Universität Rostock Traditio et Innovatio

CITlab Computational Intelligence Technology

READ

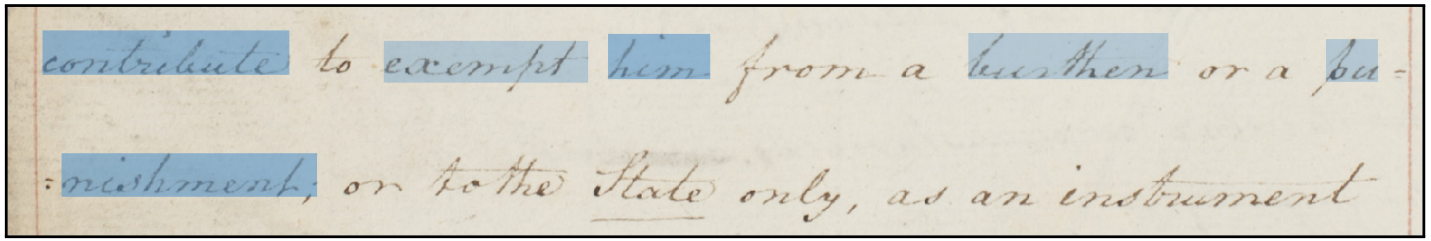
# CITlab ARGUS fer Keyword Search in Historical Handwritten Documents

Thanks to Mauricio for giving the talk.

**Tobias Strauß**
**Tobias Grüning**
**Gundram Leifert**
**Roger Labahn**

## Handwritten Scanned Document Retrieval Task 2016

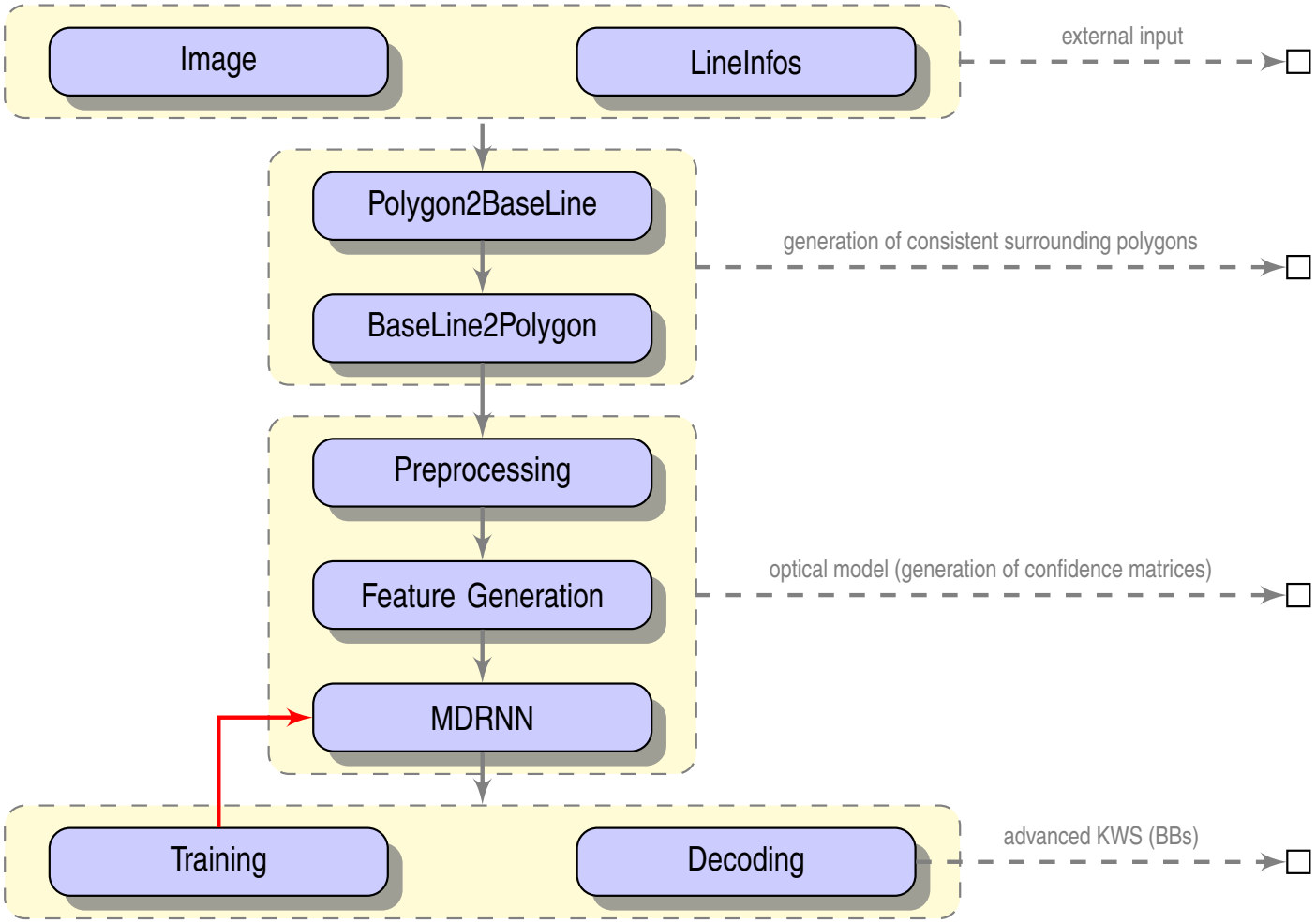
- advanced keyword spotting task




## Composition of the data sets provided by the organizers

| | #pages | #lines | #characters | polygons | baselines |
|---|---|---|---|---|---|
| training set | 363 | 9645 | 65488 | yes | no |
| development set | 433 | 10589 | 80758 | yes | no |
| test set | 200 | 6355 | — | no | yes |

