MRN: Multiplexed Routing Network for Incremental Multilingual Text Recognition

Tianlun Zheng[1,2], Zhineng Chen[1,2]*, Bingchen Huang[1,2], Wei Zhang[3] and Yu-Gang Jiang[1,2]
[1]School of Computer Science, Fudan University, China
[2]Shanghai Collaborative Innovation Center of Intelligent Visual Computing, China
[3]Gaoding AI, China
{tlzheng21, bchuang21}@m.fudan.edu.cn, {zhinchen, ygj}@fudan.edu.cn, wzhang.cu@gmail.com

Abstract

Multilingual text recognition (MLTR) systems typically focus on a fixed set of languages, which makes it difficult to handle newly added languages or adapt to ever-changing data distribution. In this paper, we propose the Incremental MLTR (IMLTR) task in the context of incremental learning (IL), where different languages are introduced in batches. IMLTR is particularly challenging due to rehearsal-imbalance, which refers to the uneven distribution of sample characters in the rehearsal set, used to retain a small amount of old data as past memories. To address this issue, we propose a Multiplexed Routing Network (MRN). MRN trains a recognizer for each language that is currently seen. Subsequently, a language domain predictor is learned based on the rehearsal set to weigh the recognizers. Since the recognizers are derived from the original data, MRN effectively reduces the reliance on older data and better fights against catastrophic forgetting, the core issue in IL. We extensively evaluate MRN on MLT17 and MLT19 datasets. It outperforms existing general-purpose IL methods by large margins, with average accuracy improvements ranging from 10.3% to 35.8% under different settings. Code is available at https://github.com/simplify23/MRN.

1. Introduction

Scene text recognition (STR) is a task aiming to read text in natural scenes. Recent advances in deep learning have significantly improved the accuracy of STR, allowing it to recognize text in the presence of font variations, distortions, and noise interference [39, 40, 43, 38, 19, 53]. As countries and cultures are more interconnected, the task of simultaneously recognizing multiple languages, i.e., multilingual text recognition (MLTR), has also become more important.

*Corresponding Author.

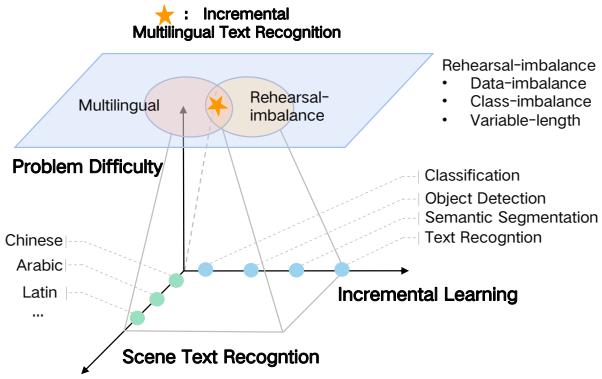

Figure 1. Incremental multilingual text recognition (IMLTR) focuses on the practical scenario where different languages are introduced sequentially. The goal is to accurately recognize the newly introduced language while maintaining high recognition accuracy for previously seen languages. IMLTR introduces a task focusing on text recognition that faces rehearsal-imbalance challenges.

Existing methods typically address this challenge by training on mixed multilingual data [8, 4, 34] or designing independent language blocks [29, 22, 24]. However, when each time a new language is added, the above methods need retraining on a dataset mixing the old and new languages. This increases the training cost [37, 46] and also may lead to an imbalance [7, 14] between old and new data.

Incremental learning (IL) is designed for scenarios where new data is continuously learned and typically, the old samples are maintained by a small ratio. The collection of old samples is referred to as the rehearsal set [51, 27], which serves as limited past memories. IL aims to learn the new data well while minimizing forgetting the past learned knowledge. Most existing studies [37, 7, 52, 28] conduct experiments on balanced datasets and maintain a constant number of classes at each learning step. However, in real-world scenarios, the number of classes and samples may differ across steps, leading to imbalanced datasets. To ad-

dress these issues, IL2M [7] alleviated class-imbalance by storing statistics of old classes rather than samples. Delange et al. De Lange et al. [14] surveyed typical IL methods on datasets and solutions with different data imbalances. Despite progress made, research on data and class imbalance is still in its infancy stage. Moreover, as illustrated in Fig. 1, there is currently no research introducing IL to STR.

We rewrite MLTR in the context of IL. Languages are treated as tasks and characters are their classes. During training, the model only observes the newly arrived language data and a small amount of data from old languages. The recognition model is expected to maintain the ability to recognize characters of all languages that it has encountered before, regardless of whether their data are still available or discarded. We term this problem incremental multilingual text recognition (IMLTR).

IMLTR poses significant challenges to IL approaches due to its unbalanced features. 1) At the dataset level, it is difficult to collect sufficient training data for minority languages such as Bangla compared to popular languages such as English and Chinese, which affects the quality of recognition models. 2) At the language level, the size of character sets varies from tens to thousands across different languages, which leads to data imbalance. 3) At the character level, the occurrence frequency of characters follows a long-tailed distribution, leading to class imbalance. In addition, IMLTR faces the problem of variable length recognition, where text instances are the recognizing unit instead of character classes. Therefore, IL methods cannot sample characters as evenly as required in the context of IMLTR, resulting in a significant fraction of characters not being included in the rehearsal data, as shown in Fig. 2. This phenomenon is summarized as rehearsal-imbalance in Fig. 1. Rehearsal-imbalance leads to catastrophic forgetting, where forgotten characters cannot be recognized. Therefore, there is an urgent need to develop new methods to overcome it.

Although the rehearsal set does not ensure full coverage of all interlingual character classes, it is still adequate for training a language domain predictor to identify the languages. Motivated by this observation, we propose a novel Multiplexed Routing Network (MRN) for IMLTR. MRN involves training a new text recognition model at each learning step and utilizing it and previously trained models for parallel feature extraction. A domain MLP router is designed to receive these features and predict the probability over the languages. Meanwhile, these features are used for character recognition in their own domain by feeding them to the multi-lingual modeling module. Finally, we fuse the results obtained at both the language domain and character levels to decode the recognized character sequence.

