




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; didier.stricker@dfki.de

² Mindgarage, Technical University of Kaiserslautern, 67663 Kaiserslautern, Germany

³ German Research Institute for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany; alain.pagani@dfki.de

⁴ Department of Computer Science, Luleå University of Technology, 971 87 Luleå, Sweden; marcus.liwicki@ltu.se

* Correspondence: khurram_azeem.hashmi@dfki.de (K.A.H.); muhammad_zeshan.afzal@dfki.de (M.Z.A.)

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



Citation: 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. *Appl. Sci.* **2021**, *11*, 7610. <https://doi.org/10.3390/app11167610>

Academic Editor: Manuel Armada

Received: 5 July 2021

Accepted: 17 August 2021

Published: 19 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Information extraction from document images is a 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] (we use the LSTM-based version 4.1.1 available at <https://github.com/tesseract-ocr/tesseract> accessed on 5 July 2021), 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 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.

We state and prove next the new work decomposition laws.

THEOREM 5. (WORK DECOMPOSITION LAWS). Under any dynamic scheduling policy, and for any subset $S \subseteq N$ of job classes,

$$(30) \sum_{j \in S} V_j^S x_j = f(S) + \frac{1}{1 - \rho^0(S)} \sum_{i \in S^c} \sum_{j \in S} \lambda_i V_i^S V_j^S x_j^0 + \frac{1 - \rho}{1 - \rho^0(S)} \sum_{j \in S} V_j^S x_j^0.$$

(b) Identity (30) can be reformulated as

$$(31) E[V^S] = f(S) - \sum_{i \in S} \rho_i(\beta_i - r_i) - \frac{\rho^0(S)}{1 - \rho^0(S)} \sum_{i \in S^c} \rho_i r_i + \frac{1}{1 - \rho^0(S)} \sum_{i \in S^c} (\lambda_i V_i^S - \rho_i) E^S[V^S] + \frac{1 - \rho(S)}{1 - \rho^0(S)} E[V^S | B^S = 0].$$

PROOF.

(a) In what follows we use the following notation: if $S, T \subseteq N, z = (z_i)_{i \in N}$ is an n -vector, and $A = (a_{ij})_{i,j \in N}$ is an $n \times n$ matrix, we shall write

$$z_S = (z_i)_{i \in S} \quad \text{and} \quad A_{ST} = (a_{ij})_{i \in S, j \in T}.$$

Let v denote the n -vector

$$v = \begin{pmatrix} V^S \\ \mathbf{0} \end{pmatrix}.$$

(a) Sample input document image.

We state and prove next the new work decomposition laws.

THEOREM 5. (WORK DECOMPOSITION LAWS). Under any dynamic scheduling policy, and for any subset $S \subseteq N$ of job classes,

$$(30) Vix = f(S) + \frac{1}{1 - \rho^0(S)} \sum_{i \in S^c} \sum_{j \in S} \lambda_i V_i^S V_j^S x_j + \frac{1 - \rho}{1 - \rho^0(S)} \sum_{j \in S} V_j^S x_j^0.$$

(b) Identity (30) can be reformulated as

$$(31) ELV = AS - \sum_{i \in S} \rho_i(\beta_i - r_i) - \text{ip Den} + \frac{1 - \rho(S)}{1 - \rho^0(S)} E[V^S | B^S = 0].$$

PROOF.

(a) In what follows we use the following notation: if $S, T \subseteq N, z = (z_i)_{i \in N}$ is an n -vector,

and $A = (a_{ij})_{i,j \in N}$ is an $n \times n$ matrix, we shall write

$$z_S = (z_i)_{i \in S} \quad \text{and} \quad A_{ST} = (a_{ij})_{i \in S, j \in T}.$$

Let v denote the n -vector

(b) Extracted information after applying OCR.

Figure 1. Visual depiction of the need to apply formula detection before extracting information in document images. We apply open source Tesseract-OCR [4] on a document image taken from Marmot dataset [9] containing mathematical formulas as illustrated in (a). Besides the textual content, the OCR system fails miserably in recognizing information from formulas as depicted in (b).

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 [10]. 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 be misinterpreted with the regular text in a document image [11].

