MDPI

Article

Cascade Network with Deformable Composite Backbone for Formula Detection in Scanned Document Images

Khurram Azeem Hashmi [1,2,3,*], Alain Pagani [3], Marcus Liwicki [4], Didier Stricker [1,3] and Muhammad Zeshan Afzal [1,2,3*]

1 Department of Computer Science, Technical University, 67663 Kaiserslautern, Germany; khurram_azeem.hashmi@dfki.de (K.A.H.); muhammad_zeshan.afzal@dfki.de, afzal.tukl@gmail.com (M.Z.A.); alain.pagani@dfki.de (A.P.); didier.stricker@dfki.de (D.S.);
2 Mindgarage, Technical University, 67663 Kaiserslautern, Germany
3 German Research Institute for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany,
4 Luleå University of Technology, 971 87 Luleå, Sweden; marcus.liwicki@ltu.se (M.L.);
* Correspondence: khurram_azeem.hashmi@dfki.de

Abstract: This paper presents a novel architecture for detecting mathematical formulas in document images, which is an important step for reliable information extraction in several domains. Recently, Cascade Mask R-CNN networks have been introduced to solve object detection in computer vision. In this paper, we suggest a couple of modifications to the existing Cascade Mask R-CNN architecture: First, the proposed network uses deformable convolutions instead of conventional convolutions in the backbone network to spot areas of interest better. Second, it uses a dual backbone of ResNeXt-101, having composite connections at the parallel stages. Finally, our proposed network is end-to-end trainable. We evaluate the proposed approach on the ICDAR-2017 POD and Marmot datasets. The proposed approach demonstrates state-of-the-art performance on ICDAR-2017 POD at a higher IoU threshold with an f1-score of 0.917, reducing the relative error by 7.8%. Moreover, we accomplished correct detection accuracy of 81.3% on embedded formulas on the Marmot dataset, which results in a relative error reduction of 30%.

Keywords: Formula detection; Cascade Mask R-CNN; Mathematical expression detection; document image analysis; deep neural networks; computer vision.

1. Introduction

Information extraction from document images is the primary need in various domains such as banking, archiving, or academia and industry in general. Research in document analysis has been trying to develop precise information extraction systems for several years [1–4]. Although state-of-the-art Optical Character Recognition (OCR) systems [5,6] recognize regular text with high accuracy, they are vulnerable to recognize information from page objects (tables, figures, mathematical formulas) in document images [7,8]. Figure 1 illustrates the problem in which an open-source OCR, Tesseract [4][1] is applied to extract the content from a document image. Besides recognizing the textual content, the OCR fails to extract the information from mathematical formulas. This shows that formula detection is a crucial preliminary step for the information extraction in such document images.

Mathematical formulas are an integral part of documents because they allow us to represent complex information concisely by exploiting mathematics capabilities. Formulas present in the documents are categorized into isolated formulas (mentioned in a separate line) and embedded formulas (inline mathematical symbols). Figure 2 exhibits the problem of detecting isolated and embedded formulas in document images.

[1] We use latest (LSTM based) version 4.1.1 available at https://github.com/tesseract-ocr/tesseract

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0165.v1

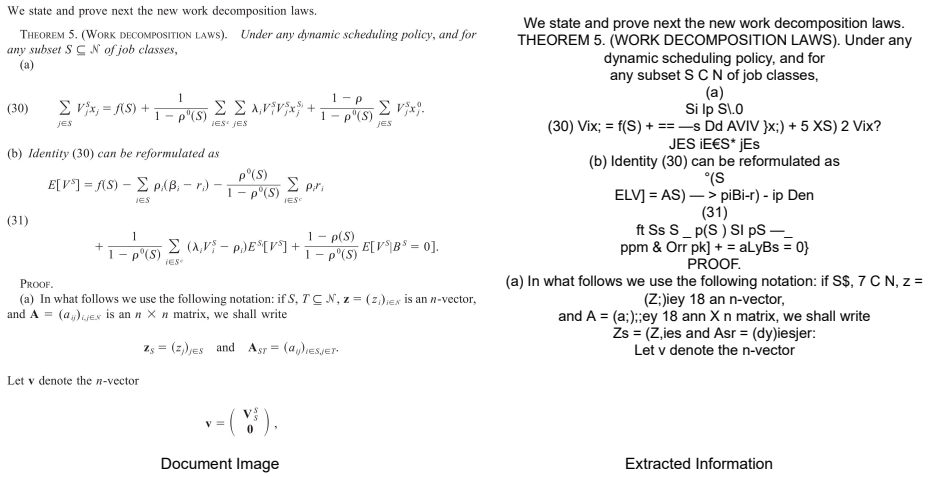


Figure 1. Illustrating the need of applying formula detection before extracting information in document images. We apply open source Tesseract-OCR [4] on a document image containing mathematical formulas. Besides the textual content, the OCR system fails miserably in recognizing information from formulas.

The task of detecting both isolated and embedded formulas in document images is a difficult problem because of the underlying low inter-class and high intra-class variance [9]. The hurdles involved in detecting isolated and embedded formulas are exhibited in Figure 2. The isolated formulas present in a document image can easily be misclassified with other page objects due to low inter-class variance with tables, algorithms, and figures. The embedded formulas contain mathematical functions (log, exp, tan), operators ($\times, +, \sigma, \%$), and variables (i, j, k). These inline expressions are prone to misinterpret with the regular text in a document image [10].

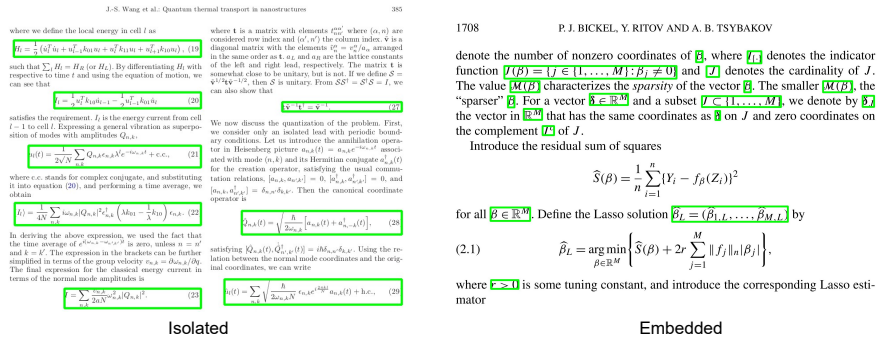


Figure 2. Instances of isolated and embedded formulas in sample document images. The green boundaries represent the ground truth regions. Separate images are used for the convenience of the readers. The isolated formulas spanning multiple lines are prone to misclassified with tables, whereas the embedding formulas confuse with the regular text.