Our contributions can be summarized as follows. First, we introduce the IMLTR task, the first effort to adapt IL to text recognition. It contributes to the exploration of other practical scenarios for text recognition. Second, we develop MRN to address the rehearsal-imbalance problem in ILMTR. It is a dynamic and scalable architecture that is compatible with various IL methods and recognition models. Third, experiments on two benchmarks show that MRN significantly outperforms existing general-purpose IL methods, achieving accuracy improvements ranging from 10.3% to 27.4% under different settings.

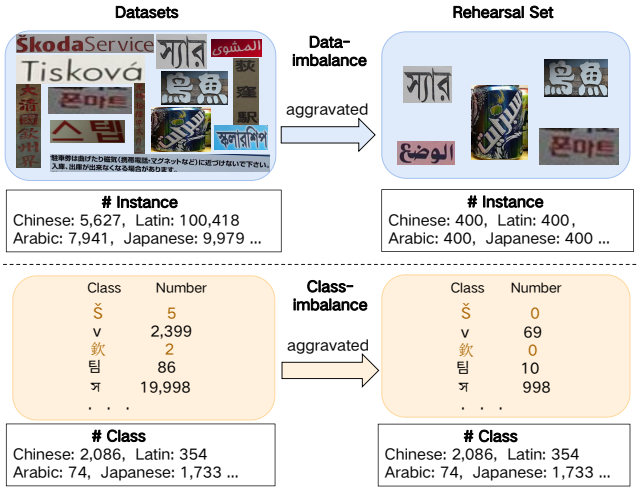

Figure 2. The showcase of rehearsal-imbalance. Data-imbalance (top) and class-imbalance (bottom) are severely aggravated from the full dataset to the rehearsal set, while the character classes to be read remain the same, making IMLTR particularly challenging.

2. Related Work

2.1. Incremental Learning (IL)

IL has received intensive research attention over the past few years. Typically, the problem is investigated in the context of image classification, where addressing catastrophic forgetting effectively and efficiently is its core issue. We can broadly classify existing efforts into three categories: regularization [30, 50, 15], rehearsal [36, 9, 1] and dynamic expansion [2, 46, 17, 28]. Regularization methods emphasize constraining weight changes, e.g., allowing only small magnitude changes from the previous weights. It suffers from the problem that the changes do not adequately describe the complex pattern shifts caused by new task learning. Rehearsal methods keep a small amount of old data when training a new task, thus retaining some prior knowledge. Studies in this category focus on the selection of old data and the way it is used. For example, iCaRL was developed to learn an exemplar-based data representation [37]. Alternatively, dynamic expansion methods dynamically create feature extraction sub-networks each associated with one specific task [23, 12, 45, 28]. Early methods required a task identifier to select the correct sub-network at test time. Unfortunately, the assumption is unrealistic as new sam-

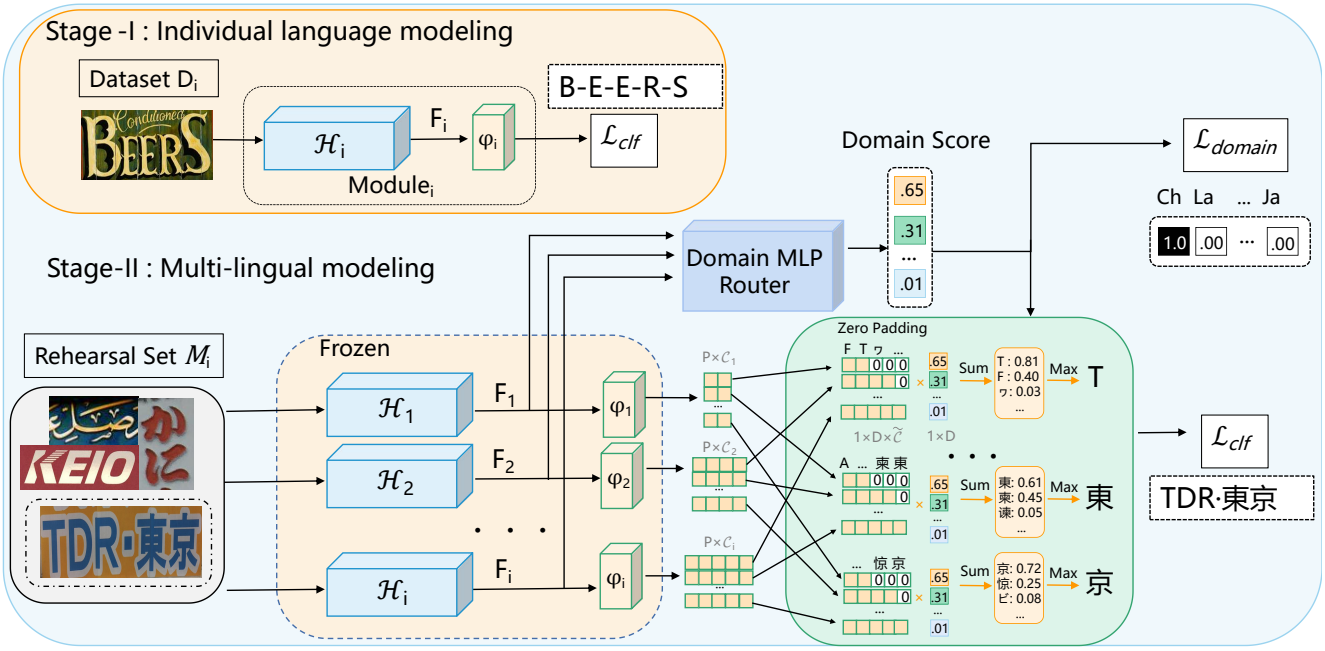

Figure 3. An overview of MRN. In stage-I, text recognizers are trained language-by-language. While in stage-II, these recognizers are frozen for feature extraction. The Domain MLP Router, which is trained based on the rehearsal set, is proposed to predict the likelihood distribution over the languages. Meanwhile, a padded classification layer is constructed, where the parallel predicted text sequences and likelihood distributions are merged to generate the decoded character sequence.

ples would not come with their task identifiers. Recently, DER [46] proposed a dynamically expandable representation by discarding the task identifier, where the classifier was finetuned on a balanced exemplar subset to mitigate the task-tendency bias. It attained impressive results. Some recent works [7, 14] studied IL in inhomogeneous or uneven datasets. However, the datasets they adopted are still ideal and cannot sufficiently describe challenges in real-world problems. Moreover, there were also some studies proposed for object detection [21, 10, 48, 47], semantic segmentation [49, 16, 51] and object retrieval [32]. Text recognition has not been studied in IL so far.

