A Multiplexed Network for End-to-End, Multilingual OCR

Jing Huang Guan Pang Rama Kovvuri Mandy Toh Kevin J Liang
Praveen Krishnan Xi Yin Tal Hassner
Facebook AI
{jinghuang,gpang,ramakovvuri,mandytoh,kevinjliang,pkrishnan,yinxi,thassner}@fb.com

Abstract

Recent advances in OCR have shown that an end-to-end (E2E) training pipeline that includes both detection and recognition leads to the best results. However, many existing methods focus primarily on Latin-alphabet languages, often even only case-insensitive English characters. In this paper, we propose an E2E approach, Multiplexed Multilingual Mask TextSpotter, that performs script identification at the word level and handles different scripts with different recognition heads, all while maintaining a unified loss that simultaneously optimizes script identification and multiple recognition heads. Experiments show that our method outperforms the single-head model with similar number of parameters in end-to-end recognition tasks, and achieves state-of-the-art results on MLT17 and MLT19 joint text detection and script identification benchmarks. We believe that our work is a step towards the end-to-end trainable and scalable multilingual multi-purpose OCR system. Our code and model will be released.

1. Introduction

Reading text in visual content has long been a topic of interest in computer vision, with numerous practical applications such as search, scene understanding, translation, navigation, and assistance for the visually impaired. In recent years, advances in deep learning have led to dramatic improvements of Optical Character Recognition (OCR), allowing reading text in increasingly diverse and challenging scene environments with higher accuracy than ever before. A common approach is to decompose the task into two sub-problems: text detection, the localization of text in visual media, and text recognition, the transcription of the detected text. While these two components were traditionally learned separately, recent works have shown that they can be learned jointly, with benefits to both modules.

As the most commonly spoken language in the world [1] and a *lingua franca* for research, the English language has been the focus of many public OCR benchmarks [35, 62, 25, 24, 61, 53] and methods [30, 33, 43, 44]. However, English (and other Latin alphabet languages) represent only a fraction of the languages spoken (and written) around the world. OCR technology is also used to study forgotten languages and ancient manuscripts, where alphabets and script styles can vary enormously [14, 15]. Thus, developing OCR capabilities in other languages is also important to ensure such technologies are accessible to everyone. Additionally, because of the increasing interconnectedness of the world and its cultures, it is important to develop OCR systems capable of recognizing text from multiple languages co-occurring in the same scene.

While many concepts and strategies from OCR on English text can be adapted to other languages, developing multilingual OCR systems is not completely straightforward. Naively training a separate system for each language is computationally expensive during inference and does not properly account for predictions made for other languages. Furthermore, previous works [44, 30] have shown that jointly learning text detection and text recognition modules is mutually beneficial; separate models lose out on the potential benefits of a shared text detection module. On the other hand, learning a unified model with single recognition head also presents problems. While uncased English only has 26 characters, many Asian languages like Chinese, Japanese, and Korean have tens of thousands of characters. Different languages/scripts can also have very different word structures or orientations. For example, vertically written text is far more common in East Asian languages like Chinese, Japanese and Korean than in Western languages, and characters in Arabic and Hindi are usually connected to each other. This variability in the number of characters as well as the wide variability in script appearance characteristics mean it is highly unlikely that a single architecture can capably maximize accuracy and efficiency over all languages/scripts, and any imbalances in the training data may result in significantly different performances between languages.

Given these challenges, we present a blend of these two approaches, incorporating each one's advantages while mit-

igating their faults. Specifically, we propose a single text detection module followed by a text recognition head for each language, with a multiplexer routing the detected text to the appropriate head, as determined by the output of a Language Prediction Network (LPN). This strategy can be seen as analogous to human perception of text. Locating the words of most languages is easy even without knowing the language, but recognizing the actual characters and words requires special knowledge: language/script identification typically precedes recognition.

Notably, this multiplexer design has important implications for real-world text spotting systems. Having language-specific text recognition heads allows custom design of the architecture depending on the difficulty and characteristics of each language, while still sharing and jointly learning the same text detection trunk. New languages can also be easily added to the system without re-training the whole model and worrying about affecting the existing languages.

Our contributions can be summarized as follows:

- We propose an end-to-end trainable multiplexed OCR model that can automatically pick the best recognition head for the detected words.
- We propose a language prediction network using masked pooled features as input and an integrated loss function with the recognition heads.
- We design a training strategy that takes advantage of the proposed losses, allows for easy extension to new languages and addresses the data imbalance problem.
- We empirically show that the multiplexed model consistently outperforms single-head model and is less prone to training data distribution bias.

2. Related work

Text spotting is commonly broken down into two subtasks: text detection and text recognition. In scenes with multiple languages, script identification is also necessary, either explicitly by learning a classification model or implicitly as a byproduct of text recognition. While these three sub-tasks were often considered individually and then chained together in the past, end-to-end methods seeking to learn all at once have recently become popular. We give a brief overview of relevant works below; see [34] for a more thorough treatment.

2.1. Text detection

Text detection is commonly the first stage of understanding text content in images. Early approaches typically consisted of human-engineered features or heuristics, such as connected components [22, 40, 67] or sliding windows [28]. The promise of early deep learning models [26] led to some of these strategies being combined with convolutional networks [63, 20], and as convolutional networks proved successful for object detection [11, 45, 16], more recent approaches have almost exclusively been using deep detection models [58]. Given the various orientations and shapes that text can take, further refinements have focused on making text detection rotation invariant [23] or switched from rectangular bounding boxes to more flexible segmentation masks [29, 44]. Character-level detection with weakly supervised learning of word-level annotations has also been shown effective [2].

