Journal of *Imaging*

MDPI

Article

Explainable Connectionist-Temporal-Classification-Based Scene Text Recognition

Rina Buoy [1,*], Masakazu Iwamura [1], Sovila Srun [2] and Koichi Kise [1]

[1] Department of Core Informatics, Graduate School of Informatics, Osaka Metropolitan University, Osaka 599-8531, Japan; masa.i@omu.ac.jp (M.I.); kise@omu.ac.jp (K.K.)
[2] Department of Information Technology Engineering, Faculty of Engineering, Royal University of Phnom Penh, Phnom Penh 12156, Cambodia; srun.sovila@rupp.edu.kh
* Correspondence: sp22676n@st.omu.ac.jp

Abstract: Connectionist temporal classification (CTC) is a favored decoder in scene text recognition (STR) for its simplicity and efficiency. However, most CTC-based methods utilize one-dimensional (1D) vector sequences, usually derived from a recurrent neural network (RNN) encoder. This results in the absence of explainable 2D spatial relationship between the predicted characters and corresponding image regions, essential for model explainability. On the other hand, 2D attention-based methods enhance recognition accuracy and offer character location information via cross-attention mechanisms, linking predictions to image regions. However, these methods are more computationally intensive, compared with the 1D CTC-based methods. To achieve both low latency and model explainability via character localization using a 1D CTC decoder, we propose a marginalization-based method that processes 2D feature maps and predicts a sequence of 2D joint probability distributions over the height and class dimensions. Based on the proposed method, we newly introduce an association map that aids in character localization and model prediction explanation. This map parallels the role of a cross-attention map, as seen in computationally-intensive attention-based architectures. With the proposed method, we consider a ViT-CTC STR architecture that uses a 1D CTC decoder and a pretrained vision Transformer (ViT) as a 2D feature extractor. Our ViT-CTC models were trained on synthetic data and fine-tuned on real labeled sets. These models outperform the recent state-of-the-art (SOTA) CTC-based methods on benchmarks in terms of recognition accuracy. Compared with the baseline Transformer-decoder-based models, our ViT-CTC models offer a speed boost up to 12 times regardless of the backbone, with a maximum 3.1% reduction in total word recognition accuracy. In addition, both qualitative and quantitative assessments of character locations estimated from the association map align closely with those from the cross-attention map and ground-truth character-level bounding boxes.

Keywords: vision Transformer; connectionist temporal classification; scene text recognition; character localization; model explainability

Citation: Buoy, R.; Iwamura, M.; Srun, S.; Kise, K. Explainable Connectionist-Temporal-Classification-Based Scene Text Recognition. *J. Imaging* **2023**, *9*, 248. https://doi.org/10.3390/jimaging9110248

Academic Editor: Ioannis Pratikakis

Received: 2 October 2023
Revised: 7 November 2023
Accepted: 10 November 2023
Published: 15 November 2023

1. Introduction

Scene text recognition (STR) identifies text in natural scenes and remains a vibrant research field due to challenging imaging conditions [1,2]. Current deep learning methods for STR typically comprise a visual feature extractor, a sequence modeler, and a decoder. The choice of decoder significantly impacts model recognition performance, latency, and explainability, given the same feature extractor and sequence modeler design. State-of-the-art (SOTA) methods categorize by their decoding of visual features into characters using connectionist temporal classification (CTC) and attention-based and Transformer decoders [3–5].

A 2D attention-based or Transformer decoder, using 2D feature maps, excels in recognition accuracy and character localization through a cross-attention mechanism. Unlike

Transformer-based object detectors, such as DETR [6] and V-DETR [7], which directly output object bounding boxes, a Transformer-based text recognizer outputs only characters. These characters can be localized via the decoder's cross-attention map. With enough inductive biases, including locality, the Transformer decoder attends only to the locations of the objects of interest [7]. Thus, the Transformer decoder generates a cross-attention map, linking predicted characters to relevant image regions. This location information yields benefits like model explainability [8–12] and text rectification [13]. Figure 1(2) exemplifies the overlaid cross-attention maps (summed across predicted characters) from a Transformer decoder, illustrating alignment between character positions and attention weights. However, it should be noted that the attention-based decoder has high latency due to an intricate attention mechanism [3,14].

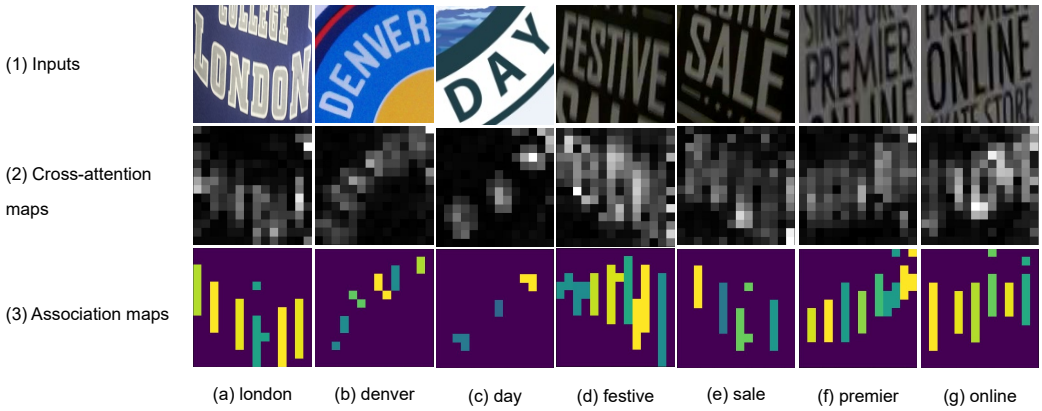


Figure 1. The cross-attention vs. the association maps. The first row consists of text images. The second and third rows consist of the cross-attention and association maps, respectively, that associate each predicted character with image regions. The last row consists of text transcriptions. The cross-attention map is obtained from a Transformer decoder, while the association map is obtained from a ViT-CTC model. Best viewed in color.

Conversely, the CTC decoder offers superior latency efficiency but often sacrifices recognition accuracy compared with the attention-based decoder [3,4,14]. The CTC decoder demands a 1D class probability distribution sequence input, prompting the common use of a 1D feature extractor in existing CTC-based methods [14–18]. However, this approach hampers the ability to establish explainable 2D spatial relationship between the predicted characters and relevant image regions. The 2D-CTC [19] method emerged to handle 2D feature maps, extending the 1D CTC algorithm to process the height dimension. However, using 2D-CTC involves a trade-off, resulting in higher inference latency and training costs, particularly with larger 2D feature maps.

For explainable character localization using a 1D CTC decoder, we introduce a ViT-CTC STR architecture that enables a 1D CTC decoder with a pretrained vision Transformer (ViT) to act as a 2D feature extractor. To incorporate the 2D feature extractor, we propose a novel marginalization-based technique that predicts 2D joint probability distributions over the height and class dimensions. By marginalizing the height dimension, we obtain a 1D class probability distribution sequence suited for a 1D CTC decoder.

Our proposed method also generates an association map, serving for character localization and model prediction explanation. This map resembles the role of a cross-attention map in the attention-based architectures but with significantly lower computational demand. Qualitative comparisons between the overlaid cross-attention and association maps are depicted in Figure 1(2),(3), respectively, showcasing alignment. Moreover, unlike 2D-CTC [19], our method maintains consistent inference latency and training cost, regardless of 2D feature map size. To quantitatively measure the alignment between character positions from the association map and the ground-truth character locations, we propose an alignment evaluation metric (AEM).

Our contributions can be summarized as follows:

1. We introduce a novel marginalization-based method for enabling a 2D feature extractor to be compatible with a 1D CTC decoder. This method yields an association map that links predicted characters to relevant image regions, enabling character localization and improving prediction explainability.
2. We derive an alignment evaluation metric (AEM) that measures the alignment between character positions from the association map and the ground-truth character locations. This metric can also be used for the cross-attention map.
3. Using our method, we experimented with the ViT-CTC architecture with various pretrained ViT backbones and a 1D CTC decoder. Our ViT-CTC models outperform the recent SOTA methods on public benchmark datasets.
4. Compared with a Transformer-decoder-based model, a ViT-CTC model offers a remarkable speed boost, surpassing the former by up to 12 times, regardless of the ViT backbone used. This speed gain comes with a maximum reduction in total word recognition accuracy of 3.1%. Hence, the ViT-CTC model is particularly attractive for low-latency, resource-constrained environments.

1.1. Related Work

In this section, we provide a brief review of common decoders in mainstream scene text recognition (STR) architectures. In addition, we also describe the recent advances of vision Transformer (ViT) architectures and their adoptions in STR, followed by model explanation through visualizations.

1.1.1. Scene Text Recognition

Scene text recognition is a variant of unsegmented sequence labeling tasks in which a 2D input stream of pixels is labeled with a sequence of characters. Other similar perceptual tasks include speech and gesture recognition [20].

Graves et al. [20] introduced the CTC algorithm, which maps a recurrent neural network (RNN) output sequence of a speech signal to a character sequence. CTC incorporates a blank token (ϵ) to handle multiple input-to-output alignments. Instead of predicting a probability of a single alignment, CTC estimates a total probability by marginalizing over all possible alignments.