2.2. Scene Text Recognition (STR)

Text recognition is a longstanding research topic in computer vision and pattern recognition. Recent efforts mainly focused on recognizing text in natural scenes, i.e., STR. The task exhibits variations like text distortion, occlusion, blurring, etc., making the recognition challenging. With the advances in deep learning, especially CNN [5, 39, 26, 40] and Transformers [38, 20, 54, 43, 18, 44], STR methods have been pushed forward significantly.

Multilingual text recognition (MLTR) is an important sub-field of STR. The most popular solution for MLTR was data-joint training [35, 8, 34, 4], where all data was gathered to train a model capable of recognizing all character classes. However, in addition to computational intensive, the approach also had the drawback of being biased toward data-rich languages, while performing poorly in minority languages where training data was scarce. As alternatives, multi-task or ensemble architectures were developed to allow data-rich languages to transfer knowledge to data-poor ones [6, 13]. They alleviated the data scarcity issue to some extent. In addition, Some studies [22, 24, 41, 29] added a script identification step to text recognition. They first identified the language domain and then selected the corresponding recognizer. Although similar to ours in the pipeline, they did not explore dependencies between languages. Moreover, none of them discussed the task within the IL framework.

3. Methodology

3.1. Incremental Multilingual Text Recognition

Our goal is to develop a unified model that can recognize text instances in different languages, with the model trained incrementally language-by-language. Mathematically, assume there are I kinds of languages $\{\mathcal{D}_1, \cdots, \mathcal{D}_I\}$, with $\mathcal{D}_i = \{(\mathbf{x}_{i,1}, y_{i,1}), \cdots, (\mathbf{x}_{i,N(i)}, y_{i,N(i)})\}$ as the training data at step i (i.e., task i), where $x_{i,j}$ is the j-th input image and $y_{i,j} \in \mathcal{C}_i$ is its label within the label set $\mathcal{C}_i$, $N(i)$ is the number of samples in set $\mathcal{D}_i$. At the i-th learning step, samples of the i-th language will be added to the training set. Therefore, the goal can be formulated as to learn

new knowledge from the set $\mathcal{D}_i$, while retaining the previous knowledge learned from old data $\{\mathcal{D}_1, \cdots, \mathcal{D}_{i-1}\}$. The label space of the model is all seen categories $\tilde{\mathcal{C}}_i = \cup_{k=1}^{i} \mathcal{C}_k$ and the model is expected to predict well on all classes in $\tilde{\mathcal{C}}_i$. Note that there may be a small overlap between label sets, i.e., $\mathcal{C}_k \cap \mathcal{C}_j \neq \emptyset$ for some k and j. To better fight against catastrophic forgetting, we discuss IMLTR in the rehearsal setting. That is, a small and fix-sized rehearsal set $\mathcal{M}_i$ with a portion of samples from $\{\mathcal{D}_1, \cdots, \mathcal{D}_{i-1}\}$ is accessible at incremental step i.

3.2. Challenge and Solution Statement

To build a recognition model to correctly recognize text instances from all currently seen languages and their character classes, let x_n be the text instance to be recognized. $y_n^t \in \tilde{\mathcal{C}}_i$ denotes the t-th character label corresponding to x_n. $T(n)$ gives the total number of characters in this instance. IMLTR differs significantly from existing IL settings. For example compared to incremental image classification, standard IL usually has $|\tilde{\mathcal{C}}_i| \leq 100$ and $T(n) = 1$ regardless of the value n. While the size of rehearsal set $\mathcal{M}_i$ is a constant (e.g., 2,000). However, in IMLTR $\mathcal{C}_i$ ranges from dozens of to thousands of character classes for different languages, and $T(n)$ belongs to (1, 25), assuming 25 as the maximized length of a character sequence. Consequently, rehearsal-imbalance becomes a prominent challenge. Due to the limited size of the rehearsal set, it is not rare that a character class appears in the full dataset but is absent from the rehearsal set, as shown in Fig. 2. Thus, the incrementally trained models are likely to forget the absent character classes, despite having learned them previously, which can ultimately hurt the recognition accuracy.

Although the rehearsal set may not be enough to train a multilingual text recognizer to identify thousands of character classes, it is still sufficient to train a language classifier to recognize the language domains present in the text instance, whose classes are a much smaller number. Once the language domains are identified, we can choose an alternative scheme that involves aggregating the results from corresponding language recognizers to perform the recognition task, thereby bypassing the rehearsal-imbalance issue.

Motivated by this, we define $\mathcal{H}_i$ and φ_i the skeleton network (all except classifier) and classifier trained at the i-th incremental step. Note that $\mathcal{H}_i$ is trained on $\mathcal{D}_i$, therefore can only recognize character classes of the i-th language in principle. Meanwhile, φ_i is set to have $\tilde{\mathcal{C}}_i$ nodes to be compatible with typical IL settings, despite not being taught to recognize character classes of other languages. Then, we can adopt an aggregating-like scheme to implement IMLTR. The learning function can be written as:

$$\sum_{k=1}^{i} \prod_{t=1}^{T(n)} \left(P\left(y_n^t | x_n; \mathcal{H}_k, \varphi_k\right) * S\left(d_n^k\right) \right), \tag{1}$$

where d_n^k is the domain score indicating x_n being classified as the k-th language. $S(\cdot)$ is the score quantization function, which can be a one-hot vector (hard-voting) or a likelihood distribution (soft-voting). Eq. 1 treats IMLTR as a weighted ensemble of recognition models trained based on different languages. By doing so, it successfully overcomes the rehearsal-imbalance issue within the IL framework.

