CurT: End-to-End Text Line Detection in Historical Documents with Transformers

Benjamin Kiessling[1,2](✉)

[1] École Pratique des Hautes Études, Paris, France
benjamin.kiessling@ephe.sorbonne.fr
[2] UMR 8546 CNRS-Université PSL (ENS-EPHE) - AOROC, Paris, France

Abstract. We present the curve transformer (CurT), a novel method of direct baseline detection that models document text line detection as set prediction of cubic Bézier curves, simplifying the layout analysis pipeline by removing the need for the laboriously hand-crafted postprocessing algorithms that are necessary with the current state of the art. CurT combines multiple appealing features: direct prediction enabling processing of material that is ill-suited for the prevailing methods adapting semantic segmentation backbones, a conceptually simple Transformer-based encoder-decoder architecture that can be extended to additional tasks beyond baseline detection, and increased computational efficiency in comparison to older approaches. In addition, we demonstrate that CurT achieves metrics that are competitive with methods based on semantic segmentation.

Training and inference code is available under Apache 2.0 license at https://github.com/mittagessen/curt.

Keywords: Document analysis · Machine learning · Text line detection · Object detection

1 Introduction

Document image analysis of historical material has seen a continued and rising interest over the last few decades, both from Computer Science researchers in search for more challenging material for their algorithms, scholars in the Human and Social Sciences aiming to apply computational analysis on ever larger corpora, and libraries and archives digitizing collections to ensure accessibility of humanity's collective cultural heritage. High quality document layout analysis is a keystone technology in any of those efforts reliant on retrodigitization on both the level of individual lines and higher order zones demarking textual and non-textual content.

While Automatic Text Recognition has achieved tremendous progress in the last decade, with typical character error rates well below 10% even for highly challenging handwritten material, these methods are uniquely dependent on

U. Porwal et al. (Eds.): ICFHR 2022, LNCS 13639, pp. 34–48, 2022.
https://doi.org/10.1007/978-3-031-21648-0_3

accurate prior segmentation as both the best performing and most widespread systems rely on segmented text lines as inputs. While segmentation-less methods have been proposed from time to time, they uniformly suffer from incomparable requirements on training time and data, are less robust regarding the nature of the texts to be recognized, or are not competitive with pre-segmenting approaches. Thus text recognition workflows are almost exclusively constructed out of a preliminary text line extraction step followed by actual text recognition with any failures in the segmentation directly translating into text recognition errors.

As can be expected for its central role, a large number of methods and paradigms, with varying focuses over the years as text recognition methods have grown in capability, have been proposed to deal with various handwritten and machine-printed historical documents. Early algorithms to extract individual characters from a page utilizing conventional computer vision methods have largely been supplanted first by hand-crafted algorithms detecting whole lines such as [1] and, later, machine learning-based approaches. Through these advances many documents are now in the reach of digitization without close human supervision but significant obstacles remain: generalization on out-of-domain documents is generally poor, especially for degraded material or different writing supports, implicit assumptions on the nature of the text, e.g. writing direction, that often don't hold for non-Latin-script documents are widespread, and methodological limitations of many methods make detection of overlapping and rotated writing difficult.

Layout analysis methods based on object detection systems offer the promise to overcome many of the conceptual limitations of these earlier systems. While established *indirect* object detection algorithms employing surrogate regression and classification problems that are highly dependent on postprocessing steps such as non-maximum suppression to collapse near-duplicate predictions, designs of anchor sets, and heuristics assigning entities to anchors are difficult to adapt to non-box shaped objects required for text line detection in historical documents, a new class of direct object detectors based on vision Transformers are much more flexible in their output data models.

2 Related Work

2.1 Transformers for Computer Vision

Transformers are a class of artificial neural network architecture that is characterised by a self-attention mechanism that learns relationships between elements of sets. In contrast to conventional recurrent neural networks that process sequences recursively and in practice can model long-term relationships only in a limited manner, Transformers are able to attend to complete sequences. This particular attention mechanism computing attention tensors across multiple heads (multi-head self-attention) along with minimal inductive biases in comparison to recurrent (sequentiality, recursion) and convolutional (translation invariance, locality) neural networks through the exclusive use of fully connected layers are the Transformer's distinguishing features. While Transformer layers can be arranged in a number of different configurations depending on task, the original and most widely used one organizes them into a encoder-decoder configuration.

Originally proposed in [32] for Natural Language Processing, Transformers have demonstrated astounding improvement on the then current state of the art for language modelling tasks such as text classification, machine translation, or question answering. The ability of Transformer networks to be effectively scaled up and trained with very large parameter counts that consistently outperform prior more lightweight models, e.g. the 340 million parameter BERT, 175 billion parameter GPT-3, up to the latest Switch transformers with up to 1.6 trillion parameters, have achieved generalization and adaptability that makes the impact of these architectures difficult to overstate.