CTC gained popularity in text recognition, leading to numerous CTC-based STR methods [14–18]. These methods typically employ a common pipeline encompassing optional rectification, a 1D convolutional feature extractor, a recurrent sequence modeler, and a 1D CTC decoder. While most CTC-based methods were initially designed for the Latin script, Gunna et al. [21] and Hu et al. [4] extended the CTC-based recognition pipeline to different Indian and Vietnamese scripts, respectively. However, a 1D CTC-based approach (using a 1D feature extractor) is unable to establish explainable 2D spatial relationships between predicted characters and relevant image areas.

To tackle this, 2D-CTC [19], an extension of the 1D CTC algorithm with the height dimension, handles 2D feature maps. However, it leads to increased inference latency and training cost, particularly based on the height of feature maps. Moreover, there is a lack of standardized, optimized 2D-CTC implementations in prevalent deep learning frameworks.

In contrast to a 1D CTC decoder, an attention-based decoder accommodates both 1D and 2D feature extractors. One-dimensional attention-based methods [4,14,22,23] substitute a CTC decoder with an attention-based one to enhance recognition performance by capturing character dependencies. Recognizing limitations in accurately predicting characters within complex and curved text, 2D attention-based methods [9,24] emerged.

As Transformer networks [10] gained prominence, the Transformer decoder became the standard attention-based decoder, leading to Transformer-decoder-based methods [25,26]. Via cross-attention mechanisms, the attention-based decoder produces a cross-attention map, associating each predicted character with relevant input image regions. The cross-attention map is widely used for visual explanations of model predictions [8–12]. Despite

its superior performance, Baek et al. [14] showed that an attention-based decoder, using the same feature extractor, yields about three times higher latency than a CTC-based decoder.

1.1.2. Vision Transformer

Transformers [10] have established themselves in natural language processing (NLP). Vision Transformers (ViT) [27] extend this architecture to vision tasks by dividing images into patches and projecting them as tokens, similar to words in NLP. The ViT's training demands are computationally efficient, but it lacks inductive biases. Addressing this, effective ViT models require substantial training data (priors). Data-efficient image Transformers [28–30] were introduced to alleviate data demands, achieving competitive outcomes against convolutional networks. ViT swiftly integrated into existing STR setups as a 2D feature extractor and sequence modeler. ViT-based STR methods [1,5,31] were subsequently proposed, displaying SOTA performance, particularly when trained on real labeled data.

1.1.3. Visual Model Explanations

To help users understand model failure and discover biases in training data, transparent models are necessary [32]. Nevertheless, deep neural networks (DNNs) behave as black boxes, making them difficult to understand. According to Junkang and Joe [32], an explanation map is a map that highlights relevant regions that contribute to a model's decision. The explanation can be obtained by using class activation mapping (CAM)-based or attention-based methods. Gradient-weighted class activation mapping (Grad-CAM) [33] is an example of CAM-based methods. Grad-CAM computes the gradients of a given class to produce a low-resolution localization map that highlights relevant image regions. Xu et al. [8] utilized an attention mechanism and visualized the attention map to show human intuition-like alignments between a model-generated caption and relevant image regions.

2. Materials and Methods

2.1. Proposed Method

In our study, ViT-CTC models leverage pretrained vision Transformers and a 1D CTC decoder. This allows our models to draw on extensive visual pretraining and exploit 2D spatial feature relationships via self-attention layers, all while retaining the low latency of a 1D CTC decoder. The introduced marginalization-based method also facilitates character localization and model prediction explanations through a novel association map that is absent in the existing 1D CTC-based methods.

In this section, we present the details of our proposed marginalization-based method in 2D class probability space. We begin by providing a concise overview of the 1D CTC algorithm and its assumptions in Section 2.1.1, followed by the detailed derivations of the proposed method in Section 2.1.2. We formulate the association map that relates each model prediction to relevant image regions in Section 2.1.3. Lastly, we derive an alignment evaluation metric (AEM) that measures the alignment between character locations estimated using the association and cross-attention maps and ground-truth character locations in Section 2.1.4.

2.1.1. Connectionist Temporal Classification (CTC)

CTC assigns a total probability of an output sequence ($\boldsymbol{Y}$) given an input sequence ($\boldsymbol{X}$) [20,34,35]. Instead of assigning a probability to the most likely alignment, CTC estimates a total probability by summing over all possible alignments between an input and output sequence. CTC introduces a blank or no-label token (ϵ) to allow the alignments and the input to have the same length. For any alignment, repeated characters are merged and blank tokens are removed to produce a final output sequence. For example, $\boldsymbol{A}_1 = (\epsilon, c, \epsilon, a, \epsilon, t)$

and $A_2 = (c, c, \epsilon, a, \epsilon, t)$ are two of the possible and valid alignments for the same word, *cat*. Mathematically, the total probability assigned by CTC is given by [20,34,35]

$$p(\boldsymbol{Y}|\boldsymbol{X}) = \sum_{\boldsymbol{A} \in \boldsymbol{S}_{\boldsymbol{X},\boldsymbol{Y}}} \prod_{t=1}^{W'} p_t(a_t|\boldsymbol{X}), \tag{1}$$

where $p(\boldsymbol{Y}|\boldsymbol{X})$ is a total probability of an $(\boldsymbol{X}, \boldsymbol{Y})$ pair. $\boldsymbol{A} = (a_1, \ldots, a_{W'})$ is an alignment and $\boldsymbol{S}_{\boldsymbol{X},\boldsymbol{Y}} = (\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n)$ is a set of possible and valid alignments between $\boldsymbol{X}$ and $\boldsymbol{Y}$. $p_t(a_t|\boldsymbol{X})$ is a conditional probability on $\boldsymbol{X}$ at a prediction frame, t. Thus, at each timestep t, a learning algorithm must produce a valid probability distribution (i.e., 1D vector) over characters. In the context of text recognition, the width dimension is treated as time while the height dimension is often collapsed by convolution and pooling layers.

Since $\boldsymbol{S}_{\boldsymbol{X},\boldsymbol{Y}}$ can be large, naive implementation is computationally inefficient. This is mitigated by dynamic programming by merging two alignments with the same output at the same t. Modern deep learning libraries have a built-in, optimized, efficient, low-level implementation of CTC. During inference, a greedy decoding scheme is used by selecting the most likely output at each prediction frame independently to obtain the highest probability alignment, $\boldsymbol{A}^*$, from which ϵ and duplicate characters are removed and merged, respectively [34]. The greedy and parallel decoding nature allows CTC to achieve low latency that is crucial in low-resource and real-time environments. $\boldsymbol{A}^*$ is given by

$$\boldsymbol{A}^* = \underset{\boldsymbol{A}}{\operatorname{argmax}} \prod_{i=1}^{T} p_t(a_t|\boldsymbol{X}). \tag{2}$$

The CTC algorithm makes the following assumptions [34]:

1. Conditional independence. The predicted characters are conditionally independent, meaning there are no dependencies between characters.
2. Monotonicity. When handling the subsequent feature vector, the current character can persist or the subsequent character must be processed.
3. Many to one. There can be multiple feature vectors corresponding to a single output character. This implies that the length of feature vectors must be greater than or equal to the length of target characters.

2.1.2. The Proposed Marginalization-Based Method

The concept of the proposed method is to handle 2D feature maps with a 1D CTC decoder without adding complexity. This is achieved by applying the marginalization rule in 2D class probability space.

Concretely, as shown in Figure 2, a ViT encoder takes an input image and produces 2D feature maps, represented by $\boldsymbol{F} = (\boldsymbol{F}_{1,1}, \ldots, \boldsymbol{F}_{H',W'})$, $\boldsymbol{F}_{i,j} \in \mathcal{R}^D$, where H', W', and D are the height, width, and embedding dimensions of the feature maps. $\boldsymbol{F}$ is directly fed to a linear layer to produce unnormalized 2D score distributions, $\boldsymbol{S} = (\boldsymbol{S}_{1,1}, \ldots, \boldsymbol{S}_{H',W'})$, $\boldsymbol{S}_{i,j} \in \mathcal{R}^C$. $\boldsymbol{S}$ is given by

$$\boldsymbol{S} = \textbf{LinearLayer}(\boldsymbol{F}), \tag{3}$$

where **LinearLayer** is a feedforward neural network. Each $\boldsymbol{S}_{i,j}$ is an unnormalized vector and C is the number of class labels. A softmax normalization is applied to $\boldsymbol{S}$ along both H' and C dimensions to produce $\boldsymbol{U} = (\boldsymbol{U}_{1,1}, \ldots, \boldsymbol{U}_{H',W'})$, $\boldsymbol{U}_{i,j} \in \mathcal{R}^C$. $\boldsymbol{U}$ is given by

$$\boldsymbol{U} = \textbf{Softmax}_{H',C}(\boldsymbol{S}), \tag{4}$$

where $\textbf{Softmax}_{H',C}$ is a softmax operator along the H' and C dimensions. A cross-section along W' is a valid 2D joint probability distribution over the H' and C dimensions. A 3D graphical illustration of $\boldsymbol{U}$ is provided in Figure 3.

Next, $\boldsymbol{U}$ is marginalized over the H' dimension to produce a sequence of valid 1D probability distributions over the C dimension, $\boldsymbol{P} = (\boldsymbol{P}_1, \ldots, \boldsymbol{P}_{W'})$, $\boldsymbol{P}_j \in \mathcal{R}^C$, that is required by a CTC decoder. $\boldsymbol{P}_j$ is given by

$$\boldsymbol{P}_j = \sum_{h=1}^{H'} \boldsymbol{U}_{h,j}, \tag{5}$$

where each $\boldsymbol{P}_j$ is a normalized class probability distribution vector. In the case of a 1D feature extractor (i.e., $H' = 1$), $\boldsymbol{U}$ is exactly $\boldsymbol{P}$. The overall text recognition workflow with the proposed method is shown in Figure 2.