3.3. Method Overview

We propose a Multiplexed Routing Network (MRN) to implement this idea. As illustrated by Fig. 3, it contains two stages, i.e., individual language modeling (stage-I) and multi-lingual modeling (stage-II). In stage-I, given $\mathcal{D}_i$ for the *i*-th language, we train its recognizer using a popular text recognition model, which can recognize the character classes seen in $\mathcal{D}_i$. The model is represented as $\mathcal{H}_i$ and φ_i. For character classes in $\tilde{\mathcal{C}}_i$ but not in $\mathcal{C}_i$, we simply truncate gradient propagation from these nodes thus the learned model still focuses on recognizing the i-th language.

Stage-II aims at building a multilingual routing network for IMLTR. Given a text instance $x_n \in \mathcal{D}_i \cup \mathcal{M}_i$, we feed it into all the learned *i* skeleton networks in parallel, while keeping the parameters of the networks frozen for targeted feature extraction. It extracts *i* sets of features, each associated with a certain language. The features are further fed into a Domain MLP Router (DM-Router) module, which is designed for domain score estimation, i.e., estimating the likelihood that the text instance belongs to the languages. Meanwhile, the *i* sets of features are fed to their respective classifiers, where the corresponding recognition character sequences are obtained. To merge their recognition, we pad the classification nodes with zeros to $|\tilde{\mathcal{C}}_i|$, ensuring that all classifiers are aligned to the same dimension. As a result, their recognized characters can be merged using weighted element-wise addition, where the weights are the domain scores estimated using DM-Router. Finally, the recognition is conducted by applying a CTC- or attention-based decoding. Since DM-Router plays a critical role in the proposed method, we provide a detailed illustration below.

3.4. Domain MLP Router

DM-Router uses features that are biased towards different language domains to discriminate the language domain of text instances. It accomplishes this by leveraging both the rehearsal set and the language data that arrives at the i-th step. While training a separate single-network classifier, which takes an image as input and outputs the language domain scores, can identify the language domains, we believe that this approach overlooks inter-domain dependencies that could be explored for better identification. For instance, different languages may have distinct appearance patterns, such as strokes, which differ significantly between Eastern Asian languages and Latin. Additionally, their features ex-

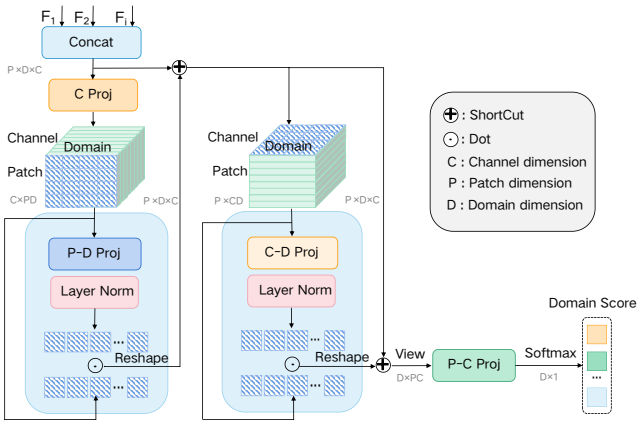

Figure 4. Detail structure of Domain MLP Router (DM-Router). Spatial-domain and Channel-domain dependencies are explored and fused to generate the language domain score distribution.

hibit different frequency distributions, which can also aid language identification.

To achieve this goal, DM-Router accepts all i sets of features extracted previously as input and mines the spatial-domain and channel-domain dependencies for better language identification. The detailed structure of DM-Router is shown in Fig. 4. Features from different skeleton networks are concatenated, generating a feature cubic with size $P \times D \times C$, where P, D, and C stand for the number of reshaped spatial patches, language domains, and feature channels, respectively. Then, a linear projection is applied along the channel dimension (C Proj), followed by reshaping the cubic from the patch-domain dimension. Next, a gated-mechanism is followed to generate the attention scores between the patch and domain. This is achieved by applying linear projection along the patch-domain dimension, followed by a layer norm and a feature dot product. We reshape the generated feature back to a feature cubic of the same size and merge it with the original cubic. The operations above explore the spatial-domain dependency. A similar operation is then applied to the merged feature cubic to explore the channel-domain dependence. In the following, the explored feature cubic gradually shrinks to a D-dimensional score vector that indicates the probability over the languages. It represents the likelihood of each language domain for the input text instance.

DM-Router is an MLP-based attention network that targets language domain weighting. Note that there are a few similar solutions in the literature. Expert Gate (E-Gate) [2] developed an expert gating network that identified which model could be employed based on image reconstruction loss. However, it might not effectively discriminate IMLTR due to some languages exhibiting character class overlapping, which can cause classification confusion. On the other hand, multilingual OCR [29] determines the languages by script recognition and selected the corresponding model for recognition. Unlike these hard-voting methods, MRN employs soft-voting, which allows for the use of knowledge from other languages. For instance, Japanese has the ability to correct Chinese to some extent, given that they share some common words and similar strokes.

Dataset	categories	Task1	Task2	Task3	Task4	Task5	Task6
		Chinese	Latin	Japanese	Korean	Arabic	Bangla
MLT17[35]	train instance	2687	47411	4609	5631	3711	3237
	test instance	529	11073	1350	1230	983	713
	train class	1895	325	1620	1124	73	112
MLT19[34]	train instance	2897	52921	5324	6107	4230	3542
	test instance	322	5882	590	679	470	393
	train class	2086	220	1728	1160	73	102

Table 1. MLT17 and MLT19 statistics in our experiments.

3.5. Training Loss

MRN has two loss terms. One for multilingual text recognition while the other for language domain prediction. The total loss function is written as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{clf}} + \alpha \mathcal{L}_{\text{domain}}, \tag{2}$$

where α is an empirical hyperparameter to balance the two.

MRN shows two advantages in dealing with rehearsal-imbalance. First, it ensures fair use of language. As previously mentioned, data distribution is uneven across different languages, and if not addressed during model training, it may lead to bias in the resulting model. By adopting language-by-language training and parameter freezing, data-rich and data-poor languages are treated equally, and class-imbalance is also alleviated. Second, MRN makes use of inter-lingual dependency in two ways: through the DM-Router described earlier, and through recognition score merging. When a character is recognized by more than one language, it receives confidence scores from each of them, allowing for the utilization of inter-lingual dependencies.