The breakthroughs in performance achieved with Transformers have caused great interest outside of the NLP domain and the computer vision community has started to adapt these models for vision and multi-modal learning tasks. The resulting systems can largely by divided into hybrid architectures combining CNN encoders and Transformer decoders and architectures replacing convolutions altogether. The Vision Transformer (ViT) [8] was one of the first showcases for a standard Transformer architecture operating on flattened image patches producing competitive results on a number of computer vision tasks, albeit requiring pre-training on the extremely large proprietary JFT dataset. DeiT [31] demonstrated training transformers on the more moderately sized ImageNet dataset with state-of-the-art results through a teacher-student distillation approach with a CNN teacher model. These fixed-scale methods perform well on sparse prediction tasks such as image classification but the quadratic complexity of self-attention limits their applicability to higher-resolution images. Multi-scale architectures that merge tokens reducing the sequence length along a cascade of hierarchical layers have been proposed as a better alternative for dense prediction, e.g. object detection or semantic segmentation. Examples of these are the Swin Transformer [19], Pyramid Vision Transformers [34], and Focal Transformers [36]. An extensive survey of self-attention and Transformer-like methods for a wide range of computer vision tasks can be found in [12].

2.2 DETR and Variants

DETR [2] is an object detector built upon a Transformer encoder-decoder architecture combined with a set-based loss that forces unique predictions for each ground-truth bounding box through bipartite matching.

The model operates on input feature maps extracted by a CNN backbone, in the originally proposed implementation ResNet-50 and ResNet-101, that are fed into a standard Transformer encoder-decoder architecture. The inputs of the decoder stage are the transformed image features from the encoder and N learned positional encodings called *object queries* that condition the decoder to produce N distinct output embeddings from the transformed image features. A simple linear projection and a 3-layer feed-forward network are used to decode the output embeddings into classes and regress the normalized bounding box coordinates $\mathbf{b} \in \{b_{cx}, b_{cy}, b_w, b_h\}$ respectively.

The architectural simplicity of DETR and lack of hand-crafted algorithms such as non-maximum suppression or anchors often required in non-direct methods such as Faster R-CNN [27], YOLOv3 [26], and SSD [18] among many others makes it an attractive design for object detection and derived tasks. Unfortunately the original design suffers from several major drawbacks. The first is the quadratic computational complexity of the attention weight computation in the Transformer encoder with regard to the input size, putting a low upper bound on the maximum input resolution which makes detection of small objects difficult.

The second is the slow convergence with the original approach requiring a very long training schedule of 500 epochs to converge on the COCO dataset, roughly 10 to 20 times slower than Faster R-CNN's typical 30 epochs. The primary reason identified in [37] for this slow convergence is the suboptimal initial initialization of the attention modules casting nearly uniform attention weights to all pixels in the input feature maps requiring many epochs to achieve sufficient sparsity for the decoder to detect object effectively. This slow convergence is exacerbated by the lack of pre-training of the Transformer.

Another contributing factor to these long training times is the instability of the bipartite matching during initial epochs, as the assignment is essentially random in the early phases of training [28].

An abundance of detection transformers variants are intended to resolve these problems. Deformable DETR [37] decouples the computational cost of the attention module from the input feature maps through a deformable attention mechanism that attends only to a small set of sampling points around a reference point while at the same time incorporating multi-scale features improving recall of small objects. Conditional DETR [22] narrows down the spatial range for localizing object regions via learning the decoder embedding conditioned on a spatial query. UP-DETR [6] pre-trains standard DETR using a multi-query patch pretext task. FP-DETR [33] proposes a way to pre-train an encoder-only Deformable DETR object detector on a classification task. [28] adapt an FCOS-like object detector [30] with an encoder-only Transformer block and a modified bounding-box specific matching scheme that improves stability in the early stages of training. DN-DETR [15] halves the training time of DETR-like methods by introducing an auxiliary denoising task that bypasses the bipartite matching to circumvent early instability while at the same time improving accuracy on baseline DETR.

2.3 Text Baseline Detection

As a representation of text lines, baselines have a long history in historical document layout analysis (see [16] for a survey of early methods) and has recently both enjoyed a resurgence in research interest and widespread use in a number of practical text recognition systems for handwritten and printed documents such as Transkribus [5] and eScriptorium [14]. While early methods employing this paradigm utilized conventional image processing methods, the challenging cBAD competitions in 2017 and 2019 have triggered the publication of a large number of deep learning-based methods.