Beyond the CTC algorithm's assumptions, our proposed method assumes horizontal or curved textline, excluding vertical orientation.

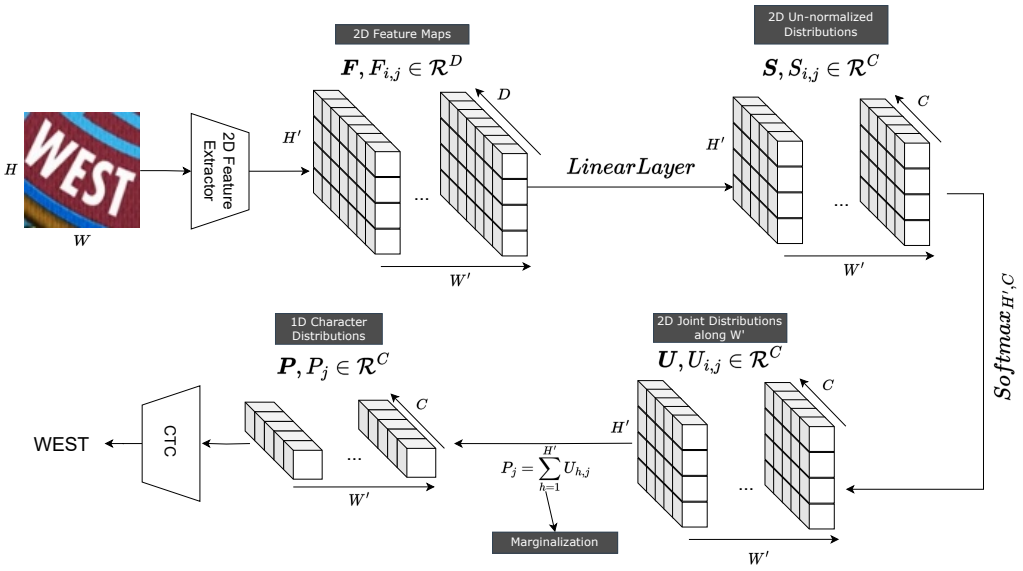


Figure 2. The proposed marginalization-based method: A 2D feature sequence, $\boldsymbol{F} = (\boldsymbol{F}_{1,1}, \ldots, \boldsymbol{F}_{H',W'})$, is produced by a 2D feature extractor such as a ViT backbone. $\boldsymbol{F}$ is fed to a linear layer to produce $\boldsymbol{S} = (\boldsymbol{S}_{1,1}, \ldots, \boldsymbol{S}_{H',W'})$ from which a softmax normalization is performed over both H' and C dimensions. Next, the normalized $\boldsymbol{U} = (\boldsymbol{U}_{1,1}, \ldots, \boldsymbol{U}_{H',W'})$ is marginalized over the H' dimension to produce $\boldsymbol{P} = (\boldsymbol{P}_1, \ldots, \boldsymbol{P}_{W'})$ that is fed to a CTC decoder. D and C are the feature and class dimensions, respectively.

2.1.3. Association Map (AM)

In the existing CTC-based methods, the height dimension is physically discarded by feature averaging or pooling layers. The proposed method preserves the height dimension, making a 2D feature extractor compatible with a CTC decoder.

Thanks to the proposed method, a cross-section along the W' dimension of $\boldsymbol{U}$ forms a valid 2D joint probability distribution over the H' and C dimensions, as shown in Figure 3. Based on $\boldsymbol{U}$, we can derive a novel association map (**AM**) that enables linking each predicted character to relevant image regions. This spatial connection serves two purposes: (1) explaining model predictions and (2) character localization.

The association map functions in the same way as the localization map of Grad-CAM [33], but without gradients, and the attention map [8], but without an attention mechanism. Concretely, given the most likely alignment, $\boldsymbol{A}^* = (a_1, \ldots, a_{W'})$, $\mathbf{AM} = (\mathrm{AM}_{1,1}, \ldots, \mathrm{AM}_{H',W'})$, $\mathrm{AM}_{i,j} \in \{0, 1\}$, is expressed as

$$\mathrm{AM}_{i,j} = \begin{cases} 1, & \text{if } \boldsymbol{U}_{i,j,\mathrm{ind}(a_j)} \geq \alpha \wedge a_j \neq \epsilon \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where j is a prediction timestep or frame. a_j is a CTC predicted character at j. $U_{i,j,\text{ind}(a_j)}$ is a probability of character, a_j, at timestep, j, and height, i. ind() is a character-to-index mapping. α is a threshold between zero and one while ϵ is a blank token required by a CTC decoder. A high α associates a predicted character, a_j, with the high probability image regions. The resulting character regions are illustrated in Figure 1(3).

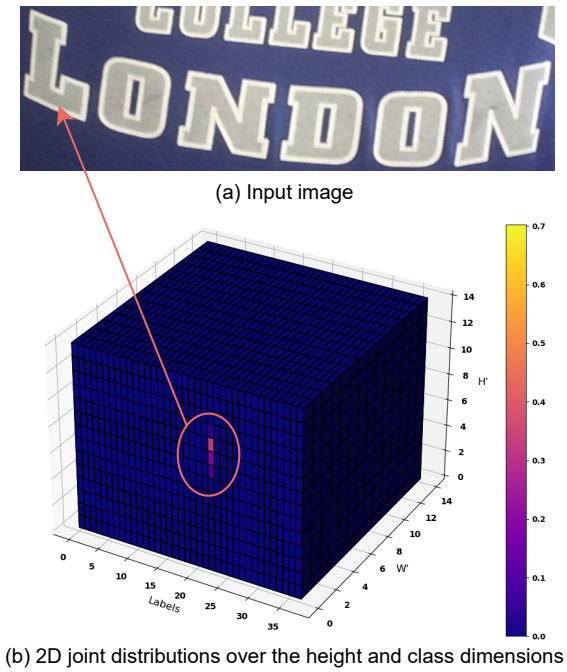


Figure 3. 3D graphical illustration of $\boldsymbol{U}$ for an input image. (**a**) Input image. (**b**) The computed $\boldsymbol{U}$. At $W' = 1$, the bright cells, responding to the character $\boldsymbol{L}$, have a high probability. Best viewed in color.

2.1.4. Alignment Evaluation Metric (AEM)

Model predictions are explicable through visualization of the association and cross-attention maps (Figure 1). We also quantitatively assess alignment between character positions in these maps and the ground truth character locations. Given the absence of explicit character coordinate predictions by the association and cross-attention maps, the intersection-over-union (IoU) metric is unsuitable. Instead, we introduce an alignment metric suitable for both association and cross-attention maps.

Concretely, given character regions R_k on the association map and GT_k as the ground-truth bounding box (depicted in Figure 4), the alignment evaluation metric (AEM) for a predicted character, k, is given by

$$\text{AEM}_k = \begin{cases} 1, & \text{if } R_k \cap \text{GT}_k \neq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The AEM for a given text of length, L, is given by

$$\text{AEM}_{TEXT} = \frac{\sum_{k=1}^{L} \text{AEM}_k}{L}. \tag{8}$$

In the case of the cross-attention map, we first sum the cross-attention map over all attention heads in the case of multi-headed attention mechanism and normalize for each predicted character, k, to obtain $\mathbf{CA} = (\text{CA}_{1,1}, \ldots, \text{CA}_{H',W'})$, $\text{CA}_{i,j} \in \mathcal{R} | 0 \leq \text{CA}_{i,j} \leq 1$. Examples of the resulting overlaid and normalized cross-attention map are given in Figure 5a. In contrast to the association map, the cross-attention map is more diffuse due to the decoder's

need to compute continuous attention weights across the entire feature maps. We filter out regions with low attention weights below the threshold, β. The filtered, binary cross-attention map in Figure 5b, $\mathbf{CAF} = (\mathrm{CAF}_{1,1}, \ldots, \mathrm{CAF}_{H',W'})$, $\mathrm{CAF}_{i,j} \in \{0,1\}$, is given by

$$\mathrm{CAF}_{i,j} = \begin{cases} 1, & \text{if } \mathrm{CA}_{i,j} \geq \beta \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

where β is between zero and one. A high β associates a predicted character, k, with the high attention weight regions. With the **CAF**, AEM_k and AEM_{TEXT} are computed, according to the above equations.

(**a**) Input image with ground-truth character bounding boxes

(**b**) Association map

Figure 4. The estimated character locations, R_k, from the association map. (**a**) Input image with ground-truth character bounding boxes, GT_k. (**b**) Estimated character regions. Best viewed in color.

(**a**) Cross-attention maps

(**b**) Filtered cross-attention maps

Figure 5. The estimated character locations, R_k, for the two predicted characters of the input image in Figure 4a, from the cross-attention maps. (**a**) Cross-attention maps. (**b**) Estimated character regions. Best viewed in color.