J.-S. Wang et al. Quantum thermal transport in nanostructures 385

where we define the local energy in cell l as

$$H_l = \frac{1}{2} (u_l^2 \ddot{u}_l + u_{l-1}^2 k_{1l} \dot{u}_l + u_l^2 k_{1l} \dot{u}_l + u_{l+1}^2 k_{2l} \dot{u}_l), \quad (19)$$

such that $\sum_l H_l = H_R$ (or H_L). By differentiating H_l with respect to time t and using the equation of motion, we can see that

$$\dot{H}_l = \frac{1}{2} (u_l^2 k_{1l} \ddot{u}_{l-1} - u_l^2 k_{1l} \ddot{u}_l) \quad (20)$$

satisfies the requirement. I_l is the energy current from cell $l - 1$ to cell l . Expressing a general vibration as superposition of modes with amplitudes $Q_{n,k}$,

$$u(t) = \frac{1}{\sqrt{2N}} \sum_n \sum_k Q_{n,k} e^{i(kn - \omega_n t)} + c.c., \quad (21)$$

where $c.c.$ stands for complex conjugate, and substituting it into equation (20), and performing a time average, we obtain

$$I_l = \frac{1}{4N} \sum_n \sum_k \omega_n Q_{n,k} \left(\frac{M_{l-1}}{2} k_{1l} - \frac{M_l}{2} k_{2l} \right) \dot{Q}_{n,k}, \quad (22)$$

In deriving the above expression, we used the fact that the time average of $e^{i(\omega_n t - \omega_n' t)}$ is zero, unless $n = n'$ and $k = k'$. The expression in the brackets can be further simplified in terms of the group velocity $v_{n,k} = \partial \omega_n / \partial k$. The final expression for the classical energy current in terms of the normal mode amplitudes is

$$I_l = \sum_n \sum_k \frac{v_{n,k}}{2N} \omega_n Q_{n,k} \dot{Q}_{n,k}. \quad (23)$$

where t is a matrix with elements $t_{n,n}^{m,m}$ where (n, n) are considered row index and (m, m) the column index. \tilde{v} is a diagonal matrix with the elements $v_n^m = v_n^m / \omega_n$ arranged in the same order as t, u_l and u_{l+1} are the lattice constants of the left and right lead, respectively. The matrix t is somewhat close to be unitary, but is not. If we define $S = S^L S^R U_L$, then S is unitary. From $S S^\dagger = S^\dagger S = I$, we can also show that

$$S^\dagger t = \tilde{v}^{-1} \tilde{v}^{-1} S^\dagger. \quad (24)$$

We now discuss the quantization of the problem. First, we consider only an isolated lead with periodic boundary conditions. Let us introduce the annihilation operator in Heisenberg picture $a_{n,k}(t) = a_{n,k} e^{-i\omega_n t}$ associated with mode (n, k) and its Hermitian conjugate $a_{n,k}^\dagger(t)$ for the creation operator, satisfying the usual commutation relations: $[a_{n,k}, a_{n',k'}] = 0$, $[a_{n,k}^\dagger, a_{n',k'}^\dagger] = 0$, and $[a_{n,k}, a_{n',k'}^\dagger] = \delta_{n,n'} \delta_{k,k'}$. Then the canonical coordinate operator $Q_{n,k}(t) = \sqrt{\frac{\hbar}{2m\omega_n}} [a_{n,k}(t) + a_{n,k}^\dagger(t)]. \quad (25)$

satisfying $[Q_{n,k}(t), Q_{n',k'}^\dagger(t)] = i\hbar \delta_{n,n'} \delta_{k,k'}$. Using the relation between the normal mode coordinates and the original coordinates, we can write

$$u(t) = \sum_n \sum_k \sqrt{\frac{\hbar}{2m\omega_n N}} e^{i(kn - \omega_n t)} a_{n,k}(t) + h.c., \quad (26)$$

(a) Defining isolated formula detection in a document image.

(b) Defining embedded formula detection in a document image.

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 highlighted in (a), spanning multiple lines, are prone to be misclassified with tables, whereas the embedded formulas depicted in (b) are confused with the regular text.

