNet	87.48
End2End Embed	89.07
DeepEmbed	90.38
HWNetV2	92.41
NormSpot	92.54
Seq2Emb	92.04
Proposed Systems	
reference system	91.88
on-the-fly deformations	93.07

Using cosine distance on PHOC estimation leads to 88.78% MAP

Comparison to SOTA: IAM dataset

Method	MAP (%)
PHOCNet	72.51
HWNet	80.61
Triplet-CNN	81.58
PHOCNet-TPP	82.74
DeepEmbed	84.25
Deep Descriptors	84.68
Zoning Ensemble PHOCNet	87.48
End2End Embed	89.07
DeepEmbed	90.38
HWNetV2	92.41
NormSpot	92.54
Seq2Emb	92.04
Proposed Systems	
reference system	91.88
on-the-fly deformations	93.07

Our approach outperforms SOTA

at the cost of computational requirements

Thank you

Acknowledgements

This research has been partially co-financed by the EU and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the calls: “RESEARCH - CREATE - INNOVATE”, project Culdile, and “OPEN INNOVATION IN CULTURE”, project Bessarion.