# System Overview



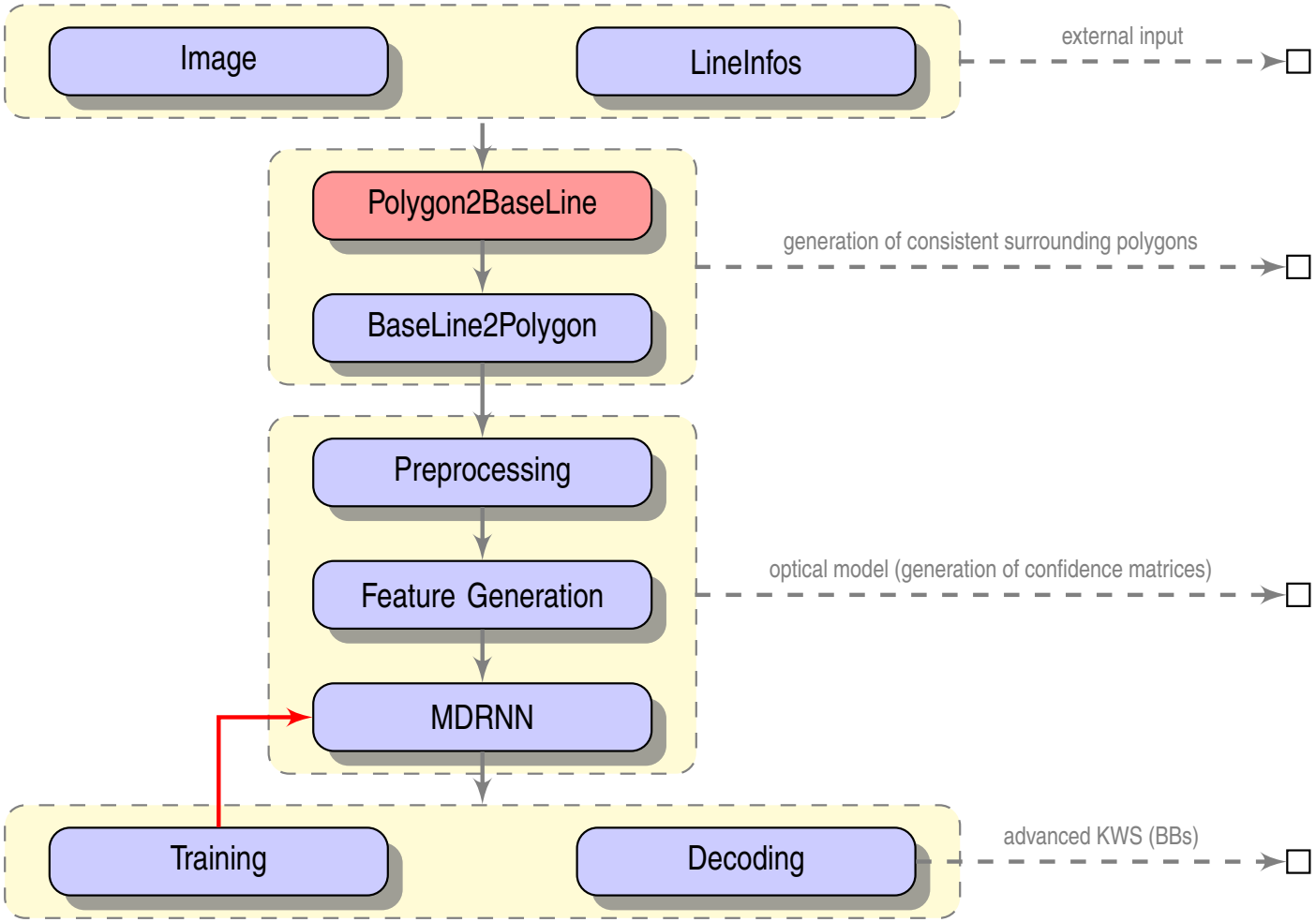
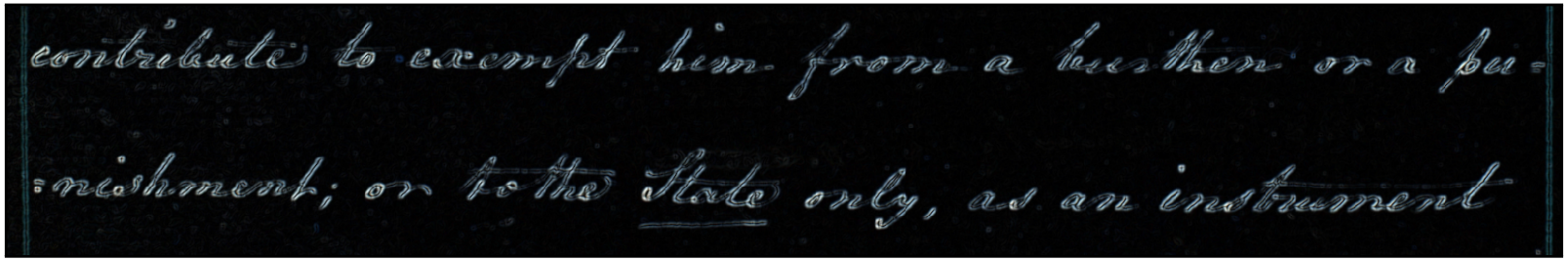


Image
LineInfos
external input
Polygon2BaseLine
BaseLine2Polygon
generation of consistent surrounding polygons
Preprocessing
Feature Generation
MDRNN
optical model (generation of confidence matrices)
Training
Decoding
advanced KWS (BBs)