Previous works employed hand-crafted features to detect formulas in documents [2,11,12]. Although these systems extract mathematical formulas, they fail to obtain effective results on generic datasets. Later, statistical learning, mainly machine learning-based methods, has advanced the performance of formula identification systems [13–15]. The recent success of deep learning-based methods on computer vision within the last decade also had an impact on the task of formula detection in scanned document images. Several deep learning-based formula detection approaches [16–19] have been presented

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0165.v1

in the past two years. They are mainly equipped with object detection algorithms such as Faster R-CNN [20], YOLO [21], SSD [22], and FPNs [23].

In recent work, Agarwal *et al.* [24] presented a method equipped with Cascade Mask R-CNN [25] to tackle the problem of table detection in document images. However, the capabilities of Cascade Mask R-CNN have not been investigated yet in the domain of mathematical formula detection in document images.

This paper presents an end-to-end data-driven approach to detect both isolated and embedded formulas in document images. The main contributions of this paper are as follows:

- We present an end-to-end trainable framework that operates on cascade Mask R-CNN equipped with a deformable composite backbone to detect both isolated and embedded formulas in document images.
- Unlike prior work, our formula detection pipeline operates on a lightweight dilation method as a pre-processing step.
- We accomplish state-of-the-art results in detecting isolated formulas on higher IoU threshold in ICDAR-2017 POD dataset [26]. Furthermore, on Marmot dataset [27], we surpass previous state-of-the-art results on embedded formulas with a huge margin and achieve identical results with prior state-of-the-art on isolated formulas.

2. Related work

Research progress in the field of document image analysis directly relates to the advances in the computer vision research community. The task of formula detection in documents is a well-studied problem [28]. Noticeable progress has been achieved in this domain by implementing custom-heuristics to deep learning-based approaches. Earlier, rule-based approaches develop character-based heuristics to identify formulas in documents [29–32]. These techniques look for special characters (e.g., “>”, “×”, “=”) that mainly exist in mathematical formulas.

Kacem *et al.* [11] came with a model based on fuzzy logic to detect mathematical symbols. The approach predicts the formula region by exploiting the features of mathematical symbols. Inoue *et al.* [2] first employed conventional OCR method to extract characters. The method treated all the remaining characters as mathematical symbols that OCR was unable to parse.

Specific OCR systems have been presented that recognize mathematical symbols based on their positions and sizes [2]. Baker *et al.* [12] segregated the lines containing formulas to the regular textual lines in order to detect isolated formulas in PDF documents.

Decision trees have been equipped to detect isolated formulas by classifying formula lines with the plain text lines [33]. Chang *et al.* [14] proposed a similar method based on the projection of the features that only works for isolated formulas in documents.

Later, machine learning-based algorithms have been proposed to alleviate the performance of formula detection systems in documents [13,34]. Liu *et al.* [15] leveraged the combination of Conditional Random Field (CRF) and Support Vector Machine (SVM) to classify sparse lines in documents. Subsequently, the method distinguished formulas from other graphical page objects such as figures and tables by applying custom heuristics.

Succeedingly, the researchers have investigated the capabilities of Deep Neural Networks (DNNs) for the problem of formula identification in document images [26,35]. To the best of our knowledge, He *et al.* [36] exploited Convolutional neural networks (CNNs) with spatial context to detect mathematical symbols in document images. Later, Gao *et al.* [37] presented a deep learning-based formula detection system in PDF documents.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0165.v1

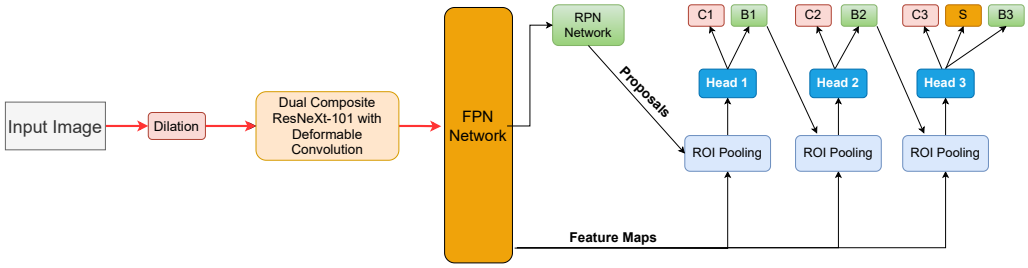


Figure 3. The presented framework is based on Cascade Mask R-CNN equipped with a deformable composite backbone applied on dilated document images. Modules B, C, and D represent bounding box, classification, and segmentation, respectively.

NLPR-PAL [26] produced the best results in the competition of POD at ICDAR -2017. The proposed a blend of connected components, SVM and Faster R-CNN [20] to detect figures, formulas, and tables in document images.

Yi *et al.* [38] published another CNNs-based approach that detects graphical page objects like tables, figures, and formulas in document images. The authors employed the dynamic programming technique instead of Non-Maximum Suppression (NMS) to refine the final candidate proposals. Semantic segmentation-based architecture like U-Net [39] has also been utilized to detect mathematical expressions in scientific document images [16].

Recently, Phong *et al.* [17] published a method equipped with YOLO [40] to detect mathematical formulas in document images. In another approach [18], SSD [22] has been exploited to detect mathematical expression in PDF documents.

Another graphical page object detection system is published by Li *et al.* [41]. The authors combined deep structure prediction with a traditional approach to detecting page objects, including formulas in document images. Younas *et al.* [19] introduced a system called *Fi-Fo* that detects figures and formulas in document images. The authors empirically established that deformable convolutions [42] with Feature Pyramid Networks (FPN) [23] is a better fit as compared to other object detection algorithms. The proposed approach heavily relied on the image transformation pre-processing techniques to produce state-of-the-art results.

3. Method

The presented approach is comprised of Cascade Mask R-CNN [43] equipped with a recently published composite backbone having deformable convolutions replaced with traditional convolution filters. Figure 3 illustrates the complete pipeline of our proposed framework. In this section, we dive deeper into each component of our proposed method.

3.1. Cascade Mask R-CNN

We treat the problem of formula detection in document images as an object detection problem on natural images. Recently, Cai and Vasconcelos [25] introduced Cascade R-CNN [25] that extends the concept of the idea of Faster R-CNN [20] by adding multi staging technique. In our approach, we incorporate the instance segmentation branch as proposed in the original Mask R-CNN [43].

As explained in Figure 3, the input image is passed through the composite ResNeXt-101 backbone, which is explained in Section 3.2. The backbone extracts the spatial features and generates feature maps. The Region Proposal Network (RPN) head estimates the possible candidate regions where formulas can be present. The first bounding box component receives the features from the RPN and creates predictions. Each of the three bounding box modules performs classification and regression. The classification score and bounding box coordinates predicted by each bounding box head $BH1$, $BH2$, and $BH3$, are denoted with $(C1, B1)$, $(C2, B2)$, and $(C3, B3)$ respectively. The output of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0165.v1

one bounding box head becomes the training input for the next head. This cascaded regression and classification method optimizes the process of differentiating false positive samples with true positives even at higher IoU thresholds. After computing the refined bounding boxes and classification scores from $BH3$, the segmentation head predicts the mask that contributes to the loss function to optimize the training further.