Previous works employed hand-crafted features to detect formulas in documents [2,12,13]. Although these systems extract mathematical formulas, they fail to obtain effective results on

denote the number of nonzero coordinates of β , where $\mathbb{1}_J$ denotes the indicator function $\mathbb{1}_J(\beta) = \{j \in \{1, \dots, M\} : \beta_j \neq 0\}$ and J denotes the cardinality of J . The value $|\mathbb{M}(\beta)|$ characterizes the sparsity of the vector β . The smaller $|\mathbb{M}(\beta)|$, the “sparser” β . For a vector $\beta \in \mathbb{R}^M$ and a subset $J \subseteq \{1, \dots, M\}$, we denote by β_J the vector in \mathbb{R}^M that has the same coordinates as β on J and zero coordinates on the complement J^c of J .

Introduce the residual sum of squares

$$\hat{S}(\beta) = \frac{1}{n} \sum_{i=1}^n \{Y_i - f_\beta(Z_i)\}^2$$

for all $\beta \in \mathbb{R}^M$. Define the Lasso solution $\hat{\beta}_L = (\hat{\beta}_{1,L}, \dots, \hat{\beta}_{M,L})$ by

$$(2.1) \quad \hat{\beta}_L = \arg \min_{\beta \in \mathbb{R}^M} \left\{ \hat{S}(\beta) + 2r \sum_{j=1}^M \|f_j\| |\beta_j| \right\},$$

where $r > 0$ is some tuning constant, and introduce the corresponding Lasso estimator

generic datasets. Later, statistical learning, mainly machine learning-based methods, advanced the performance of formula identification systems [14–16]. 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 [17–20] have been presented in the past two years. They are mainly equipped with object detection algorithms such as Faster R-CNN [21], YOLO [22], SSD [23], and FPNs [24].

In recent work, Agarwal et al. [25] presented a method equipped with Cascade Mask R-CNN [26] 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 a 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 a higher IoU threshold in the ICDAR-2017 POD dataset [27]. Furthermore, on the Mar-mot dataset [9], 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 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 developed 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. [12] introduced 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 a 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. [13] 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. [15] proposed a similar method based on the projection of the features that only works for isolated formulas in documents.

Later, machine learning-based algorithms were proposed to alleviate the performance of formula detection systems in documents [14,34]. Liu et al. [16] 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, researchers have investigated the capabilities of Deep Neural Networks (DNNs) for the problem of formula identification in document images [27,35]. 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.

NLPR-PAL [27] produced the best results in the competition of POD at ICDAR-2017. They proposed a blend of connected components, SVM, and Faster R-CNN [21] to detect figures, formulas, and tables in document images.

Yi et al. [38] published another CNN-based approach that detects graphical page objects such as 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 such as U-Net [39] has also been utilized to detect mathematical expressions in scientific document images [17].

Recently, Phong et al. [18] published a method equipped with YOLO [40] to detect mathematical formulas in document images. In another approach [19], SSD [23] was exploited to detect mathematical expressions in PDF documents.

Another graphical page object detection system was 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. [20] 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) [24] are 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.

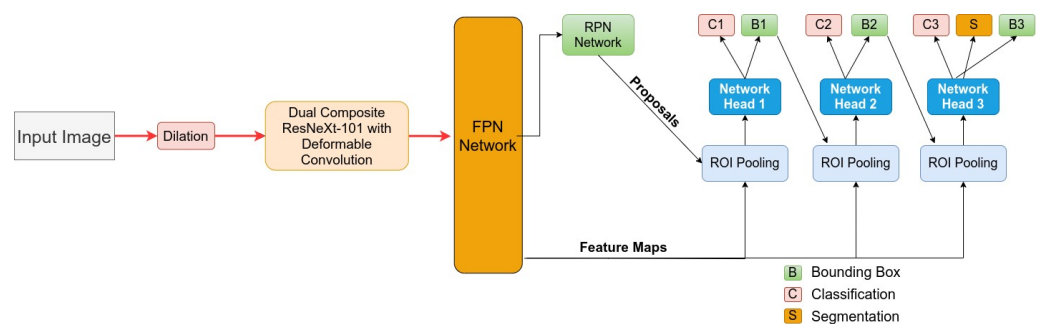


Figure 3. The presented framework is based on Cascade Mask R-CNN equipped with a deformable composite backbone applied on dilated document images. Modules B, C, and D represent bounding box, classification, and segmentation, respectively.

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 [26] introduced Cascade R-CNN [26] that extends the concept of the idea of Faster R-CNN [21] 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, BH_1 , BH_2 , and BH_3 , are denoted with (C_1, B_1) , (C_2, B_2) , and (C_3, B_3) , respectively. The output of 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. Figure 4 illustrates the architecture of our dual composite backbone.