2.2. Text recognition

Once text has been localized through detection, the region is often cropped and then fed to a text recognition system to be read as a character/word sequence.

Like text detection, text recognition methods have a long history predating the popular use of deep learning [7, 42, 52, 46], but most recent methods use neural networks. Connectionist temporal classification (CTC) [13] methods use recurrent neural networks to decode features (recently mostly convolutional) into an output sequence [54, 18]. Another common framework for text recognition is the Seq2Seq encoder-decoder framework [56] that is often combined with attention [4], which is used by [27, 49]. [21] frames the problem as a V-way image classification problem, where V is the size of a pre-defined vocabulary.

2.3. Script identification

Text spotting in multilingual settings often requires script identification to determine a language for text recognition. Early works focused on identifying the language of scripts in simple environments like documents [19, 57, 5], primarily with traditional pre-deep learning methods.

As with other vision tasks, convolutional architectures proved especially effective [48, 12]. Script identification in natural scenes began with Shi *et al*. [51], who cropped and labeled text in images from Google Street View, then trained convolutional neural networks with a specialized multi-stage pooling layer for classification; the authors achieved further gains in accuracy with densely extracted local descriptors combined with discriminitive clustering. Fujii *et al*. [10] proposed a line-level script identification method casting the problem as a sequence-to-label problem. E2E-MLT [6] is a multilingual end-to-end text spotting system that forgoes script identification and performs text recognition directly. They proposed to use a CNN to classify the script at the cropped-word level that preserves the aspect ratio. Another common approach for script identification comes after the text recognition step, which infers the language by identifying the most frequent language occurrences of the characters in the text [3]. The resulting more challenging text recognition task leads to somewhat

hampered model performance. We find that performing script identification in our proposed Language Prediction Network (LPN), with the masked pooled features of the detected words as input, to multiplex the text recognition heads leads to significantly higher accuracy for the script identification task, compared to the majority-vote approach.

2.4. Text spotting

While many early works focused on one of the aforementioned tasks in isolation, end-to-end (E2E) text spotting systems have also been proposed.

Some learn text detection and recognition submodules separately, linking the two independent systems together for the final product [40, 21, 31]. However, the learning tasks of text detection and recognition are mutually beneficial: recognition can provide additional feedback to detection and remove false positives, while detection can provide augmentations for recognition. As such, recent works learn these two jointly [32, 44, 29, 30]. E2E-MLT [6] proposed an E2E text spotting method evaluated on multilingual settings, but does not explicitly incorporate any specific model components adapted for multiple languages, instead dealing with characters from all languages in the same recognition head; this is the approach taken by most E2E systems for multilingual settings like ICDAR-MLT [39, 38] and CRAFTS [3].

3. Methodology

The multiplexed model shares the same detection and segmentation modules as Mask TextSpotter V3 [30] (Figure 1). A ResNet-50 [17] backbone with a U-Net structure [47] is used to build the Segmentation Proposal Network (SPN). Similar to [30, 64], the Vatti clipping algorithm [60] is used to shrink the text regions with a shrink ratio r to separate neighboring text regions. Once the segmentation proposals are generated, hard RoI masking [30] is used to suppress the background and neighboring text instances.

The recognition model for Mask TextSpotter V3 [30] comprises of a Character Segmentation Module and a Spatial Attention Module [29] adapted for text recognition. We only use the Spatial Attention Module in our model due to the following reasons: (1) using both modules does not scale when expanding the character set from 36 to 10k; (2) the Character Segmentation Module requires character-level annotations to supervise the training and the order of the characters cannot be obtained from the segmentation maps; (3) in our experiments on Latin-only model, disabling the Character Segmentation Module has a minimal effect on the final recognition results.

To extend the model from Latin-only to multilingual, there are two directions: (1) treat all languages and characters as if they belong to the same language with all characters, and use a single recognition head to handle all of them; (2) build separate recognition heads to handle words from different languages, and then pick/combine the predictions from them. We choose approach (2) since it is much more flexible when we train the model without worrying about data imbalance across different languages, and has greater potential for future extension, e.g., incorporating language model into the recognition.

3.1. Language prediction network

To automatically select the recognition module appropriate for a given script, we propose a Language Prediction Network (LPN), as detailed in Figure 2. The input of the LPN is the masked pooled feature from the detection and segmentation modules, with size $256 \times 32 \times 32$. We apply a standard classification network with two 2×2 convolutional layers with rectified linear unit (ReLU) activations and a 2×2 max pooling layer in between, followed by two fully connected (FC) layers with a ReLU in between. This network produces an output vector of size L, which can be converted into probabilities using a Softmax, where $L = N_{lang}$ is the number of language classes we would like the model to support.

Note that in practice, the number of different recognition heads in the model N_{rec} does not necessarily have to equal the number of supported languages N_{lang}, particularly in the case of shared alphabets. For example, the Latin alphabet is used for Germanic (e.g. English, German) and Romance (e.g. French, Italian) languages, meaning LPN predictions for any of these languages can be routed to a singular Latin recognition head. For simplicity, in the following sections we assume $N_{lang} = N_{rec}$.

Finally, we note that previous work suggested network decision making mechanisms [65]. These methods were proposed for very different applications than the one considered here. Importantly, the decision mechanisms they described were not network-based and so not end-to-end trainable with other network components.