4. Experiments

4.1. Datasets and Implementation Details

ICDAR 2017 MLT (MLT17) [35] has 68,613 training instances and 16,255 validation instances, which are from 6 scripts and 9 languages: Chinese, Japanese, Korean, Bangla, Arabic, Italian, English, French, and German. The last four use Latin script. The samples are from natural scenes with challenges like blur, occlusion, and distortion. We use the validation set for test due to the unavailability of test data. Tasks are split by scripts and modeled sequentially. Special symbols are discarded at the preprocessing step as with no linguistic meaning.

ICDAR 2019 MLT (MLT19) [34] has 89,177 text instances coming from 7 scripts. Since the inaccessibility of test set, we randomly split the training instances to 9:1 script-by-script, for model training and test. To be consistent with

	MLT17							MLT19						
Model : CRNN (TPAMI'17) [39]														
	T1	T2	T3	T4	T5	T6	AVG	T1	T2	T3	T4	T5	T6	AVG
Bound	-	-	-	-	-	-	92.1	-	-	-	-	-	-	84.9
Baseline	91.1	51.7	51.0	37.2	29.3	22.3	47.1	85.1	49.6	46.5	35.5	27.6	20.7	44.2
LwF (TPAMI'17)[31]	91.1	53.7	53.4	38.2	29.7	23.7	48.3	85.1	51.6	49.2	36.5	27.7	22.0	45.3
EWC (PNAS'17)[30]	91.1	56.5	50.4	37.2	30.5	21.5	47.9	85.1	55.5	46.3	35.8	28.8	19.9	45.2
WA (CVPR'20) [52]	91.1	54.6	48.7	38.2	28.5	23.1	47.4	85.1	52.2	44.3	36.7	26.8	21.6	44.4
DER (CVPR'21)[46]	91.1	76.3	55.8	46.4	39.3	35.8	57.5	85.1	75.2	40.4	45.1	36.6	34.2	52.8
MRN	91.1	**88.6**	**77.2**	**73.7**	**69.8**	**69.8**	**78.4**	85.1	**85.1**	**73.2**	**68.3**	**65.3**	**65.5**	**73.7**
Model : TRBA (ICCV'19) [3]														
	T1	T2	T3	T4	T5	T6	AVG	T1	T2	T3	T4	T5	T6	AVG
Bound	-	-	-	-	-	-	94.9	-	-	-	-	-	-	90.5
Baseline	91.3	49.6	47.3	36.1	28.6	24.0	46.1	85.4	49.4	44.0	34.8	27.4	23.1	44.0
LwF (TPAMI'17)[31]	91.3	55.7	38.8	28.7	22.6	18.7	42.6	85.4	54.2	35.0	27.2	20.5	17.0	39.9
EWC (PNAS'17)[30]	91.3	50.4	43.6	33.1	25.6	21.9	44.3	85.4	49.4	40.6	31.7	24.8	20.6	42.1
WA (CVPR'20) [52]	91.3	45.4	41.8	30.7	23.5	19.6	42.1	85.4	44.0	37.9	29.2	21.6	18.1	39.4
DER (CVPR'21)[46]	91.3	60.1	53.0	38.8	31.4	28.6	50.5	85.4	60.7	50.3	37.2	30.3	28.1	48.7
MRN	91.3	**87.9**	**75.8**	**72.2**	**71.5**	**68.7**	**77.9**	85.4	**84.5**	**73.2**	**67.8**	**66.7**	**64.8**	**73.7**
Model : SVTR-Base (IJCAI'22) [18]														
	T1	T2	T3	T4	T5	T6	AVG	T1	T2	T3	T4	T5	T6	AVG
Bound	-	-	-	-	-	-	90.1	-	-	-	-	-	-	83.2
Baseline	90.6	32.5	40.5	30.8	24.5	19.9	39.8	84.8	31.3	37.0	29.2	22.6	19.1	37.3
LwF (TPAMI'17)[31]	90.6	28.0	38.4	29.9	24.1	18.3	38.2	84.8	27.0	34.6	28.4	22.3	17.0	35.7
EWC (PNAS'17)[30]	90.6	33.0	41.2	31.1	24.6	20.0	40.1	84.8	31.3	37.7	29.5	22.6	19.0	37.5
WA (CVPR'20) [52]	90.6	28.0	37.9	30.4	24.8	19.8	38.6	84.8	26.7	34.6	28.3	22.6	18.6	35.9
DER (CVPR'21)[46]	90.6	74.5	55.7	55.0	49.5	45.7	61.8	84.8	71.6	52.9	52.2	46.6	43.6	58.6
MRN	90.6	**86.4**	**73.9**	**65.6**	**63.4**	**58.1**	**73.0**	84.8	**83.7**	**69.4**	**64.4**	**57.8**	**53.1**	**68.9**

Table 2. Accuracy (%) of different text recognizers and incremental learning methods on MLT17 and MLT19. *Baseline* denotes the model trained solely based on the rehearsal set and language data arrived at that step. The language incremental order is introduced in Sec. 4.1.

MLT2017 dataset, we discard the Hindi script and also special symbols. Statistics of the two datasets are shown Tab. 1.

Height and width of the images are scaled uniformly to 32×256. The maximum length of a character sequence is set to 25. All models, each corresponding to a language domain, are trained with 10,000 iterations, using the Adam optimizer and the one-cycle learning rate scheduler [42] with a maximum learning rate of 0.0005. The batch size is set to 256. To mitigate the dataset variance, in each batch we evenly sample training samples from both datasets, that is, half from MLT17 and half from MLT19. A random order for the six languages is employed, which is Chinese, Latin, Japanese, Korean, Arabic, Bangla. Other orders will be discussed later. For the rehearsal setting, we limit the rehearsal size to 2000 samples unless specified. We conduct the experiments using two NVIDIA RTX 3090 GPUs.

4.2. Comparison with Existing Methods

We equip MRN with different text recognizers and combine them with different IL methods. Specifically, we consider three typical STR schemes: CTC-based (CRNN [39]), attention-based (TRBA [3]), and ViT-based (SVTR [18]). Meanwhile, four popular IL methods are chosen, i.e., Lwf [31], EWC [30], WA [52] and DER [46]. All models retain their original settings, except for the removal of the auxiliary loss of DER, which reduces its performance in our task.