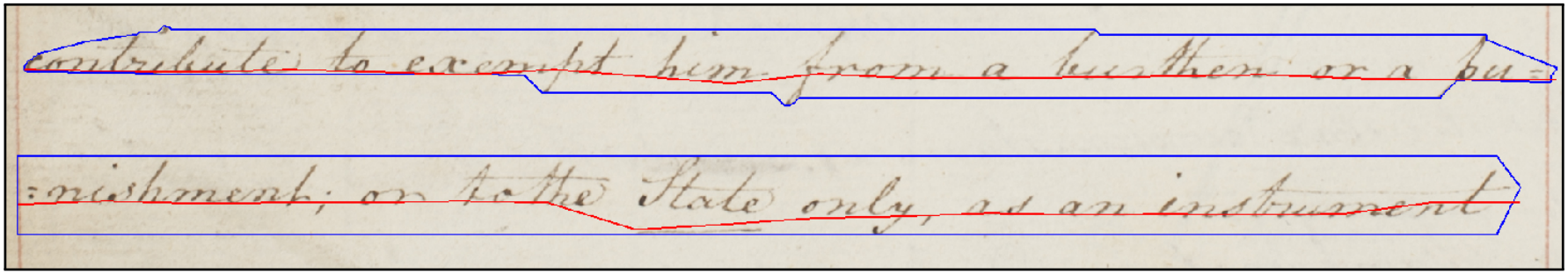
# Polygon2BaseLine

1. Convolving the input image with Sobel-Kernels:




2. Evaluation of statistics on the sobel image to calculate the baselines

## Module Output (Baselines) is calculated given the polygons



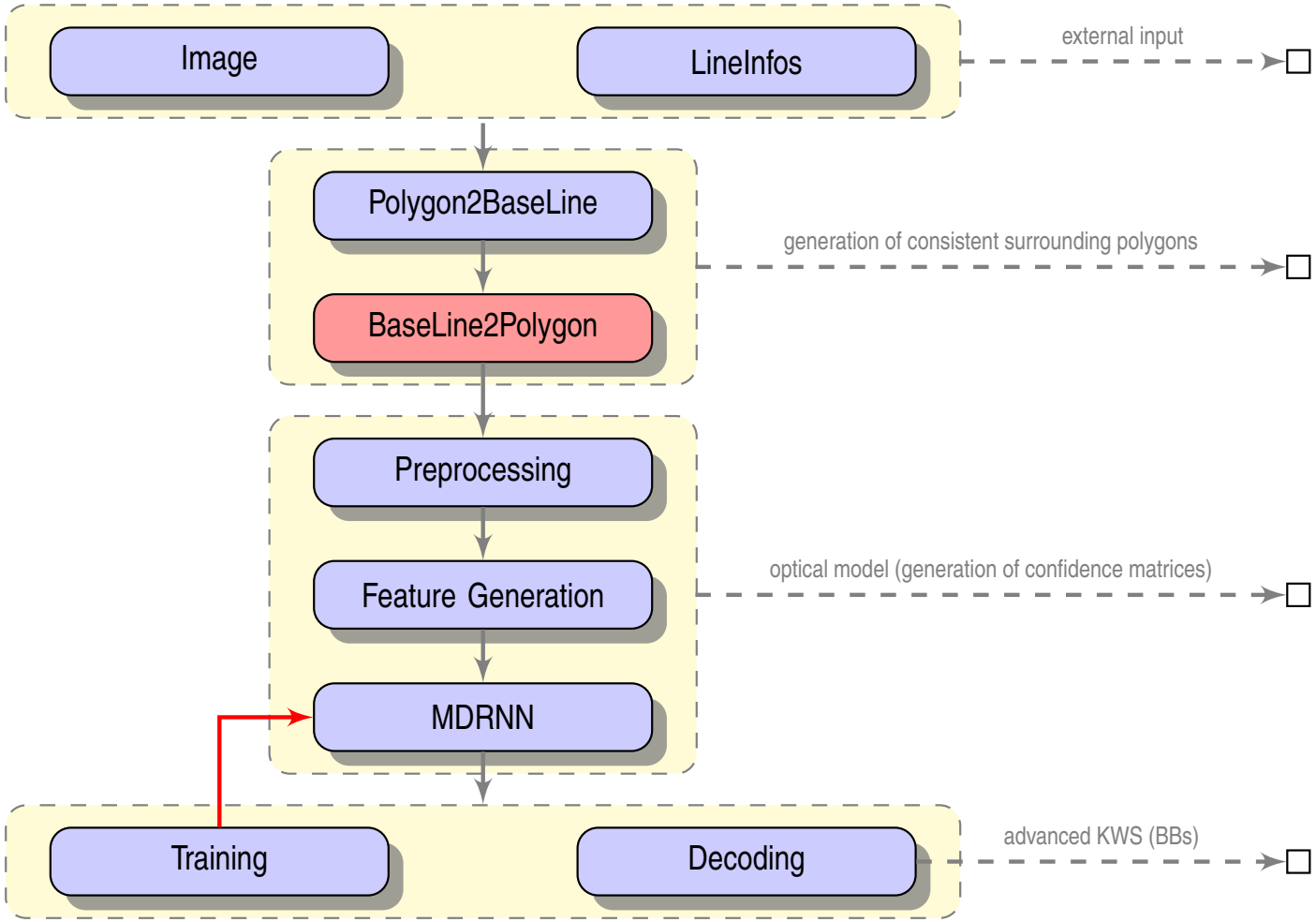
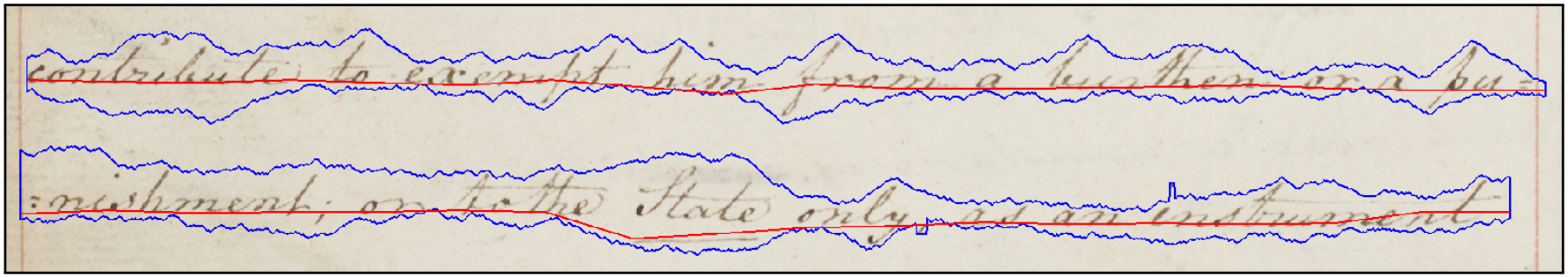


Image
LineInfos
external input
Polygon2BaseLine
BaseLine2Polygon
generation of consistent surrounding polygons
Preprocessing
Feature Generation
MDRNN
optical model (generation of confidence matrices)
Training
Decoding
advanced KWS (BBs)

# BaseLine2Polygon

1. Work on the sobel image
2. For each baseline calculate a medial seam (by maximizing sobel activity)
3. For each medial seam calculate two separating seams by using dynamic programming to calculate an optimal path from ”left to right”
4. Merge both separating seams to get a surrounding polygon per baseline