3.2. Multiplexer with disentangled loss

Since a few datasets (e.g., MLT [39]) provide ground truth annotations for the language of a particular text, we can train both the LPN and the recognition heads in parallel with a disentangled loss, i.e., computing the loss terms for each of the heads and the LPN in parallel and then directly adding them up:

$$L_{disentangled} = \alpha_{lang} L_{lang} + \sum_{r \in R} \alpha_{seq(r)} L_{seq(r)} \quad (1)$$

where L_{lang} is the loss for LPN, R is the set of recognition heads, and $L_{seq(r)}$ is the loss for the recognition head r. α_{lang} and $\alpha_{seq(r)}$ are weighting hyper-parameters. In our experiment, we set $\alpha_{lang} = 0.02$ in the first few thousands of iterations in the first training stage (4.2) and $\alpha_{lang} = 1$

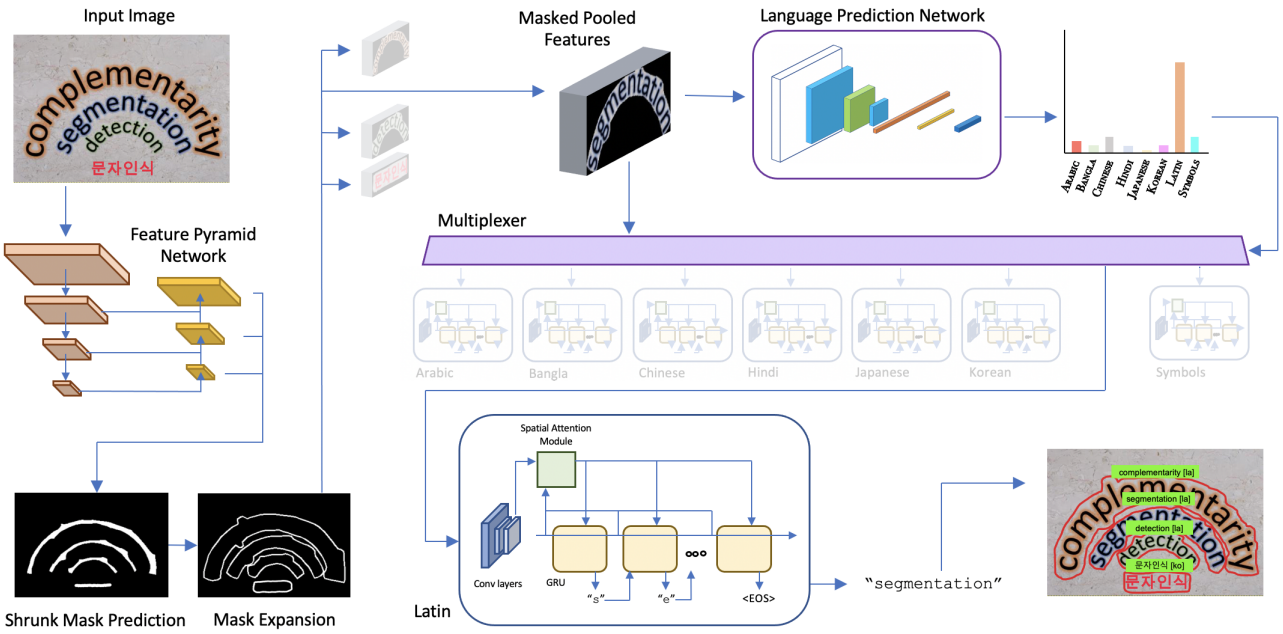

Figure 1. **M3 TextSpotter.** The proposed M3 TextSpotter shares the same detection and segmentation trunk with Mask TextSpotter v3 [30], but incorporates a novel Language Prediction Network (LPN). The output of the LPN then determines which script's recognition head the multiplexer selects.

after that; we use $\alpha_{seq(r)} = 0.5$ for all recognition heads throughout the training.

Language prediction is a standard N-way classification problem, so the language prediction loss in Equation 1 can be computed using a cross entropy loss:

$$L_{lang} = -\sum_{l=1}^{N_{lang}} I(l = l_{gt}) \log p(l), \tag{2}$$

where $I(l = l_{gt})$ is the binary indicator (0 or 1) if the language matches the ground truth, and $p(l)$ is the probability inferred by the LPN that the word belongs to language l.

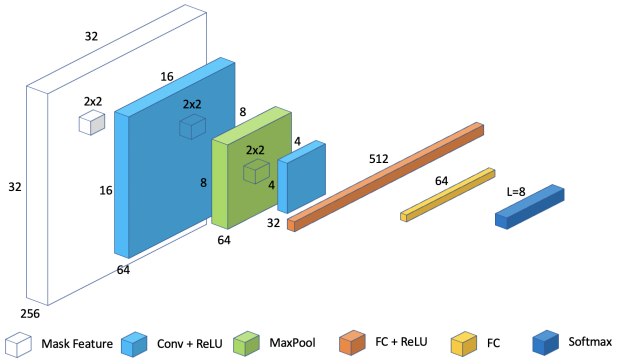

Figure 2. **Language Prediction Network.** In this figure $L = N_{lang} = 8$, denoting the eight scripts (Arabic, Bengali, Chinese, Hindi, Japanese, Korean, Latin and Symbol) supported by our default model in this paper.

Similar to [29], we use the negative log likelihood as the text recognition loss $L_{seq(r)}$:

$$L_{seq} = -\frac{1}{T}\sum_{t=1}^{T} \log p(y_t = c_t), \tag{3}$$

where $p(y_t = c_t)$ is the predicted probability of character at position t of the sequence, and T is the length of the sequence of character labels. We use $T = 32$ for all the recognition heads in this paper, but it can be customized to account for different distributions of word length across the languages - for example, since there's typically no space between the Chinese and Japanese words, we can use bigger T for these languages.