3.2. Composite Backbone

We employ a robust and novel dual backbone architecture to extract the possible spatial features to detect formulas in document images. The performance of any object detection algorithm depends on the quality of the feature map it receives from the feature extraction network [44]. In this paper, we implement a dual backbone-based network [45] in which the first backbone is the assistance backbone, and the other is known as the lead backbone. Both of the backbones are compositely connected to each other so that the assistant backbone's output features are treated as input features for the lead backbone. Figures 4 illustrates the architecture of our dual composite backbone.

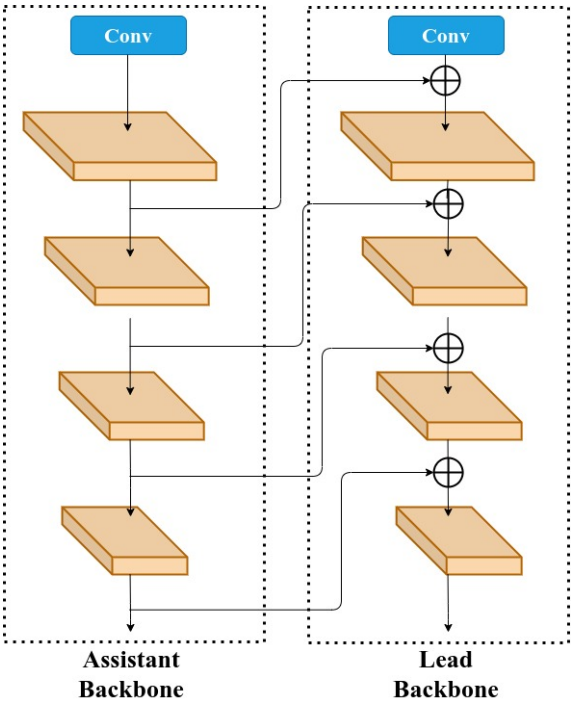


Figure 4. Visual explanation of the employed backbone (CBNet) in our framework. We utilize a dual ResNeXt-101 backbone, in which there are composite connections between parallel stages of the adjacent assistant and lead backbone. Moreover, we replace the conventional convolutions in ResNeXt101 with deformable convolution.

For the conventional convolutional network with single backbone, the output of $l-1-th$ stage is propagated as input to the $l-th$ stage which is given by:

$$x^l = F^l(x^{l-1}), l \geq 2 \tag{1}$$

where F^l represents the non-linear function on $l-th$ level. Contrary to this, our backbone network receives input from prior levels and parallel level of the assistant backbone. Therefore, the input of a lead backbone bl at stage l is the product of output of lead backbone at $l-1th$ stage and parallel $l-th$ stage of assistant backbone ba. Mathematically it is explained in [45] as:

$$x^l_{bl} = F^l_{bl}((x^{l-1}_k) + g(x^l_{ba})), l \geq 2 \tag{2}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0165.v1

Where g defines the composite connection between the lead and assistant backbone, these composite connections enable the lead backbone to extract essential spatial features. As explained in Figure 3, we propagate the output of the final lead backbone to the region proposal network of our Cascade Mask R-CNN.

3.3. Deformable Convolution

We incorporate deformable convolution filters [42] instead of the conventional convolutions that exists in the ResNeXt-101 architecure [46]. The Convolutional neural networks extract the important spatial features that are essential to perform the required task. Based on the hierarchy, convolutional layers discover different features [47]. Convolutional layers present at the bottom search for crude features such as sharp edges or the gradients, whereas the layers at higher levels look for the abstract components such as complete object [48]. The conventional convolution operation has the same effective receptive field for all the neurons. The 2D convolution comprises of two parts: 1) The first step samples the input feature map through a grid R; 2) Aggregation of samples values multiplied by the weight $\mathbf{w}$. For conventional convolution, the output of feature map y for each position p_o is elaborated in [42] as follows:

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \times \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n) \tag{3}$$

where x represents the input feature map, p_n iterates over the locations in a grid R that can be defined as $R = (-1,-1),(-1,0),(-1,1),(0,-1),(0,0),(0,1),(1,-1),(1,0),(1,1)$ for a 3×3 convolutional layer. The effective receptive field of such a filter is restricted to these nine positions.

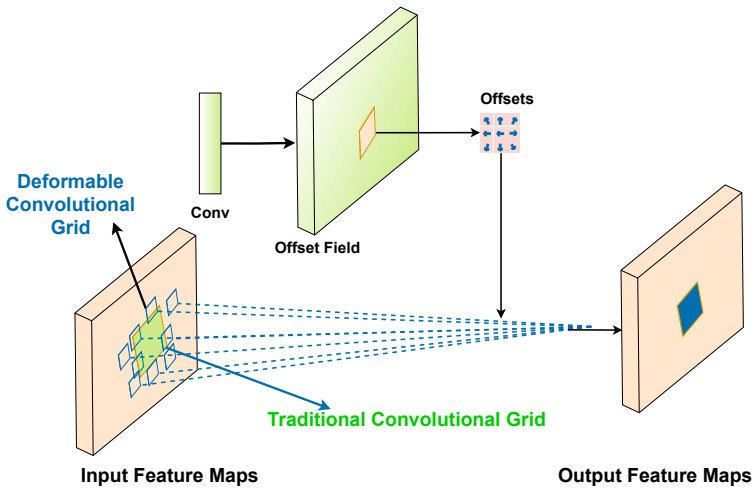


Figure 5. Demonstration of deformable convolution.

In case of deformable convolution, an additional offset represented as $\Delta(p_n)$ is added, which deforms the filter's receptive field by augmenting the predefined offsets. Hence, the Equation 3 as explained in [42] is transformed into

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \times \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta\mathbf{p}_n) \tag{4}$$

This modification makes the sampling process irregular with an offset value of $p_n + \Delta(p_n)$. As these offsets are differentiable and fractional, bilinear interpolation is

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0165.v1

used to implement them. Considering, $p = p_0 + p_n + \Delta(p_n)$, the bilinear interpolation is implemented as follows:

$$\mathbf{x}(\mathbf{p}) = \sum_{\mathbf{q}} \mathbf{G}(\mathbf{q}, \mathbf{p}) \times \mathbf{x}(\mathbf{q}) \tag{5}$$

where q iterates over all the possible places on the input feature map x and the symbol G represents the bilinear interpolation kernel. It is vital to mention that G is a two-dimensional kernel that can be further divided into two one-dimensional kernels. It is mathematically explained as:

$$\mathbf{G}(\mathbf{q}, \mathbf{p}) = \mathbf{g}(\mathbf{q_x}, \mathbf{p_x}) \times \mathbf{g}(\mathbf{q_y}, \mathbf{p_y}) \tag{6}$$

where g is explained as $g(a, b) = max(0, 1 - |a - b|)$. It is important to note that Equation 5 is more efficient since $G(q, p)$ is zero for most of the $\mathbf{q}$s. We refer our readers to [42,49] for a detailed explanation about deformable convolutions.