2.2. Datasets

2.2.1. Synthetic Datasets

Training on large-scale synthetic data is a common practice in STR. Four major synthetic datasets are MJSynth (MJ) [36], SynthText (ST) [37], SynthAdd (SA) [9], and SynthTiger [38]. The synthetic training set comprises 8.5M images from 50% of MJSynth, 50% of SynthText, 100% of SynthAdd, and 10% of SynthTiger. The mixing ratio is around 4:3:1.3:1. Combining different training sources is to increase diversity of training data. Some samples from the training datasets are shown in Figure 6a.

2.2.2. Real Datasets

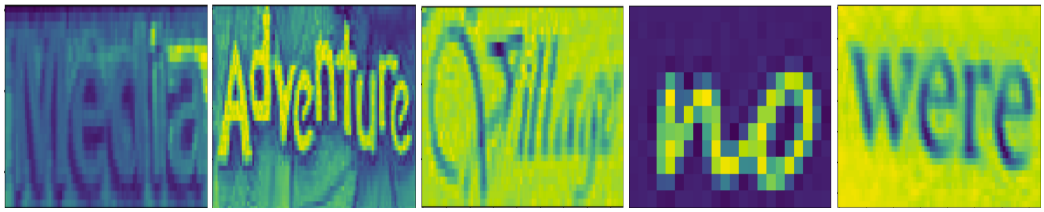
The evaluation datasets include the test sets of street view text (SVT) [39], IIIT5k-Words (IIIT) [40], ICDAR2013 (IC13) [41], ICDAR2015 (IC15) [42], SVT perspective (SVTP) [43], and CUTE80 (CT) [44]. Detailed descriptions of these datasets can be referred to [14,45].

COCO-Text (COCO) [46], RCTW [47], Uber-Text [48], ArT [49], LSVT [50], ReCTS [51], TextOCR [52], and OpenImages V5 [24] are small-scale, real labeled datasets. We used an aggregated, processed version of COCO-Text, RCTW, Uber-Text, ArT, LSVT, and ReCTS provided by Baek et al. [45]. For TextOCR and OpenImages V5, we used the processed versions provided by Yang et al. [5].

The fine-tuning datasets comprise the training sets of SVT, IIIT, IC03, IC13, IC15, and the real labeled datasets above. The idea of introducing the fine-tuning datasets

based on real labeled data is to identify whether our ViT-CTC models have any inherent weaknesses or if there are any blindspots in the training datasets [53]. The fine-tuning datasets comprise 2.4M labeled images. A few samples from the fine-tuning datasets are shown in Figure 6b.

(**a**) Synthetic training samples



(**b**) Real fine-tuning samples

Figure 6. Sample training and fine-tuning images. (**a**) Sample images from the training datasets. (**b**) Sample images from the fine-tuning datasets.

2.2.3. Synthetic Character-Level Annotation Dataset

Character-level annotations are not available with the existing datasets. Thus, to quantitatively evaluate the character locations derived from the association and cross-attention maps, we use SynthTiger (https://github.com/clovaai/synthtiger, accessed on 1 August 2023) to synthetically generate a small dataset of 446 scene text images with character-level bounding boxes. A few samples of the generated images with character-level annotations are given in Figure 7.

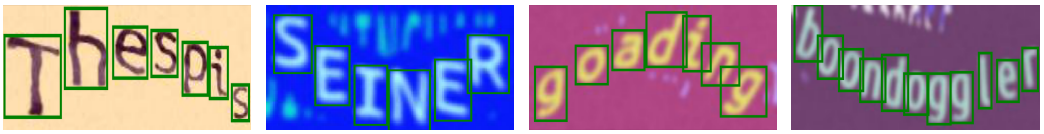


Figure 7. Sample text images with character-level annotations.

2.3. Experiment Design

We experimented with different backbones, including three variants of DeiT-III [29] (DeiT-Small, DeiT-Medium, and DeiT-Base) and a CaiT-Small [30]. The assessment of the backbone's complexity impact on recognition performance can be achieved by employing the DeiT-Small, DeiT-Medium, and DeiT-Base backbones. Furthermore, the inclusion of CaiT-Small enables us to compare the recognition performance of different ViT architectures.

The details of these four pretrained ViT backbones are shown in Table 1. For an input image of 224 $\times$ 224 pixels, the output feature maps are 14 $\times$ 14 $\times$ D, and D is the embedding dimension, which is provided in the same table for each ViT backbone. For each pretrained ViT backbone, we setup two ViT-CTC models, employing both the baseline feature averaging method (FA) [5] and the proposed marginalization method (M), presented in Section 2.1.1. In FA, feature maps are arithmetically averaged along the height or vertical dimension to produce a 1D feature sequence for a character classifier and a CTC decoder. As a result, it does not provide character location information.

Similarly, for recognition performance and latency comparison purposes, we also setup the Transformer-decoder-based models that are also based on the same ViT backbones, while the specifications of the Transformer decoder are provided in Table 2. It should be

noted that only our ViT-CTC models using the proposed marginalization method and the Transformer-decoder-based models can offer character locations in addition to recognition. The estimated character locations are qualitatively and quantitatively evaluated against the ground-truth locations.

Table 1. Specifications of the pretrained ViT backbones.

ViT Model	Params	GFLOPs	Size	Emb. Dim, D	Acc@1 (INet-1k)
DeiT-Small [30]	22.2M	4.6	224×224	384	81.4
DeiT-Medium [30]	38.8M	8.0	224×224	512	83.0
DeiT-Base [30]	86.6M	17.5	224×224	768	83.8
CaiT-Small [29]	47.0M	9.4	224×224	384	83.5

Table 2. Specifications of the Transformer decoder.

Parameters	Value
Hidden/Embedding Dimension	Encoder's Emb. Dim.
Decoder Stacks	3
Attention Heads	8
Dropout	0.1
Feed-forward Dimension	Encoder's Emb. Dim.

In the case of a CTC decoder, the character set comprises 37 characters, encompassing case-insensitive letters, numbers, and a blank token denoted as ϵ. On the other hand, for a Transformer decoder, the character set consists of 39 characters, including case-insensitive letters, numbers, and three distinct special tokens (PADDING: zero padding; EOS: end of sentence; SOS: start-of-sentence). The input images were resized to 224×224 pixels.

The training strategy comprised two phases: (1) training on the synthetic datasets and (2) fine-tuning on the real datasets. These two phases of training allow us to identify models' weaknesses or training datasets' blindspots during evaluation [53]. The training process lasted for 50 iterations. During each iteration, 300,000 images were randomly selected, and a batch of 64 images was used for training without any data augmentation to ensure a fair comparison with the SOTA methods [4,14]. In addition, because the synthetically generated training images were already augmented during generation, additional data augmentation, such as [54,55], may affect the recognition accuracy negatively [45]. The total training is equivalent to around two epochs on all of the training data. The fine-tuning phase followed the same settings as before, but it only lasted for 30 iterations, which is approximately equivalent to three epochs over the entire fine-tuning dataset. The cyclic learning schedules between 10^{-4} and 10^{-5} and between 10^{-5} and 10^{-6} were used for the training and fine-tuning phases, respectively. For all the models, pretrained ViT weights [29,30] were used with a gradient clip of ten.

3. Results

In this section, we present the experimental outcomes and important analyses. To evaluate the performance of our ViT-CTC models using the proposed method (M), we begin by providing the ablation analyses of the encoder complexities and architectures in Section 3.1, followed by comparing their accuracy with the baseline and SOTA-based methods that do not provide character locations in Sections 3.2 and 3.3. In Section 3.4, we compare with the baseline Transformer-decoder-based models that provide character locations via the cross-attention map. Lastly, we provide the qualitative and quantitative evaluation of character location derived from the proposed method and the cross-attention map.

3.1. Ablation Analyses of the Encoder Complexities and Architectures

In this section, we present the ablation analyses concerning ViT-based feature extractor complexities since the feature extractor is the main component in the proposed method. We

utilize three variants of DeiT backbones (namely DeiT-S, DeiT-M, and DeiT-B) and explore different encoder architectures employing a CaiT-S backbone.

Table 3 demonstrates that increasing the complexity of the ViT-based feature extractor, specifically transitioning from DeiT-S to DeiT-M and DeiT-B, results in higher total word recognition accuracy for both synthetic and real training data. However, these improvements are accompanied by larger model sizes and heightened computational demands, as indicated in Table 1. Table 3 also shows that despite having a much smaller model size and computational demand, the CaiT-S model achieves a comparable total recognition accuracy with the DeiT-B model for both synthetic and real training data.

Table 3. Word recognition accuracy (%) of the ablation results of the encoder complexities and architectures with the proposed method (M). FT: fine-tuning on real data. Bold: highest.

(**a**) Methods trained on synthetic training data (S).							
Method	**IIIT**	**SVT**	**IC13**	**IC15**	**SVTP**	**CUTE**	**Total**
DeiT-S + M (Ours)	91.4	85.5	91.3	75.3	76.7	82.2	85.3
DeiT-M + M (Ours)	92.5	**87.8**	**92.2**	76.6	**79.5**	81.9	86.6
DeiT-B + M (Ours)	93.0	86.9	92.2	**78.6**	79.1	84.0	**87.3**
CaiT-S + M (Ours)	**93.5**	86.9	91.9	77.6	77.8	**85.4**	87.2
(**b**) Methods trained on real labeled training data (R).							
Method	**IIIT**	**SVT**	**IC13**	**IC15**	**SVTP**	**CUTE**	**Total**
DeiT-S + M + FT (Ours)	94.6	89.2	95.4	81.5	83.1	91.3	89.9
DeiT-M + M + FT (Ours)	95.0	92.3	95.2	83.5	84.0	90.9	90.9
DeiT-B + M + FT (Ours)	95.9	**92.6**	**96.1**	84.4	84.3	**92.7**	**91.7**
CaiT-S + M + FT (Ours)	**96.1**	90.6	95.4	**84.9**	**85.4**	**92.7**	**91.7**

3.2. Recognition Accuracy Comparison with the Baseline Feature Averaging

In this section, we perform a comparison to assess the recognition accuracy of our ViT-CTC models using both the proposed method (M) and the baseline feature averaging (FA). Since FA does not yield character localization, the comparison in this section primarily centers around the recognition accuracy between the two methods.

