3.1 - Preprocessing

Text Preprocessing

- normalisations
  - capitalisation
  - abbreviations
- replacing irrelevant/duplicate symbols
  - e.g. minus vs en dash vs em dash
  - formatting indicators, comments
- collect the target alphabet

Image Segmentation

- often via DL models
  - e.g. “Laypa” → https://github.com/stefanklut/laypa
- rectangular vs polygonal masks

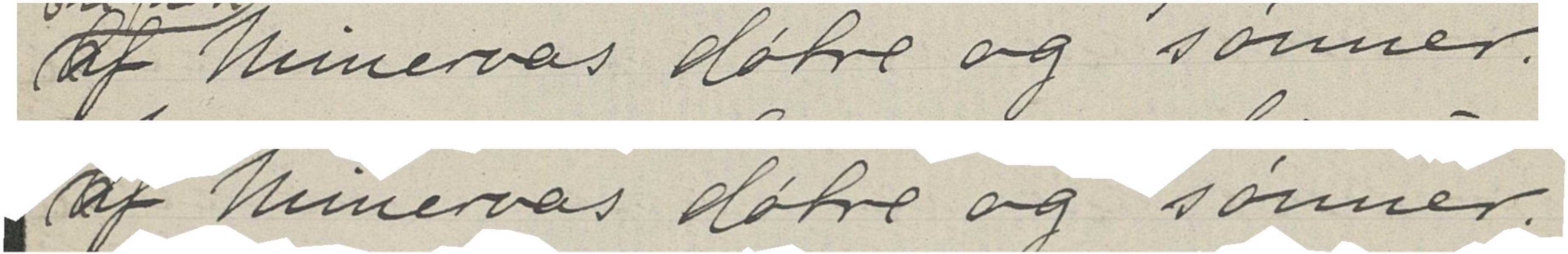


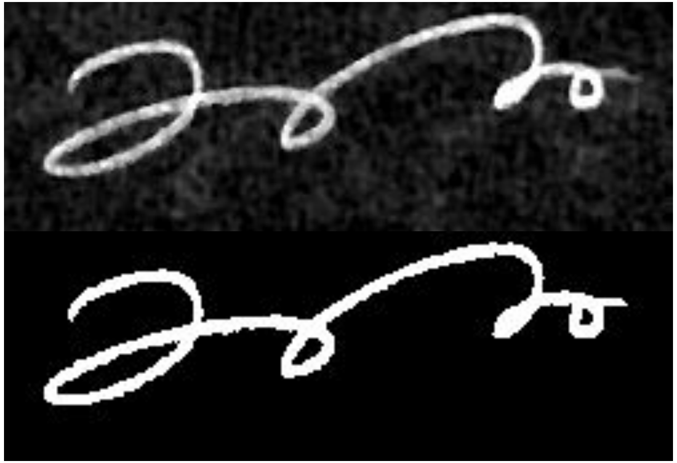
Image Preprocessing - General Goal

- improve readability
- reduce noise
- historically also: increase uniformity
  - e.g. deslanting/deskewing

- overall: **emphasise relevant features** (ink vs background)

Image Preprocessing - Colour Spaces

- colour rarely carries relevant information
  - may even “distract”
- focus on intensity, instead
  - traditionally: binarisation
    - problem: ink vs background = fuzzy
  - modern: greyscale
- inversion
  - ink: dark → bright



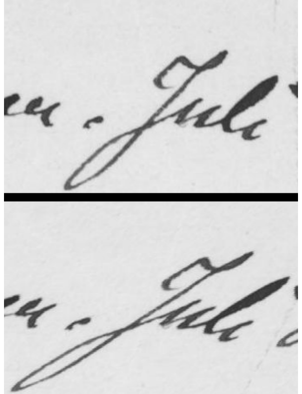
3.2 - Augmentation

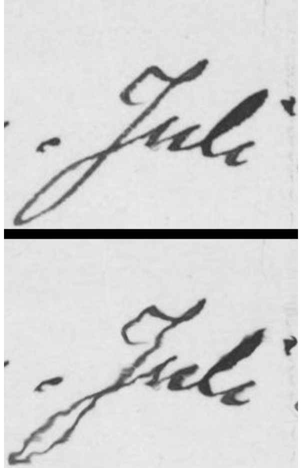
Augmentations

- increase visual variety in training set without collecting new data
- create modified copies of original training images
- include in training set

Simulating Writing Style Variations

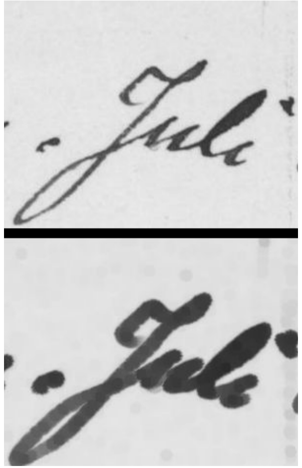
- affine transformations
  - rotate
  - scale
  - shear
  - ...
- elastic transformation
  - Simard, Patrice Y., David Steinkraus, and John C. Platt. "Best practices for convolutional neural networks applied to visual document analysis." ICDAR. Vol. 3. No. 2003. 2003.

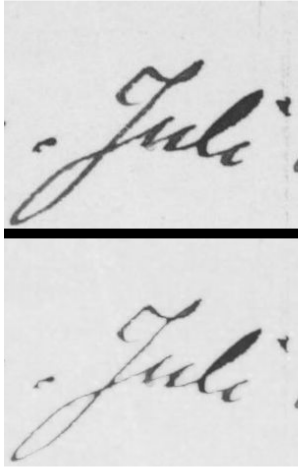




Simulating Pen Differences

- morphology:
  - greyscale dilation
  - greyscale erosion





Hands-On Time!

Exploring Augmentations

Using `demo/aug_demo.py`, experiment with:

- different augmentations
- parameter variations
- different combinations of augmentations

Observe the impact on handwritten text, e.g.:

- which settings create plausible new images?

Which augmentations/settings would you use/avoid for HTR? Why?

Integrate your chosen augmentations into `htr/utils/run_utils.py` ll.61