3.4. Image Transformation and Prepossessing

Document images mainly consist of textual regions. There exist a variable amount of gap between textual components. This gap not only separates the textual components but also provides a higher level of semantic representation. We can think of formula detection as a semantic labeling task where a textual unit is labeled as a formula or other text depending upon its contents. In order to group closely related regions, we apply dilation transformation on the images. The dilation transformation converts the input images to semantically enriched representation. It is crucial to understand that this grouping cannot replace the actual image content. Therefore we concatenate the prepossessed images with the original images. This concatenation increases the number of input channels. The deep neural network processes this combination.

3.4.1. Dilation Transformation

The dilation transformation is used to thicken the black regions in the input image. Since this transformation works on binary images, the input images are binarized first. The black pixel represents the characters, and the white pixels describe the background in the binarized images. Therefore, this transformation thickens the characters. Figure 6 depicts the output of dilation transformation on one of the sample images. We use a structuring element of 2×2. We tried different sizes of the structuring elements. However, 2×2 produces the optimal results.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0165.v1

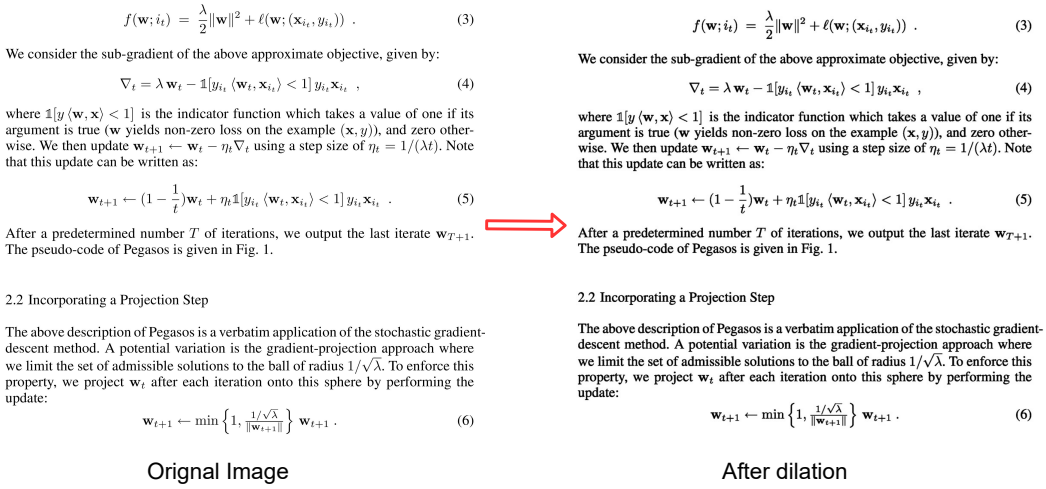


Figure 6. Example of a document image before and after our pre-processing method. The dilation process facilitates our feature extraction network by increasing the boundaries of foreground pixels which results in reducing the number of background pixels.

Table 1. Summary of the main statistics of the employed datasets.

Datasets	ICDAR-17		Marmot	
	Train	Test	Train	Test
Number of Images	1600	817	330	70
Number of Isolated Formulas	3534	1929	1322	253
Number of Embedded Formulas	-	-	6951	956

4. Experimental Results

4.1. Datasets

We employed the well-known publicly available formula detection datasets to conduct our experiments. This section elaborates these datasets, and their summary is presented in Table 1.

4.1.1. ICDAR-17

ICDAR-17 is the result of the recent competition about graphical Page Object Detection (POD) [26] in document images at ICDAR in 2017. There are 2417 document images in the dataset having annotations for figures, formulas, and tables in document images. In addition, the dataset contains a variety of isolated formulas present on the single and multi-column document images. For the experiments, we have used 1600 images for training and 817 images for testing purposes. Recently, Younas *et al.* [19] published the corrected version of this dataset which leads to more formulas in the dataset. Therefore, we have employed the revised version of the dataset in our experiments for direct comparison with state-of-the-art results.

4.1.2. Marmot

Mormot [50] is fairly a smaller dataset consisting of 400 scanned document images. However, the dataset contains annotations for isolated and embedded mathematical equations. There are 1575 isolated formulas varying from 4 to 20 formulas per document image, whereas 7907 embedded formulas with an average of almost 20 embedded formulas per document image.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0165.v1

4.2. Model Configuration

We implement the proposed method in Pytorch by leveraging the MMdetection object detection pipeline [51]. Our composite backbone ResNeXt-101 [46] is pre-trained on MS-COCO dataset [52]. The pre-trained feature extraction network facilitates our object detection algorithm to adapt from the domain of natural scenes to documents. We scaled the input document images to 1200×800 but maintained the original aspect ratio. The training starts with a learning rate of 0.0025, which is reduced after every 8th epoch. We train the network for a total of 20 epochs for both of the datasets. The IoU threshold values for cascaded bounding boxes are set to [0.5, 0.6, 0.7]. We employed three different anchor ratios of [0.5, 1.0, 2.0] with only one anchor scale of [8] since FPN [23] itself performs the multi-scale detection owing to their top-down architecture. We operated with a batch size of 1 to train our network. The models for both of the datasets are trained on NVIDIA GeForce RTX 101 Ti GPU with 12 GB memory.

4.3. Evaluation Metrics

For ICDAR-2017 POD, we work with the same evaluation criteria as elaborated in the ICDAR-2017 POD competition [26]. For the Marmot dataset, we follow the identical criteria of computing detection accuracy as explained in [10] to have direct comparisons. We report results by employing the following metrics:

4.3.1. Precision

The precision [53] defines the ratio of positive samples over all the predicted samples. Mathematically, it is given by:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{7}$$

4.3.2. Recall

The recall [53] calculates the ratio of positive samples in predictions over all the positive samples present in the ground truth. It is explained as follows:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{8}$$

4.3.3. F1-Score

The metrics f1-score [53] is the measure that is computed by taking harmonic mean of precision and recall. The formula for f1-score is:

$$\text{F1-Score} = \frac{2\times \text{ Precision } \times \text{ Recall}}{\text{Precision} + \text{Recall}} \tag{9}$$

4.3.4. Mean Average Precision (mAP)

The mean average precision also referred to as mAP score is calculated by averaging maximum precision over various recall thresholds. Mathematically, it is explained in [52] as follows:

$$\text{mAP} = \frac{1}{N}\sum_{r=1}^{N} AP_r \tag{10}$$

where AP_r is the average precision on a recall level r.