## Module Output (Polygons) is calculated given the baselines



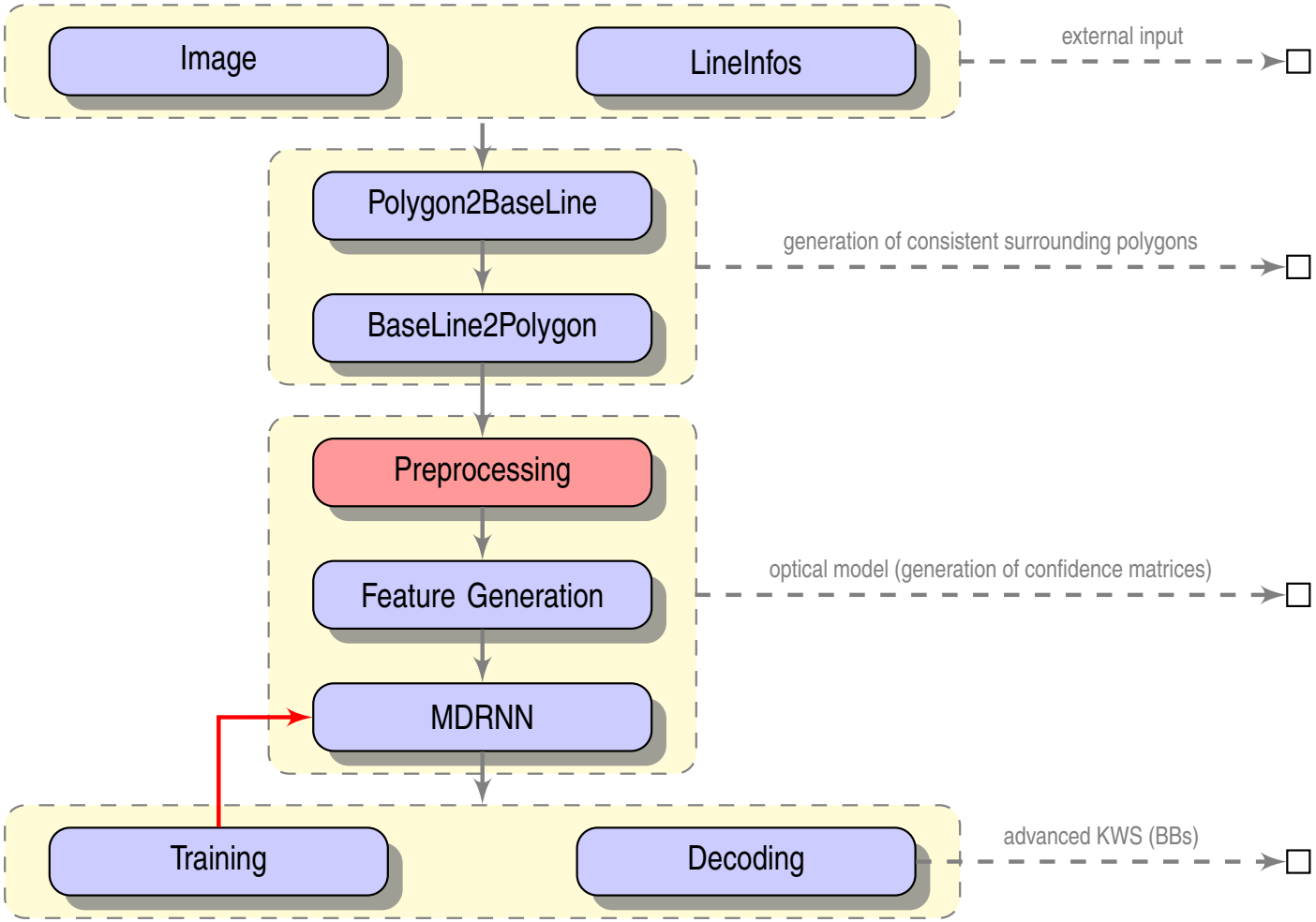


Image
LineInfos
external input
Polygon2BaseLine
BaseLine2Polygon
generation of consistent surrounding polygons
Preprocessing
Feature Generation
MDRNN
optical model (generation of confidence matrices)
Training
Decoding
advanced KWS (BBs)

## Preprocessing

Input: Line image with surrounding polygons

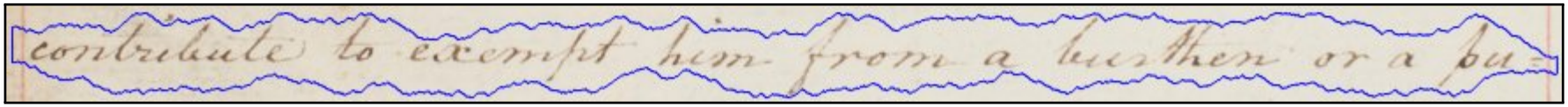
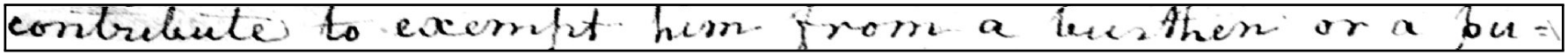


Image normalization: contrast enhancement and global size normalization

Writing normalization: deskewing and deslanting

Output: Normalized line image to fixed image height
with mainbody centered in the middle of the image



