doi:10.20944/preprints202208.0287.v1

Article

Continual Learning for Table Detection in Document Images

Mohammad Minouei [1,2,*], Khurram Azeem Hashmi [1,2], Mohammad Reza Soheili [3], Muhammad Zeshan Afzal [1,2], and Didier Stricker [1,2]

[1] Department of Computer Science, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany; khurram_azeem.hashmi@dfki.de (K.A.H.); muhammad_zeshan.afzal@dfki.de (M.Z.A.); didier.stricker@dfki.de (D.S.)
[2] German Research Institute for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany
[3] Department of Electrical and Computer Engineering, Kharazmi University, Tehran, Iran; soheili@khu.ac.ir (M.R.S)
[*] Correspondence: mohammad.minouei@dfki.de (M.M)

Abstract: The growing amount of data demands methods that can gradually learn from new samples. However, it is not trivial to continually train a network. Retraining a network with new data usually results in a known phenomenon, called "catastrophic forgetting." In a nutshell, the performance of the model drops on the previous data by learning from the new instances. This paper explores this issue in the table detection problem. While there are multiple datasets and sophisticated methods for table detection, the utilization of continual learning techniques in this domain was not studied. We employed an effective technique called experience replay and performed extensive experiments on several datasets to investigate the effects of catastrophic forgetting. Results show that our proposed approach mitigates the performance drop by 15 percent. To the best of our knowledge, this is the first time that continual learning techniques are adopted for table detection, and we hope this stands as a baseline for future research.

Keywords: table detection; document layout analysis; continual learning; incremental learning; experience replay

1. Introduction

Tables in documents present crucial contents, concisely and compactly. Since tables come in different layouts, even modern optical character recognition-based methods [1–3] cannot yield satisfactory performance when extracting information from tables. Therefore, before retrieving tabular information, identifying their position in a document is essential, mainly referred to as *table detection*.

Table detection has been a challenging and open problem for the last couple of decades [4,5]. The fundamental obstacle is the object confusion between tables and other graphical elements in a document that share a similar appearance [6]. Two factors mainly cause this confusion. First, the inherent *high intra-class variance* due to several distinctive designs and scales in tables. Second, *low inter-class variance* due to similar appearance to other graphical page objects (figures, algorithms, and formulas).

Earlier approaches have addressed such issues by exploiting the external meta-data in documents [7]. Unfortunately, the performance of these methods is usually fouled on scanned document images. Recently, deep learning-based techniques have given a fresh blood to the field [8]. We observe a direct association between the advances of table detection and object detection in computer vision [9–12]. The idea of formulating a table as an object and the document image as a natural scene, has produced state-of-the-art results on several publicly available table detection datasets [13,14].

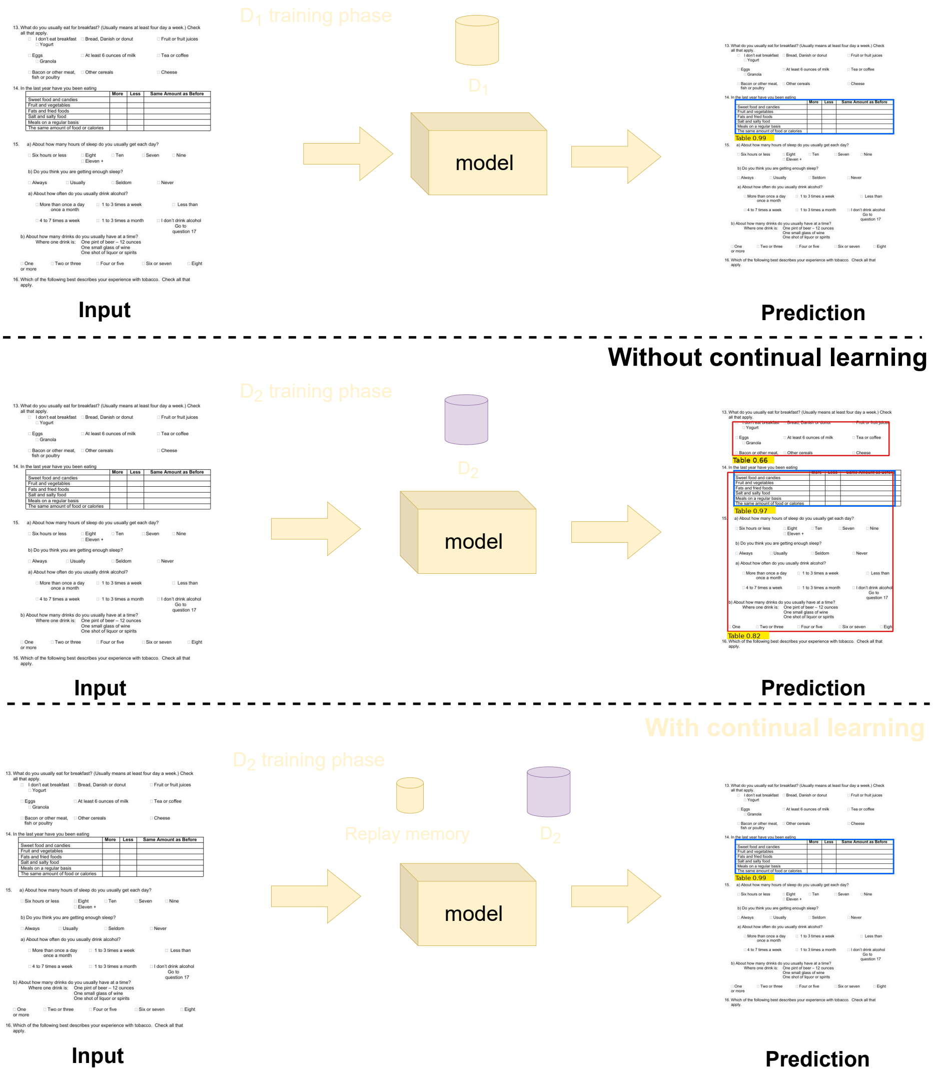


Figure 1. Illustrative sketch of the continual learning usage. After the first training phase (top row), conventional methods take the same approach for the next datasets (middle row). CL methods can involve previous knowledge in the latest trains by replaying them for the model (bottom row). Blue represents true positive, and red denotes false positive.

Although these modern deep learning-based methods are impressive in many ways, they have some drawbacks. For one, they are data-hungry and only work best when they are exposed to all the possible samples. However, in real-life situations, data scarcity is the norm. Another issue is that artificial neural networks, unlike their natural counterparts (human brain), are not necessarily able to *add* to their knowledge with the new data that is fed to them. The problem is *catastrophic forgetting* [15]. This phenomenon is associated with the degradation in the model's performance on prior cases when it is adopted to a new set of data. The primary reason for catastrophic forgetting is that learning necessitates changing the neural network's weights. However, these changes result in forgetting the prior knowledge. Many researchers studied ways to learn from new data over time [16]. In the literature, different names are used to describe these techniques (e.g. *life-long learning, incremental learning, on-line learning*), but here we refer to them as continual learning (CL). The idea of CL is to keep learning from incoming data while preserving the knowledge gained from previous trains [17]. Figure 1 illustrates the fundamental difference between conventional approaches and our proposed method that leverages continual learning. As it shows, retraining a model on new data results in lower performance. However, a continual learning method preserve the prior knowledge while learning the new.