Baselines are a term originating from typography referring to a virtual polyline on which most characters of a text line rest upon or hang from. Despite not being universal, or in fact sometimes not being located at the bottom of the line as is the case with Hebrew and various Indict scripts, analogues that serve the same purpose for text recognition purposes can often be devised for material lacking true baselines, e.g. approximate centerlines for Chinese characters. While not sufficient by themselves in most cases, baselines in combination with bounding polygons can be ingested by modern line-based text recognizers with minimal adaptation while at the same time requiring only modest effort for manual annotation.

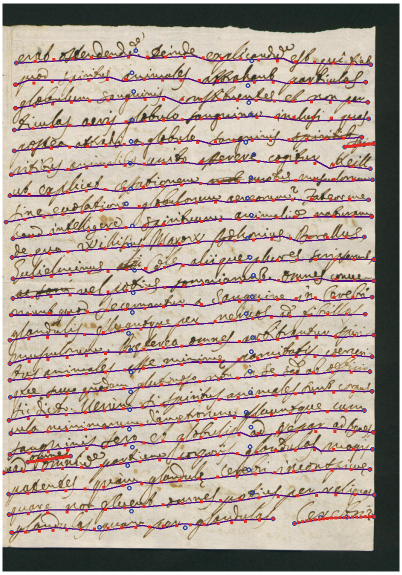

Fig. 1. Example taken from the cBAD dataset visualizing the originally annotated baseline (purple), least squares fitted Bézier curve control points (blue circles), and Bézier curve interpolated at 20 equally spaced points.

The dominant approach to baseline detection with trainable methods employing artificial neural networks is pixel-wise classification on single or multi-scale feature maps to label baseline pixels or some derivation thereof such as corpus height lines, toplines, or centerlines. These are sometimes augmented by auxiliary labels to improve accuracy or enable computation of other line characteristics such as orientation. In a postprocessing step baseline instances are extracted through grouping of baseline pixels, typically through thresholding, local connectedness, skeletonization, or interline distance estimation. Examples of this class of systems are ARU-Net [11], dhSegment [23], and BLLA [13]. While sufficiently powerful for many applications, semantic segmentation-based text line extraction has limits: most systems lack a way to determine text line orientation, have a tendency to merge and/or split close lines, are conceptually unable to deal with intersecting lines, and often impose further limitations on possible line shapes in the postprocessing stage.

3 Contribution

The main contribution of this work is a document layout analysis system based on object detection paradigms that is:

1. largely **postprocessing free** having one parameter during inference, a simple threshold of the objectness score.

2. almost **unconstrained** with regard to shape, orientation, and overlapping of the **type of text lines** to be extracted.
3. **extensible** towards other tasks, e.g. text line boundary and region detection, document classification, or reading order determination.

4 The CurT Model

CurT is heavily inspired by the DETR system for object detection. As such it has the same fundamental components: a set prediction loss forcing unique matching between predicted and ground truth baseline curves, and an architecture that predicts in a single pass a set of objects and models their relation.

For the reasons described above the verbatim method proposed in the seminal paper is largely unsuitable as a layout analysis system for historical documents. The limits imposed by the computational and memory complexity on input image feature size cause suboptimal performance on the detection of small text lines and the long training times required for convergence make learning for new material, a frequent requirement when considering the variety of historical writing one might want to segment, impractical.

From the variants presented above we adapt Conditional DETR [22] and modify it to better reflect our data model.

4.1 Text Line Data Model

The principal difference between the output of an off-the-shelf object detector and our text line extractor is the modelisation of the detected objects instances as polylines that are placed typically on the bottom of the line corpus, the baseline. As DETR-style models regress object instances encoded with a fixed dimensionality a flexible, fixed-length line representation that is able to deal with arbitrarily shaped text is needed. While directly regressing the end points of line segments of a polyline is possible, the high output dimensionality required to accurately model complex text shapes and the smoothness of handwriting makes this approach unappealing. Bézier curves on the other hand are able to represent complex shapes with a low, fixed number of control points.

A Bézier curve represents a parametric curve $c(t)$ that uses the Bernstein Polynomial as its basis:

$$c(t) = \sum_{i=0}^{n} b_i B_{i,n}(t), 0 \leq t \leq 1 \tag{1}$$

with n being the degree of the curve, b_i the i-th control point, and $B_{i,n}(t)$ the Bernstein basis polynomial:

$$B_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, i = 0, \ldots, n \tag{2}$$

To find the appropriate degree of the Bézier representation, a number of pages from the pre-training dataset (see Sect. 5.1) where sampled and the manually annotated polylines therein fit to curves of differing degrees. Cubic Bézier curves $n = 3$ were sufficient to model the lines in these document images to a sufficient degree (Fig. 1).