To compute $L_{seq}(r)$, i.e., L_{seq} for different recognition heads that support different character sets, we need to ignore the unsupported characters in the loss computation:

$$L_{seq(r)} = -\frac{1}{T}\sum_{t=1}^{T} I(c_t \in C_r) \log p(y_t = c_t), \tag{4}$$

where C_r is the character set supported by recognition head r, c_t is the ground truth character at step t, $I(c_t \in C_r) = 1$ if c_t is supported and $I(c_t \in C_r) = 0$ if c_t is not supported.

3.3. Multiplexer with integrated loss

While the multiplexer with disentangled loss could serve as a good initialization for model training, such an approach

has a few limitations. First, the training of the language predictor requires explicit ground truth annotations of the language at the word level, which can be inaccurate and is not always available outside of curated datasets. Secondly, the disentangled total loss does not reflect the actual prediction of the model at inference time, especially when there are shared characters across multiple recognition heads. Finally, despite having a mechanism to ignore labels, it is counter-productive to train the recognition heads for the wrong language with unsupported words.

To address these problems, we propose an integrated loss that combines results from the language prediction head and the recognition heads during training. To enforce consistency between training and testing, we can use a hard integrated loss:

$$L_{hard-integrated} = \alpha_{seq(r)} L_{seq(\arg\max_{1 \le l \le N_{rec}} p(l))} \quad (5)$$

With a hard integrated loss, we pick exactly one recognition head for each word, selecting and using the loss of the head that has the maximum probability as predicted by the language prediction network. This loss better matches the operation of the text spotting system during inference and avoids involving irrelevant recognition heads during training. Our ablation study (Section 4.3) shows that it outperforms an alternative soft integrated loss (Equation 7).

Note that directly using the default sequence recognition loss (Equation 4) in the integrated losses does not work due to the handling of the unsupported characters: unsupported characters will always contribute 0 to the loss while supported characters contribute a positive value to the total loss, no matter how good the actual prediction is. To resolve this problem, we can assign a large penalty factor β to unsupported characters:

$$L_{seq(r)} = -\frac{1}{T}\sum_{t=1}^{T}[I(c_t \in C_r)\cdot\log p(y_t = c_t) + I(c_t \notin C_r)\cdot\beta] \quad (6)$$

We set the penalty to $\beta = -12$ in our experiments.

4. Experiments

We validate the effectiveness of our multilingual multiplexer design with a series of experiments, evaluating the proposed Multiplexed Mask TextSpotter on multilingual scene text from the MLT17 [39] and MLT19 [38] datasets. In addition to these two datasets, we also take advantage of several other public OCR datasets for training. We report results for text detection, end-to-end script identification, and end-to-end multilingual recognition tasks. We also show the results of an ablation study comparing our multiplexed multi-headed approach with a single combined recognition head approach.

4.1. Datasets

ICDAR 2017 MLT dataset (MLT17) [39] was introduced as a part of ICDAR 2017 Robust Reading Competition for the problem of multi-lingual text detection and script identification. It contains 7200 training, 1800 validation and 9000 test images in 9 languages representing 6 different scripts equally. The dataset contains multi-oriented scene text that is annotated using quadrangle bounding boxes.

ICDAR 2019 MLT dataset (MLT19) [38] was introduced as a part of ICDAR 2019 Robust Reading Competition extending ICDAR 2017 MLT dataset for the problem of multi-lingual text detection and script identification. It contains 10000 training and 10000 test images in 10 languages representing 7 different scripts. The dataset also contains multi-oriented scene text that is annotated using quadrangle bounding boxes. It also provides a synthetic dataset (SynthTextMLT) [6] that provides ∼273k synthetic data in 7 scripts. There are many errors for Hindi images in SynthTextMLT, so we filtered out any Hindi images containing non-Hindi characters (likely errors) when using it.

Total-Text dataset [8], presented at ICDAR 2017 is a comprehensive scene text dataset for text detection and recognition. It contains 1255 training and 300 test images in English language. The dataset contains wide variety of horizontal, multi-oriented and curved text annotated at word-level using polygon bounding boxes.

ICDAR 2019 ArT dataset (ArT19) [9] was introduced as a part of ICDAR 2019 Robust Reading Competition. It contains 5603 training and 4563 test images in English and Chinese languages. The dataset is a combination of Total-Text [8] and SCUT-CTW1500 [68] datasets. The dataset contains highly challenging arbitrarily shaped text that is annotated using arbitrary number of polygon vertices. Since this dataset contains the testing images from Total Text, we deliberately filtered them out in our training so that our model weights remain valid for future training/evaluation on the Total Text benchmark.

ICDAR 2017 RCTW dataset (RCTW17) [50] was introduced as a part of ICDAR 2017 Robust Reading Competition on Reading Chinese Text in the Wild. It contains 8034 train and 4229 test images, focusing primarily on scene text in Chinese.

ICDAR 2019 LSVT dataset (LSVT19) [55] was introduced as a part of ICDAR 2019 Robust Reading Competition on Large-scale Street View Text with Partial Labeling. It is one of the largest OCR datasets, containing 30000 train and 20000 test images. The dataset is primarily street view text in Chinese, but also has about 20% of its labels in English words.