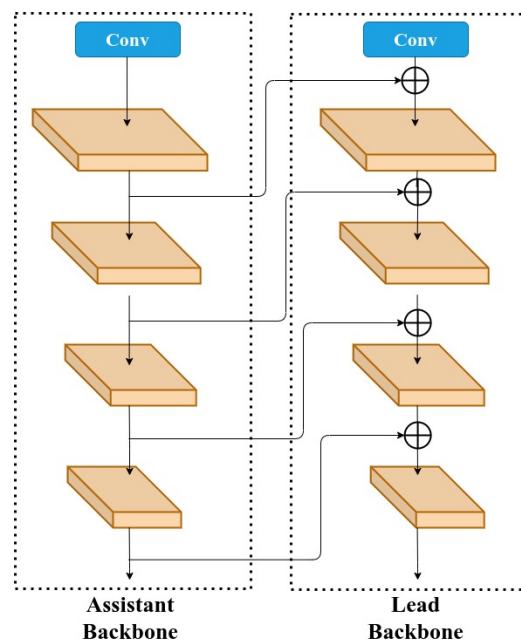


Figure 4. Visual explanation of the employed backbone (CBNet) in our framework. We utilize a dual ResNeXt-101 backbone, in which there are composite connections between parallel stages of the adjacent assistant and lead backbone. Moreover, we replace the conventional convolutions in ResNeXt101 with deformable 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 $(l - 1)$ th stage and parallel $l - th$ stage of assistant backbone ba . Mathematically, it is explained in [45] as

$$x_{bl}^l = F_{bl}^l((x_k^{l-1}) + g(x_{ba}^l)), l \geq 2 \tag{2}$$

where g defines the composite connection between the lead and assistant backbone, and these composite connections enable the lead backbone to extract essential spatial features. Table 1 outlines the architectural details of the employed dual ResNeXt-101 backbone

network. 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.

Table 1. Architectural details of the employed dual composite ResNeXt-101 backbone network. DCN represents the incorporation of deformable convolution.

| Stage | Output | DCN | ResNeXt-101 (32 × 4d) |
|-------|-----------|-----|---|
| conv1 | 112 × 112 | ✗ | 7 × 7, 64, stride 2 |
| | | - | 3 × 3 max pooling, stride 2 |
| conv2 | 56 × 56 | ✗ | 1 × 1, 128 |
| | | | 3 × 3, 128, C = 32 × 3 |
| | | | 1 × 1, 128 |
| conv3 | 28 × 28 | ✓ | 1 × 1, 256 |
| | | | 3 × 3, 256, C = 32 × 4 |
| | | | 1 × 1, 512 |
| conv4 | 14 × 14 | ✓ | 1 × 1, 512 |
| | | | 3 × 3, 512, C = 32 × 23 |
| | | | 1 × 1, 1024 |
| conv5 | 7 × 7 | ✓ | 1 × 1, 1024 |
| | | | 3 × 3, 1024, C = 32 × 3 |
| | | | 1 × 1, 2048 |
| | 1 × 1 | ✗ | global average pool 1000-d fc, softmax |

3.3. Deformable Convolution

We incorporate deformable convolution filters [42] instead of the conventional convolutions that exist in the ResNeXt-101 architecture [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 is comprised of two parts: (1) the first step samples the input feature map through a grid R , and (2) aggregation of sample values is multiplied by the weight \mathbf{w} . For conventional convolution, the output of feature map y for each position p_0 is elaborated in [42] as follows:

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

where \mathbf{x} represents the input feature map, and 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.

In the 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, Equation (3), as explained in [42], is transformed into

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

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

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

where \mathbf{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

$$\mathbf{G}(\mathbf{q}, \mathbf{p}) = \mathbf{g}(\mathbf{q}_x, \mathbf{p}_x) \times \mathbf{g}(\mathbf{q}_y, \mathbf{p}_y) \quad (6)$$

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 of deformable convolutions. Figure 5 depicts the architecture of deformable convolution. In order to convert our composite backbone network into its deformable counterparts, we replace conventional convolutional operation with deformable convolution in the higher-level layers that are from stage C_3 to stage C_5 . Table 1 highlights the presence of deformable convolution in the backbone network.