Converting the original polyline points to a cubic Bézier curve is done with a standard least squares fitting of the curve control points; the first and last points of the polyline are set as the first and last control points respectively. As polylines with a low number of line segments result in inaccurately parametrized curves, ground-truth lines are interpolated to contain at least 8 points.

4.2 Curve Detection Set Prediction Loss

As DETR and its derivations, CurT infers a set of N predictions of objects, in our case baseline curves encoded as cubic Bézier curves, and their associated class in a single decoder pass. Typically the number of possible text line predictions N is somewhat larger than the actual number of lines found in the image, so the possible classes are padded with a no object class $\varnothing$.

The loss operates in two stages: a matching phase where each predicted object (curve) is assigned to a ground truth object through an optimal bipartite matching computed with the Hungarian algorithm, followed by a loss optimizing the object-specific curve losses. The overall structure is similar to the set prediction loss proposed in [2] but modified to account for the prediction of baseline curves instead of bounding boxes.

Given the ground truth y and $\hat{y} = \{\hat{y}_i\}_{i=1}^{N}$, the set of N predictions with y padded with $\varnothing$ (no object) if smaller than N, we find an optimal bipartite matching between these two sets as a permutation of N elements $\sigma \in \mathfrak{S}_N$ with the lowest cost:

$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg\min} \sum_{i}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) \tag{3}$$

where $\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$ is a pair-wise matching cost between ground truth y_i and a prediction with index $\sigma(i)$.

The matching cost is a linear combination of the class prediction and the similarity between predicted and ground truth curves. Let the i-th ground truth element be $y_i = (t_i, c_i)$ where t_i is the target class label (which may be $\varnothing$) and $c_i \in [0, 1]^8$ is a vector defining the coordinates of the four control points relative to the image size.

For such an element y_i, we define the matching cost as $\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$ as $-\mathbb{1}_{\{t_i \neq \varnothing\}} \alpha \mathcal{L}_{\text{focal}}(t_i, \hat{p}_{\sigma(i)}(t_i)) + \mathbb{1}_{\{t_i \neq \varnothing\}} \beta \ell_1(c_i, \hat{c}_{\sigma(i)})$ given for a prediction with index $\sigma(i)$ the probability of class t_i as $\hat{p}_{\sigma(i)}(t_i)$ and the curve prediction as $\hat{c}_{\sigma(i)}$. $\alpha = 1.0$ and $\beta = 5.0$ are free parameters defining the relative weight between class and curve score in the matching cost.

Once a optimal matching has been computed, the Hungarian loss is computed on the matched pairs. Similar to the matching cost it is a linear combination of class prediction focal loss and the curve loss:

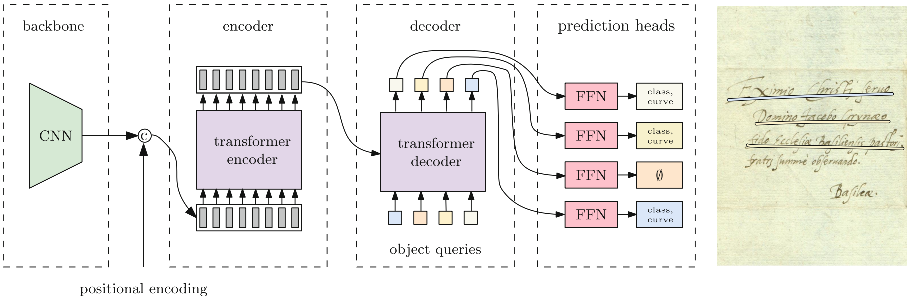

Fig. 2. CurT operates in a standard Transformer encoder-decoder configuration on feature maps computed with a conventional convolutional backbone. The feature maps from the backbone are flattened and positional encodings are concatenated to it before being passed into the encoder. The decoder then takes as input a fixed number of learned positional embeddings, called *object queries* that are mapped with a linear projection into reference points (see Fig. 3) and the encoder embeddings. Its output embeddings are then decoded into separate class scores and curve regressions with a shared feed-forward network (FFN).

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[\gamma \mathcal{L}_{\text{focal}}(t_i, \hat{p}_{\hat{\sigma}(i)}(t_i)) + \mathbb{1}_{\{t_i \neq \varnothing\}} \epsilon \ell_1(c_i, \hat{c}_{\hat{\sigma}}(i)) \right] \tag{4}$$

where $\hat{\sigma}$ is the optimal assignment computed in the first step (3) and $\gamma = 1.0$ and $\epsilon = 5.0$ are the relatives weights accorded to classification and regression losses respectively.

The component losses in both the matching and loss computation phase are focal loss with loss weight of four [17] for object classification and ℓ_1 distance for the curve regression.