4.3.5. Intersection Over Union (IOU)

The metrics Intersection over union [54] estimates the amount of predicted region intersecting with the ground truth region. It is explained as follows:

$$\text{IoU(A,B)} = \frac{\text{Area of Overlap region}}{\text{Area of Union region}} = \frac{|A \cap B|}{|A \cup B|} \tag{11}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0165.v1

4.3.6. Detection Accuracy

We report results on the Marmot dataset using the metrics of detection accuracy. As explained in [10], we classify the prediction into correct and partial correct based on the IoU value:

1. **Correct**: the predicted bounding box is considered correct when the IoU score between the predicted formula region and the ground truth is equal or greater than 0.5.
2. **Partial**: when the IoU score between the inferred and the ground truth formula region is in the interval (0; 0.5), the detection is categorized as partial.

4.4. Result and Discussion

We report results on the datasets of ICDAR-2017 POD [26] and Marmot [27] to demonstrate the effectiveness of the proposed method. This section analyzes the qualitative and quantitative performance of our approach by highlighting the strengths and weaknesses. Furthermore, it compares the presented results with prior state-of-the-art methods.

4.4.1. ICDAR-17

We follow the evaluation protocol as elaborated in ICDAR-2017 POD [26]. We first calculate the number of true positives, false positives, and false negatives from the complete test set. We then compute the precision, recall, and F1-Score as calculated in the prior methods [19,41]. Moreover, we also report the mAP score by evaluating the performance of our method on the test set. Following the criteria of the competition, we present results on the IoU threshold of 0.6 and 0.8. It is essential to emphasize that we have employed the recently published corrected version of the dataset [19]. Therefore, only the methods that have reported results on the corrected version of the dataset are directly comparable with our approach.

Table 2 presents the results that are achieved by our proposed end-to-end method with and without incorporating the pre-processing technique. After setting an IoU threshold of 0.6, we achieve a precision of 0.95, recall 0.948, f1-score 0.949, and mAP of 0.97 without the inclusion of the pre-processing method. The results further improve with an average of almost 0.04 after employing the proposed pre-processing. Upon increasing the IoU threshold value to 0.8, our network reaches a precision of 0.914, recall of 0.912, f1-score of 0.913, and mAP score of 0.949 in the absence of pre-processing method presence of pre-processing advances the results with an average difference of 0.04.

Figure 7 depicts the qualitative performance of our proposed system. Out of 1929 isolate formulas present in the test set, our cascade formula detection network correctly predicted the region for 1836 formulas at an IoU threshold of 0.6. Moreover, it is vital to mention that even at a higher IoU threshold of 0.8, the system identified correct boundaries for 1767 formulas present in the test set. We also observe some rare cases of false positive and false negative samples, which are exhibited in Figure 7(b) and (c).

Comparison with State-of-the-art Methods

By looking at Table 2, it is evident that our hybrid method of cascade network leveraging deformable composite backbone with lightweight pre-processing has outperformed the prior state-of-the-art method [19] on a higher IoU threshold of 0.8 with an average f1-score of 0.917, thus reducing the relative error by 7.8%. Furthermore, we achieve an almost identical f1-score at an IoU threshold of 0.6. It is essential to emphasize that the previous state-of-the-art work [19] depends on the heavy pre-processing pipeline consisting of distance transform, connected components analysis (CCA) applied on greyscale images. However, our generic data-driven method operates on the lightweight dilation technique to produce better results.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0165.v1

Table 2. Quantitative analysis of the presented work with existing state-of-the-art methods on the ICDAR-2017 POD dataset. † represents the results that are not directly comparable with our method because they are not evaluated on the revised version of the dataset.

Method	ICDAR-2017 POD							
	IoU = 0.6				IoU = 0.8			
	Precision	Recall	F1-Score	AP	Precision	Recall	F1-Score	AP
NLPR-PAL [26]†	0.901	0.929	0.915	0.839	0.888	0.916	0.902	0.816
Li et al. [41]†	0.935	0.331	0.489	0.312	0.877	0.310	0.459	0.274
Fi-Fo Detector Non Deformable [19]	0.910	0.927	0.918	0.953	0.860	0.877	0.868	0.928
Fi-Fo Detector Deformable [19]	**0.957**	**0.952**	**0.954**	0.949	0.913	0.908	0.910	0.898
Ours (Without Pre-Processing)	0.950	0.948	0.949	0.97	0914	0.912	0.913	0.949
Ours (Complete Method)	0.954	**0.952**	0.953	**0.97.5**	**0.918**	**0.916**	**0.917**	**0.954**

Table 3. Performance comparison between our method and previous state-of-the-art approaches on the Marmot dataset.

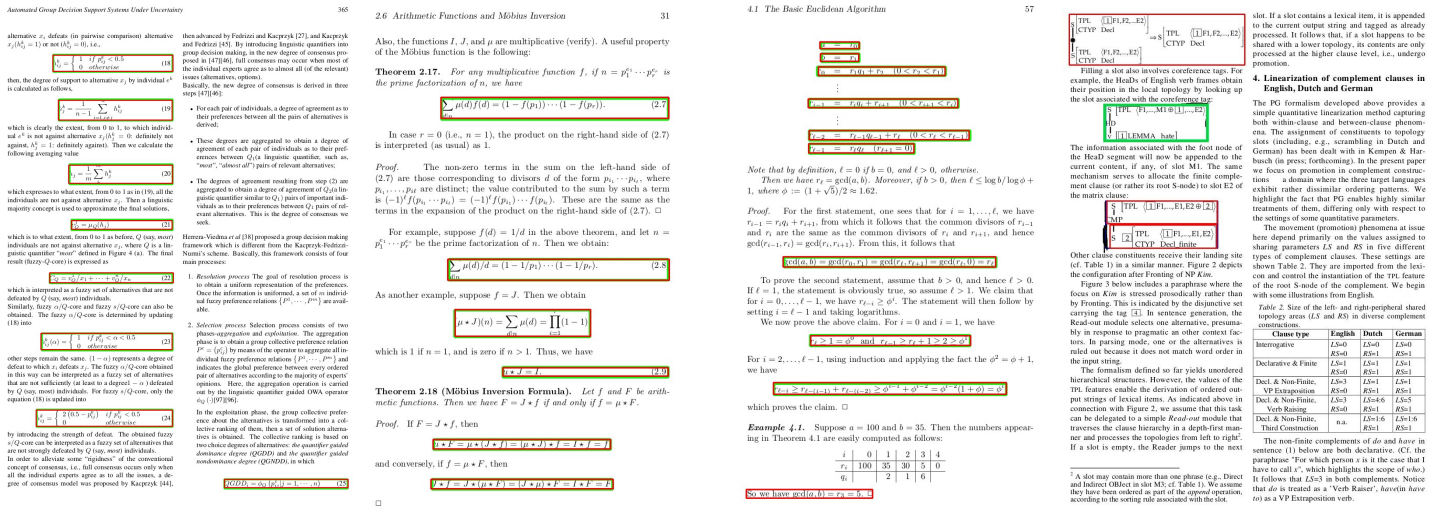
Method	Formula	Correct (%)	Partial (%)	Total
Chu et al. [55]	Isolated	26.87	44.87	71.76
	Embedded	1.74	28.87	30.61
Phong et al. [10]	Isolated	50.37	39.14	91.18
	Embedded	22.9	58.45	81.35
Phong et al. [17]	Isolated	**93**	-	-
	Embedded	73	-	-
Ours (Without Pre-processing)	Isolated	92.5	4.64	97.14
	Embedded	80.6	6.23	86.83
Ours (Complete)	Isolated	**93**	**4.86**	**97.86**
	Embedded	**81.3**	**6.77**	**88.07**