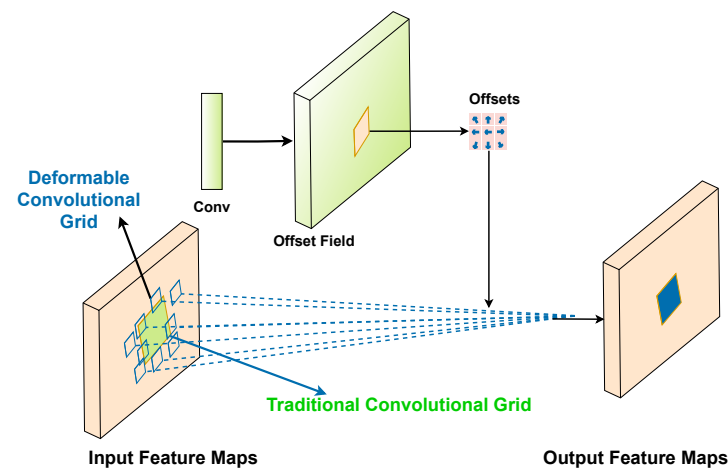


Figure 5. This figure illustrates a visual demonstration of a deformable convolution. The green grid depicts the conventional 3×3 convolutional operation, whereas the blue boxes highlight the effective receptive field of a similar 3×3 deformable convolution.

3.4. Image Transformation and Preprocessing

Document images mainly consist of textual regions. There exists 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 preprocessed images with the original images. This concatenation increases the number of input channels. The deep neural network processes this combination.

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.

$$f(\mathbf{w}; i_t) = \frac{\lambda}{2} \|\mathbf{w}\|^2 + \ell(\mathbf{w}; (\mathbf{x}_{i_t}, y_{i_t})) . \tag{3}$$

We consider the sub-gradient of the above approximate objective, given by:

$$\nabla_t = \lambda \mathbf{w}_t - \mathbb{1}[y_{i_t} \langle \mathbf{w}_t, \mathbf{x}_{i_t} \rangle < 1] y_{i_t} \mathbf{x}_{i_t} , \tag{4}$$

where $\mathbb{1}[y \langle \mathbf{w}, \mathbf{x} \rangle < 1]$ is the indicator function which takes a value of one if its argument is true (\mathbf{w} yields non-zero loss on the example (\mathbf{x}, y)), and zero otherwise. We then update $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \nabla_t$ using a step size of $\eta_t = 1/(\lambda t)$. Note that this update can be written as:

$$\mathbf{w}_{t+1} \leftarrow (1 - \frac{1}{t}) \mathbf{w}_t + \eta_t \mathbb{1}[y_{i_t} \langle \mathbf{w}_t, \mathbf{x}_{i_t} \rangle < 1] y_{i_t} \mathbf{x}_{i_t} . \tag{5}$$

After a predetermined number T of iterations, we output the last iterate \mathbf{w}_{T+1} . The pseudo-code of Pegasos is given in Fig. 1.

2.2 Incorporating a Projection Step

The above description of Pegasos is a verbatim application of the stochastic gradient-descent method. A potential variation is the gradient-projection approach where we limit the set of admissible solutions to the ball of radius $1/\sqrt{\lambda}$. To enforce this property, we project \mathbf{w}_t after each iteration onto this sphere by performing the update:

$$\mathbf{w}_{t+1} \leftarrow \min \left\{ 1, \frac{1/\sqrt{\lambda}}{\|\mathbf{w}_{t+1}\|} \right\} \mathbf{w}_{t+1} . \tag{6}$$

(a) Sample input document image.

$$f(\mathbf{w}; i_t) = \frac{\lambda}{2} \|\mathbf{w}\|^2 + \ell(\mathbf{w}; (\mathbf{x}_{i_t}, y_{i_t})) . \tag{3}$$