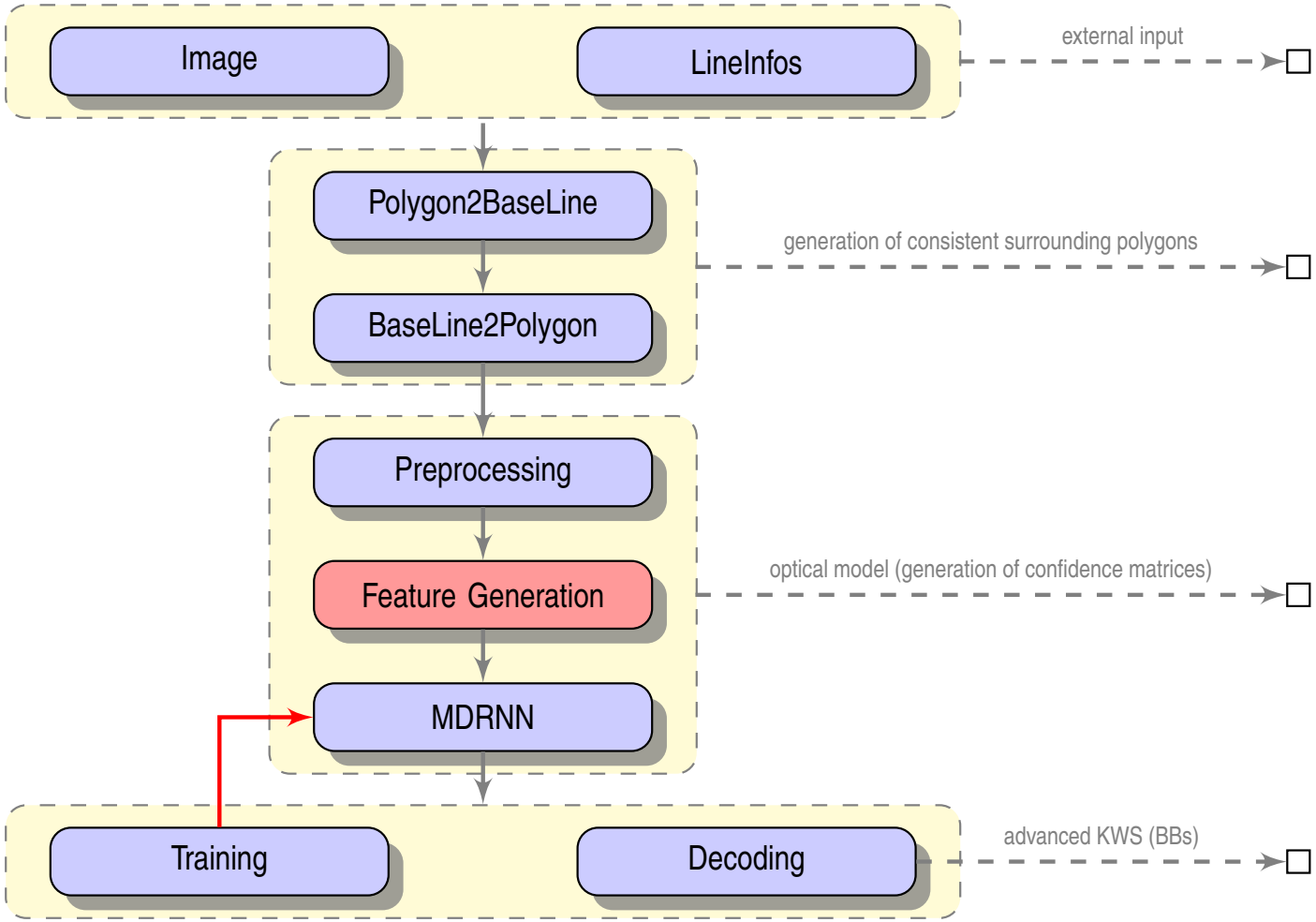
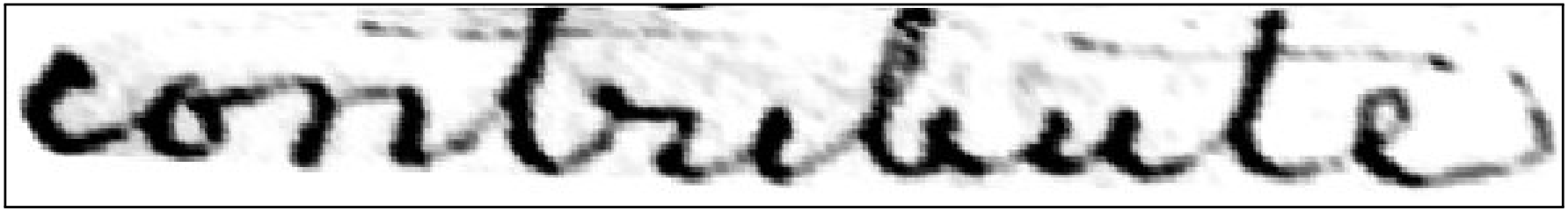


Image
LineInfos
external input
Polygon2BaseLine
BaseLine2Polygon
generation of consistent surrounding polygons
Preprocessing
Feature Generation
MDRNN
optical model (generation of confidence matrices)
Training
Decoding
advanced KWS (BBs)