With the advent of deep learning, we see that enormous datasets are annually introduced for table detection [18,19]. In order to demonstrate state-of-the-art results, all prior methods require extensive training on the new datasets, hence overriding the previous knowledge. To address these limitations, we propose a novel end-to-end trainable table detection method that leverages a continual learning mechanism. The method is robust to the inclusion of novel datasets and exhibits impressive performance on the previously trained data. To investigate the effectiveness of continual learning, we treat the identification of tables as an object detection problem and employ Faster R-CNN [9] and PVT [20] to detect tables in document images.

The primary contributions of this work are threefold:

- To the best of our knowledge, this work is the first attempt to incorporate a continual learning-based method for table detection.
- Extensive experiments are conducted on considerably large datasets with more than 900K images combined.
- The presented method is able to reduce the forgetting effect by a 15 percent margin.

The rest of the paper is organized as follows: in Section 2, the previous attempts for table localization are reviewed, followed by an analysis of CL exploitation. Our proposed methodology is elaborated in Section 3. The results of the experiments that demonstrate the effect of CL are presented in Section 4 and Section 5 concludes the paper.

2. Related Work

Table detection has been an open problem for a few decades. This section discusses earlier rule-based methods, followed by the recent deep learning approaches. Finally, we highlight current methods that have incorporated continual learning in various fields.

2.1. Rule-Based Table Detection

The incorporation of rule-based methods for table detection originates from the work of Itonori et al. [21]. They proposed heuristics that capitalized the position of text blocks and ruling lines to detect tabular boundaries in documents. In a similar direction, [22] exploits ruling lines in tables to tell them from regular text. Pyreddy and Croft [4] presented another method based on custom heuristics that detects structural elements and filters the tables in a document. Similarly, specific tabular layouts and grammars have been designed to identify tabular structures [23,24]. For the comprehensive summarization of rule-based methods, we refer our readers to [7,25–28].

2.2. Table Detection with Deep Neural Networks

Research in table analysis has experienced a remarkable improvement with the introduction of neural networks [6,10,12–14,29–37]. Hashmi et al. have thoroughly surveyed these studies in [8], where they also covered the heuristic detectors.

Hao et al. [38] applied Convolutional Neural Networks (CNNs) to retrieve spatial features from a document and merged them with PDF meta-data to detect tables. Gilani et al. [10] formulated table detection as object detection and used a generic detector, Faster R-CNN [9], to detect tables in document images. In a similar line of work, [29] incorporates deformable networks [39] to tackle varied tabular layouts. Saha et al. [40] exploited Mask R-CNN [41] to improve overall performance table, figure, and formula detection in document images. Cascade Mask R-CNN [42] has also been employed to enhance the performance of table detection with a conventional backbone network [13] and recursive feature pyramid networks [14]. The recently proposed Hybrid Task Cascade network [11] was merged with deformable convolutions to produce impressive table detection results [12].

Besides object detection-based methods, earlier methods leveraged the concepts of Fully Convolutional Network (FCN) [43,44] and Graph Neural Network (GNN) [37,45] to resolve table detection in document images. It is essential to emphasize that all prior works in the field of table detection heavily rely on dataset specific training to yield satisfactory results.

2.3. Applications of Continual Learning

Catastrophic forgetting has been tackled by researchers with various techniques in different domains [15,46]. In some works, a regularization technique is applied to different parts of the model like the loss function, learning rate, and the optimizer; and in others, a form of dynamic architecture or parameters isolation is practiced for learning different tasks continually. Some means of rehearsal process is also exercised [47]. For image classification, Kirkpatrick et al. [48] introduced the elastic weight consolidation (EWC) to alleviate forgetting. In their approach, any modification to the important weights of the network is penalized. In [49], authors present an EWC-based method for incremental object detection. According to their findings, when the annotations of the old classes are missing, EWC misclassifies previous categories as background. Therefore, they proposed the pseudobounding process for annotating old classes on the new set of images.

Rebuffi et al. [50] proposed iCaRL for image classification. In iCaRL, a set of examplars for each class is selected dynamically and used for replay with a knowledge distillation technique [51]. In [52], authors proposed deep model consolidation (DMC) for incremental learning that can be applied in both image classification and object detection. In their approach, a double distillation loss is used to combine the two models that one is trained on the old classes and the other is trained on the new ones.

In [53], authors developed a variant of Fast R-CNN [54] with a class agnostic region proposal [55] for object detection in a class incremental setting. A distillation loss was also used to reduce forgetting when training on the new objects with a frozen copy of the learned model on the previous set of classes. RODEO [56] applied the experience replay procedure using a buffer memory comprising of compressed representations of the past samples. Another recent work that applied experience replay is presented by Shieh et al. [57]; In their method images from the former task are concatenated with the new samples for class incremental object detection.

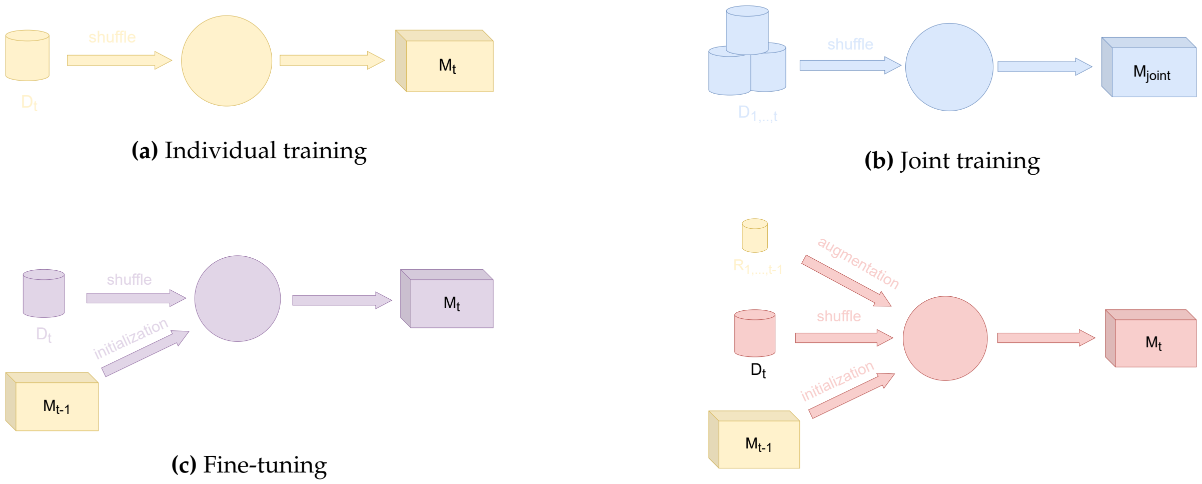
Considering the performance and adaptability of the experience replay approach, we employed it for detecting tables in a continual manner. It is expected that by satisfying the shortage of lifelong learning for table-detector networks, the employment of emerging datasets becomes easier for this step of OCR pipeline. The following sections will explain and examine the proposed solution.

3. Experimental Setup

The purpose of this study is to continuously train a network with the new data while preserving prior knowledge. In recent years, multiple datasets have been published for table detection with existing demands for more labeled data; thereby, we define the continual learning for table detection as follows. Suppose $D_{1,2,\ldots,t-1}$ is an array of multiple datasets, and M_{t-1} is a model that has been trained on them. At the event of introducing a new dataset at time, t, different scenarios are possible. Figure 2 displays four of the possible ways of consuming the new dataset. In the following, we will describe them along our proposed experiments.

3.1. Independent training

This is the conventional way of training in which a model is trained on each dataset. The Algorithm 1 shows the straightforward batch training procedure which is used here. The results of this experiment will show the upper bound of possible learning with current data and architecture. Figure 2a presents this training process.