4.4.2. Marmot

We follow similar evaluation criteria to report results on the marmot dataset in order to draw a direct comparison with the prior work. Our network separately detects the isolated and embedded formulas in a document image due to their variable sizes between isolated and embedded formulas. Table 3 summarizes the performance of our method on the Marmot dataset. As explained in Section 4.3.6, we calculate the accuracies of correct and partial detections. Our proposed mathematical formula identification system achieves the correct detection accuracy of 93% and 92.5% on isolated formulas with and without incorporating the pre-processing method, respectively. In embedded formulas, the system obtains the correct detection accuracy of 81.3% and 80.6% equipped with and without the proposed dilation method, respectively.

The qualitative performance analysis of the presented method on marmot dataset is exhibited in Figures 8, 9, and 10. We predict correct regions for 236 out of 253 formulas present in the test set in detecting isolated formulas. In the case of embedded formulas, the network is able to precisely detect 777 out of the 956 formulas from the test set.

 doi:10.20944/preprints202107.0165.v1



(a) True positives (b) True positives and False Positives (c) False positives and False negatives

Figure 7. Performance evaluation on the proposed method on the ICDAR-2017-POD dataset. The green color depicts the ground truth, while red denotes the predicted bounding boxes. In this figure, (a) represents two instances of true positives. Part (b) highlights an example of both true positives and a single false positive case, whereas (c) shows three false positives with one false negative.

Lemma A2: *If* $\bar{\kappa} < \kappa_L$ *the Democratic party wins for sure and picks* $\bar{v} = 1$ *and* $v_D = v$.

Proof: This follows by observing that for $\bar{\kappa} < \kappa_L$ the Democrats win for sure and hence pick their ideal policy.■

Now define:

$$-\kappa_H = -\kappa_L + \frac{\Delta}{(1 + T_v(\bar{v}))} .$$

Lemma A3: *For* $\bar{\kappa} \in (\kappa_L, \kappa_H)$, $\underline{v} < v_D^* < \overline{v} = v_R^*$

Proof: First, we show for all $\bar{\kappa} < \kappa_H$ the Republicans will pick $v_R = \bar{v}$. To see this, observe that at $v_R = \bar{v}$ and $v_D = v$ the change in the payoff of the Republican party from a small increase in v is:

$$\left[\frac{1}{2} - \xi\left[-\kappa + \underline{v} - \overline{v}\right]\right](1 + T_v(\bar{v})) + \xi\left[\Delta + \overline{v} - \underline{v}\right] \;>\; \left[\frac{1}{2} - \xi\left[-\kappa_H + \underline{v} - \overline{v}\right]\right](1 + T_v(\bar{v})) + \xi\Delta \;=\; 0$$

from the definition of κ_L. Moreover, Assumption 1 implies that this inequality holds for all $v_D > v$.

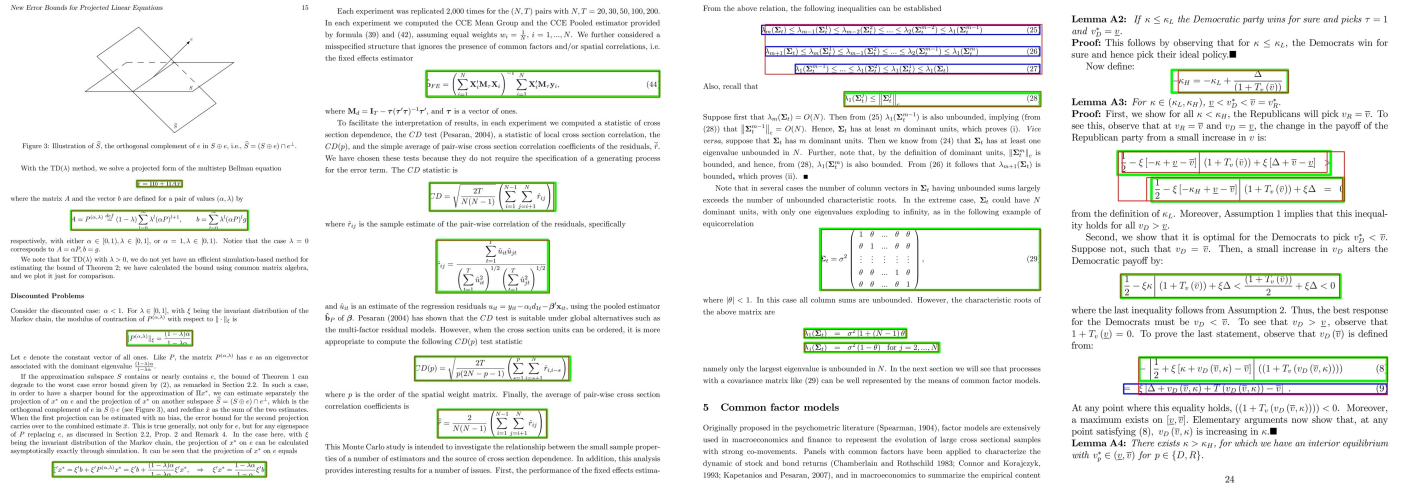
Second, we show that it is optimal for the Democrats to pick $v_D < v$. Suppose not, such that $v_D = \bar{v}$. Then, a small increase in v_D alters the Democratic payoff by:

Figure 9. Instances of correct detections of embedded formulas in a document image taken from the Marmot dataset. The green color highlights the ground truth, whereas the predictions are marked with red color.