# Feature Generation

Input: Normalized line image



Output: 3-dimensional frequency feature

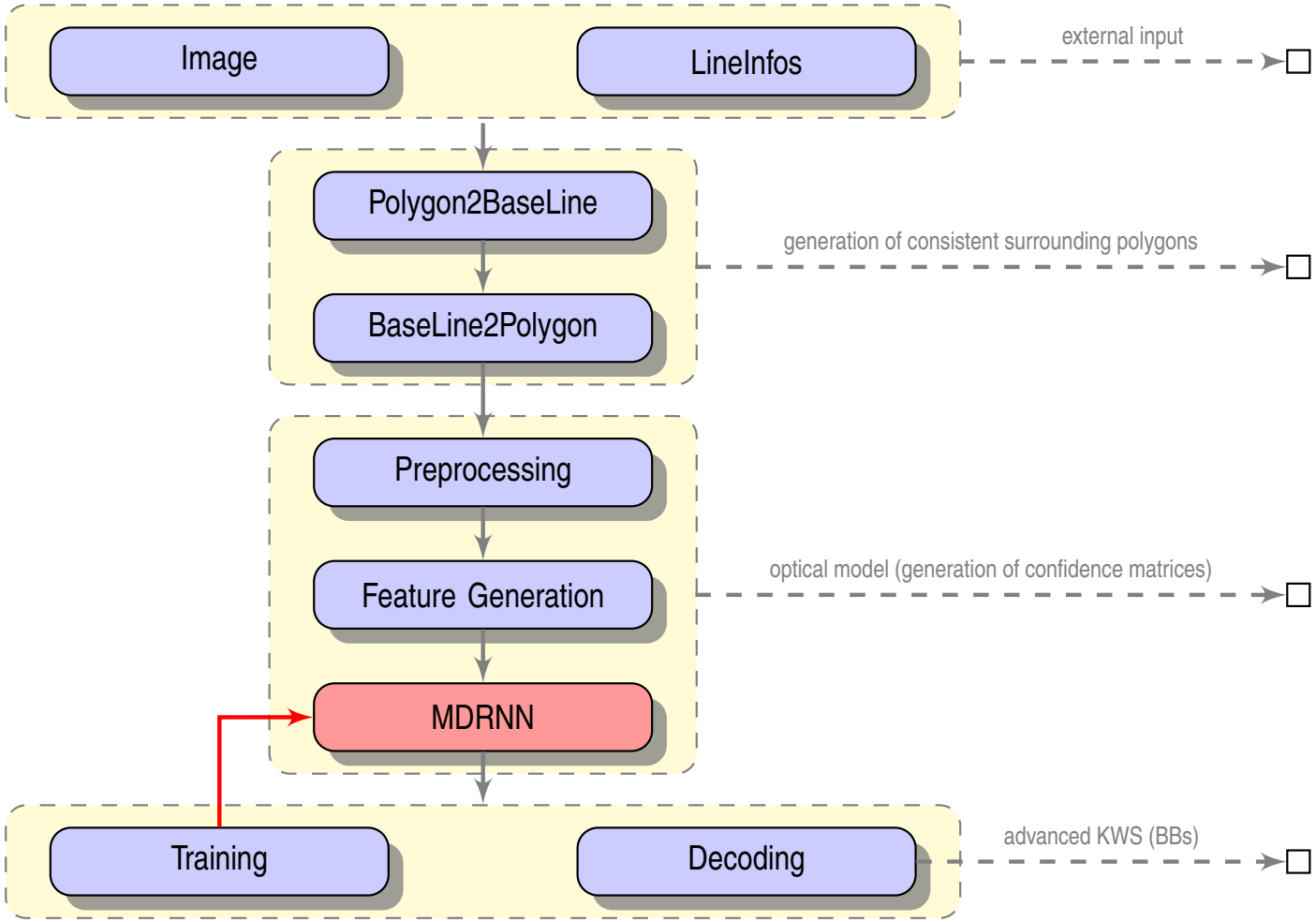
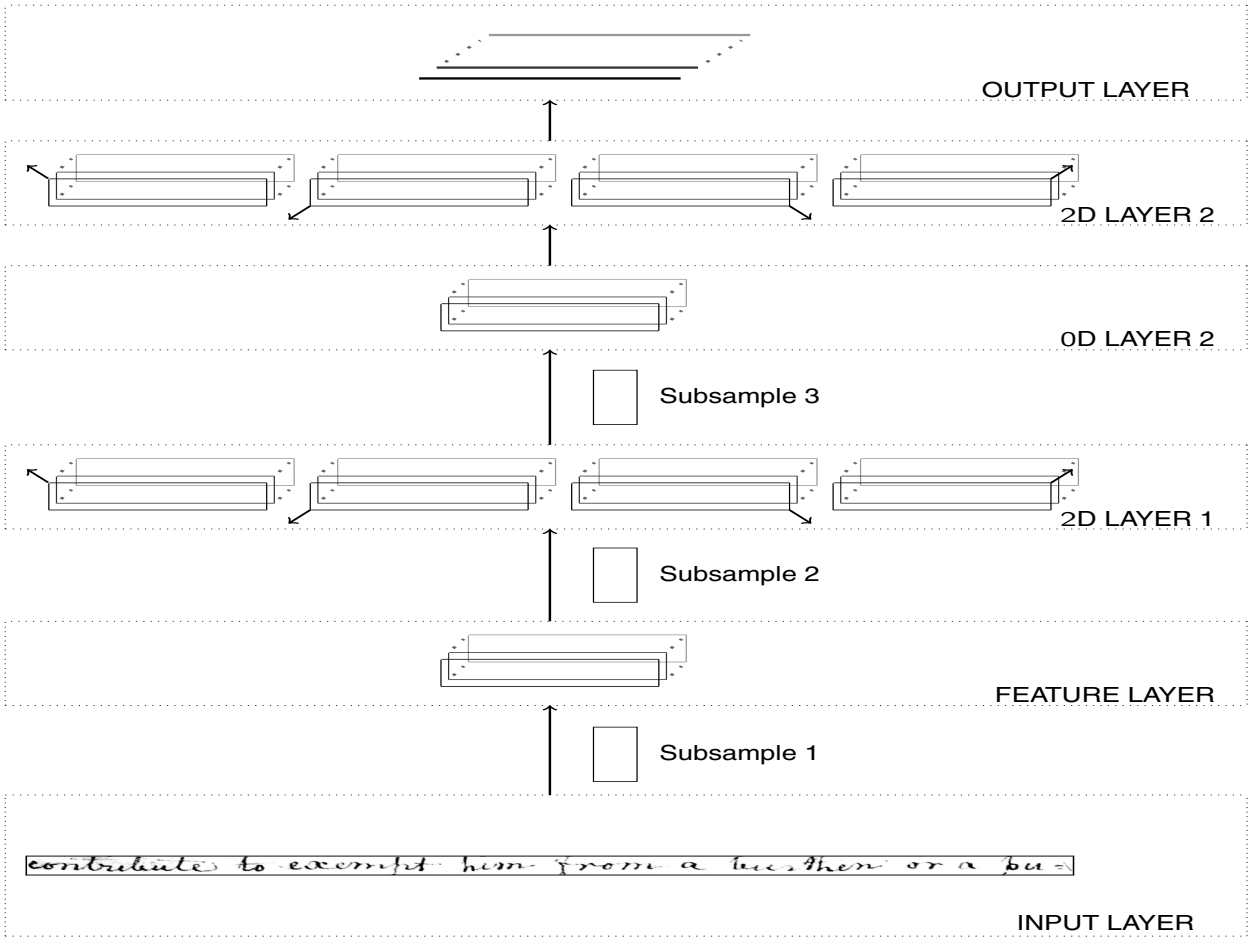


Image
LineInfos
external input
Polygon2BaseLine
BaseLine2Polygon
generation of consistent surrounding polygons
Preprocessing
Feature Generation
MDRNN
optical model (generation of confidence matrices)
Training
Decoding
advanced KWS (BBs)

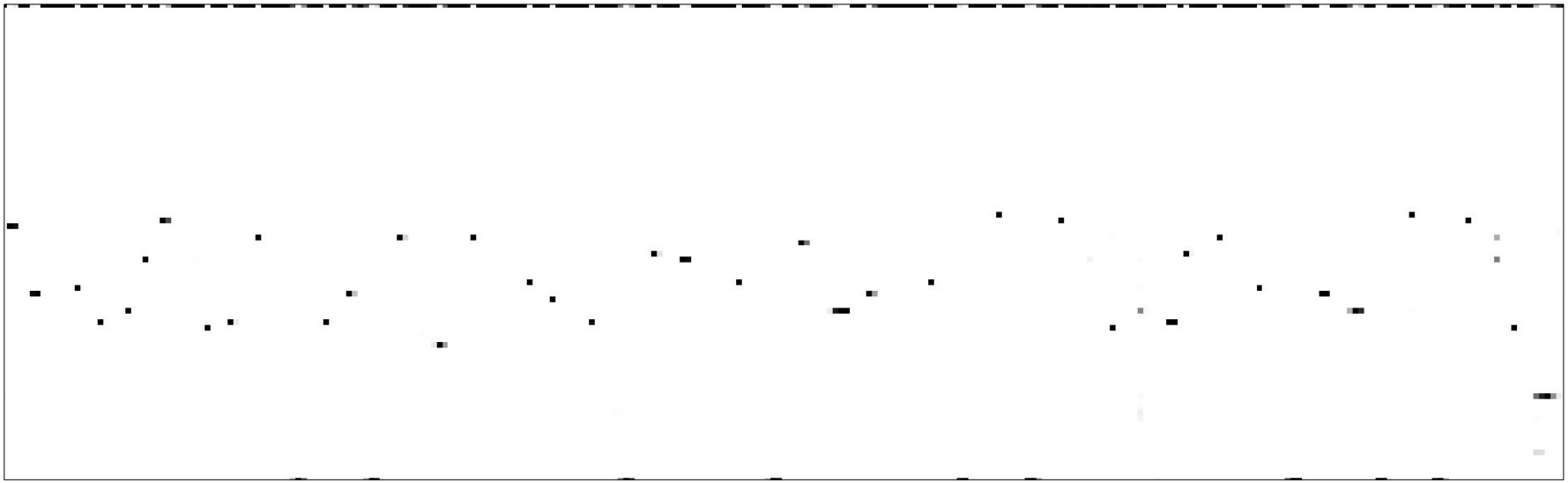
# Multidimensional Recurrent Neural Network