(a) Individual training

(b) Joint training

(c) Fine-tuning

(d) Experience replay

Figure 2. The training setup. a) In the individual training approach, the model is trained on a new dataset. b) In the joint training, the model is trained on all available datasets. c) In the fine-tuning approach, the model is trained on a new dataset with the initial parameters obtained from training on the previous datasets. d) In the experience replay approach, at first, the model is initialized with the parameters attained from the previous learning stages on the former datasets; then, the model is trained on a new dataset and a replay memory (that is randomly selected from the former datasets).

3.2. Joint training

In the joint training, all the available datasets are exploited. This setup acts as an upper bound of the learning capability of the model using all the available data. As presented in Figure 2b, all the available samples are shuffled before the batch training.

3.3. Fine-tuning

The classical fine-tuning procedure is implemented for this experiment. As Figure 2c shows, during the training of D_t, a pre-trained model on previous datasets, M_{t-1}, is employed for initializing the parameters of the model. Afterwards, the model is retrained with a lower learning rate on the new instances. Since this setup will result in catastrophic forgetting, its performance is the lower bound for the learners.

3.4. Training with Experience Replay

The last experiment is the continual learning technique that we devised for our task, called *experience replay* (Figure 2d). In this approach, $R_{1,2,...,t-1}$ is a small memory that is dedicated to images of the prior datasets. These images are then presented to the model while the model is trained with the new data. To be precise, each batch contains samples from both D_t and $R_{1,2,...,t-1}$.

Algorithm 2 describes our batch training with experience replay. Assume that the training procedure is supposed to proceed for D_t and we have the prior data and the trained model at our disposal. The algorithm begins by initializing the replay memory, $R_{1,2,...,t-1}$. It is a random selection of images from $D_1, D_2, \ldots, D_{t-1}$. In each iteration of training, a mini-batch is chosen from the D_t and another from $R_{1,2,...,t-1}$. These batches are then concatenated in one batch, and one step of gradient descent is taken by them.

The number of images in $R_{1,2,...,t-1}$ will be equal to one percent of the number of training samples in D_t. In this manner, we can ensure that the memory is neither too small nor too large for preserving the past knowledge while learning the new ones. Its images are selected randomly from D_is with respect to their size. If s_{D_i} designates the number of training samples in dataset D_i, then the number of images from dataset D_i that reside in $R_{1,2,...,t-1}$ are achieved from (1) and represented by C_{D_i}.

$$C_{D_i} = \lceil \frac{s_{D_i}}{\sum_{j=1}^{t-1} s_{D_j}} \times \frac{1}{100} \times s_{D_t} \rceil \quad (1)$$

Algorithm 1 Batch training

```
Require: Learning rate: $\nu_k$
Require: Initial weights from ImageNet
  inputs: dataset $D_t$, batch size $bs$
  function BATCHTRAINING($D_t, bs$)
    for each iteration do
      $B \overset{bs}{\leftarrow} D_t$                                   ▷ sample a mini-batch from $D_t$
      $\mathbf{g} \leftarrow \frac{1}{n} \sum_i^n \nabla_{\mathbf{w}} L(\mathbf{B})$     ▷ compute the gradient for the current batch, $B$
      $\mathbf{w} \leftarrow \mathbf{w} - \nu_k \mathbf{g}$                 ▷ update the current weights
    end for
  end function
```

Algorithm 2 Batch training with Experience replay

```
Require: Formerly trained model
Require: Learning rate: $\nu_k$
  inputs: array of prior datasets $D_{1,2,\ldots,t-1}$, new dataset $D_t$, batch size $bs$, and memory sample size $ms$
  function BATCHTRAININGWITHREPLAY($D_{1,2,\ldots,t-1}, D_t, bs, ms$)
    $R \leftarrow randomChoice(D_{1,2,\ldots,t-1})$                 ▷ allocate samples from previous datasets
    for each iteration do
      $B_n \overset{bs}{\leftarrow} D_t$                                  ▷ sample a mini-batch from $D_t$
      $B_r \overset{ms}{\leftarrow} R_{1,2,\ldots,t-1}$                    ▷ sample a mini-batch from the memory
      $\mathbf{g} \leftarrow \frac{1}{bs+ms} \sum_i^{bs+ms} \nabla_{\mathbf{w}} L(\mathbf{B_n} \bigcup \mathbf{B_r})$   ▷ stack two minibatches and compute the gradient
      $\mathbf{w} \leftarrow \mathbf{w} - \nu_k \mathbf{g}$                 ▷ update the current weights
    end for
  end function
```

3.5. Networks

In order to validate the proposed approach, two state-of-the-art architectures are selected: Faster R-CNN [9] and Pyramid Vision Transformer (PVT) [20,58] combined with Sparse R-CNN [59]. Faster R-CNN is considered as a classic baseline in many previous works, thereby it was our first choice. Next, the PVT architecture was chosen which is one of the recent SOTA detectors.

3.5.1. Faster R-CNN

Faster R-CNN is a two-stage detector proposed in 2015 [60]. Contrary to Fast R-CNN [61], which uses a selective search algorithm to find the region of objects, the Faster R-CNN utilizes a module called region proposal network (RPN) to approximate the possible locations of each object in the image. Using RPN, Faster R-CNN can effectively cut the prediction time and improve the accuracy. Moreover, the RPN allows the Faster R-CNN to be end-to-end trainable. After detecting the possible region of interests (ROIs), the feature map of each ROI is fed to a fully connected network consisting of two branches at the final layer; One branch is a softmax layer to predict the class of objects and the other is a box-regression layer for computing the coordinates. The architecture of Faster R-CNN is seen in Figure 3.

3.5.2. PVT+Sparse R-CNN

Pyramid Vision Transformer made the first convolution-free object detector possible [20]. As proposed in [58], the combination of PVT with Sparse R-CNN creates a strong end-to-end method for object detection. In PVT, a progressive shrinking pyramid extracts multi-scale features. Similar to traditional CNNs, as the network grows in depth, the output resolution progressively shrinks. Moreover, an efficient attention layer was designed to further reduce the computation cost.

Given the Figure 4, the structure of PVT consists of one main stage repeated four times in order to simulate the pyramid approach, which reduces the size of the feature maps. Inspired by the transformer idea in language translators, the input image must be tokenized. Therefore, the mentioned stage in PVT converts the input to some patches as a dictionary of tokens. So, at the first stage, the input image of size $H \times W \times 3$ is split into $\frac{H \times W}{4^2}$ patches; then the patches are flattened, and a linear projection process is applied to the patches in order to attain embedded equivalents. Afterwards, the embedded patches and their positions are input to another block called transformer encoder which includes a spatial-reduction attention (SRA) layer. Ultimately, by reshaping the output of the transformer encoder block, the feature map F_1 is obtained. By taking a closer look at Figure 4, we can observe that by applying the mentioned processes to the output of the previous stage, the new feature maps F_2, F_3, and F_4 are produced. After that, these feature maps are fed into a sparse R-CNN [59] for object detection. Unlike the conventional RPN which requires thousands of anchor boxes, sparse R-CNN relies on a small set of learnable proposal boxes. These predictions are then refined in multiple stages.

