

# Layout Analysis Help

## Transkribus LA – Layout Analysis Method

### General Information

Transkribus LA is a baseline detection algorithm that uses an ARU-Net as outlined in the first stage of this paper: <https://arxiv.org/abs/1802.03345>. It roughly works as follows:

- Stage 1: a neural net (i.e. an ARU-Net) is used to produce “mask-images” indicating either baselines or separators of baselines (i.e. small vertical lines at the beginning/end of each baselines) – each pixel value of those images indicates the accuracy that a baseline/separator is present at this point of the input image
- Stage 2: from the mask-images produced in stage 1, the final baselines are created as sorted sets of points according to several heuristics (e.g. that a baseline shall not be too curved or that it should only contain points with a certain accuracy). This is also referred to as the postprocessing stage.

Instead of using the postprocessing of the above paper however (which was implemented in the CITlabAdvancedLA but had to be dropped due to licensing issues), we implemented our own postprocessing phase to produce the final baselines. This is an ongoing process and we hope to be able to fix all major issues as fast as possible.

### Neural Net setting

Allows to choose a specific neural network (i.e. a trained ARU-Net) to be used for finding baselines.

Choose “Preset” if you are not sure about the options or when working on a new dataset.

Note that text-regions are currently *not* used as training information. They are created in a pure unsupervised fashion after the final baselines are detected.

### Postprocessing Settings

Those settings apply to the second phase of the baseline detection, i.e. the postprocessing phase of the above paper. Currently all parameters can be changed freely by the user, however we aim to find sets of optimal parameters for certain types of documents (e.g. newspapers, ‘regular’ handwritten documents etc) for easier usage.

The current parameters are:

- Minimal baseline length:
  - The minimum length for a baseline in pixels – detected baselines below this length are dropped.
- Baseline accuracy threshold:

- The threshold for binarization of the baseline mask images. Higher values enforce higher accuracy in the detected baselines. Ranges between 0 and 255.
- Try to reduce the threshold if you have low resolution images and no or only a few baselines are detected. Bear in mind however, that the results can get noisy for lower thresholds.
- Separator threshold:
  - Threshold for using the trained separator images. Ranges between 0 and 255. If threshold is exceeded, nearby baselines are getting merged. If set to  $\leq 0$ , separators are not used at all.
  - The separator images are small vertical lines drawn beside each baseline during training, which indicate the start and end of each baseline – they should *not* be confused with actual separators in printed document images.
  - Usually, low values are sufficient to prevent a connection between nearby baselines. Use e.g. 1 to use separator information “sometimes” and larger values to use them pretty much all the time.
- Max-dist for merging:
  - If the distance exceeds this *fraction* of the *width* of the image, baselines will *not* get merged.
  - The algorithm produces a set of smaller baselines in the initial phase. Then it tries to merge nearby baselines but only if the distance is smaller than this threshold.
- Max-dist for clustering: ALPHA
  - If the distance exceeds this *fraction* of the *width* of the image, baselines will *not* get clustered to regions. If set to  $\leq 0$ , no region clustering will be performed.
  - This parameter is only valid for producing Text-Regions after all baselines are detected. Nearby baselines are getting clustered according to the distance of their leftmost point. Larger values here lead to larger Text-Regions.
  - General note on Text-Region clustering: the algorithm currently used is just an unsupervised clustering of the baselines, i.e. it is not trained on user input. Also, it's a very simple approach and thus the regions produced may not be useful at all. We are aiming to improve region detection in the future by using graph neural networks.