## Output of the Network: Confidence Matrix (ConfMat)

Input: 3-dimensional frequency feature

Output: Confidence of a character at a specific position (high confidence = black)



Transcription: “contribute to exempt him from a burthen or a biu-”

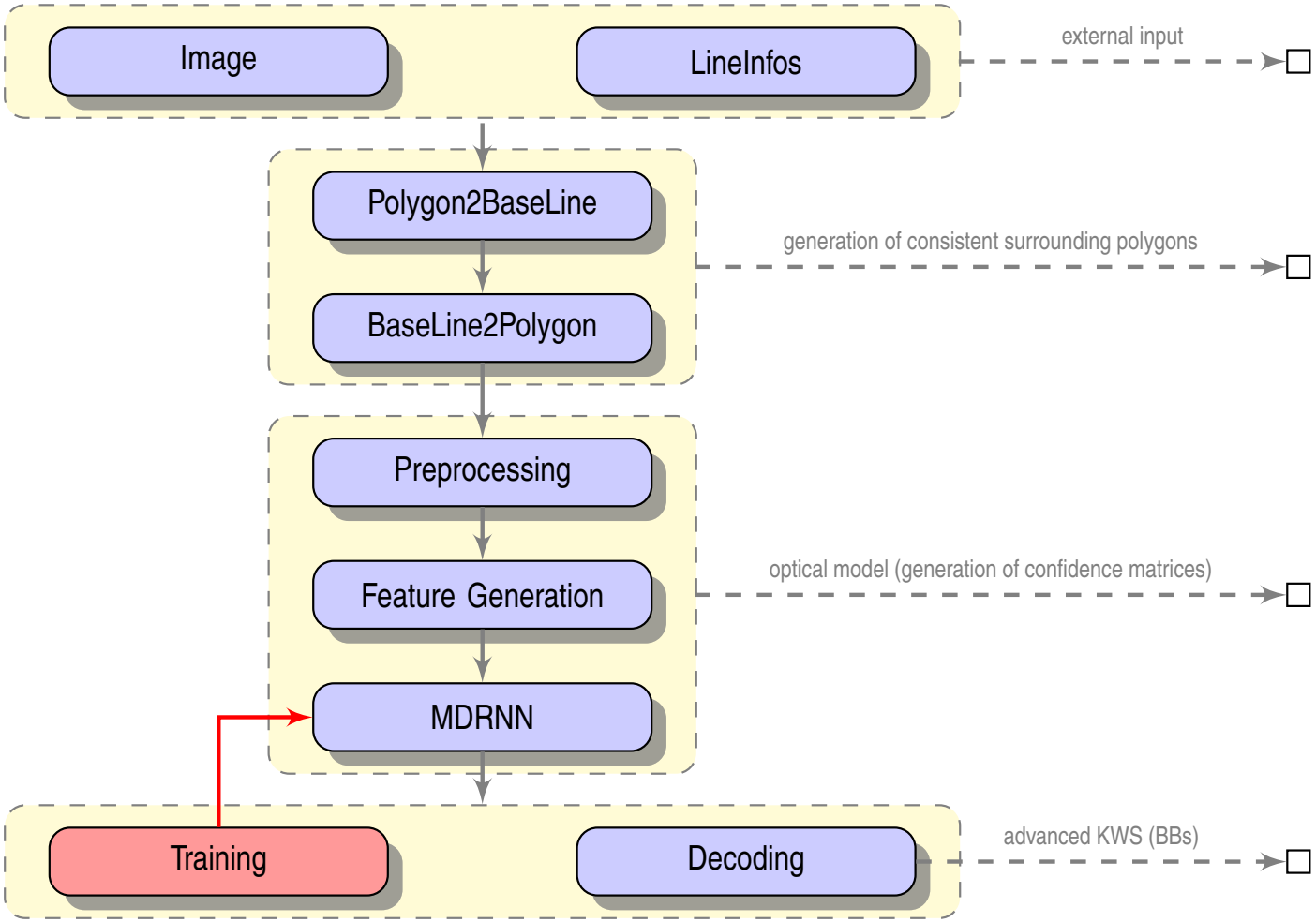


Image
LineInfos
external input
Polygon2BaseLine
BaseLine2Polygon
generation of consistent surrounding polygons
Preprocessing
Feature Generation
MDRNN
optical model (generation of confidence matrices)
Training
Decoding
advanced KWS (BBs)

## Network Training

- The network is trained using CTC as error Function and an extension of Nesterov's Accelerated Gradient Descent with momentum $\mu = 0.9$ for parameter update.
- One epoch contains $10.000$ training samples (lines) from the training set. No lines of the devel or the test set were used for training at all.
- A learning rate $\lambda = 5e - 4$ is used for 19 epochs using the original images.
- For 3 epochs noise to preprocess parameter and network activation are added with the same learning rate.
- For additional 19 epochs we set $\lambda = 5e - 5$ and add degradation (pixel noise, blur, cross outs,...) to the image.