As indicated in Tables 4a,b, there are minimal distinctions in terms of recognition accuracy between the two methods, regardless of source of training data (i.e., real and synthetic). The findings can be distilled into three primary points. Firstly, the proposed method, offering both model explainability and character location information, does not lead to any loss of recognition accuracy. Secondly, the utilization of a 2D feature extractor such as a ViT backbone improves the recognition accuracy of a CTC decoder, whereas the majority of CTC-based methods depend on a tailored 1D feature extractor. Thirdly, the utilization of real labeled data, albeit limited, results in a substantial recognition performance improvement compared with relying solely on synthetic training data.

3.3. Recognition Accuracy Comparison with the SOTA CTC-Based Methods

Similar to the preceding section, this section compares the recognition accuracy of our ViT-CTC models using our proposed method (M) with the SOTA CTC-based methods lacking character location information. Among the SOTA methods in Table 5, only the DiG-ViT [5] and GTC [4] models use real labeled data for training. The other models use solely synthetic data for training. The table suggests that integrating real labeled data can improve recognition accuracy on benchmark datasets. However, various factors like backbone architecture, training iterations, and data augmentation also play a significant role in this improvement. Among these methods, only ViTSTR [1] and DiG-ViT employ a ViT backbone; the rest rely on convolutional backbones. DiG-ViT employs the feature averaging technique to convert 2D feature maps to 1D for a CTC decoder. GTC [4] uses an attention-based decoder to guide a CTC decoder.

Table 4. Word recognition accuracy (%) comparison between the proposed method (M) and the baseline feature averaging (FA). FT: fine-tuning on real data. Bold: highest.

(**a**) Methods trained on synthetic training data (S).							
Method	**IIIT**	**SVT**	**IC13**	**IC15**	**SVTP**	**CUTE**	**Total**
DeiT-S + FA	91.4	86.4	89.6	74.2	75.8	79.1	84.7
DeiT-M + FA	92.0	87.3	91.4	77.4	78.9	82.2	86.4
DeiT-B + FA	93.1	**88.7**	**92.9**	77.3	**79.7**	85.7	87.4
CaiT-S + FA	**94.3**	87.2	92.5	**79.5**	79.4	**87.1**	**88.2**
DeiT-S + M (Ours)	91.4	85.5	91.3	75.3	76.7	82.2	85.3
DeiT-M + M (Ours)	92.5	87.8	92.2	76.6	79.5	81.9	86.6
DeiT-B + M (Ours)	93.0	86.9	92.2	78.6	79.1	84.0	87.3
CaiT-S + M (Ours)	93.5	86.9	91.9	77.6	77.8	85.4	87.2
(**b**) Methods trained on real labeled training data (R).							
Method	**IIIT**	**SVT**	**IC13**	**IC15**	**SVTP**	**CUTE**	**Total**
DeiT-S + FA + FT	95.0	88.4	94.2	81.6	82.0	88.5	89.6
DeiT-M + FA + FT	95.5	91.2	95.4	83.4	83.4	92.0	91.0
DeiT-B + FA + FT	95.9	92.1	95.9	83.9	84.2	92.7	91.5
CaiT-S + FA + FT	96.0	92.3	95.8	84.5	84.7	**93.7**	**91.7**
DeiT-S + M + FT (Ours)	94.6	89.2	95.4	81.5	83.1	91.3	89.9
DeiT-M + M + FT (Ours)	95.0	92.3	95.2	83.5	84.0	90.9	90.9
DeiT-B + M + FT (Ours)	95.9	**92.6**	**96.1**	84.4	84.3	92.7	**91.7**
CaiT-S + M + FT (Ours)	**96.1**	90.6	95.4	**84.9**	**85.4**	92.7	**91.7**

Table 5. Word recognition accuracy (%) comparison between the proposed method (M) and the SOTA CTC-based methods. FT: fine-tuning on real data. Size: parameters in millions. M: the proposed method. Bold: highest.

(**a**) Methods trained on synthetic training data (S).								
Method	**Size**	**IIIT**	**SVT**	**IC13**	**IC15**	**SVTP**	**CUTE**	**Total**
CRNN [15]	8.3	82.9	81.6	89.2	69.4	70.0	65.5	78.5
STAR-Net [18]	48.7	87.0	86.9	91.5	76.1	77.5	71.7	83.5
GRCNN [17]	4.6	84.2	83.7	88.8	71.4	73.6	68.1	80.1
Rosetta [16]	44.3	84.3	84.7	89.0	71.2	73.8	69.2	80.3
TRBC [14]	48.7	87.0	86.9	91.5	76.1	77.5	71.7	83.5
ViTSTR-S [1]	21.5	85.6	85.3	90.6	75.3	78.1	71.3	82.5
ViTSTR-B [1]	85.8	86.9	87.2	91.3	76.8	**80.0**	74.7	84.0
DeiT-S + M (Ours)	21.6	91.4	85.5	91.3	75.3	76.7	82.2	85.3
DeiT-M + M (Ours)	38.9	92.5	**87.8**	**92.2**	76.6	79.5	81.9	86.6
DeiT-B + M (Ours)	**85.7**	93.0	86.9	**92.2**	**78.6**	79.1	84.0	**87.3**
CaiT-S + M (Ours)	46.5	**93.5**	86.9	91.9	77.6	77.8	**85.4**	87.2
(**b**) Methods trained on real labeled training data (R).								
Method	**Size**	**IIIT**	**SVT**	**IC13**	**IC15**	**SVTP**	**CUTE**	**Total**
GTC [4]	-	96.0	91.8	93.2	79.5	85.6	91.3	90.1
DiG-ViT-T (CTC) [5]	20.0	93.3	89.7	92.5	79.1	78.8	83.0	87.7
DiG-ViT-S (CTC) [5]	36.0	95.5	91.8	95.0	84.1	83.9	86.5	91.0
DiG-ViT-B (CTC) [5]	52.0	95.9	92.6	95.3	84.2	85.0	89.2	91.5
DeiT-S + M + FT (Ours)	21.6	94.6	89.2	95.4	81.5	83.1	91.3	89.9
DeiT-M + M + FT (Ours)	38.9	95.0	92.3	95.2	83.5	84.0	90.9	90.9
DeiT-B + M + FT (Ours)	**85.7**	95.9	**92.6**	**96.1**	84.4	84.3	**92.7**	**91.7**
CaiT-S + M + FT (Ours)	46.5	**96.1**	90.6	95.4	**84.9**	**85.4**	**92.7**	**91.7**

Focusing on the models trained only on synthetic data (S), Table 5a shows that our ViT-CTC models using the proposed method (M) outperform the SOTA CTC-based methods,

such as TRBC (TPS-ResNet-BiLSTM-CTC) [14], in recognition accuracy (bold numbers in the table). This recognition accuracy improvement is attributed to the advanced feature extraction of pretrained ViT backbones. Meanwhile, when considering methods trained or fine-tuned on real labeled data (R), Table 5b shows that our ViT-CTC models slightly outperform the SOTA DiG-ViT models (bold numbers in the table). Thus, regardless of the training data source, our ViT-CTC models with the proposed method (M) consistently show superior or comparable performance to the SOTA CTC-based methods.

3.4. Recognition Accuracy and Efficiency Comparison with the Baseline Transformer-Decoder-Based Models

Earlier sections evaluated our proposed ViT-CTC models' recognition accuracy against the CTC-based methods that lack character localization. Now, we jointly compare recognition accuracy and latency with a Transformer-decoder-based architecture that can associate predicted characters with relevant image regions.

A CTC decoder is acknowledged for its faster inference but lower recognition accuracy compared with a Transformer decoder that learns an implicit language model [3–5,14]. This section quantitatively assesses the trade-off between the two decoders in terms of both latency and recognition accuracy.

Tables 6a,b compare the recognition accuracy of our ViT-CTC models using our proposed method against Transformer-decoder-based models. Regardless of the training data source, the Transformer-decoder-based models consistently achieved higher recognition accuracy on benchmark datasets due to their ability to capture character dependencies through implicit language modeling that is absent in a CTC decocder.

However, this recognition accuracy advantage was offset by increased latency, as shown in Figure 8 and Table 7. The inference time of a Transformer decoder is directly tied to the number of decoded characters, while a CTC decoder maintains a constant inference time. Quantitatively, the inference speed of a CTC decoder surpasses a Transformer decoder by up to 12 times, making it more appealing in low-latency and low-resource scenarios.

Table 6. Word recognition accuracy (%) comparison with the baseline Transformer-decoder-based models. FT: fine-tuning on real data. Size: parameters in millions. Tr. Dec.: Transformer decoder. M: the proposed method. Bold: highest.

(**a**) Methods trained on synthetic training data (S).