ICDAR 2013 dataset (IC13) [25] was introduced as part of the ICDAR 2013 Robust Reading Competition. It contains 229 training and 233 test images in English language. The dataset contains high-resolution, horizontal text annotated

Figure 3. **Qualitative results on MLT19.** The polygon masks predicted by our model are shown over the detected words. The transcriptions from the selected recognition head and the predicted languages are also rendered in the same color as the mask. The language code mappings are: ar - Arabic, bn - Bengali, hi - Hindi, ja - Japanese, ko - Korean, la - Latin, zh - Chinese, symbol - Symbol.

Table 1. **Parameter number comparison between multiplexed model vs. single-head model.** The total number of the multiplexed model is the sum of the parameter numbers for each individual recognition heads as well as the LPN. The parameters for detection, segmentation and mask feature extraction are not included here.

Head	Charset Size	Embed Size	Hidden Size	Parameters
Arabic	80	100	224	1.15M
Bengali	110	100	224	1.16M
Chinese	5200	200	224	3.36M
Hindi	110	100	224	1.16M
Japanese	2300	200	224	2.13M
Korean	1500	200	224	1.79M
Latin	250	150	256	1.49M
Symbol	60	30	64	0.21M
LPN	-	-	-	0.11M
Multiplexed	-	-	-	12.5M
Single-Head	9000	400	512	12.6M

at word-level using rectangular bounding boxes.

ICDAR 2015 dataset (IC15) [24] was introduced as part of the ICDAR 2015 Robust Reading Competition. It contains 1000 training and 500 test images in English language. The dataset contains multi-oriented scene text annotated at word-level using quadrangle bounding boxes.

4.2. Training strategy

Prior to training, we go through the annotations of the aforementioned datasets to obtain a character set (charset) for each of the eight scripts. Since digits and common punctuation marks appear in all languages, we append them to all character sets. The final number of characters for each recognition head are listed in the column Charset Size of Table 1. The choice of the parameters is based on the following heuristics: we use bigger embedding size (200) for Chinese/Japanese/Korean recognition heads, as they have much bigger character sets; Latin has relatively larger character sets than the remaining scripts as well as much more data, so we use an embedding size of 150 and a bigger hidden layer size of 256 to capture more sequential relationship among the characters. We order each character set by the frequencies of individual characters and map them to consecutive indices for each recognition head, respectively.

We initialize the detection, segmentation, and mask feature extraction weights from the officially published weights released by Mask TextSpotter v3 [30]. For the recognition weights, we discard the Character Segmentation Module, and initialize each of the individual heads with the sequence recognition head with spatial attention module with zero-padding or narrowing, since the dimensions of character sizes, embed layer sizes, and hidden layer sizes are different from the original weights.

In the first stage of training, we train the model end-to-end using the disentangled loss on datasets (MLT and SynthTextMLT) with ground truth annotations for languages.

Table 2. **Quantitative detection results on MLT17.** Note that (1) our model supports Hindi, which is not required by MLT17. (2) CharNet H-88 has 89.21M parameters, which is 3x heavier than CharNet R-50 that is more comparable to our backbone.

Method	F	P	R
Lyu et al. [36]	66.8	83.8	55.6
FOTS [32]	67.3	81	57.5
CRAFT [2]	73.9	80.6	68.2
CharNet R-50 [66]	73.42	77.07	70.10
CharNet H-88 [66]	**75.77**	81.27	**70.97**
Multiplexed TextSpotter	72.42	**85.37**	62.88

This leads to quicker initial convergence of both the LPN and the recognition heads, as a randomly initialized LPN is unlikely to be able to correctly identify scripts, severely hampering each of the recognition heads from learning, and poorly performing recognition heads deprives the LPN of feedback on its routing.

In the second stage of training, we switch to the hard integrated loss. This enables training on all datasets, as explicit ground truth language annotations are no longer necessary for learning.

For the third and final stage of training, we freeze most of the network, including detection, segmentation, mask feature extraction, language prediction networks and all but one individual recognition heads, and train the specific recognition head with only data from this one script. This step would have been impossible if we use the single combined head for all languages, and it greatly resolves the data imbalance problem across different languages.

4.3. Ablation study

Multiplexed model vs. single combined head. In order to make a fair comparison between the multiplexed model versus a single-head model, we estimated the individual as well as the total number of characters to be supported by the eight types of scripts, and adjusted the embedding and hidden layer sizes such that the total number of parameters are roughly the same between the multiplexed model (including the Language Prediction Network) and the single combined-head model (Table 1).

Note that for the multiplexed model, we use a limited set of hidden layer sizes and embedding layer sizes in our experiment. However, these hyper-parameters, or even the underlying architectures, can be further customized based on the importance, difficulty and characteristics of the scripts/languages.

From the experiments in detection (Table 2, Table 3), and end-to-end recognition (Table 6), we can see that the multiplexed model consistently outperforms the single combined head model. Moreover, the multiplexed model can provide extra signal of language identification results (Table 4 and Table 5) based on visual information, which is not directly available from the single-head model. There are some approaches that can infer the language during post-processing, however, they will need extra language model information to identify the language if the characters are not exclusive to certain languages.

Hard vs. soft integrated loss. There is a legitimate concern on whether the hard integrated loss is differentiable. Instead of using the $\arg\max$, we can also employ a soft relaxation to directly multiply the probabilities of each language with the loss from each recognition head and sum them up, yielding the following soft integrated loss function:

$$L_{soft-integrated} = \sum_{r=1}^{N_{rec}} p(r) \cdot \alpha_{seq(r)} L_{seq(r)} \qquad (7)$$

The hard integrated loss can be seen as a special case of soft integrated loss, where only one of $p(r)$ is 1 while all others are 0. In our experiments, however, using hard integrated loss gives about 10% better results in terms of H-mean than using soft integrated loss under the same number of iterations. This can be explained by that the hard integrated loss aligns more with the expected behavior of the model during inference time.