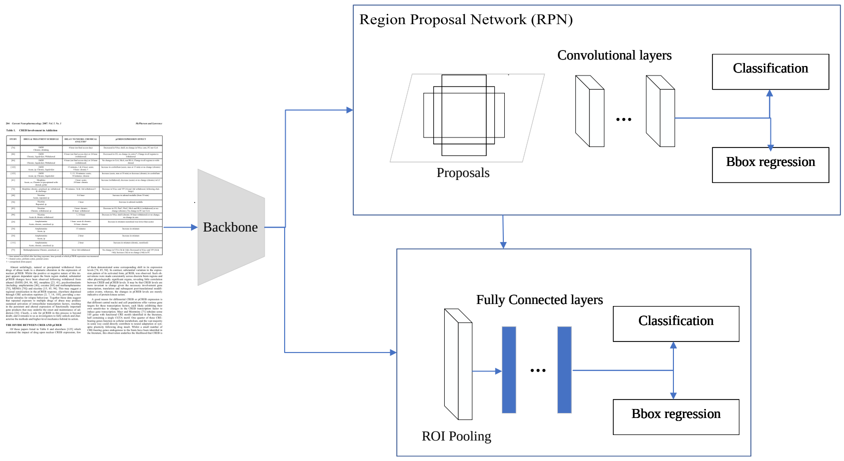


Figure 3. The structure of Faster R-CNN. The workflow consists of feeding the input image to CNN network. Afterwards, potential object regions are found with the RPN. The final step is ROI pooling that extract features for each region and feed them to classifier and the bbox regressor.

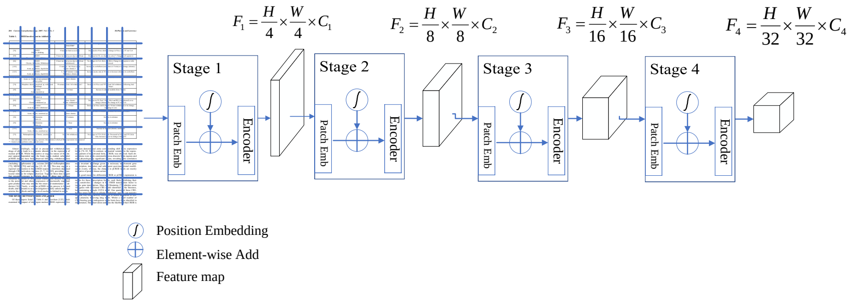

$F_1 = \frac{H}{4} \times \frac{W}{4} \times C_1$
$F_2 = \frac{H}{8} \times \frac{W}{8} \times C_2$
$F_3 = \frac{H}{16} \times \frac{W}{16} \times C_3$
$F_4 = \frac{H}{32} \times \frac{W}{32} \times C_4$
Stage 1
Stage 2
Stage 3
Stage 4
Patch Emb
Encoder
Position Embedding
Element-wise Add
Feature map

Figure 4. The schema of pyramid vision transformer (PVT). Each stage's output is passed to the next layer while the first two dimensions are halved (rows and columns). The finished map will be 16 times smaller than the input yet with a greater depth.

3.6. Implementation Details

In every evaluation, both the Faster-RCNN and the PVT were trained using the publicly available **mmdetection** toolbox [62]. The training procedure and environment were the same for all experiments. The networks were trained on 8 GPUs with four images per GPU. In the ER setting, the batch size is also four per GPU, among which one is from the replay memory.

Table 1. The number of utilized images in four employed datasets.

Set	TableBank	PubLayNet	PubTables1m	FinTabNet	Joint
Train	261K	86K	461K	48K	856K
Test	8K	4K	57K	6K	71K

The ResNet-50 [63] is used as the backbone network for Faster-RCNN and for the Sparse R-CNN the PVTv2-B2 [58] is chosen. The backbones of the models were pre-trained on the imageNet [64]. If not mentioned otherwise, all the default configurations are used from the reference implementations.

In the IT scenario, the model is trained for three epochs. The initial learning rate is 10^{-4} which decays with a factor of 10 for every epoch. In FT and ER, the added datasets were fine-tuned for one epoch with a learning rate of 10^{-5}.

To prevent overfitting in ER scenario, data augmentation is applied on images of replay memory. Four types of augmentation are used from the *Image corruptions* library [65], namely motion blur, jpeg compression, Gaussian noise, and brightness. The augmentation methods were chosen so that they simulate the common real-life scenarios.

3.7. Datasets

In total, we utilized four modern publicly available datasets: TableBank, PubLayNet, PubTablets1M, and FinTabNet. Table 1 summarizes datasets' statistical information.

- **TableBank** [18]
  TableBank has been collected from the *arXiv* database [66], containing more than 417K labeled document images. This dataset comes with two splits, Word and Latex. We combined both for training.
- **PubLayNet** [67]
  PubLayNet is another large-scale dataset that covers the task of layout analysis in documents. Contrary to manual labeling, this dataset has been collected by automatically annotating the document layout of PDF documents from the PubMed Central$^{\text{TM}}$ database. PubLayNet comprises 360K document samples containing text, title, list, figure, and table. All document samples from the PubLayNet dataset that contain tabular information were excluded for our experiments.
- **PubTables1M** [68]
  This dataset is currently the largest and most complete dataset that addresses all three fundamental tasks of table analysis. For our experiments, we include the annotations of table detection from this dataset that consists of more than 500K annotated document pages. Furthermore, we unify the annotations for various tabular boundaries in this dataset with a single class of tables to conduct joint training.
- **FinTabNet** [69]
  We employ FinTabNet to increase samples' diversity. FinTabNet is derived from the PubTabNet [19] and contains complex tables from financial reports. This dataset comprises 70K document samples with annotations of tabular boundaries and tabular structures.

4. Evaluations

In this section the numeric results of the experiments will be reported. As stated in Section 3, the four experiments are: independent training (IT), joint training (JT), fine-tuning (FT), and experience replay (ER). These experiments were performed with two famous models named Faster R-CNN and PVT for table detection.

In the IT, one model was trained for each individual train-set and tested against the corresponding test-set. In JT, the train-sets of all four datasets are shuffled and the models are trained on the compiled set of samples. The resultant network is tested separately on the four test-sets of the four datasets. In FT and ER, one network was trained with the four datasets in sequence. Unlike IT, the model won't be tested until it finishes training with all four train-sets. The obtained network will then be tested on the four test-sets with the same order. It is expected that FT approach forgets the first dataset more severely. It should be mentioned that the effect of the order will be studied in a further subsection. As mentioned, the proposed method, ER, takes the same path as FT, with the difference that it consumes a subset of the previous datasets while being fed with the new samples. The four reported results are obtained in a similar manner to FT's. It is expected that ER suffers less from catastrophic forgetting and, ideally, reaches the performance of JT.

4.1. Evaluation Metrics

Being a sub-class of object detection, our task is evaluated with same performance criteria. The followings are the definition of the common evaluation metrics.

- **Precision**
  The number of true positives divided by the total number of positive predictions is known as precision. This is a metric for determining the accuracy of a model. The precision is calculated through Equation (2).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

- **Recall**
  To indicate the rate of missed positive predictions, a metric named Recall (or Sensitivity) is commonly used. Recall could be obtained through dividing correct positive predictions (TP) by all positive predictions (TP + FP). Equation (3) shows the mathematical definition.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

- **Precision-Recall curve**
  For all possible thresholds, the precision-recall (PR) curve plots precision versus recall. A good object detector has a high recall rate as well as a high precision rate.
- **Intersection Over Union (IOU)**
  As written in Equation (4), the overlap of a predicted bounding box versus the correct one for an object is measured by the intersection over union of the two boxes.

$$\text{IoU(A,B)} = \frac{\text{The Overlapping area}}{\text{The Union area}} = \frac{|A \cap B|}{|A \cup B|} \tag{4}$$

- **Mean Average Precision (mAP)**
  The mean average precision (mAP) is a widely used parameter for evaluating object detection models. It is the area value under the precision-recall curve for each class, and the mAP is computed by averaging all average precision for all classes. Equation (5) formulates the metric.

$$\text{mAP} = \frac{1}{N}\sum_{r=1}^{N} AP_r \tag{5}$$

Where AP_r is the average precision r.