Method	Size	IIIT	SVT	IC13	IC15	SVTP	CUTE	Total
DeiT-S + Tr. Dec.	26.1	93.7	88.9	92.4	80.0	80.6	86.8	88.3
DeiT-M + Tr. Dec.	46.2	94.1	89.6	92.6	**81.5**	82.8	83.6	89.0
DeiT-B + Tr. Dec.	**103.4**	94.8	**90.3**	92.9	81.0	**85.1**	87.5	89.6
CaiT-S + Tr. Dec.	50.9	**94.9**	**90.3**	**94.2**	81.3	83.4	**89.9**	**89.9**
DeiT-S + M (Ours)	21.6	91.4	85.5	91.3	75.3	76.7	82.2	85.3
DeiT-M + M (Ours)	38.9	92.5	87.8	92.2	76.6	79.5	81.9	86.6
DeiT-B + M (Ours)	85.7	93.0	86.9	92.2	78.6	79.1	84.0	87.3
CaiT-S + M (Ours)	46.5	93.5	86.9	91.9	77.6	77.8	85.4	87.2

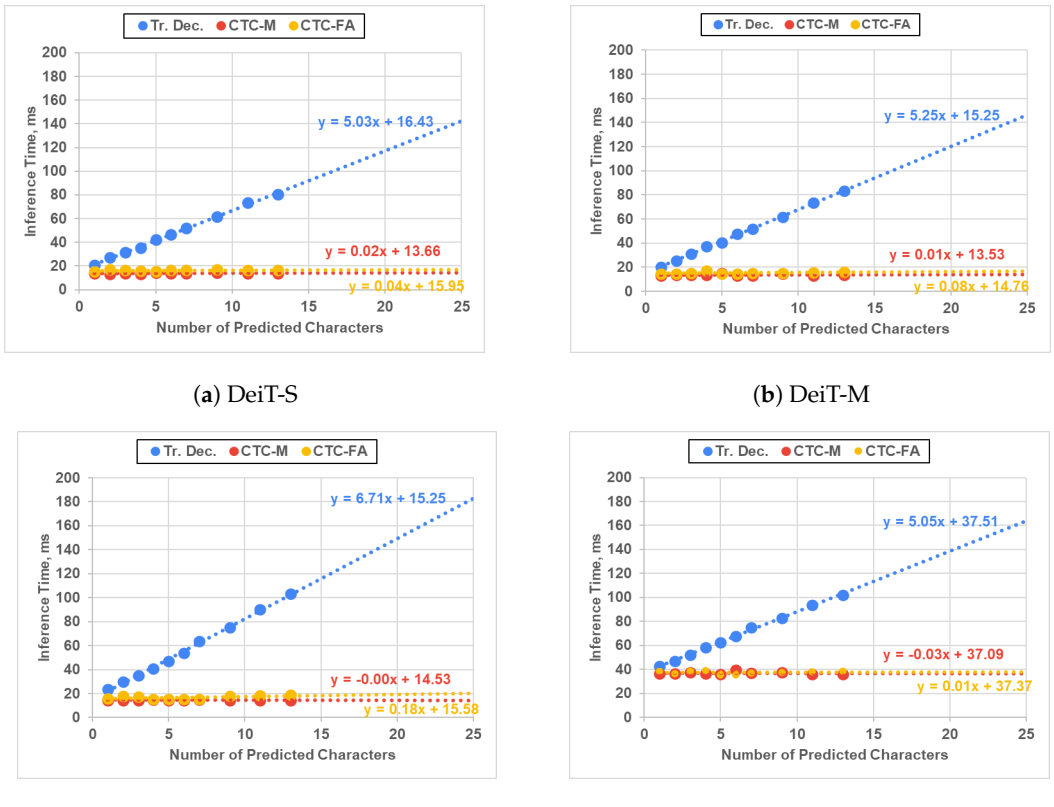
(**b**) Methods trained on real labeled training data (R).

Method	Size	IIIT	SVT	IC13	IC15	SVTP	CUTE	Total
DeiT-S + Tr. Dec. + FT	26.1	96.8	93.0	96.7	86.3	87.8	94.8	93.0
DeiT-M + Tr. Dec. + FT	46.2	97.0	94.0	97.1	86.3	89.3	95.1	93.4
DeiT-B + Tr. Dec. + FT	**103.4**	**98.0**	94.6	**97.5**	**86.9**	**90.5**	95.1	**94.2**
CaiT-S + Tr. Dec. + FT	50.9	97.4	**94.9**	97.1	86.5	89.5	**95.8**	93.7
DeiT-S + M + FT (Ours)	21.6	94.6	89.2	95.4	81.5	83.1	91.3	89.9
DeiT-M + M + FT (Ours)	38.9	95.0	92.3	95.2	83.5	84.0	90.9	90.9
DeiT-B + M + FT (Ours)	85.7	95.9	92.6	96.1	84.4	84.3	92.7	91.7
CaiT-S + M + FT (Ours)	46.5	96.1	90.6	95.4	84.9	85.4	92.7	91.7

Table 7. Maximum inference time comparison. Bold: highest. FA: feature averaging. M: the proposed method. Tr. Dec.: Transformer decoder.

Method	GFLOPs	Time (ms)
DeiT-S + Tr. Dec.	4.9	142
DeiT-M + Tr. Dec.	8.5	146
DeiT-B + Tr. Dec.	**18.7**	**183**
CaiT-S + Tr. Dec.	9.6	164
DeiT-S + FA	4.6	17
DeiT-M + FA	8.0	17
DeiT-B + FA	17.5	20
CaiT-S + FA	9.4	38
DeiT-S + M (Ours)	4.6	14
DeiT-M + M (Ours)	8.0	14
DeiT-B + M (Ours)	17.5	15
CaiT-S + M (Ours)	9.4	36

Considering both latency and recognition accuracy, Figure 9 summarizes the trade-off between a CTC decoder and a Transformer decoder using different ViT backbones. With the same ViT backbone, the CTC decoder outperforms the Transformer decoder significantly in terms of efficiency, with a speed advantage of up to 12 times. However, this speed gain is countered by a maximum reduction in overall word recognition accuracy of 3.1%.



(**a**) DeiT-S (**b**) DeiT-M

(**c**) DeiT-B (**d**) CaiT-S

Figure 8. Inference time comparison between our ViT-CTC models and the Transformer-decoder-based models on an RTX 2060 GPU. Trendlines are projected to the maximum number of characters (i.e., 25) [1]. Tr. Dec.: Transformer decoder. CTC-M: CTC decoder with the proposed method. CTC-FA: CTC decoder with feature averaging. Best viewed in color.

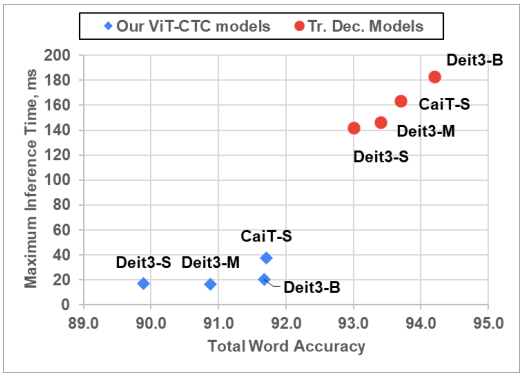


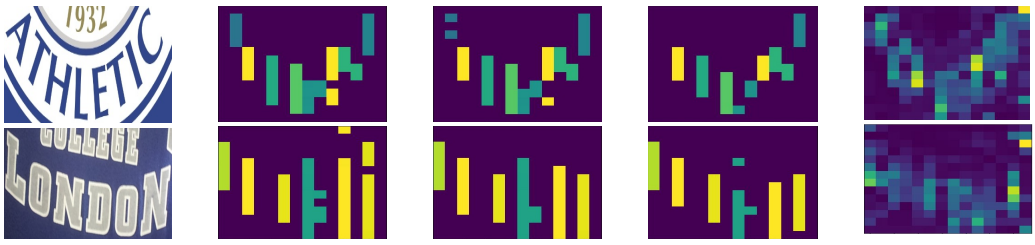
Figure 9. Maximum inference time vs. recognition accuracy comparisons between the ViT-CTC models using the proposed method and the Transformer-decoder-based models on an RTX 2060 GPU. Tr. Dec.: Transformer decoder. Best viewed in color.

3.5. Qualitative Evaluation of Association Map

Until now, we have examined our ViT-CTC models' recognition performance and efficiency in comparison to the CTC and Transformer-decoder-based models. This section shifts focus to the significance of the association map, denoted as **AM**, which is a key output of our proposed method. The detailed derivation of the **AM** can be found in Section 2.1.3. Utilizing an **AM** enables the establishment of explainable 2D spatial relationships between the model's predictions and relevant image regions. This spatial link is crucial for understanding the model's predictions and localization. The **AM** generated by our proposed method corresponds to the cross-attention map formed by the cross-attention module within the Transformer decoder. This module selectively incorporates relevant features for adaptive character predictions.

Figure 10 displays the association maps corresponding to different α values for two examples where text from the top intrudes. Instead of '1932' and 'COLLEGE', the ground-truth words are 'ATHLETIC' and 'LONDON'. The ViT-CTC model accurately predicts both words. Examination of the association maps reveals the model accurately linking the correctly predicted characters with the relevant lower regions containing 'ATHLETIC' and 'LONDON', as opposed to upper regions with '1932' and 'COLLEGE'. Thus, association maps not only explain the model's predictions but also offer localization for those predictions.