Comparison with State-of-the-art Methods

We compare our results with earlier approaches on the Marmot dataset in Table 3. From the table, It is evident that our cascade network with a deformable composite backbone has clearly out-smarted the prior state-of-the-method [17] in detecting embedded formulas while accomplished identical results in the case of isolated formulas. We reduce the relative error of 30% by achieving the detection accuracy of 81.3% on embedded formulas. Another point that is worth mentioning is the partial detection accuracy. The proposed system partly predicts 4.86% from the remaining 7% missing isolated formulas, which makes the total detection accuracy of 97.86%. For embedded formulas, we achieve partial detection accuracy of 6.77%, which adds up to 88.07% total detection accuracy. Therefore, our network only missed 2.14% and 11.93% of isolated formulas and embedded formulas from the test set, respectively. The reduced number of

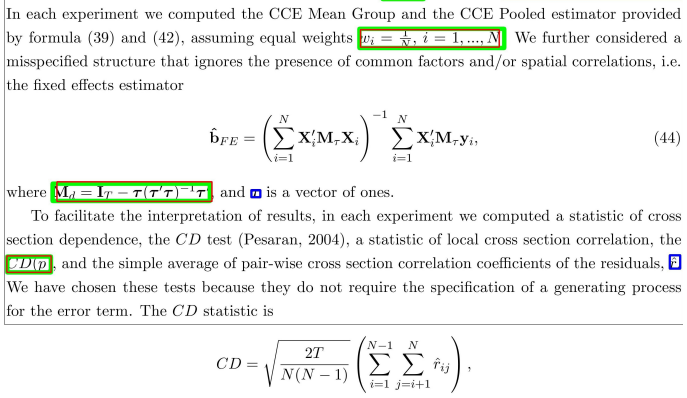
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 6 July 2021 doi:10.20944/preprints202107.0165.v1



(a) Correct Detections (b) Partial Detection

Figure 8. Instances of correct and partial detection of isolated formulas on Marmot dataset. The green color represents the correct detections, whereas the partial and missed detections are highlighted with red and blue colors, respectively. Part (a) depicts a couple of samples of correct detection in which an IoU score between ground truth and predicted region is greater or equal to 0.5, whereas part (b) illustrates few cases of partial and missed detection.

missed detections in isolated and embedded formulas demonstrate the superiority of the proposed method.



Each experiment was replicated 2,000 times for the (N, T) pairs with $N, T = 20, 30, 50, 100, 200$. In each experiment we computed the CCE Mean Group and the CCE Pooled estimator provided by formula (39) and (42), assuming equal weights $w_i = \frac{1}{N}, i = 1, ..., N$. We further considered a misspecified structure that ignores the presence of common factors and/or spatial correlations, i.e. the fixed effects estimator

$$\hat{\mathbf{b}}_{FE} = \left(\sum_{i=1}^{N} \mathbf{X}_i' \mathbf{M}_\tau \mathbf{X}_i \right)^{-1} \sum_{i=1}^{N} \mathbf{X}_i' \mathbf{M}_\tau \mathbf{y}_i, \tag{44}$$

where $\mathbf{M}_d = \mathbf{I}_T - \boldsymbol{\tau}(\boldsymbol{\tau}'\boldsymbol{\tau})^{-1}\boldsymbol{\tau}'$, and $\boldsymbol{\tau}$ is a vector of ones.

To facilitate the interpretation of results, in each experiment we computed a statistic of cross section dependence, the CD test (Pesaran, 2004), a statistic of local cross section correlation, the $CD(p)$, and the simple average of pair-wise cross section correlation coefficients of the residuals, $\bar{r}$. We have chosen these tests because they do not require the specification of a generating process for the error term. The CD statistic is

$$CD = \sqrt{\frac{2T}{N(N-1)}} \left(\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \hat{r}_{ij} \right),$$

where $\hat{r}_{ij}$ is the sample estimate of the pair-wise correlation of the residuals, specifically

Figure 10. Example of partial and missed detections of embedded formulas in a document image taken from the Marmot dataset. While green color highlights the correct predictions, partial and missed detections are marked with red and blue colors, respectively.

5. Conclusion and Future Work

We introduce an end-to-end trainable network for the detection of formulas in document images. Our proposed method follows high-level architectural principles of traditional object detection approaches. Specifically, it exploits dilated document images fed in Cascade Mask R-CNN equipped with a deformable composite dual backbone network. The proposed modifications help the network to achieve better generalization and detection performance. We achieve state-of-the-art performance on a higher IoU threshold with an f1-score of 0.917 on the ICDAR-2017 POD dataset. Furthermore, we reduce the relative error by 30% in detecting embedded formulas on the Marmot dataset with the correct detection accuracy of 81.3%. Not only do we improve the quantitative accuracy, but we also observe an outstanding improvement in terms of

false-positive rates. Moreover, the presented work empirically establishes that without relying on heavy pre-processing pipelines, it is possible to achieve a state-of-the-art formula detection system in scanned document images.

For future work, we expect that a deeper backbone would be able to perform better in terms of both IoU and false positives. Moreover, the experiments can be extended to detect various graphical page objects like figures, charts, titles, and headings in document images.

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

Funding: The work leading to this publication has been partially funded by the European project INFINITY under Grant Agreement ID 883293.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Not applicable.

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

References

1. Kieninger, T.; Dengel, A. The t-recs table recognition and analysis system. International Workshop on Document Analysis Systems. Springer, 1998, pp. 255–270.
2. Inoue, K.; Miyazaki, R.; Suzuki, M. Optical recognition of printed mathematical documents. Proc. Third Asian Technology Conf. Math, 1998, pp. 280–289.
3. 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. 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW). IEEE, 2019, Vol. 5, pp. 116–121.
4. Smith, R. An overview of the Tesseract OCR engine. Ninth international conference on document analysis and recognition (ICDAR 2007). IEEE, 2007, Vol. 2, pp. 629–633.
5. Azawi, M.A.; Afzal, M.Z.; Breuel, T.M. Normalizing historical orthography for OCR historical documents using LSTM. Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing, 2013, pp. 80–85.
6. Mokhtar, K.; Bukhari, S.S.; Dengel, A. OCR Error Correction: State-of-the-Art vs an NMT-based Approach. 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). IEEE, 2018, pp. 429–434.
7. Mahdavi, M.; Zanibbi, R.; Mouchere, H.; Viard-Gaudin, C.; Garain, U. ICDAR 2019 CROHME+ TFD: Competition on recognition of handwritten mathematical expressions and typeset formula detection. 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019, pp. 1533–1538.
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. *arXiv preprint arXiv:2104.14272* **2021**.
9. 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.
10. Phong, B.H.; Hoang, T.M.; Le, T.L. A hybrid method for mathematical expression detection in scientific document images. *IEEE Access* **2020**, *8*, 83663–83684.
11. Kacem, A.; Belaïd, A.; Ahmed, M.B. Automatic extraction of printed mathematical formulas using fuzzy logic and propagation of context. *International Journal on Document Analysis and Recognition* **2001**, *4*, 97–108.
12. Baker, J.B.; Sexton, A.P.; Sorge, V. Towards Reverse Engineering of PDF Documents. *DML 2011 Towards a Digital Mathematics Library* **2011**.
13. Jin, J.; Han, X.; Wang, Q. Mathematical Formulas Extraction. Icdar. Citeseer, 2003, pp. 1138–1141.
14. Chang, T.Y.; Takiguchi, Y.; Okada, M. Physical structure segmentation with projection profile for mathematic formulae and graphics in academic paper images. Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). IEEE, 2007, Vol. 2, pp. 1193–1197.
15. Liu, Y.; Bai, K.; Gao, L. An efficient pre-processing method to identify logical components from pdf documents. Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, 2011, pp. 500–511.
16. Ohyama, W.; Suzuki, M.; Uchida, S. Detecting mathematical expressions in scientific document images using a u-net trained on a diverse dataset. *IEEE Access* **2019**, *7*, 144030–144042.