4.2. Results

Table 2 summarizes the results of these experiments. By taking a close look at the values of FT and ER, it can be observed that the proposed approach effectively prevents the models from forgetting previous datasets. To emphasize the contrast, parenthesized

Table 2. The mAP results of different experiments on multiple test-sets. IT is the Independent training, JT is the Joint training, FT is the Fine-tuning, and ER is the Experience replay. the values written in the parentheses in the ER experiment demonstrate the difference in the mAP metrics between the ER and FT approaches. Acronyms TB, PN, PT, and FN denote TableBank, PubLayNet, PubTables, and FinTabNet respectively.

Experiment	Train-set / Test-set	Faster R-CNN+ResNet	Sparse R-CNN+PVT
IT	TB / TB	95.7	96.2
JT	$\{TB \cup PN \cup PT \cup FN\}$ / TB	94.1	94.7
FT	$TB \rightarrow PN \rightarrow PT \rightarrow FN$ / TB	74.2	76.4
ER	$TB \rightarrow PN^{R_1} \rightarrow PT^{R_{1,2}} \rightarrow FN^{R_{1,2,3}}$ / TB	89.6(+15.4)	90.7(+14.3)
IT	PN / PN	97.6	97.4
JT	$\{TB \cup PN \cup PT \cup FN\}$ / TB / PN	97.4	97.5
FT	$TB \rightarrow PN \rightarrow PT \rightarrow FN$ / PN	90.5	90.6
ER	$TB \rightarrow PN^{R_1} \rightarrow PT^{R_{1,2}} \rightarrow FN^{R_{1,2,3}}$ / PN	93.7(+3.2)	92.5(+1.9)
IT	PT / PT	98.9	99
JT	$\{TB \cup PN \cup PT \cup FN\}$ / PT	98.4	98.7
FT	$TB \rightarrow PN \rightarrow PT \rightarrow FN$ / PT	97.2	98
ER	$TB \rightarrow PN^{R_1} \rightarrow PT^{R_{1,2}} \rightarrow FN^{R_{1,2,3}}$ / PT	97.4(+0.2)	98.2(+0.2)
IT	FN / FN	90	91.3
JT	$\{TB \cup PN \cup PT \cup FN\}$ / FN	88.4	92.7
FT	$TB \rightarrow PN \rightarrow PT \rightarrow FN$ / FN	90	93.3
ER	$TB \rightarrow PN^{R_1} \rightarrow PT^{R_{1,2}} \rightarrow FN^{R_{1,2,3}}$ / FN	90	93.1

values in the ER rows show the mAP-gain by ER in comparison to FT. In particular, the mAP for the proposed method on the TableBank is about 15 percent higher than FT. We see that the PVT demonstrates a better performance than the Faster R-CNN in almost all experiments.

The precision-recall curves for ER and FT with the PVT architecture are presented in Figure 5. It is evident that as IOU thresholds rise, FT curves plummet. This is further illustrated with IOU values of 0.9 and 0.95 (in green and red, respectively). Moreover, the fact that the older datasets are more prone to forgetting is again apparent in the figures. This is evident from the TableBank's curves in Figures 5b-5a and in Figures 5h-5g that correspond to the most recent dataset.

Figure 6 presents common pitfalls ahead of FT. In some cases the model inaccurately detects the bounding boxes and in others we see frequent samples of false-positives. In contrast, the ER approach has led to a better performance and prevented the model from forgetting.

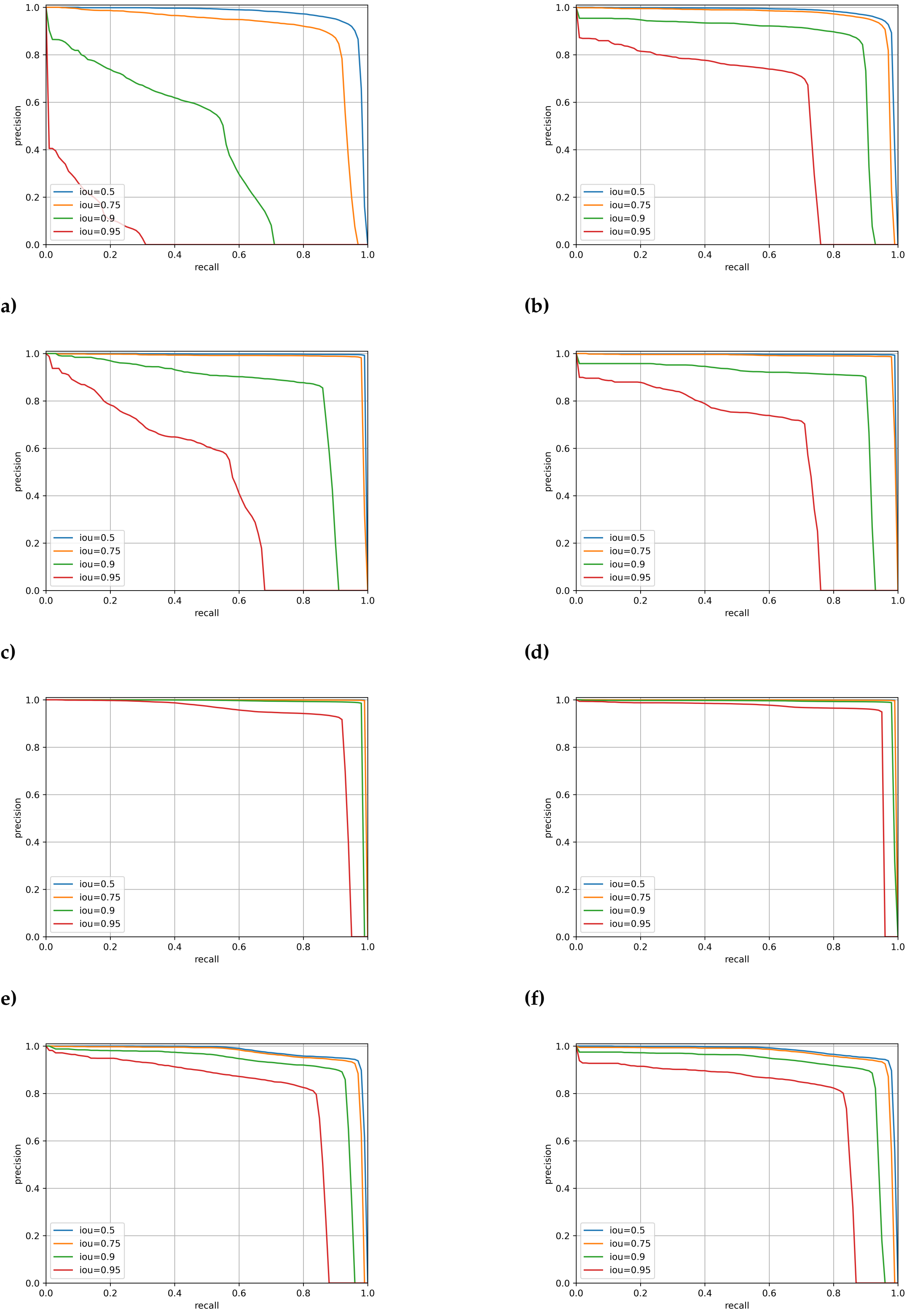
4.3. The Effect of Datasets Order

It is clear that the inherent differences between the datasets is the cause of catastrophic forgetting. Nonetheless, the results showed that the performance drop of the network, is harsher on the older samples. To investigate this, we repeated the experiments in Section 3 with a different sequence of datasets during the training phase. In the initial experiments, the order of the datasets was: TableBank, PubLayNet, PubTables, and FinTabNet. However, for the second trial, the sequence is changed to PubTables, PubLayNet, TableBank, and FinTabNet. The rest of the settings is equal to the previous experiments.