4.4. Text detection task

Text detection precision (P) and recall (R) for our Multiplexed TextSpotter and several baselines for on MLT17 [39] and MLT19 [38] are shown in Tables 2 and 3, respectively. Note that our model is not fine-tuned on MLT17, which contains one fewer language (Hindi), but still manages to achieve comparable results as other SOTA methods and the highest precision.

In Table 3, we also show the language-wise F-measure results. Our method beats entries from all published methods on the leaderboard including CRAFTS [3], except the result reported in the paper version of CRAFTS [3] is higher. For language-wise evaluation, our method shows the best result for all languages except Hindi, with especially large improvements in Arabic, Chinese, and Korean. Interestingly, we find that the single-head Mask TextSpotter performs slightly better than the Multiplexed Mask TextSpotter in Latin. We hypothesize that this is because of the higher prevalence of Latin words in the MLT dataset, due to the inclusion of 4 languages with Latin alphabets: English, French, German, and Italian. Thus, the single-head model greatly favoring Latin leads to stronger Latin performance, to the detriment of the other languages. This demonstrates that the single-head model is more vulnerable to training data distribution bias. By contrast, the Multiplexed Mask TextSpotter achieves more equitable performance due to its design.

Table 3. **Quantitative detection results on MLT19 with language-wise performance.** All numbers are from the official ICDAR19-MLT website except CRAFTS (paper), which comes from their paper [3].

Method	F	P	R	AP	Arabic	Latin	Chinese	Japanese	Korean	Bangla	Hindi
PSENet [64]	65.83	73.52	59.59	52.73	43.96	65.77	38.47	34.47	51.73	34.04	47.19
RRPN [37]	69.56	77.71	62.95	58.07	35.88	68.01	33.31	36.11	45.06	28.78	40.00
CRAFTS [3]	70.86	81.42	62.73	56.63	43.97	72.49	37.20	42.10	54.05	38.50	**53.50**
CRAFTS (paper) [3]	**75.5**	81.7	**70.1**	-	-	-	-	-	-	-	-
Single-head TextSpotter [30]	71.10	83.75	61.76	58.76	51.12	**73.56**	40.41	41.22	56.54	39.68	49.00
Multiplexed TextSpotter	72.66	**85.53**	63.16	**60.46**	**51.75**	73.55	**43.86**	**42.43**	**57.15**	**40.27**	51.95

Table 4. **Joint text detection and script identification results on MLT17.** Note that our general model supports Hindi, which is not required by MLT17, but still achieves the best result.

Method	F	P	R	AP
E2E-MLT [6]	58.69	64.61	53.77	-
CRAFTS [3]	68.31	74.52	**63.06**	54.56
Multiplexed TextSpotter	**69.41**	**81.81**	60.27	**56.30**

Table 5. **Joint text detection and script identification results on MLT19.** All Task 3 numbers taken from the official ICDAR19-MLT website.

Method	F	P	R	AP
CRAFTS [3]	68.34	78.52	**60.50**	53.75
Single-head TextSpotter [30]	65.19	75.41	57.41	51.98
Multiplexed TextSpotter	**69.42**	**81.72**	60.34	**56.46**

Table 6. **End-to-end recognition results on MLT19.** All numbers are from the official ICDAR19-MLT website except CRAFTS (paper), which comes from [3].

Method	F	P	R
E2E-MLT [6]	26.5	37.4	20.5
RRPN+CLTDR [37]	33.8	38.6	30.1
CRAFTS [3]	51.7	65.7	42.7
CRAFTS (paper) [3]	**58.2**	**72.9**	**48.5**
Single-head TextSpotter [30]	39.7	71.8	27.4
Multiplexed TextSpotter	48.2	68.0	37.3

4.5. End-to-end script identification task

Table 4 and Table 5 show the end-to-end language identification results on MLT17 [39] and MLT19 [38], respectively. The proposed Multiplexed Mask TextSpotter achieves the best F-score (H-mean), precision, and average precision. Note that we didn't fine-tune our model on the MLT17 dataset for the MLT17 benchmark, which contains one fewer language (Hindi), but still managed to outperform existing methods in all metrics but recall. Also, we implemented a post-processing step for the single-head Mask TextSpotter that infers the language from the recognized words similar to [6, 3], and the results again show that the multiplexed model with an LPN outperforms the single-head model with post processing. This can be explained by the fact that our language prediction network infers the script based on visual cues directly, while the post-processing-based method could suffer from noisy text recognition results.

4.6. End-to-end multilingual recognition task

Table 6 shows the end-to-end multilingual recognition benchmark results on MLT19 [38]. Our method outperforms all methods except CRAFTS [3]. We think that the difference in performance mainly comes from: (a) their ResNet-based feature extraction module with 24 Conv layers in their recognition head, as opposed to our feature extraction module with only 5 Conv layers, (b) the orientation estimation/link representation for vertical text that is common in East Asian languages, (c) the TPS-based rectification, and (d) the use ofthe ReCTS dataset [69] containing 20K additional training images. All these improvements are orthogonal to the proposed multiplexed framework and they can be combined. For example, the multiplexed model enables the potential improvement by allowing specific recognition heads to be customized to accommodate the vertical text. Regardless, we observe that the multiplexed model strongly outperforms the single-head model, demonstrating the effectiveness of the proposed multiplexed framework. Figure 3 shows some qualitative visualization results of end-to-end recognition with the proposed Multiplexed TextSpotter on the MLT19 test set. We see that the proposed model is able to successfully detect and recognize text from multiple languages within the same scene.