17. Phong, B.H.; Dat, L.T.; Yen, N.T.; Hoang, T.M.; Le, T.L. A deep learning based system for mathematical expression detection and recognition in document images. 2020 12th International Conference on Knowledge and Systems Engineering (KSE). IEEE, 2020, pp. 85–90.
18. Mali, P.; Kukkadapu, P.; Mahdavi, M.; Zanibbi, R. ScanSSD: Scanning Single Shot Detector for Mathematical Formulas in PDF Document Images. *arXiv preprint arXiv:2003.08005* **2020**.
19. Younas, J.; Siddiqui, S.A.; Munir, M.; Malik, M.I.; Shafait, F.; Lukowicz, P.; Ahmed, S. Fi-Fo Detector: Figure and Formula Detection Using Deformable Networks. *Applied Sciences* **2020**, *10*, 6460.
20. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497* **2015**.
21. Huang, Y.; Yan, Q.; Li, Y.; Chen, Y.; Wang, X.; Gao, L.; Tang, Z. A YOLO-based table detection method. 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019, pp. 813–818.
22. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. European conference on computer vision. Springer, 2016, pp. 21–37.
23. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection, 2017, [arXiv:cs.CV/1612.03144].
24. Agarwal, M.; Mondal, A.; Jawahar, C. CDeC-Net: Composite Deformable Cascade Network for Table Detection in Document Images. *arXiv preprint arXiv:2008.10831* **2020**.
25. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6154–6162.
26. Gao, L.; Yi, X.; Jiang, Z.; Hao, L.; Tang, Z. ICDAR2017 competition on page object detection. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2017, Vol. 1, pp. 1417–1422.
27. Fang, J.; Tao, X.; Tang, Z.; Qiu, R.; Liu, Y. Dataset, ground-truth and performance metrics for table detection evaluation. 2012 10th IAPR International Workshop on Document Analysis Systems. IEEE, 2012, pp. 445–449.
28. Lin, X.; Gao, L.; Tang, Z.; Baker, J.; Sorge, V. Mathematical formula identification and performance evaluation in PDF documents. *International Journal on Document Analysis and Recognition (IJDAR)* **2014**, *17*, 239–255.
29. Fateman, R.J.; Tokuyasu, T.; Berman, B.P.; Mitchell, N. Optical character recognition and parsing of typeset mathematics1. *Journal of Visual Communication and Image Representation* **1996**, *7*, 2–15.
30. Lee, H.J.; Wang, J.S. Design of a mathematical expression understanding system. *Pattern Recognition Letters* **1997**, *18*, 289–298.
31. Toumit, J.Y.; Garcia-Salicetti, S.; Emptoz, H. A hierarchical and recursive model of mathematical expressions for automatic reading of mathematical documents. Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318). IEEE, 1999, pp. 119–122.
32. Garain, U.; Chaudhuri, B. A syntactic approach for processing mathematical expressions in printed documents. Proceedings 15th International Conference on Pattern Recognition. ICPR-2000. IEEE, 2000, Vol. 4, pp. 523–526.
33. Chowdhury, S.; Mandal, S.; Das, A.K.; Chanda, B. Automated segmentation of math-zones from document images. Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings. Citeseer, 2003, pp. 755–759.
34. Drake, D.M.; Baird, H.S. Distinguishing mathematics notation from English text using computational geometry. Eighth international conference on document analysis and recognition (ICDAR'05). IEEE, 2005, pp. 1270–1274.
35. 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.
36. He, W.; Luo, Y.; Yin, F.; Hu, H.; Han, J.; Ding, E.; Liu, C.L. Context-aware mathematical expression recognition: An end-to-end framework and a benchmark. 2016 23rd International Conference on Pattern Recognition (ICPR). IEEE, 2016, pp. 3246–3251.
37. Gao, L.; Yi, X.; Liao, Y.; Jiang, Z.; Yan, Z.; Tang, Z. A deep learning-based formula detection method for PDF documents. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2017, Vol. 1, pp. 553–558.
38. Yi, X.; Gao, L.; Liao, Y.; Zhang, X.; Liu, R.; Jiang, Z. CNN based page object detection in document images. 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2017, Vol. 1, pp. 230–235.
39. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.
40. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* **2018**.
41. Li, X.H.; Yin, F.; Liu, C.L. Page object detection from pdf document images by deep structured prediction and supervised clustering. 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018, pp. 3627–3632.
42. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks, 2017, [arXiv:cs.CV/1703.06211].
43. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.
44. Zhao, Z.Q.; Zheng, P.; Xu, S.t.; Wu, X. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems* **2019**, *30*, 3212–3232.
45. Liu, Y.; Wang, Y.; Wang, S.; Liang, T.; Zhao, Q.; Tang, Z.; Ling, H. Cbnet: A novel composite backbone network architecture for object detection. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, Vol. 34, pp. 11653–11660.
46. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. *arXiv preprint arXiv:1611.05431* **2016**.

47. Yosinski, J.; Clune, J.; Nguyen, A.; Fuchs, T.; Lipson, H. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579* **2015**.
48. 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.
49. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable convnets v2: More deformable, better results. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9308–9316.
50. Lin, X.; Gao, L.; Tang, Z.; Lin, X.; Hu, X. Performance evaluation of mathematical formula identification. 2012 10th IAPR International Workshop on Document Analysis Systems. IEEE, 2012, pp. 287–291.
51. 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**.
52. Lin, T.Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context, 2014. cite arxiv:1405.0312Comment: 1) updated annotation pipeline description and figures; 2) added new section describing datasets splits; 3) updated author list.
53. Powers, D.M. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv:2010.16061* **2020**.
54. Blaschko, M.B.; Lampert, C.H. Learning to localize objects with structured output regression. Eur. Conf. Comput. vision. Springer, 2008, pp. 2–15.
55. Chu, W.T.; Liu, F. Mathematical formula detection in heterogeneous document images. 2013 conference on technologies and applications of artificial intelligence. IEEE, 2013, pp. 140–145.