4.3 CurT Architecture

The overall architecture of CurT depicted in Fig. 2 is based on a modification of the Conditional DETR [22] variant of the originally proposed Detection Transformer. It contains three main components: a convolutional backbone extracting an input feature representation, an encoder-decoder Transformer, and two feed forward networks predicting line class and the Bézier curve control points respectively.

Backbone. The backbone network is a conventional CNN backbone generating a lower-resolution feature map $\mathbf{f} \in \mathbb{R}^{C \times H \times W}$ from an input image $\mathbf{x} \in \mathbb{R}^{3 \times H_0 \times W_0}$. Resnet-50 (used by DETR and most variants), SegFormer [35] (a multi-head attention-based architecture originally devised for efficient semantic segmentation), and EfficientNetv2 [29] were evaluated informally as possible backbones. All 3 architectures produce output feature maps of size $\mathbf{f} \in \mathbb{R}^{C \times \frac{H}{32} \times \frac{W}{32}}$

with variable C and higher resolution feature maps available from earlier layers. While the tests were only performed on a shortened training cycle, backbone choice and configuration seems to not impact the training loss of the model drastically apart from slower convergence with the SegFormer backbone and steeper drop in loss during early training with larger segmentation maps (and concomitant higher memory consumption at same input image resolution).

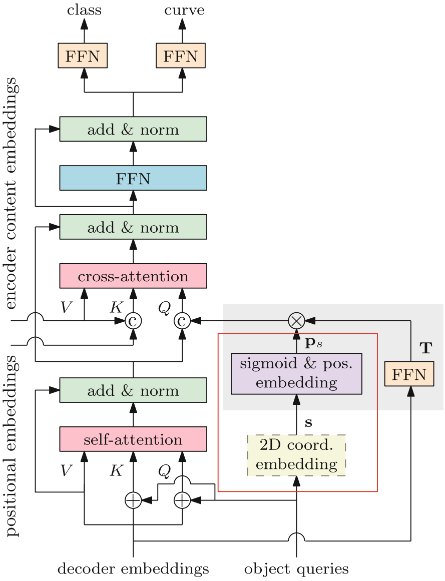

Fig. 3. A depiction of one decoder layer in CurT. The difference to the Conditional DETR decoder lies in the dimensionality of the reference points $\mathbf{s} \in \mathbb{R}^8$ with $\mathbf{p}_s \in [0, 1]^8$ used in the conditional spatial query construction (red box) and the output regression feed-forward network. $\mathbf{s}$ represents the unnormalized reference points, $\mathbf{p_s}$ the normalized reference points including the fixed positional encoding. Raw object query dimensionality remains unchanged. Original figure from [22].

Thus, the backbone model chosen is an EfficientNetV2-L pretrained on Imagenet 21k with the last layer feature maps $\mathbf{f} \in \mathbb{R}^{640 \times \frac{H_0}{32} \times \frac{W_0}{32}}$ being used as the input embeddings of the Transformer part of the model. This choice optimizes convergence speed, memory consumption, and computational requirements.

Transformer Encoder. The CurT encoder follows the standard Transformer encoder layer construction of a multi-head attention module and a feed forward network (FFN).

Encoder inputs are reduced in dimensionality with a 1×1 convolution of the input feature map $\mathbf{f}$ with $C = 640$ to an embedding feature map $\mathbf{z_0} \in \mathbb{R}^{d \times H \times W}$ with $d = 256$. As the standard Transformer encoder layers expect a sequence the two-dimensional map $\mathbf{z_0}$ is collapsed into a $d \times HW$ tensor. As with most other applications of the Transformer architecture in vision a fixed sinusoidal positional encoding [24] is applied to the encoder inputs to account for the Transformer's permutation invariance.

Transformer Decoder. The CurT decoder cross-attention mechanism in the decoder layers is largely identical to the construction of the Conditional DETR decoder, which modifies the DETR decoder cross-attention by decoupling queries into a content and spatial part by decoding the object queries into explicit reference points which are then concatenated to the decoder embeddings with the aim to accelerate training.

The primary difference in the construction of our decoder is the use of multiple reference points in the construction of the conditional spatial query (seegray-shaded box of Fig. 3) from object queries and decoder embeddings. Each object query is decoded not into a single center reference point $\mathbf{s_c} \in \mathbb{R}^2$ but four separate reference points $\mathbf{s} \in \mathbb{R}^8$. This is motivated in part by the formulation of the curve regression (see Sect. 4.3) but is primarily intended to aid the spatial attention mechanism to more easily deliminate the spatial extent of the baseline, similar to how singular reference points translate the attention to the extremities of the object box in the original architecture.