As α increases, the association maps maintain high probability regions while discarding those below α, as seen in Figure 10d. Compared with the Transformer decoder's cross-attention maps in Figure 10e, overall alignments are observed. These alignments validate the accuracy and reliability of the association maps from our proposed method that does not reply on a computationally-intensive cross-attention mechanism



(**a**) Input (**b**) $\alpha = 0.80$ (**c**) $\alpha = 0.90$ (**d**) $\alpha = 0.99$ (**e**) Cross-attention

Figure 10. Association maps for different values of α. The color bars show image regions, corresponding to predicted characters. Best viewed in color.

3.6. Quantitative Evaluation of Association Maps

In this section, we quantitatively evaluate our ViT-CTC models' association map and the Transformer-decoder-based models' cross-attention map. Employing Equation (6) for the association map and Equation (9) for the cross-attention map, we calculate alignment evaluation metrics (AEMs) using Equation (8). This was performed using different threshold values α and β, respectively, on the synthetic dataset with character-level annotations, as detailed in Section 2.2.3. To ensure fairness, only image samples correctly recognized by both our ViT-CTC and the Transformer-decoder-based models were included in the evaluation.

Figure 11 depicts that the average alignment evaluation metric (AEM) of the cross-attention map remains stable across different β values, showing good alignment accuracy with the ground-truth character locations. In contrast, the average AEM of the association map exhibits slight sensitivity to α, particularly at higher values. For $\alpha \leq 0.95$, the average AEM of the association map remains above 98% accuracy, signifying strong alignment between the estimated and ground-truth character locations. Thus, the association map is comparable to the cross-attention map in localizing the predicted characters, while the former has a significantly lower computational demand.

Figure 12 compares the estimated character locations from the association and cross-attention maps with the ground-truth bounding boxes in a few highly curved text images. Both methods' estimated character locations closely align with the ground-truth positions.

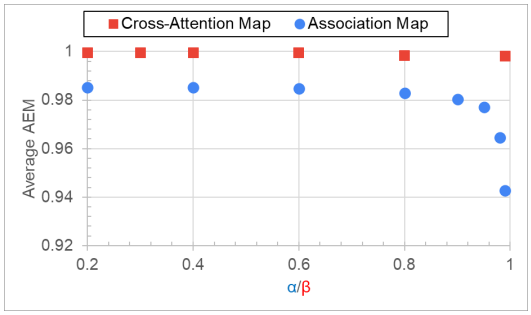
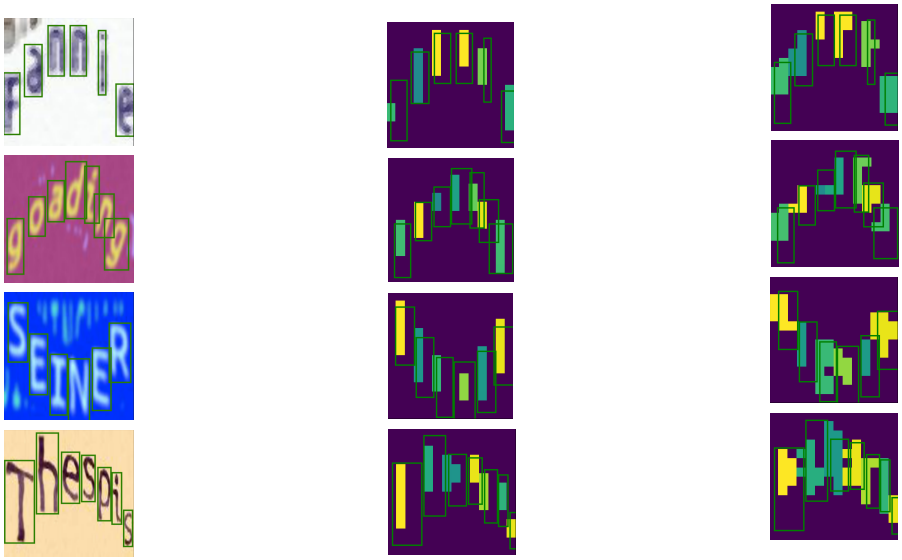


Figure 11. The average AEMs of the association and cross-attention maps as a function of α and β, respectively. Best viewed in color.



(**a**) Input (**b**) Association Map (**c**) Cross-Attention Map

Figure 12. Illustrations of the estimated character locations from the association ($\alpha = 0.8$) and cross-attention ($\beta = 0.5$) maps vs. the ground-truth character locations. Best viewed in color.

4. Limitations and Future Work

Since a CTC decoder is many to one, the pretrained ViT backbone must produce 2D feature maps, the width of which must be greater than or equal to the length of text in an input image. For a ViT-CTC backbone that takes an input image of 224 × 224 pixels and returns 14 × 14 feature maps, it can predict at most 14 characters. Moreover, due to its reliance on left-to-right alignments, a CTC decoder is unable to recognize vertical or highly oriented text images.

Furthermore, due to the sizable receptive field of 16 × 16 pixels in the pretrained ViT backbones employed in this research, the character locations they generate exhibit low resolution.

Thus, future experiments will consider other pretrained ViT or hybrid CNN-Transformer backbones that output dense feature maps, increasing the number of predicted characters and enhancing the resolution of the resulting association map. We will also explore two potential applications of the association maps. Firstly, the association map can guide a Transformer decoder to counter attention drift in long textline images. Secondly, estimated character locations can aid text rectification for highly curved text images.

5. Conclusions

In this paper, we propose a marginalization-based method that enables a 2D feature extractor with a 1D CTC decoder by predicting an output sequence of 2D joint probability distributions over the height and class dimensions. The height dimension is marginalized to suit a 1D CTC decoder. In addition, the proposed method yields an association map that can be used to determine character locations and explain model predictions.

The experimental results show that our ViT-CTC models outperform the recent CTC-based SOTA methods on the public benchmark datasets in terms of recognition accuracy. Compared with a Transformer-decoder-based model, a ViT-CTC model has a maximum reduction in total word recognition accuracy of 3.1%, regardless of the ViT backbone. However, a ViT-CTC model exhibits a substantial speed improvement, surpassing a Transformer-decoder-based model by up to 12 times. Both the qualitative and quantitative evaluations of the character locations estimated from the association map closely correspond with those estimated using the cross-attention map and the ground-truth character-level bounding boxes.

Author Contributions: Conceptualization, R.B.; Data curation, R.B.; Formal analysis, R.B.; Funding acquisition, M.I., S.S. and K.K.; Investigation, R.B.; Methodology, R.B.; Project administration, M.I., S.S. and K.K.; Resources, M.I.; Software, R.B.; Supervision, M.I., S.S. and K.K.; Validation, R.B.; Visualization, R.B.; Writing—original draft, R.B.; Writing—review & editing, M.I., S.S. and K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by JSPS Kakenhi Grant Number 22H00540 and RUPP-OMU/HEIP.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Atienza, R. Vision transformer for fast and efficient scene text recognition. In *Document Analysis and Recognition—ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10. 2021, Proceedings, Part III*; Springer: Cham, Switzerland, 2021; pp. 319–334.
2. Liao, M.; Zhang, J.; Wan, Z.; Xie, F.; Liang, J.; Lyu, P.; Yao, C.; Bai, X. Scene text recognition from two-dimensional perspective. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 8714–8721. [CrossRef]
3. Diaz, D.; Qin, S.; Ingle, R.; Fujii, Y.; Bissacco, A. Rethinking Text Line Recognition Models. *arXiv* **2021**, arXiv:104.07787.
4. Hu, W.; Cai, X.; Hou, J.; Yi, S.; Lin, Z. GTC: Guided Training of CTC towards efficient and accurate scene text recognition. *Proc. Aaai Conf. Artif. Intell.* **2020**, *34*, 11005–11012. [CrossRef]