5. Conclusion

We propose a multiplexed network for end-to-end multilingual OCR. To our knowledge, this is the first end-to-end framework that trains text detection, segmentation, language identification and multiple text recognition heads in an end-to-end manner. The framework provides flexibility on freezing any part of the model and focus on training the other parts. The multiplexed pipeline is particularly useful when we need to support new languages, remove existing languages (e.g. for deployment in special scenarios that require only a subset of languages and/or have limited hardware resource), or improve/upgrade recognition model on certain languages, without worrying about harming the other existing languages of the model. We achieve state-of-the-art performance on the joint text detection and script identification tasks of both MLT19 and MLT17 benchmarks. Our experiments also show that with similar number of total parameters, the multiplexed model can achieve better results than a single unified recognition model with similar architectures. As future work, we plan to leverage task similarities [41, 59] to explore grouping related languages into single recognition heads.

References

[1] English. *Ethnologue (22nd ed.)*, 2019. 1

[2] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019. 2, 7

[3] Youngmin Baek, Seung Shin, Jeonghun Baek, Sungrae Park, Junyeop Lee, Daehyun Nam, and Hwalsuk Lee. Character region attention for text spotting. In *European Conference on Computer Vision*, pages 504–521. Springer, 2020. 2, 3, 7, 8

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 2

[5] Andrew Busch, Wageeh W Boles, and Sridha Sridharan. Texture for script identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1720–1732, 2005. 2

[6] Michal Bušta, Yash Patel, and Jiri Matas. E2e-mlt-an unconstrained end-to-end method for multi-language scene text. In *Asian Conference on Computer Vision*, pages 127–143. Springer, 2018. 2, 3, 5, 8

[7] Xilin Chen, Jie Yang, Jing Zhang, and Alex Waibel. Automatic detection and recognition of signs from natural scenes. *IEEE Transactions on image processing*, 13(1):87–99, 2004. 2

[8] Chee Kheng Ch'ng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 935–942. IEEE, 2017. 5

[9] Chee Kheng Chng, Yuliang Liu, Yipeng Sun, Chun Chet Ng, Canjie Luo, Zihan Ni, ChuanMing Fang, Shuaitao Zhang, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, Chee Seng Chan, and Jin Lianwen. Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1571–1576. IEEE, 2019. 5

[10] Yasuhisa Fujii, Karel Driesen, Jonathan Baccash, Ash Hurst, and Ashok C Popat. Sequence-to-label script identification for multilingual ocr. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 161–168. IEEE, 2017. 2

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[12] Lluis Gomez, Anguelos Nicolaou, and Dimosthenis Karatzas. Improving patch-based scene text script identification with ensembles of conjoined networks. *Pattern Recognition*, 67:85–96, 2017. 2

[13] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006. 2

[14] Tal Hassner, Malte Rehbein, Peter A Stokes, and Lior Wolf. Computation and palaeography: potentials and limits. *Dagstuhl Reports*, 2(9):184–199, 2012. 1

[15] Tal Hassner, Robert Sablatnig, Dominique Stutzmann, and Ségolène Tarte. Digital palaeography: New machines and old texts (dagstuhl seminar 14302). *Dagstuhl Reports*, 4(7), 2014. 1

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 770–778, 2016. 3

[18] Pan He, Weilin Huang, Yu Qiao, Chen Change Loy, and Xiaoou Tang. Reading scene text in deep convolutional sequences. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3501–3508, 2016. 2

[19] Judith Hochberg, Patrick Kelly, Timothy Thomas, and Lila Kerns. Automatic script identification from document images using cluster-based templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):176–181, 1997. 2

[20] Weilin Huang, Yu Qiao, and Xiaoou Tang. Robust scene text detection with convolution neural network induced mser trees. In *European conference on computer vision*, pages 497–511. Springer, 2014. 2

[21] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International journal of computer vision*, 116(1):1–20, 2016. 2, 3

[22] Anil K Jain and Bin Yu. Automatic text location in images and video frames. *Pattern recognition*, 31(12):2055–2076, 1998. 2

[23] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. R2cnn: rotational region cnn for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*, 2017. 2

[24] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE, 2015. 1, 6

[25] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493. IEEE, 2013. 1, 5

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 2012. 2

[27] Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2231–2239, 2016. 2

[28] Jung-Jin Lee, Pyoung-Hean Lee, Seong-Whan Lee, Alan Yuille, and Christof Koch. Adaboost for text detection in natural scene. In *2011 International Conference on Document Analysis and Recognition*, pages 429–434. IEEE, 2011. 2

[29] Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 2, 3, 4

[30] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 1, 3, 4, 6, 8

[31] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. *arXiv preprint arXiv:1611.06779*, 2016. 3

[32] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5676–5685, 2018. 3, 7

[33] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9809–9818, 2020. 1

[34] Shangbang Long, Xin He, and Cong Yao. Scene text detection and recognition: The deep learning era. *International Journal of Computer Vision*, pages 1–24, 2020. 2

[35] Simon M Lucas, Alex Panaretos, Luis Sosa, Anthony Tang, Shirley Wong, and Robert Young. Icdar 2003 robust reading competitions. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 682–687. Citeseer, 2003. 1