In Tab. 2, we give the results at different incremental steps, where the language is added one-by-one and the average accuracy of different methods is reported. *Bound*, the model trained using all training data, is also listed as the oracle for reference. As can be seen, MRN consistently outperforms all the compared methods by significant margins under different settings, no matter which recognizer is employed. When looking into the general-purpose IL methods, their accuracy mostly decreased rapidly as the incremental steps due to the affection of rehearsal-imbalance. DER has the highest accuracy among them, as its dynamic expansion architecture has certain advantages in fighting against catastrophic forgetting. However, there is still a clear accuracy gap between DER and our MRN, and the gap widens as the incremental step increases. We attribute the accuracy improvement achieved by MRN to two factors. First, IMLTR is a task that differs significantly from image classification, where most IL methods have been experimented on. These methods do not well accommodate the challenge raised by IMLTR. For example, the rehearsal-imbalance issue. Sec-

ond, MRN develops an elegant pipeline that implements the recognition in a domain routing and result fusion manner. It works particularly well for scenarios where incremental tasks exhibit significant differences.

When comparing the recognizers, MRN equipped with CRNN has the highest overall accuracy. The result is interesting as CRNN has a simpler architecture and generally performs worse than the other two methods on typical STR tasks. We attribute this to parameter freezing, where the feature extraction backbone (e.g., $\mathcal{H}_i$) and the decoder cannot be jointly optimized. Therefore, advanced models are more severely affected, while the simpler one is less affected and can better mitigate catastrophic forgetting.

model	MLT17		MLT19	
	Avg	Last	Avg	Last
None	64.8	37.9	60.8	35.6
MLP	68.5	60.5	65.3	56.3
CycleMLP[11]	75.5	63.5	71.1	60.0
ViP[25]	76.4	62.6	72.2	59.6
gMLP[33]	77.5	68.2	73.1	64.2
DM-Router	**78.4**	**69.8**	**73.8**	**65.5**

Table 3. Performance comparisons on different MLP models.

4.3. Ablation Study

We perform a series of controlled experiments to gain a deeper understanding of MRN. CRNN is employed as the text recognizer unless specified.

Effectiveness of DM-Router: There are multiple ways to deduce the language domain scores. We enumerate several of them that have been used in existing studies, as shown in Tab. 3. *None* denotes no dependence is explored, which corresponds to the worst result. It, in turn, demonstrates the necessity of utilizing language dependence. Among the rest competitors, MLP enables a naive learning mechanism while the remaining three are based on more advanced MLP-like models, which are typically more effective. Despite this, DM-Router attains the highest accuracy among the methods. The results clearly demonstrate the rationality of the DM-Router structure in terms of language dependence exploration.

Influence of the size of the rehearsal set: We conduct analytical experiments to evaluate the influence of the rehearsal size on the accuracy of Lwf [31], DER [46] and MRN. Fig. 4 shows the accuracy under different rehearsal sizes. As anticipated, increasing the rehearsal set size leads to accuracy gains, as more past memories are retained. We observe that larger gains are obtained in LwF and DER, particularly DER. This reveals the accuracy of general-purpose IL methods is largely affected by the rehearsal size in IMLTR, while MRN is less affected. MRN has already achieved relatively high accuracy, and the performance of MRN in identifying language domains is less affected by the rehearsal size. The results indicate that MRN is robust to rehearsal scarcity and can better fight against data imbalance.

Size	Method	MLT17		MLT19	
		Avg	Last	Avg	Last
2k	LwF[31]	48.3	23.7	45.4	22.0
	DER[46]	57.5	35.8	52.8	34.2
	MRN	**78.4**	**69.8**	**73.8**	**65.5**
3k	LwF[31]	52.2	24.9	48.8	23.6
	DER[46]	60.9	42.0	58.7	40.6
	MRN	**80.2**	**72.7**	**75.4**	**68.2**
4k	LwF[31]	55.5	27.5	52.2	26.1
	DER[46]	66.4	48.7	63.8	46.6
	MRN	**81.5**	**75.0**	**76.5**	**70.6**

Table 4. Ablation study on the size of the rehearsal set.

Order	Method	MLT17		MLT19	
		Avg	Last	Avg	Last
O1	LwF[31]	48.3	23.7	45.4	22.0
	DER[46]	57.5	35.8	52.8	34.2
	MRN	**78.4**	**69.8**	**73.8**	**65.5**
O2	LwF[31]	46.9	23.8	43.1	22.9
	DER[46]	63.1	39.1	58.7	39.6
	MRN	**80.5**	**65.3**	**74.1**	**61.5**
O3	LwF[31]	57.7	34.7	55.7	34.2
	DER[46]	69.6	41.3	65.7	38.2
	MRN	**82.9**	**70.6**	**78.3**	**66.0**

Table 5. Ablation study on language order.

Sampling Strategy	MLT17		MLT19	
	Avg	Last	Avg	Last
Confidence	56.4	43.8	54.0	41.2
Length	71.0	50.3	66.6	48.9
Frequency	72.6	56.6	67.8	53.7
Random	**78.4**	**69.8**	**73.8**	**65.5**

Table 6. Ablation study on rehearsal sampling strategy.

Influence of language incremental order: In addition to the order in Sec. 4.1 (O1), we assess two other orders as follows: 1) Arabic, Chinese, Latin, Japanese, Bangla, Korean (O2); 2) Latin, Arabic, Bangla, Chinese, Japanese, Korean (O3). The two orders either alternate the three Eastern Asia languages, which have large vocabularies and show more stroke commonalities, or group them together at the end. We also include Lwf and DER for comparison.