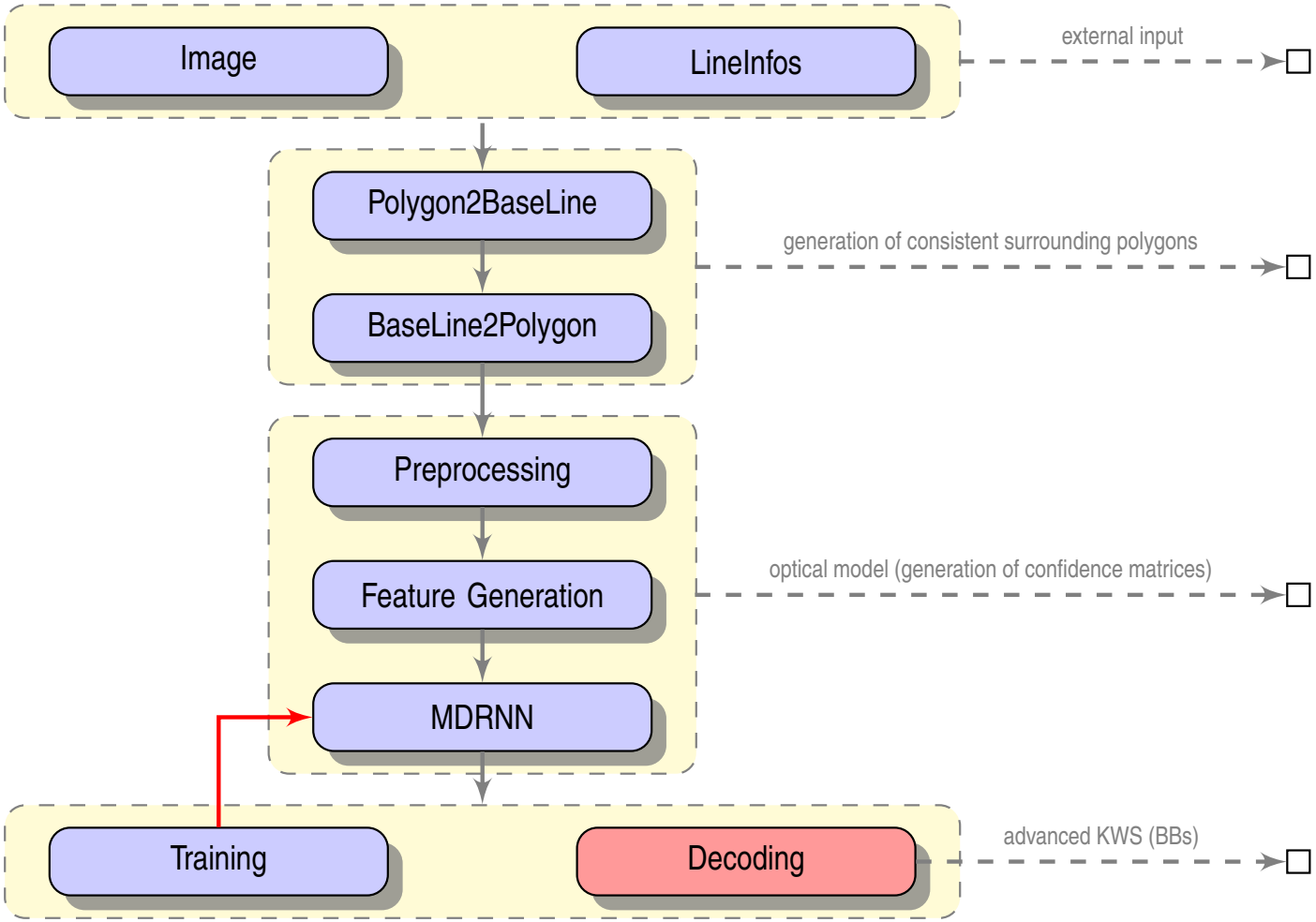


Image
LineInfos
external input
Polygon2BaseLine
BaseLine2Polygon
generation of consistent surrounding polygons
Preprocessing
Feature Generation
MDRNN
optical model (generation of confidence matrices)
Training
Decoding
advanced KWS (BBs)

# Decoding I

## Find words regions

For any keyword $\omega$ and any line image $X$

1. Let $r(\omega) := .*\omega.*$
2. Find the most likely path $\pi^*$ matching $r(\omega)$
3. Calculate the likelihood

$$\mathrm{P}_{s:e}(\pi^*(\omega) \mid X) := \prod_{t=s}^{e} y_{t,\pi_t^*}$$

   of $\omega$. $(s, e)$ are determined by the subpath $\pi_{s:e}$ which is related to $\omega$.
4. Accept $\omega$ if $\mathrm{P}_{s:e}(\pi^*(\omega) \mid X)$ exceeds a certain threshold

# Decoding II

## Hyphens

ConfMats of interest end on hyphen symbols such as `-:=`.

1. Find ConfMats matching $\boldsymbol{r} :=$ `.* [A-Za-z]+ [-:=]`
2. Combine those columns of the ConfMat which match `[A-Za-z]+` with the ConfMat of the next line
3. Search in the resulting new ConfMat for keywords

# Decoding III

## Score

Reestimate the likelihood of the neural net by:

$$s(\omega) := \frac{1}{N} \frac{\mathrm{P}_{\mathcal{T}}(\omega)}{\mathrm{P}_{\mathcal{S}}(\omega)} \mathrm{P}_{s:e}(\pi^*(\omega) \mid \boldsymbol{X})$$

$$N := \sum_{\omega' \in \mathcal{A}^*} \frac{\mathrm{P}_{\mathcal{T}}(\omega')}{\mathrm{P}_{\mathcal{S}}(\omega')} \mathrm{P}_{s:e}(\pi^*(\omega') \mid \boldsymbol{X})$$

where $\mathrm{P}_{\mathcal{T}}(\omega)$ is the uni-gram prior probability of $\omega$ in the corpus and $\mathrm{P}_{\mathcal{S}}(\omega)$ is the prior probability estimated by the network.

# Decoding IV

## Multi word query $\omega_{1:n} := \omega_1, \ldots, \omega_n$

Search all words $\omega_i$ individually and combine their score by

$$s(\omega_{1:n}) := \left( \prod_{i=1}^{n} s(\omega_i) \right)^{\frac{1}{n}} .$$

# Decoding V

## Implicit network priors $P_{\mathcal{S}}$

Estimate the network priors $P_{\mathcal{S}}$ by

- `abs`: the real uni-grams $P_{\mathcal{S}}(\boldsymbol{z}) = P_{\mathcal{T}}(\boldsymbol{z})$
- `prior`: a constant $P_{\mathcal{S}}(\boldsymbol{z}) \propto 1$
- `da`: character priors $P_{\mathcal{S}}(\boldsymbol{z}) \propto \left(\prod_{i=1}^{|\boldsymbol{z}|} P(z_i)\right)^c$
  where $P(z_i)$ is the character prior and $0 < c \leq 1$.

## Results: global average precision (gAP) on segment level

| | `abs` | `prior` | `da` | baseline |
|---|---|---|---|---|
| devel set | 94.81 | **95.36** | 94.99 | 74.2 |
| test set | 33.89 | 43.20 | **46.74** | 14.4 |

# Decoding VI

## CTC vs. path probability

Compare path probability $P_{s:e}(\pi^*(\omega) \mid \boldsymbol{X})$ with CTC-probability i.e. the sum over all feasible paths related to $\omega$ in the range from $s$ to $e$.

## Results: gAP on segment level

| | abs | | prior | | da | | |
|---|---|---|---|---|---|---|---|
| | path | ctc | path | ctc | path | ctc | baseline |
| devel set | 94.81 | 94.89 | 95.36 | **95.42** | 94.99 | 95.04 | 74.2 |
| test set | 33.89 | 33.98 | 43.20 | 43.72 | 46.74 | **47.13** | 14.4 |

**Thank you for your attention.**

Sorry for not being here.