The results of this trial are presented in Table 3. These results support the previous ones, and the effect of forgetting is once again apparent. By comparing the results of the models on PubTables1M in the first and second trials (Tables 2 and 3), we can infer that the performance of models on the preceding datasets drops more drastically.

4.4. Comparison with State-of-the-Arts

The current SOTA methods heavily rely on particular datasets for training and evaluation. However, in this study we conducted the experiments on multiple datasets in sequence. To this end, some of the datasets were altered and the training procedure were



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 5. Precision-recall (PR) curves for FT (left column) and ER (right column) with PVT on four datasets. Each row corresponds to a dataset (from top to bottom): TableBank, PubLayNet, PubTables, FinTabNet. Different IOU threshold are demonstrated with blue, orange, green, and red which correspond to 50%, 75%, 90%, and 95%, respectively.

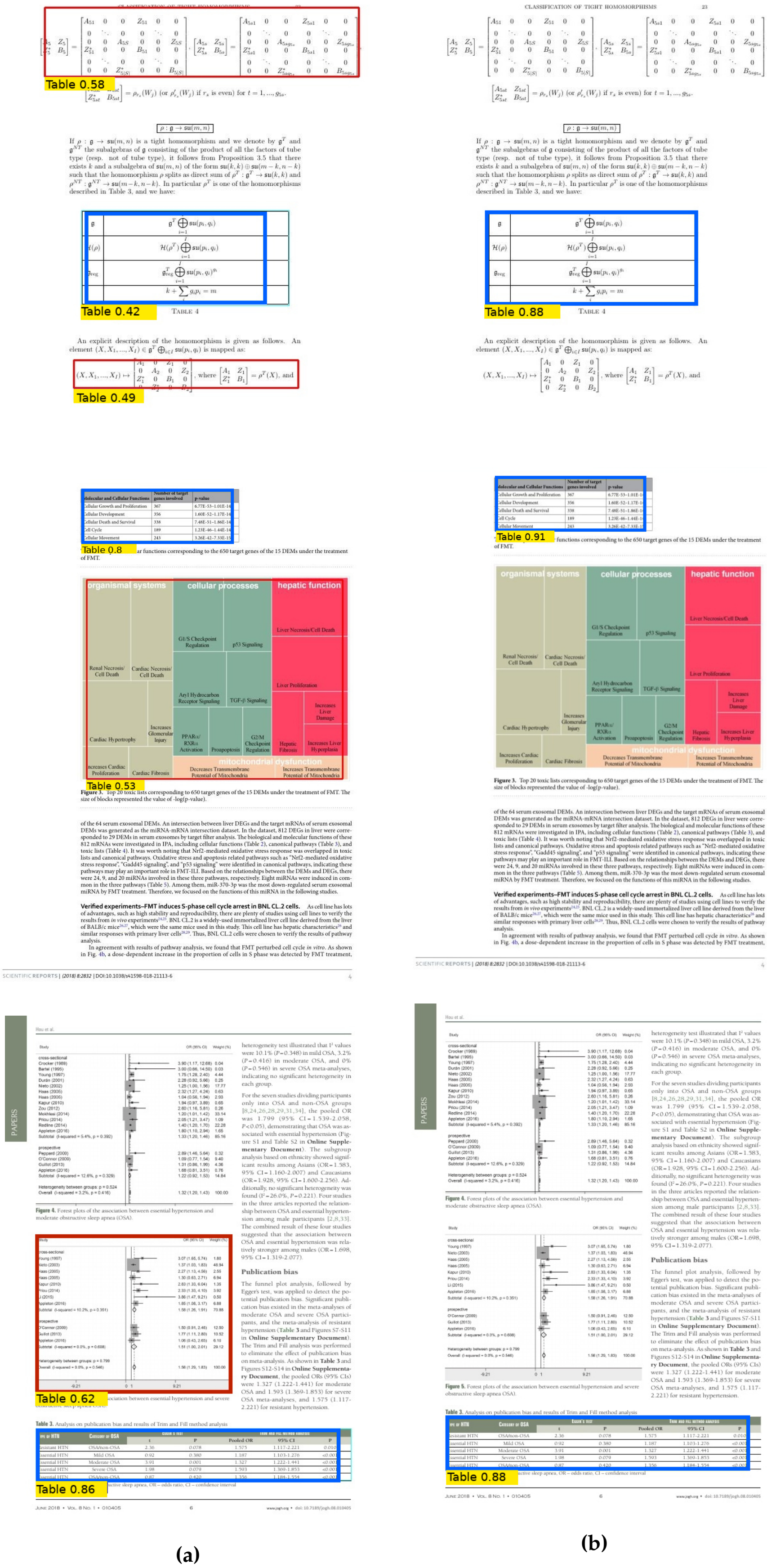


Figure 6. The qualitative results using two methods: (a) Fine-tuning, (b) Experience replay. Blue represents true positive, and red denotes false positive. The samples are from TableBank, PubLayNet, and PubTables datasets, respectively. The Experience replay method maintains the performance but the Fine-tuning approach suffers from false detection and inaccurate bounding boxes.

Table 3. The mAP results of different experiments on multiple test-sets. FT is the Fine-tuning, and ER is the Experience replay. Acronyms TB, PN, PT, and FN denote TableBank, PubLayNet, PubTables, and FinTabNet respectively.

Experiment	Train-set / Test-set	Faster R-CNN+ResNet	Sparse R-CNN+PVT
FT	$PT \rightarrow PN \rightarrow TB \rightarrow FN$ / PT	96.1	97.4
ER	$PT \rightarrow PN^{R_1} \rightarrow TB^{R_{1,2}} \rightarrow FN^{R_{1,2,3}}$ / PT	97.7(+1.6)	98.5(+1.1)
FT	$PT \rightarrow PN \rightarrow TB \rightarrow FN$ / PN	91.2	93.4
ER	$PT \rightarrow PN^{R_1} \rightarrow TB^{R_{1,2}} \rightarrow FN^{R_{1,2,3}}$ / PN	94.2(+3)	94.4(+1)
FT	$PT \rightarrow PN \rightarrow TB \rightarrow FN$ / TB	76.5	79.8
ER	$PT \rightarrow PN^{R_1} \rightarrow TB^{R_{1,2}} \rightarrow FN^{R_{1,2,3}}$ / TB	87.9(+11.4)	90.7(+10.9)
FT	$PT \rightarrow PN \rightarrow TB \rightarrow FN$ / FN	89.5	92.9
ER	$PT \rightarrow PN^{R_1} \rightarrow TB^{R_{1,2}} \rightarrow FN^{R_{1,2,3}}$ / FN	89.1	93

different than the customary. Hence, results of this study are not directly comparable to the previous SOTA. Nevertheless, a few of them are reported for the convenience of the reader. Table 4 quotes the published values.

Table 4. The mAP results of SOTA methods and our continual methods. * indicates that the results are not directly comparable. Acronyms TB, PN, PT, and FN denote TableBank, PubLayNet, PubTables, and FinTabNet respectively.