Tab. 5 gives the results and two observations. First, O3 shows the best accuracy, while O2 also performs better than O1. It is because the three Eastern Asia languages are more difficult to recognize due to their large vocabulary sizes, therefore introducing them later leads to a better average accuracy. Meanwhile, putting them together also reduces the

Method	Select	Model	Voting	MLT17		MLT19		Params (M)	FLOPs (G)
				Avg	Last	Avg	Last		
Baseline	–	–	–	47.1	22.3	44.2	20.7	9.5	3.5
DER[46]	–	–	–	57.5	35.8	52.8	34.2	33.8	12.3
E-Gate[2]	Re-Const.	Autoencoder	Hard	37.2	15.2	34.8	14.2	32.5	12.2
E-Gate[2]	Stacking	Autoencoder	Hard	62.7	15.2	59.3	14.2	35.5	12.4
MRN	Stacking	DM-Router	Hard	74.4	62.9	69.9	57.7	33.5	12.4
MRN	Stacking	DM-Router	Soft	**78.4**	**69.8**	**73.8**	**65.5**	33.5	12.4

Table 7. Comparisons on different routing strategies.

oscillation during parameter learning and generates a better model, due to their stroke commonalities. The experiment suggests that careful selection of the order of languages can attain better accuracy. Second, O1 shows the largest accuracy gaps between MRN and other methods. This is because in O1, the large vocabulary languages appear earlier, while the rehearsal set is fix-sized, resulting in the most severe class imbalance among the three orders. The result indicates that MRN can better handle class imbalance.

Influence of rehearsal sampling strategy: The determination of text instances being sampled to the rehearsal set is an issue also worthy of ablating. Tab. 6 gives the accuracy of four sampling strategies, i.e., *Confidence* that selects instances with the highest recognition scores, *Length* that selects instances with the largest number of characters, *Frequency* that selects instances with the most frequently occurred characters, and *Random* adopted in our MRN that randomly selects the instances. Interestingly, *Random* gives the best accuracy. We attribute the reason to: the rehearsal set obtained from *Confidence* or *Frequency* cannot fully represent the true data distribution, where difficult or less occurred instances are excluded. *Length*, to some extent, overlooks the varying-length characteristic of IMLTR. On the contrary, *Random*, despite simple, well mimics the underlying data distribution and well handles the variable length challenge.

Comparison on routing strategy: We compare MRN with E-Gate and its variants. E-Gate [2] treats different subnetworks as experts, and each time selects the most appropriate one for inference. In Tab. 7 we provide the model details. Raw E-Gate performs poorly in IMLTR. When stacking is used to build feature extractors, the accuracy improves significantly and outperforms DER. We also evaluate MRN with hard-voting. It reports a worse result. Compared to other routing strategies, our MRN shows clear superiority in terms of accuracy, while incurring only a negligible cost in parameters and computational complexity.

4.4. Qualitative Results Analysis

Fig. 5 gives several recognition results of MRN. It correctly read instances of different languages, even with the presence of common recognition difficulties. More importantly, MRN also recognizes character classes that are not present in the rehearsal set. These results again demonstrate that our MRN is effective in handling rehearsal-imbalance and can generalize well to unseen character classes.

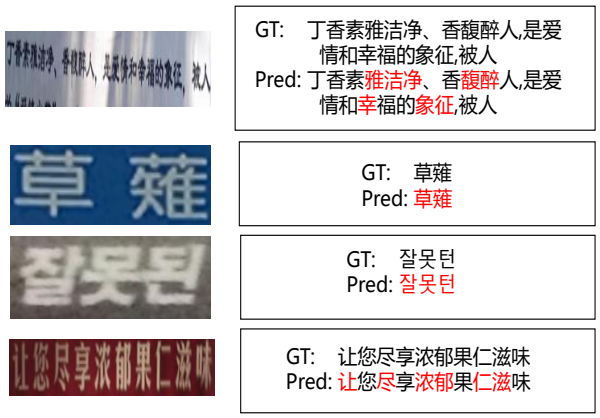

Figure 5. Illustrative recognition examples, where red denotes characters that are absent from the rehearsal set.

5. Conclusion

In this work, we introduce a new task called incremental multilingual text recognition (IMLTR). IMLTR handles text recognition in an incremental learning setting, therefore is suitable for applications like streaming data processing. IMLTR faces a distinct problem of rehearsal-imbalance, including data imbalance, class imbalance, and variable character length. To address this challenge, we designed a Multiplexed Routing Network (MRN) that first trains a multilanguage correlated DM-router to weight the language domains, and then votes the separately trained recognition branches for final text recognition. Experiments on public benchmarks show that MRN significantly outperforms existing general-purpose IL methods by large margins. As the first attempt to apply IL to multilingual text recognition, we hope that this work will broaden the applications of text recognition and inspire further research in this area.

Acknowledgments This project was supported by National Key R&D Program of China (No. 2022YFB3104703) and in part by the National Natural Science Foundation of China (No. 62172103)

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, pages 139–154, 2018. 2

[2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *CVPR*, July 2017. 2, 5, 8

[3] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *ICCV*, pages 4714–4722, 2019. 6

[4] Youngmin Baek, Seung Shin, Jeonghun Baek, Sungrae Park, Junyeop Lee, Daehyun Nam, and Hwalsuk Lee. Character region attention for text spotting. In *ECCV*, pages 504–521. Springer, 2020. 1, 3

[5] Jinfeng Bai, Zhineng Chen, Bailan Feng, and Bo Xu. Chinese image text recognition on grayscale pixels. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1380–1384. IEEE, 2014. 3

[6] Jinfeng Bai, Zhineng Chen, Bailan Feng, and Bo Xu. Image character recognition using deep convolutional neural network learned from different languages. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 2560–2564. IEEE, 2014. 3

[7] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *ICCV*, pages 583–592, 2019. 1, 2, 3

[8] Michal Bušta, Yash Patel, and Jiri Matas. E2e-mlt-an unconstrained end-to-end method for multi-language scene text. In *ACCV*, pages 127–143. Springer, 2018. 1, 3

[9] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, pages 233–248, 2018. 2

[10] Li Chen, Chunyan Yu, and Lvcai Chen. A new knowledge distillation for incremental object detection. In *IJCNN*, pages 1–7. IEEE, 2019. 3

[11] Shoufa Chen, Enze Xie, Chongjian GE, Runjian Chen, Ding Liang, and Ping Luo. CycleMLP: A MLP-like architecture for dense prediction. In *ICLR*, 2022. 7

[12] Mark Collier, Efi Kokiopoulou, Andrea Gesmundo, and Jesse Berent. Routing networks with co-training for continual learning. *ICML*, 2020. 2