For a detailed description of the operation of the Transformer encoder and decoder we defer to [2,32] and [22] respectively.

Curve Regression. Following the regression scheme of Conditional DETR the control points of a candidate curve are predicted from each decoder layer as follows:

$$\mathbf{c} = \text{sigmoid}(\text{FFN}(\mathbf{g}) + \mathbf{s}) \tag{5}$$

where $\mathbf{g}$ is the decoder embedding, $\mathbf{c} \in [0,1]^8$ an eight-dimensional vector of the normalized curve control points, and $\mathbf{s} \in \mathbb{R}^8$ the unnormalized coordinates of the reference points. This differs from the originally proposed:

$$\mathbf{b} = \text{sigmoid}(\text{FFN}(\mathbf{g}) + [\mathbf{s_c}^\top 0 \ 0]^\top) \tag{6}$$

with $\mathbf{b} = [b_{cx} b_{cy} b_w b_h] \in [0,1]^4$ where the reference point only impacts the regression of the bounding box center point and extremities of the bounding box are completely regressed from the decoder embeddings.

The FFN in the curve regressor is a three-layer multi-layer-perceptron with ReLU activation function, a hidden dimension of d (256 per default as per above) and output dimension of eight.

Line Class Prediction. The classification score for each candidate curve is directly predicted from the decoder embeddings through an FNN followed by a softmax activation from each decoder layer:

$$\mathbf{t} = \text{softmax}(\text{FFN}(\mathbf{g})) \tag{7}$$

5 Experiments

5.1 Dataset and Evaluation Protocol

We perform experiments on the standard cBAD 2019 baseline detection dataset, containing 755 training, 778 validation, and 1511 test images. As this dataset is insufficient in size to train a CurT model from scratch, an auxiliary dataset of 38k annotated handwritten and machine-printed page images is assembled from the HTR-United repository [3], the NewsEye project [9], the Kuzushiji cursive Japanese dataset [4] with automatically annotated baselines, and an additional set of non-public data including highly challenging material from the Princeton

Geniza Project. The quality and annotation standards vary widely across this large dataset, often only containing annotations for parts of the text, a mixture of top-, center-, and baselines, and a variety of ontologies for text line and region classes. As there is little coherence across the chosen datasets and the standard cBAD evaluation scheme for text line detection disregards text line classification, line classes are merged into one default class and regions are suppressed.

5.2 Implementation Details

Strong augmentation is applied during training, with inputs being resized randomly to a longest edge size between 900 px and 1800 px, random rectangular crops followed by resizing, and random photometric distortion.

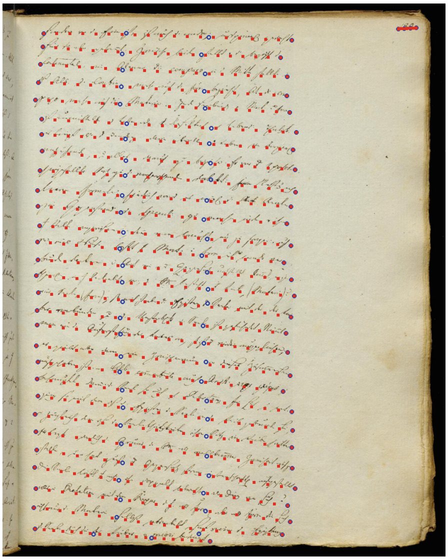

Fig. 4. Example output for a page taken from the cBAD dataset visualizing the Bézier curve control points (blue circles) and Bézier curve interpolated at 20 equally spaced points.

The number of object queries is increased to 1000 from the original 100 to account for the higher number of text lines on a typical page in comparison to objects annotated in the COCO 2017 images. As in other applications the number of queries is chosen to be in large excess of the possible number of objects in any input ($\mu = 7.19$ for objects per image for COCO 2017 resulting in 10–40 times the number of object queries for DETR and variants), following the same approach would increase computational requirements considerably for text line detection as the mean number of lines per page is 54.3 ($\sigma = 123.8$) with a small number of pages containing above 500 and even 1000 lines in comparison to the maximum 63 objects in a COCO image.

By default models are pretrained on the large general dataset of 38k page images for 100 epochs and then fine-tuned for an additional 50 epochs on the target dataset. The model is trained using the AdamW optimizer [20] with base learning rate of 10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay of 10^{-4} with a lower learning rate of 10^{-5} for the convolutional backbone. Learning rate is scheduled according to a single cosine cycle with an initial warmup over 8000 training steps as the more widely used fixed schedule with a 10-fold decay after 80% of epochs results in convergence at very high losses for the text line detection task. The number of encoder and decoder layers is set to 3 respectively. Traditional dropout with $p = 0.1$ is applied to the transformer. Auxiliary losses are computed on the output embedding decoded with the prediction heads sharing weights at each decoder layer.