Method	TB[mAP]	PN[mAP]	PT[mAP]	FN[mAP]
CDeC-Net [31]	96.5	97.8	-	-
CasTabDetectoRS [14]	95.3	-	-	-
DETR [68]	-	-	96.6	-
Faster R-CNN+ResNet (ours)*	89.6	93.7	97.4	90
Sparse R-CNN+PVT (ours)*	90.7	92.5	98.2	93.1

5. Conclusion and Future Work

Continual learning-based methods have shown promising results in computer vision [16,47] and natural language processing [70]. In this paper, we present an incremental learning method for table detection in documents. To the best of our knowledge, this work is the first attempt to investigate the capabilities of a continual learning-based method in the field of document image analysis. We conduct experiments with four different settings: independent training, joint training, fine-tuning, and training with an experience replay mechanism. While the conventional independent training is considered as the upper bound for performance, results with fine-tuning are taken as the lower bound. With the employed continual learning method (ER), we achieve a significant increase of 15% on the results achieved by fine-tuning. Furthermore, the performance from our method is comparable to previous state-of-the-art dataset-specific training methods.

Our evaluations of the modern table detection datasets demonstrate the potential to address the major challenge of relying on dataset-specific training in document image analysis. Moreover, it is an important step in developing robust table detection methods for different domains. We hope that this study serves as an inspiration for future research in continual learning for document image analysis.

Author Contributions: Methodology, M.M, writing—original draft preparation, M.M., K.A.H.; writing—review and editing, K.A.H., and M.R.S.; supervision and project administration, M.R.S., M.Z.A. and D.S. All authors have read and agreed to the submitted version of the manuscript.

Funding: Not applicable.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Zhao, Z.; Jiang, M.; Guo, S.; Wang, Z.; Chao, F.; Tan, K.C. Improving deep learning based optical character recognition via neural architecture search. IEEE Congr. Evol. Computation (CEC), 2020, pp. 1–7.
2. Hashmi, K.A.; Ponnappa, R.B.; Bukhari, S.S.; Jenckel, M.; Dengel, A. Feedback Learning: Automating the Process of Correcting and Completing the Extracted Information. Int. Conf. Document Anal. Recognit. Workshops (ICDARW), Sydney, Australia, September 20–25, 2019, Vol. 5, pp. 116–121.
3. van Strien, D.; Beelen, K.; Ardanuy, M.C.; Hosseini, K.; McGillivray, B.; Colavizza, G. Assessing the Impact of OCR Quality on Downstream NLP Tasks. ICAART (1), 2020, pp. 484–496.
4. Pyreddy, P.; Croft, W.B. Tintin: A system for retrieval in text tables. Proc. 2nd ACM Int. Conf. Digit. libraries, 1997, pp. 193–200.
5. Hashmi, K.A.; Pagani, A.; Liwicki, M.; Stricker, D.; Afzal, M.Z. Cascade Network with Deformable Composite Backbone for Formula Detection in Scanned Document Images. *Applied Sciences* **2021**, *11*, 7610.
6. Bhatt, J.; Hashmi, K.A.; Afzal, M.Z.; Stricker, D. A Survey of Graphical Page Object Detection with Deep Neural Networks. *Applied Sciences* **2021**, *11*, 5344.
7. Zanibbi, R.; Blostein, D.; Cordy, J.R. A survey of table recognition. *Document Anal. Recognit.* **2004**, *7*, 1–16.
8. Hashmi, K.A.; Liwicki, M.; Stricker, D.; Afzal, M.A.; Afzal, M.A.; Afzal, M.Z. Current Status and Performance Analysis of Table Recognition in Document Images with Deep Neural Networks. *IEEE Access* **2021**.
9. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv:1506.01497* **2015**.
10. Gilani, A.; Qasim, S.R.; Malik, I.; Shafait, F. Table detection using deep learning. 14th IAPR Int. Conf. document Anal. Recognit. (ICDAR), Kyoto, Japan, November 9–15, 2017, Vol. 1, pp. 771–776.
11. Chen, K.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Shi, J.; Ouyang, W.; others. Hybrid task cascade for instance segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 4974–4983.
12. Nazir, D.; Hashmi, K.A.; Pagani, A.; Liwicki, M.; Stricker, D.; Afzal, M.Z. HybridTabNet: Towards better table detection in scanned document images. *Applied Sciences* **2021**, *11*, 8396.
13. Prasad, D.; Gadpal, A.; Kapadni, K.; Visave, M.; Sultanpure, K. CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents. Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit. Workshops, Seattle, WA, USA, June 14–19, 2020, pp. 572–573.
14. Hashmi, K.A.; Pagani, A.; Liwicki, M.; Stricker, D.; Afzal, M.Z. CasTabDetectoRS: Cascade Network for Table Detection in Document Images with Recursive Feature Pyramid and Switchable Atrous Convolution. *Journal of Imaging* **2021**, *7*, 214.
15. French, R.M. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences* **1999**, *3*, 128–135.
16. Delange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2021**.
17. Biesialska, M.; Biesialska, K.; Costa-jussà, M.R. Continual lifelong learning in natural language processing: A survey. *arXiv preprint arXiv:2012.09823* **2020**.
18. Li, M.; Cui, L.; Huang, S.; Wei, F.; Zhou, M.; Li, Z. Tablebank: Table benchmark for image-based table detection and recognition. Proc. The 12th Lang. Resour. Eval. Conf., Marseille, France, May 11–16, 2020, pp. 1918–1925.
19. Zhong, X.; ShafieiBavani, E.; Yepes, A.J. Image-based table recognition: data, model, and evaluation. *arXiv:1911.10683* **2019**.
20. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 568–578.
21. Itonori, K. Table structure recognition based on textblock arrangement and ruled line position. Proc. 2nd Int. Conf. Document Anal. Recognit. (ICDAR'93), Tsukuba City, Japan, October 20-22, 1993, pp. 765–768.
22. Chandran, S.; Kasturi, R. Structural recognition of tabulated data. Proc. 2nd Int. Conf. Document Anal. Recognit. (ICDAR'93), sukuba, Japan, October 20–22, 1993, pp. 516–519.
23. Green, E.; Krishnamoorthy, M. Recognition of tables using table grammars. Proc. 4th Annu. Symp. Document Anal. Inf. Retrieval, 1995, pp. 261–278.
24. Pivk, A.; Cimiano, P.; Sure, Y.; Gams, M.; Rajkovič, V.; Studer, R. Transforming arbitrary tables into logical form with TARTAR. *Data & Knowl. Eng.* **2007**, *3*, 567–595.
25. e Silva, A.C.; Jorge, A.M.; Torgo, L. Design of an end-to-end method to extract information from tables. *Int. J. Document Anal. Recognit. (IJDAR)* **2006**, *8*, 144–171.
26. Khusro, S.; Latif, A.; Ullah, I. On methods and tools of table detection, extraction and annotation in PDF documents. *J. Inf. Sci.* **2015**, *41*, 41–57.
27. Coüasnon, B.; Lemaitre, A. Handbook of Document Image Processing and Recognition, chapter Recognition of Tables and Forms. *D. Doermann and K. Tombre, Eds. London, U.K.: Springer* **2014**, pp. 647–677.
28. Embley, D.W.; Hurst, M.; Lopresti, D.; Nagy, G. Table-processing paradigms: a research survey. *Int. J. Document Anal. Recognit. (IJDAR)* **2006**, *8*, 66–86.