[13] Jia Cui, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, Tom Sercu, Kartik Audhkhasi, Abhinav Sethy, Markus Nussbaum-Thom, and Andrew Rosenberg. Knowledge distillation across ensembles of multilingual models for low-resource languages. In *ICASSP*, pages 4825–4829, 2017. 3

[14] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2022. 1, 2, 3

[15] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, pages 5138–5146, 2019. 2

[16] Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *CVPR*, pages 4040–4050, 2021. 3

[17] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *CVPR*, pages 9285–9295, 2022. 2

[18] Yongkun Du, Zhineng Chen, Caiyan Jia, Xiaoting Yin, Tianlun Zheng, Chenxia Li, Yuning Du, and Yu-Gang Jiang. SVTR: scene text recognition with a single visual model. In *IJCAI*, 2022. 3, 6

[19] Shancheng Fang, Zhendong Mao, Hongtao Xie, Yuxin Wang, Chenggang Yan, and Yongdong Zhang. Abinet++: Autonomous, bidirectional and iterative language modeling for scene text spotting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1

[20] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *CVPR*, pages 7094–7103, 2021. 3

[21] Tao Feng, Mang Wang, and Hangjie Yuan. Overcoming catastrophic forgetting in incremental object detection via elastic response distillation. In *CVPR*, pages 9427–9436, 2022. 3

[22] Yasuhisa Fujii, Karel Driesen, Jonathan Baccash, Ash Hurst, and Ashok C Popat. Sequence-to-label script identification for multilingual ocr. In *ICDAR*, volume 1, pages 161–168. IEEE, 2017. 1, 3

[23] Siavash Golkar, Michael Kagan, and Kyunghyun Cho. Continual learning via neural pruning. *NIPS*, 2019. 2

[24] 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. 1, 3

[25] Qibin Hou, Zihang Jiang, Li Yuan, Ming-Ming Cheng, Shuicheng Yan, and Jiashi Feng. Vision permutator: A permutable mlp-like architecture for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 7

[26] Wenyang Hu, Xiaocong Cai, Jun Hou, Shuai Yi, and Zhiping Lin. Gtc: Guided training of ctc towards efficient and accurate scene text recognition. In *AAAI*, volume 34, pages 11005–11012, 2020. 3

[27] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3957–3966, 2021. 1

[28] Bingchen Huang, Zhineng Chen, Peng Zhou, Jiayin Chen, and Zuxuan Wu. Resolving task confusion in dynamic expansion architectures for class incremental learning. In *AAAI*, volume 37, pages 908–916, 2023. 1, 2

[29] Jing Huang, Guan Pang, Rama Kovvuri, Mandy Toh, Kevin J Liang, Praveen Krishnan, Xi Yin, and Tal Hassner. A multiplexed network for end-to-end, multilingual ocr. In *CVPR*, pages 4547–4557, 2021. 1, 3, 5

[30] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 2, 6

[31] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 6, 7

[32] An-An Liu, Haochun Lu, Heyu Zhou, Tianbao Li, and Mohan Kankanhalli. Balanced class-incremental 3d object classification and retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 2023. 3

[33] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *NIPS*, 34:9204–9215, 2021. 7

[34] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Cheng-lin Liu, et al. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In *ICDAR*, pages 1582–1587. IEEE, 2019. 1, 3, 5

[35] 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 *ICDAR*, volume 1, pages 1454–1459. IEEE, 2017. 3, 5

[36] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. 2

[37] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017. 1, 2

[38] Fenfen Sheng, Zhineng Chen, and Bo Xu. Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition. In *ICDAR*, pages 781–786, 2019. 1, 3

[39] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304, 2017. 1, 3, 6

[40] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2035–2048, 2018. 1, 3

[41] Baoguang Shi, Cong Yao, Chengquan Zhang, Xiaowei Guo, Feiyue Huang, and Xiang Bai. Automatic script identification in the wild. In *ICDAR*, pages 531–535. IEEE, 2015. 3

[42] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019. 6

[43] Yuxin Wang, Hongtao Xie, Shancheng Fang, Jing Wang, Shenggao Zhu, and Yongdong Zhang. From two to one: A new scene text recognizer with visual language modeling network. In *ICCV*, pages 14174–14183, 2021. 1, 3

[44] Yuxin Wang, Hongtao Xie, Shancheng Fang, Mengting Xing, Jing Wang, Shenggao Zhu, and Yongdong Zhang. Petr: Rethinking the capability of transformer-based language model in scene text recognition. *IEEE Transactions on Image Processing*, 31:5585–5598, 2022. 3

[45] Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *ICLR*, 2020. 2

[46] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *CVPR*, pages 3014–3023, 2021. 1, 2, 3, 6, 7, 8

[47] Dongbao Yang, Yu Zhou, Xiaopeng Hong, Aoting Zhang, and Weiping Wang. One-shot replay: Boosting incremental object detection via retrospecting one object. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 3127–3135, 2023. 3

[48] Dongbao Yang, Yu Zhou, Aoting Zhang, Xurui Sun, Dayan Wu, Weiping Wang, and Qixiang Ye. Multi-view correlation distillation for incremental object detection. *Pattern Recognition*, 131:108863, 2022. 3

[49] Guanglei Yang, Enrico Fini, Dan Xu, Paolo Rota, Mingli Ding, Moin Nabi, Xavier Alameda-Pineda, and Elisa Ricci. Uncertainty-aware contrastive distillation for incremental semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 3

[50] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, pages 3987–3995. PMLR, 2017. 2

[51] Chang-Bin Zhang, Jia-Wen Xiao, Xialei Liu, Ying-Cong Chen, and Ming-Ming Cheng. Representation compensation networks for continual semantic segmentation. In *CVPR*, pages 7053–7064, 2022. 1, 3

[52] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *CVPR*, pages 13208–13217, 2020. 1, 6

[53] Tianlun Zheng, Zhineng Chen, Jinfeng Bai, Hongtao Xie, and Yu-Gang Jiang. Tps++: Attention-enhanced thin-plate spline for scene text recognition. *IJCAI*, 2023. 1

[54] Tianlun Zheng, Zhineng Chen, Shancheng Fang, Hongtao Xie, and Yu-Gang Jiang. Cdistnet: Perceiving multi-domain character distance for robust text recognition. *arXiv preprint arXiv:2111.11011*, 2021. 3