5.3 Overall Performance

We report precision, recall, and F-value averaged over the 1511 test set images of the cBAD 2019 dataset in Table 1. Baseline results are from the winning method of the complex track of the cBAD 2017 competition. The metrics were computed with the standard schema described in [10]. A sample from the output on the test set is shown in Fig. 4.

Table 1. CurT text baseline detection performance on cBAD 2019 dataset (values for other methods from [7])

Method	Precision	Recall	F-value
Baseline (DMRZ-17)	0.773	0.743	0.758
TJNU	0.852	0.885	0.868
UPVLC	0.911	0.902	0.907
DMRZ	0.925	0.905	0.915
Planet	0.937	0.926	0.931
CurT	0.909	0.908	0.908

As shown by the competitive results in comparison to semantic segmentation-based methods our approach is able to detect text baselines effectively under various challenging conditions such as faded ink, degraded writing surfaces, and variously oriented lines.

5.4 Ordered Prediction

In addition to models trained with the set loss described above, an alternative formulation without bipartite matching was also evaluated. The chief purpose is to determine the ability of the system to learn a basic reading order in addition to text line detection by enforcing that the prediction at $\hat{y}_i, i \leq N$ corresponds to the y_i in the original ground truth. The basic assumption underlying this experiment is that reading order can be determined using fixed geometric relationships, i.e. that the spatial attention conditioned on the object queries is sufficient to determine a basic reading order.

While such a basic system would evidently be insufficient for practical purposes without the introduction of additional semantic depth like the distinction of headings, notes, insertions, main text, etc. its capabilities would be in line with the current state of the art of heuristics, learned rule based systems [21], and recent neural approaches [25].

An obvious challenge for this approach to ordered prediction is that object query utilization is highly dependent on the spatial frequency of baselines in the source document, i.e. object queries need to attend to areas of the document occurring earlier in the reading order for documents with a high number of text lines and later areas for sparse documents. As somewhat expected CurT failed to converge for this considerably more challenging task.

5.5 Further Extensions

A straightforward next target for a text line detection system is the extension to region detection and text line boundary detection. In DETR these tasks were analogously modelled as panoptic segmentation, predicting pixel-wise maps for both stuff and thing classes in COCO with a multi-attentional (M) mask head that predicts $M \times N$ attention maps simultaneously from the decoder embeddings. These attention maps are then upsampled through a FPN-like architecture incorporating multi-scale feature maps from the convolutional backbone network, followed by a classification layer to produce the final output pixel maps.

The drawback of simultaneous prediction of all segmentation maps is the linear increase of memory consumption with the number of object queries, in addition to the high base memory requirements for high resolution inputs. A future extension to CurT is a mask head predicting regions and text line boundaries sequentially.

6 Conclusion

This work presents the first attempt to adapt a modern direct object detection system for the task of text baseline detection in historical documents. The capabilities of this approach are demonstrated on the widely used cBAD 2019 dataset where the proposed method was shown to perform well. While only rudimentarily explored at this time, the proposed framework offers the perspective to solve a number of ancillary tasks to document layout analysis such as region detection and text line boundary detection, or reading order computation.

References

1. Breuel, T.M.: Two geometric algorithms for layout analysis. In: Lopresti, D., Hu, J., Kashi, R. (eds.) DAS 2002. LNCS, vol. 2423, pp. 188–199. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45869-7_23
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 213–229. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_13
3. Chagué, A., Clérice, T., Romary, L.: HTR-united : mutualisons la vérité de terrain! In: DHNord2021 - Publier, partager, réutiliser les données de la recherche : les data papers et leurs enjeux. MESHS, Lille, France (2021)
4. Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., Ha, D.: Deep learning for classical Japanese literature. CoRR abs/1812.01718 (2018)
5. Colutto, S., Kahle, P., Hackl, G., Mühlberger, G.: Transkribus. a platform for automated text recognition and searching of historical documents. In: 15th International Conference on eScience, eScience 2019, San Diego, CA, USA, 24–27 September 2019, pp. 463–466. IEEE (2019)
6. Dai, Z., Cai, B., Lin, Y., Chen, J.: UP-DETR: unsupervised pre-training for object detection with transformers. CoRR abs/2011.09094 (2020)
7. Diem, M., Kleber, F., Sablatnig, R., Gatos, B.: cBAD: ICDAR 2019 competition on baseline detection. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1494–1498 (2019)