[36] Pengyuan Lyu, Cong Yao, Wenhao Wu, Shuicheng Yan, and Xiang Bai. Multi-oriented scene text detection via corner localization and region segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7553–7563, 2018. 7

[37] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 20(11):3111–3122, 2018. 8

[38] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Cheng-lin Liu, and Jean-Marc Ogier. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1582–1587. IEEE, 2019. 3, 5, 7, 8

[39] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1454–1459. IEEE, 2017. 3, 5, 7, 8

[40] Lukáš Neumann and Jiří Matas. Real-time scene text localization and recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3538–3545. IEEE, 2012. 2, 3

[41] Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. Leep: A new measure to evaluate transferability of learned representations. In *International Conference on Machine Learning*, pages 7294–7305. PMLR, 2020. 8

[42] Shigueo Nomura, Keiji Yamanaka, Osamu Katai, Hiroshi Kawakami, and Takayuki Shiose. A novel adaptive morphological approach for degraded character image segmentation. *Pattern Recognition*, 38(11):1961–1975, 2005. 2

[43] Liang Qiao, Sanli Tang, Zhanzhan Cheng, Yunlu Xu, Yi Niu, Shiliang Pu, and Fei Wu. Text perceptron: Towards end-to-end arbitrary-shaped text spotting. *arXiv preprint arXiv:2002.06820*, 2020. 1

[44] Siyang Qin, Alessandro Bissacco, Michalis Raptis, Yasuhisa Fujii, and Ying Xiao. Towards unconstrained end-to-end text spotting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4704–4714, 2019. 1, 2, 3

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2

[46] Jose A Rodriguez-Serrano, Florent Perronnin, and France Meylan. Label embedding for text recognition. In *BMVC*, pages 5–1, 2013. 2

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Int. Conf. on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3

[48] Nabin Sharma, Ranju Mandal, Rabi Sharma, Umapada Pal, and Michael Blumenstein. Icdar2015 competition on video script identification (cvsi 2015). In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1196–1200. IEEE, 2015. 2

[49] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4168–4176, 2016. 2

[50] Baoguang Shi, Cong Yao, Minghui Liao, Mingkun Yang, Pei Xu, Linyan Cui, Serge Belongie, Shijian Lu, and Xiang Bai. Icdar2017 competition on reading chinese text in the wild (rctw-17). In *ICDAR*, 2017. 5

[51] Baoguang Shi, Cong Yao, Chengquan Zhang, Xiaowei Guo, Feiyue Huang, and Xiang Bai. Automatic script identification in the wild. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 531–535. IEEE, 2015. 2

[52] Cunzhao Shi, Chunheng Wang, Baihua Xiao, Yang Zhang, Song Gao, and Zhong Zhang. Scene text recognition using part-based tree-structured character detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2961–2968, 2013. 2

[53] Amanpreet Singh, Guan Pang, Mandy Toh, Jing Huang, Tal Hassner, and Wojciech Galuba. Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1

[54] Bolan Su and Shijian Lu. Accurate scene text recognition based on recurrent neural network. In *Asian Conference on Computer Vision*, pages 35–48. Springer, 2014. 2

[55] Yipeng Sun, Zihan Ni, Chee-Kheng Chng, Yuliang Liu, Canjie Luo, Chun Chet Ng, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, Chee Seng Chan, and Lianwen Jin. Icdar 2019 competition on large-scale street view text with partial labeling – rrc-lsvt. In *ICDAR*, 2019. 5

[56] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014. 2

[57] TN Tan. Rotation invariant texture features and their use in automatic script identification. *IEEE Transactions on pattern analysis and machine intelligence*, 20(7):751–756, 1998. 2

[58] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *European conference on computer vision*, pages 56–72. Springer, 2016. 2

[59] Anh T Tran, Cuong V Nguyen, and Tal Hassner. Transferability and hardness of supervised classification tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1395–1405, 2019. 8

[60] Bala R Vatti. A generic solution to polygon clipping. *Communications of the ACM*, 35(7):56–64, 1992. 3

[61] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016. 1

[62] Kai Wang and Serge Belongie. Word spotting in the wild. In *European Conference on Computer Vision*, pages 591–604. Springer, 2010. 1

[63] Tao Wang, David J Wu, Adam Coates, and Andrew Y Ng. End-to-end text recognition with convolutional neural networks. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, pages 3304–3308. IEEE, 2012. 2

[64] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9336–9345, 2019. 3, 8

[65] Yue Wu, Tal Hassner, KangGeon Kim, Gerard Medioni, and Prem Natarajan. Facial landmark detection with tweaked convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):3067–3074, 2017. 3

[66] Linjie Xing, Zhi Tian, Weilin Huang, and Matthew R Scott. Convolutional character networks. In *Int. Conf. Comput. Vis.*, 2019. 7

[67] Xu-Cheng Yin, Xuwang Yin, Kaizhu Huang, and Hong-Wei Hao. Robust text detection in natural scene images. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):970–983, 2013. 2

[68] Liu Yuliang, Jin Lianwen, Zhang Shuaitao, and Zhang Sheng. Detecting curve text in the wild: New dataset and new solution. *arXiv preprint arXiv:1712.02170*, 2017. 5

[69] Rui Zhang, Yongsheng Zhou, Qianyi Jiang, Qi Song, Nan Li, Kai Zhou, Lei Wang, Dong Wang, Minghui Liao, Mingkun Yang, et al. Icdar 2019 robust reading challenge on reading chinese text on signboard. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1577–1581. IEEE, 2019. 8