Fast Approximate Modelling of the Next Combination Result for Stopping the Text Recognition in a Video

Konstantin Bulatov Nadezhda Fedotova Vladimir V. Arlazarov

E-mail: kbulatov@smartengines.com

ICPR 2020 25th INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION Milan, Italy 10 | 15 January 2021

Mobile text recognition

Context & Goal

Mobile DAR systems:

Mobile document data extraction needs to be performed offline and in real time

Camera-based systems: ability to use video stream to increase recognition quality

Per-frame recognition results combination leads to a stopping problem

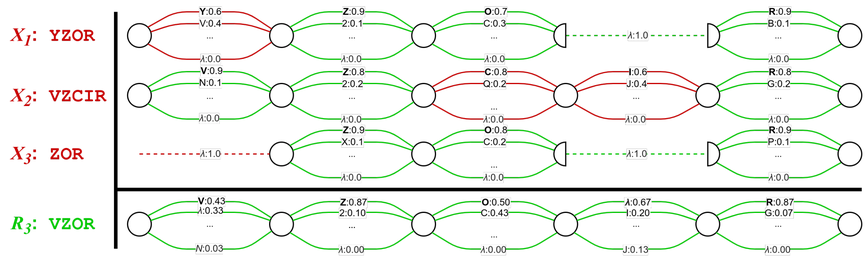
Goal: construct and evaluate computationally efficient methods for stopping the text recognition in a video

Existing methods

Text recognition result:

$$X = (x_{jk}) \in [0.0, 1.0]^{M \times K}, \quad \forall j \sum_{k=1}^{K} x_{jk} = 1.0$$

ROVER per-frame combination procedure:



Stopping: modelling of the next combination result: threshold the estimated distance (generalized Levenshtein) to the next combination result.

Proposed approximations & methods

Naive alignment:

During the test combination $R(X_1, \ldots, X_n, X_i)$ we will assume that the rows of X_i will be aligned with the same rows of R_n as on the stage i – skipping the costly alignment.

The length of $R(X_1, \ldots, X_n, X_i)$ stays the same as the length of R_n.

Naive Levenshtein:

The alignment between R_n and the combination $R(X_1, \ldots, X_n, X_i)$ is direct – each j-th row of R_n is aligned with the j-th row of $R(X_1, \ldots, X_n, X_i)$, thus the generalized Levenshtein distance between them is a sum of distances in terms of the row metric – skipping costly distance computation.

Additional accumulators:

$$Y_n = (y_{ijk}) \in [0.0, 1.0]^{n \times S_n \times (K+1)}; \quad A_{jk} = \sum_{i=1}^{n} y_{ijk}$$

y_{ijk} is a membership value of class k from a row of input X_i which was aligned and merged into the j-th row of the current combined result R_n.

Method A:

$$\hat{\Delta}_n \approx \frac{1}{n+1}\left(\delta + \sum_{i=1}^{n}\sum_{j=1}^{S_n}\sum_{k=0}^{K}\Delta_{ijk}\right)$$

$$\Delta_{ijk} = \frac{|A_{jk} - n \cdot y_{ijk}|}{2n(n+1)}$$

Complexity: $O(nS_nK)$

Method B:

$$L_{jk} \subset \{1, \ldots, n\} : \forall i \in L_{jk} : n \cdot y_{ijk} < A_{jk}$$

$$B_{jk} = \sum_{i \in L_{jk}} y_{ijk} \qquad \sum_{i=1}^{n} \Delta_{ijk} = \frac{A_{jk}|L_{jk}| - nB_{jk}}{n(n+1)}$$

L_{jk} using treaps: complexity $O(S_nK \log n)$

Expected distance modelling:

$$\hat{\Delta}_n = \frac{1}{n+1}\left(\delta + \sum_{i=1}^{n} \rho(R_n, R(X_1, \ldots, X_n, X_i))\right)$$

The method was shown to achieve a lower mean error level given the same mean number of processed frames and vice versa – lower mean number of processed frames for reaching the same mean error.

Downside: high complexity. If M is the maximal length of individual results X_i and S_n is the length of the combined resuilt R_n, the complexity of making the stopping decision at stage n: $O(nS_nK(S_n + M))$

Experimental datasets

ID documents:

MIDV-500 and MIDV-2019: text fields of identity documents. Recognition using Smart IDReader [1].