29. Siddiqui, S.A.; Malik, M.I.; Agne, S.; Dengel, A.; Ahmed, S. Decnt: Deep deformable cnn for table detection. *IEEE Access* **2018**, *6*, 74151–74161.
30. Schreiber, S.; Agne, S.; Wolf, I.; Dengel, A.; Ahmed, S. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. 14th IAPR Int. Conf. document Anal. Recognit. (ICDAR), Kyoto, Japan, November 9–15, 2017, Vol. 1, pp. 1162–1167.
31. Agarwal, M.; Mondal, A.; Jawahar, C. CDeC-Net: Composite Deformable Cascade Network for Table Detection in Document Images. *arXiv:2008.10831* **2020**.
32. Hashmi, K.A.; Stricker, D.; Liwicki, M.; Afzal, M.N.; Afzal, M.Z. Guided Table Structure Recognition through Anchor Optimization. *arXiv:2104.10538* **2021**.
33. Huang, Y.; Yan, Q.; Li, Y.; Chen, Y.; Wang, X.; Gao, L.; Tang, Z. A YOLO-based table detection method. Int. Conf. Document Anal. Recognit. (ICDAR), Sydney, Australia, September 20–25, 2019, pp. 813–818.
34. Casado-García, Á.; Domínguez, C.; Heras, J.; Mata, E.; Pascual, V. The benefits of close-domain fine-tuning for table detection in document images. Int. Workshop Document Anal. Sys. Springer, Cham, 2020, pp. 199–215.
35. Arif, S.; Shafait, F. Table detection in document images using foreground and background features. Digit. Image Computing : Techn. Appl. (DICTA), 2018, pp. 1–8.
36. Sun, N.; Zhu, Y.; Hu, X. Faster R-CNN based table detection combining corner locating. Int. Conf. Document Anal. Recognit. (ICDAR), Sydney, Australia, September 20–25, 2019, pp. 1314–1319.
37. Qasim, S.R.; Mahmood, H.; Shafait, F. Rethinking table recognition using graph neural networks. Int. Conf. Document Anal. Recognit. (ICDAR), Sydney, Australia, September 20–25, 2019, pp. 142–147.
38. Hao, L.; Gao, L.; Yi, X.; Tang, Z. A table detection method for pdf documents based on convolutional neural networks. 12th IAPR Workshop Document Anal. Sys. (DAS), Santorini, Greece, April 11–14, 2016, pp. 287–292.
39. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. Proc. IEEE Int. Conf. Comput. vision, 2017, pp. 764–773.
40. Saha, R.; Mondal, A.; Jawahar, C. Graphical object detection in document images. Int. Conf. Document Anal. Recognit. (ICDAR), Sydney, Australia, September 20–25, 2019, pp. 51–58.
41. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. Proc. IEEE Int. Conf. Comput. vision, 2017, pp. 2961–2969.
42. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. Proc. IEEE Conf. Comput. vision pattern Recognit., 2018, pp. 6154–6162.
43. Kavasidis, I.; Palazzo, S.; Spampinato, C.; Pino, C.; Giordano, D.; Giuffrida, D.; Messina, P. A saliency-based convolutional neural network for table and chart detection in digitized documents. *arXiv:1804.06236* **2018**.
44. Paliwal, S.S.; Vishwanath, D.; Rahul, R.; Sharma, M.; Vig, L. Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. Int. Conf. Document Anal. Recognit. (ICDAR), Sydney, Australia, September 20–25, 2019, pp. 128–133.
45. Holeček, M.; Hoskovec, A.; Baudiš, P.; Klinger, P. Table understanding in structured documents. Int. Conf. Document Anal. Recognit. Workshops (ICDARW), Sydney, Australia, September 20–25, 2019, Vol. 5, pp. 158–164.
46. Grossberg, S. How does a brain build a cognitive code? *Studies of mind and brain* **1982**, pp. 1–52.
47. Qu, H.; Rahmani, H.; Xu, L.; Williams, B.; Liu, J. Recent Advances of Continual Learning in Computer Vision: An Overview. *arXiv preprint arXiv:2109.11369* **2021**.
48. Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; others. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* **2017**, *114*, 3521–3526.
49. Liu, L.; Kuang, Z.; Chen, Y.; Xue, J.H.; Yang, W.; Zhang, W. Incdet: In defense of elastic weight consolidation for incremental object detection. *IEEE transactions on neural networks and learning systems* **2020**, *32*, 2306–2319.
50. Rebuffi, S.A.; Kolesnikov, A.; Sperl, G.; Lampert, C.H. iCaRL: Incremental Classifier and Representation Learning. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
51. Hinton, G.; Vinyals, O.; Dean, J.; others. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* **2015**, *2*.
52. Zhang, J.; Zhang, J.; Ghosh, S.; Li, D.; Tasci, S.; Heck, L.; Zhang, H.; Kuo, C.C.J. Class-incremental learning via deep model consolidation. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 1131–1140.
53. Shmelkov, K.; Schmid, C.; Alahari, K. Incremental learning of object detectors without catastrophic forgetting. Proceedings of the IEEE international conference on computer vision, 2017, pp. 3400–3409.
54. Girshick, R. Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015.
55. Zitnick, C.L.; Dollár, P. Edge boxes: Locating object proposals from edges. European conference on computer vision. Springer, 2014, pp. 391–405.
56. Acharya, M.; Hayes, T.L.; Kanan, C. Rodeo: Replay for online object detection. *arXiv preprint arXiv:2008.06439* **2020**.
57. Shieh, J.L.; Haq, M.A.; Karam, S.; Chondro, P.; Gao, D.Q.; Ruan, S.J.; others. Continual learning strategy in one-stage object detection framework based on experience replay for autonomous driving vehicle. *Sensors* **2020**, *20*, 6777.
58. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pvtv2: Improved baselines with pyramid vision transformer. *Computational Visual Media* **2022**, *8*, 1–10.

59. Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C.; others. Sparse r-cnn: End-to-end object detection with learnable proposals. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14454–14463.
60. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2017**, *39*, 1137–1149. doi:10.1109/TPAMI.2016.2577031.
61. Girshick, R. Fast r-cnn. Proc. IEEE Int. Conf. Comput. vision, 2015, pp. 1440–1448.
62. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; Zhang, Z.; Cheng, D.; Zhu, C.; Cheng, T.; Zhao, Q.; Li, B.; Lu, X.; Zhu, R.; Wu, Y.; Dai, J.; Wang, J.; Shi, J.; Ouyang, W.; Loy, C.C.; Lin, D. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv preprint arXiv:1906.07155* **2019**.
63. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
64. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, pp. 248–255.
65. Michaelis, C.; Mitzkus, B.; Geirhos, R.; Rusak, E.; Bringmann, O.; Ecker, A.S.; Bethge, M.; Brendel, W. Benchmarking Robustness in Object Detection: Autonomous Driving when Winter is Coming. *arXiv preprint arXiv:1907.07484* **2019**.
66. arXiv.org e-Print archive. https://arxiv.org/, 2022.
67. Zhong, X.; Tang, J.; Yepes, A.J. Publaynet: largest dataset ever for document layout analysis. Int. Conf. Document Anal. Recognit. (ICDAR), Sydney, Australia, September 20–25, 2019, pp. 1015–1022.
68. Smock, B.; Pesala, R.; Abraham, R.; Redmond, W. PubTables-1M: Towards comprehensive table extraction from unstructured documents. *arXiv preprint arXiv:2110.00061* **2021**.
69. Zheng, X.; Burdick, D.; Popa, L.; Zhong, X.; Wang, N.X.R. Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. Proc. IEEE/CVF Winter Conf. Appl. Comput. Vision, 2021, pp. 697–706.
70. Huang, Y.; Zhang, Y.; Chen, J.; Wang, X.; Yang, D. Continual learning for text classification with information disentanglement based regularization. *arXiv preprint arXiv:2104.05489* **2021**.