8. Dosovitskiy, A., et al.: An image is worth 16 × 16 words: transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021. OpenReview.net (2021)
9. Doucet, A., et al.: NewsEye: a digital investigator for historical newspapers. In: Estill, L., Guiliano, J. (eds.) 15th Annual International Conference of the Alliance of Digital Humanities Organizations, DH 2020, Ottawa, Canada, 20–25 July 2020, Conference Abstracts (2020)
10. Gruning, T., Labahn, R., Diem, M., Kleber, F., Fiel, S.: READ-BAD: a new dataset and evaluation scheme for baseline detection in archival documents. In: 13th IAPR International Workshop on Document Analysis Systems, DAS 2018, Vienna, Austria, 24–27 April 2018, pp. 351–356. IEEE Computer Society (2018)
11. Grüning, T., Leifert, G., Strauß, T., Michael, J., Labahn, R.: A two-stage method for text line detection in historical documents. Int. J. Doc. Anal. Recogn. **22**(3), 285–302 (2019). https://doi.org/10.1007/s10032-019-00332-1
12. Khan, S.H., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: a survey. CoRR abs/2101.01169 (2021)
13. Kiessling, B.: A modular region and text line layout analysis system. In: 17th International Conference on Frontiers in Handwriting Recognition, ICFHR 2020, Dortmund, Germany, 8–10 September 2020, pp. 313–318. IEEE (2020)
14. Kiessling, B., Tissot, R., Stokes, P.A., Ezra, D.S.B.: eScriptorium: an open source platform for historical document analysis. In: 2nd International Workshop on Open Services and Tools for Document Analysis, OST@ICDAR 2019, Sydney, Australia, 22–25 September 2019, pp. 19. IEEE (2019)
15. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: DN-DETR: accelerate DETR training by introducing query denoising. CoRR abs/2203.01305 (2022)
16. Likforman-Sulem, L., Zahour, A., Taconet, B.: Text line segmentation of historical documents: a survey. Int. J. Doc. Anal. Recogn. **9**(2–4), 123–138 (2007). https://doi.org/10.1007/s10032-006-0023-z
17. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, 22–29 October 2017, pp. 2999–3007. IEEE Computer Society (2017)
18. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
19. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. CoRR abs/2103.14030 (2021)
20. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019. OpenReview.net (2019)
21. Malerba, D., Ceci, M., Berardi, M.: Machine learning for reading order detection in document image understanding. In: Marinai, S., Fujisawa, H. (eds.) Machine Learning in Document Analysis and Recognition, Studies in Computational Intelligence, vol. 90, pp. 45–69. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-76280-5_3
22. Meng, D., et al.: Conditional DETR for fast training convergence. In: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, 10–17 October 2021, pp. 3631–3640. IEEE (2021)
23. Oliveira, S.A., Seguin, B., Kaplan, F.: dhSegment: a generic deep-learning approach for document segmentation. In: 16th International Conference on Frontiers in Handwriting Recognition, ICFHR 2018, Niagara Falls, NY, USA, 5–8 August 2018, pp. 7–12. IEEE Computer Society (2018)

24. Parmar, N., et al.: Image transformer. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018. Proceedings of Machine Learning Research, vol. 80, pp. 4052–4061. PMLR (2018)
25. Quirós, L., Vidal, E.: Reading order detection on handwritten documents. Neural Comput. Appl. **34**(12), 9593–9611 (2022). https://doi.org/10.1007/s00521-022-06948-5
26. Redmon, J., Farhadi, A.: YOLOv3: an incremental improvement. CoRR abs/1804.02767 (2018)
27. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, 7–12 December 2015, Montreal, Quebec, Canada, pp. 91–99 (2015)
28. Sun, Z., Cao, S., Yang, Y., Kitani, K.: Rethinking transformer-based set prediction for object detection. CoRR abs/2011.10881 (2020)
29. Tan, M., Le, Q.V.: EfficientNetV2: smaller models and faster training. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 10096–10106. PMLR (2021)
30. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: fully convolutional one-stage object detection. In: ICCV (2019)
31. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 10347–10357. PMLR (2021)
32. Vaswani, A., et al.: Attention is all you need. In: NeurIPS (2017)
33. Wang, W., Cao, Y., Zhang, J., Tao, D.: FP-DETR: detection transformer advanced by fully pre-training. In: International Conference on Learning Representations (2022)
34. Wang, W., et al.: Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, 10–17 October 2021, pp. 548–558. IEEE (2021)
35. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: simple and efficient design for semantic segmentation with transformers. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, 6–14 December 2021, virtual, pp. 12077–12090 (2021)
36. Yang, J., et al.: Focal self-attention for local-global interactions in vision transformers. CoRR abs/2107.00641 (2021)
37. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: deformable transformers for end-to-end object detection. CoRR abs/2010.04159 (2020)