5. Yang, M.; Liao, M.; Lu, P.; Wang, J.; Zhu, S.; Luo, H.; Tian, Q.; Bai, X. Reading and writing: Discriminative and generative modeling for self-supervised text recognition. In Proceedings of the 30th ACM International Conference On Multimedia, Lisboa, Portugal, 10–14 October 2022.
6. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference On Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 213–229.
7. Shen, Y.; Geng, Z.; Yuan, Y.; Lin, Y.; Liu, Z.; Wang, C.; Hu, H.; Zheng, N.; Guo, B. V-DETR: DETR with Vertex Relative Position Encoding for 3D Object Detection. *arXiv* **2023**, arXiv:2308.04409.
8. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of the 32nd International Conference On Machine Learning, Lille, France, 6–11 July 2015; pp. 2048–2057. Available online: https://proceedings.mlr.press/v37/xuc15.html (accessed on 1 March 2023).
9. Li, H.; Wang, P.; Shen, C.; Zhang, G. Show, attend and read: A simple and strong baseline for irregular text recognition. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 8610–8617. [CrossRef]
10. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2017; Volume 30.
11. Chefer, H.; Gur, S.; Wolf, L. Transformer interpretability beyond attention visualization. In Proceedings of the 2021 IEEE/CVF Conference On Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021.
12. Caron, M.; Touvron, H.; Misra, I.; Jegou, H.; Mairal, J.; Bojanowski, P.; Joulin, A. Emerging properties in self-supervised Vision Transformers. In Proceedings of the 2021 IEEE/CVF International Conference On Computer Vision (ICCV), Montreal, BC, Canada, 10–17 October 2021.
13. Baek, Y.; Shin, S.; Baek, J.; Park, S.; Lee, J.; Nam, D.; Lee, H. Character region attention for text spotting. In Proceedings of Computer Vision—ECCV 2020, Glasgow, UK, 23–28 August 2020; pp. 504–521.
14. Baek, J.; Kim, G.; Lee, J.; Park, S.; Han, D.; Yun, S.; Oh, S.; Lee, H. What is wrong with scene text recognition model comparisons? dataset and model analysis. In Proceedings of the 2019 IEEE/CVF International Conference On Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
15. Shi, B.; Bai, X.; Yao, C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2298–2304. [CrossRef] [PubMed]
16. Borisyuk, F.; Gordo, A.; Sivakumar, V. Rosetta: Large scale system for text detection and recognition in images. In Proceedings of the 24th ACM SIGKDD International Conference On Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018.
17. Wang, J.; Hu, X. Gated Recurrent Convolution Neural Network for OCR. *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2017; Volume 30.
18. Liu, W.; Chen, C.; Wong, K.; Su, Z.; Han, J. Star-net: A spatial attention residue network for scene text recognition. In Proceedings of the British Machine Vision Conference 2016, York, UK, 19–22 September 2016.
19. Wan, Z.; Xie, F.; Liu, Y.; Bai, X.; Yao, C. 2D-CTC for Scene Text Recognition. *arXiv* **2019**, arXiv:1907.09705.
20. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist Temporal Classification. In Proceedings of the 23rd International Conference On Machine Learning—ICML '06, Pittsburgh, PA, USA, 25–29 June 2006.
21. Gunna, S.; Saluja, R.; Jawahar, C. Transfer learning for scene text recognition in Indian languages. In Proceedings of the Document Analysis and Recognition—ICDAR 2021 Workshops, Lausanne, Switzerland, 5–10 September 2021; pp. 182–197.
22. Shi, B.; Wang, X.; Lyu, P.; Yao, C.; Bai, X. Robust scene text recognition with automatic rectification. In Proceedings of the 2016 IEEE Conference On Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
23. Shi, B.; Yang, M.; Wang, X.; Lyu, P.; Yao, C.; Bai, X. Aster: An attentional scene text recognizer with flexible rectification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 2035–2048. [CrossRef] [PubMed]
24. Krylov, I.; Nosov, S.; Sovrasov, V. Open Images V5 Text Annotation and Yet Another Mask Text Spotter. In Proceedings of the Asian Conference On Machine Learning, ACML 2021, Virtual, 17–19 November 2021; Volume 157, pp. 379–389.
25. Lee, J.; Park, S.; Baek, J.; Oh, S.; Kim, S.; Lee, H. On recognizing texts of arbitrary shapes with 2D self-attention. In Proceedings of the 2020 IEEE/CVF Conference On Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020.
26. Li, M.; Lv, T.; Cui, L.; Lu, Y.; Florêncio, D.; Zhang, C.; Li, Z.; Wei, F. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. *arXiv* **2021**, arXiv:2109.10282.
27. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv*, **2021**, arXiv:2010.11929.
28. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jegou, H. Training data-efficient image transformers & distillation through attention. In Proceedings of the 38th International Conference On Machine Learning, Virtual, 18–24 July 2021; Volume 139, pp. 10347–10357. Available online: https://proceedings.mlr.press/v139/touvron21a.html (accessed on 1 March 2023).
29. Touvron, H.; Cord, M.; Sablayrolles, A.; Synnaeve, G.; J'egou, H. Going deeper with Image Transformers. In Proceedings of the 2021 IEEE/CVF International Conference On Computer Vision (ICCV), Montreal, BC, Canada, 10–17 October 2021; pp. 32–42.
30. Touvron, H., Cord, M. & Jegou, H. DeiT III: Revenge of the ViT. *arXiv* **2022**, arXiv:2204.07118.

31. Liu, H.; Wang, B.; Bao, Z.; Xue, M.; Kang, S.; Jiang, D.; Liu, Y.; Ren, B. Perceiving stroke-semantic context: Hierarchical contrastive learning for robust scene text recognition. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 1702–1710. [CrossRef]
32. An J.; Joe, I. Attention Map-Guided Visual Explanations for Deep Neural Networks. *Appl. Sci.* **2022**, *12*, 3846. [CrossRef]
33. Selvaraju, R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the 2017 IEEE International Conference On Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
34. Hannun, A. Sequence Modeling with CTC. *Distill* **2017**, *2*, e8. [CrossRef]
35. Jurafsky, D.; Martin, J. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*; Pearson: London, UK, 2022.
36. Jaderberg, M.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition. *arXiv* **2014**, arXiv:1406.2227.
37. Gupta, A.; Vedaldi, A.; Zisserman, A. Synthetic data for text localisation in natural images. In Proceedings of the 2016 IEEE Conference On Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
38. Yim, M.; Kim, Y.; Cho, H.; Park, S. Synthtiger: Synthetic Text Image Generator towards better text recognition models. In Proceedings of the Document Analysis and Recognition—ICDAR 2021 Workshops, Lausanne, Switzerland, 5–10 September 2021; pp. 109–124.
39. Wang, K.; Babenko, B.; Belongie, S. End-to-end scene text recognition. In Proceedings of the 2011 International Conference On Computer Vision, Barcelona, Spain, 6–13 November 2011.
40. Mishra, A.; Alahari, K.; Jawahar, C. Scene text recognition using higher order language priors. In Proceedings of the British Machine Vision Conference 2012, Surrey, UK, 3–7 September 2012.
41. Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; Bigorda, L.; Mestre, S.; Mas, J.; Mota, D.; Almazan, J.; Heras, L.; et al. ICDAR 2013 robust reading competition. In Proceedings of the 2013 12th International Conference On Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013.
42. Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V.R.; Lu, S.; et al. ICDAR 2015 competition on robust reading. In Proceedings of the 2015 13th International Conference On Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015.
43. Phan, T.; Shivakumara, P.; Tian, S.; Tan, C. Recognizing text with perspective distortion in natural scenes. In Proceedings of the 2013 IEEE International Conference On Computer Vision, Sydney, Australia, 1–8 December 2013.
44. Risnumawan, A.; Shivakumara, P.; Chan, C.; Tan, C. A robust arbitrary text detection system for natural scene images. *Expert Syst. Appl.* **2014**, *41*, 8027–8048. [CrossRef]
45. Baek, J.; Matsui, Y.; Aizawa, K. What if we only use real datasets for scene text recognition? Toward scene text recognition with fewer labels. In Proceedings of the 2021 IEEE/CVF Conference On Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021.
46. Veit, A.; Matera, T.; Neumann, L.; Matas, J.; Belongie, S. COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images. *arXiv* **2016**, arXiv:1601.07140.
47. Shi, B.; Yao, C.; Liao, M.; Yang, M.; Xu, P.; Cui, L.; Belongie, S.; Lu, S.; Bai, X. ICDAR2017 competition on reading Chinese text in the wild (RCTW-17). In Proceedings of the 2017 14th IAPR International Conference On Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017.
48. Zhang, Y.; Gueguen, L.; Zharkov, I.; Zhang, P.; Seifert, K.; Kadlec, B. Uber-Text: A Large-Scale Dataset for Optical Character Recognition from Street-Level Imagery. In Proceedings of the SUNw: Scene Understanding Workshop—CVPR 2017, Honolulu, HI, USA, 21–26 July 2017.
49. Chng, C.; Ding, E.; Liu, J.; Karatzas, D.; Chan, C.; Jin, L.; Liu, Y.; Sun, Y.; Ng, C.; Luo, C.; et al. ICDAR2019 robust reading challenge on arbitrary-shaped text-RRC-art. In Proceedings of the 2019 International Conference On Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019.
50. Sun, Y.; Karatzas, D.; Chan, C.; Jin, L.; Ni, Z.; Chng, C.; Liu, Y.; Luo, C.; Ng, C.; Han, J.; et al. ICDAR 2019 competition on large-scale street view text with partial labeling - RRC-LSVT. In Proceedings of the 2019 International Conference On Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019.
51. Zhang, R.; Yang, M.; Bai, X.; Shi, B.; Karatzas, D.; Lu, S.; Jawahar, C.; Zhou, Y.; Jiang, Q.; Song, Q. et al. ICDAR 2019 robust reading challenge on reading Chinese text on Signboard. In Proceedings of the 2019 International Conference On Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019.
52. Singh, A.; Pang, G.; Toh, M.; Huang, J.; Galuba, W.; Hassner, T. TextOCR: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In Proceedings of the 2021 IEEE/CVF Conference On Computer Vision and Pattern Recognition (CVPR), Virtual, 19–25 June 2021.
53. Liu, N.; Schwartz, R.; Smith, N. Inoculation by fine-tuning: A method for analyzing challenge datasets. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), Minneapolis, MN, USA, 3–5 June 2019.

54. Andriyanov, N.; Andriyanov, D. Pattern recognition on radar images using augmentation. In Proceedings of the 2020 Ural Symposium On Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), Yekaterinburg, Russia, 14 May 2020; pp. 0289–0291.
55. Buslaev, A.; Iglovikov, V.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A. Albumentations: Fast and flexible image augmentations. *Information* **2020**, *11*, 125. [CrossRef]