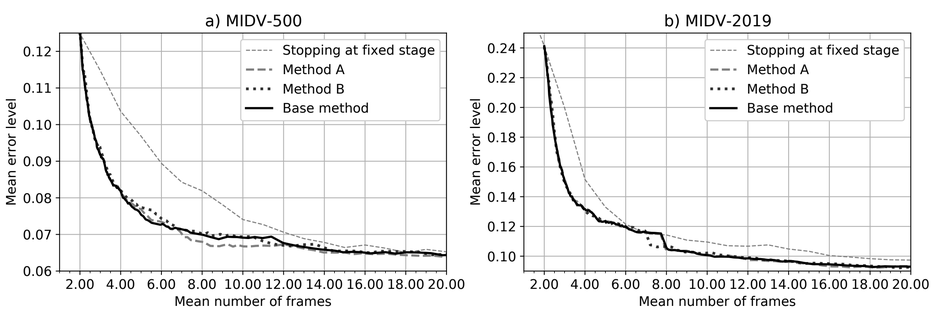
Arbitrary text:

ICDAR 2015 Train and Youtube Video Dataset. Recognition using published model [2].

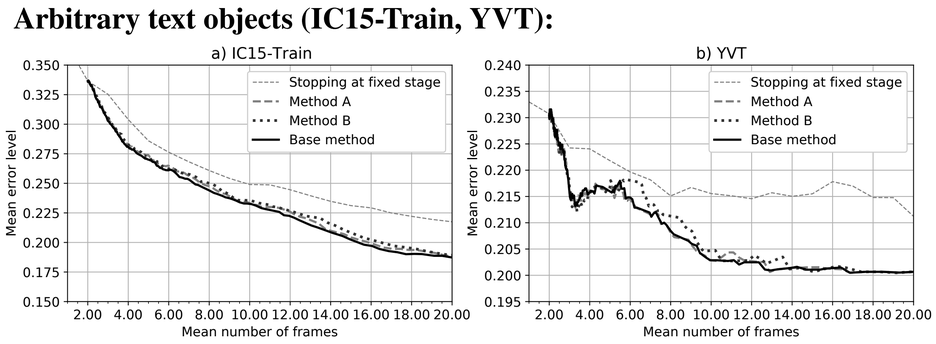
Expected performance profiles

Plotted mean error level (= distance to the ground truth) of the combined result against the mean number of consumed frames before stopping.

Identity document texts (MIDV):



Arbitrary text objects (IC15-Train, YVT):



Timing

Total time to perform text recognition results combination and making the stopping decision using the base method and the proposed optimizations.

Measured using prototype implementation (Python 3.7.4, AMD Ryzen 9 3950X).

Identity document texts (MIDV):

Method	Time on stage n (in seconds)				
	$n = 5$	$n = 10$	$n = 15$	$n = 20$	$n = 25$
Base	0.228	0.448	0.678	0.906	1.143
Method A	**0.022**	**0.022**	**0.023**	**0.024**	**0.024**
Method B	0.027	0.030	0.032	0.033	0.034

Arbitrary text objects (IC15-Train, YVT):

Method	Time on stage n (in seconds)				
	$n = 5$	$n = 10$	$n = 15$	$n = 20$	$n = 25$
Base	0.024	0.050	0.078	0.107	0.136
Method A	**0.002**	**0.003**	**0.003**	**0.003**	**0.003**
Method B	0.004	0.004	0.005	0.005	0.005

Conclusion

1. Proposed 2 approx. modelling schemes for text recognition in a video, which allow to compute the estimated distance to the next result and perform stopping decision.
2. Both methods evaluated on open datasets.
3. Optimizations do not decrease quality, but significantly increase the computational performance.

References

[1] Y.S. Chernyshova, A.V. Sheshkus, V.V. Arlazarov. "Two-step CNN framework for text line recognition in camera-captured images,", IEEE Access, v. 8, 2020.

[2] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, H. Lee. "What is wrong with scene text recognition model comparisons? Dataset and model analysis," ICCV 2019.