We consider the sub-gradient of the above approximate objective, given by:

$$\nabla_t = \lambda \mathbf{w}_t - \mathbb{1}[y_{i_t} \langle \mathbf{w}_t, \mathbf{x}_{i_t} \rangle < 1] y_{i_t} \mathbf{x}_{i_t} , \tag{4}$$

where $\mathbb{1}[y \langle \mathbf{w}, \mathbf{x} \rangle < 1]$ is the indicator function which takes a value of one if its argument is true (\mathbf{w} yields non-zero loss on the example (\mathbf{x}, y)), and zero otherwise. We then update $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t - \eta_t \nabla_t$ using a step size of $\eta_t = 1/(\lambda t)$. Note that this update can be written as:

$$\mathbf{w}_{t+1} \leftarrow (1 - \frac{1}{t}) \mathbf{w}_t + \eta_t \mathbb{1}[y_{i_t} \langle \mathbf{w}_t, \mathbf{x}_{i_t} \rangle < 1] y_{i_t} \mathbf{x}_{i_t} . \tag{5}$$

After a predetermined number T of iterations, we output the last iterate \mathbf{w}_{T+1} . The pseudo-code of Pegasos is given in Fig. 1.

2.2 Incorporating a Projection Step

The above description of Pegasos is a verbatim application of the stochastic gradient-descent method. A potential variation is the gradient-projection approach where we limit the set of admissible solutions to the ball of radius $1/\sqrt{\lambda}$. To enforce this property, we project \mathbf{w}_t after each iteration onto this sphere by performing the update:

$$\mathbf{w}_{t+1} \leftarrow \min \left\{ 1, \frac{1/\sqrt{\lambda}}{\|\mathbf{w}_{t+1}\|} \right\} \mathbf{w}_{t+1} . \tag{6}$$

(b) Processed image after dilation.

Figure 6. Visual comparison of a document image before (shown in (a)) and after the pre-processing method (depicted in (b)). The dilation process facilitates our feature extraction network by increasing the boundaries of foreground pixels, reducing the number of background pixels.

3.5. Datasets

We employed the well-known publicly available formula detection datasets to conduct our experiments. This section elaborates upon these datasets, and their summary is presented in Table 2.

Table 2. 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 |

3.5.1. ICDAR-17

ICDAR-17 is the result of a recent competition in graphical page object detection (POD) [27] 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. [20] 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.

3.5.2. Marmot

Marmot [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 there are 7907 embedded formulas with an average of almost 20 embedded formulas per document image.

4. Experimental Results

4.1. 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 the 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 eighth 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 [24] itself performs the multi-scale detection owing to its top-down architecture. We operated with a batch size of one to train our network. The models for both of the datasets are trained on an NVIDIA GeForce RTX 101 Ti GPU with 12 GB memory.

4.2. Evaluation Metrics

For ICDAR-2017 POD, we work with the same evaluation criteria as elaborated in the ICDAR-2017 POD competition [27]. For the Marmot dataset, we follow the identical criteria of computing detection accuracy as explained in [11] to have direct comparisons. We report results by employing the following metrics.

4.2.1. Precision

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

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (7)$$

4.2.2. Recall

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:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (8)$$

4.2.3. F1-Score

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

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

4.2.4. 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:

$$\text{mAP} = \frac{1}{N} \sum_{r=1}^N AP_r \quad (10)$$

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

4.2.5. 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:

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

4.2.6. Detection Accuracy

We report results on the Marmot dataset using the metrics of detection accuracy. As explained in [11], we classify the prediction into correct and partially 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 to 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.

4.3. Result and Discussion

We report the results on the datasets of ICDAR-2017 POD [27] and Marmot [9] 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.

4.3.1. ICDAR-17

We follow the evaluation protocol as elaborated in ICDAR-2017 POD [27]. 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 [20,41]. 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 [20]. Therefore, only the methods that have reported results on the corrected version of the dataset are directly comparable with our approach.

Table 3 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 of 0.948, f1-score of 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, and the presence of pre-processing advances the results with an average difference of 0.04. For the completeness of the paper, we compute the f1-score of the proposed method on different IoU thresholds ranging from 0.5 to 1.0. Figure 7 illustrates the performance of our approach in terms of f1-score.

Figures 8 and 9 depict 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 9.

Table 3. 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.

| ICDAR-2017 POD | | | | | | | | |
|---------------------------------------|-----------|--------|----------|--------|-----------|--------|----------|-------|
| Method | IoU = 0.6 | | | | IoU = 0.8 | | | |
| | Precision | Recall | F1-Score | AP | Precision | Recall | F1-Score | AP |
| NLPR-PAL [27] † | 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 [20] | 0.910 | 0.927 | 0.918 | 0.953 | 0.860 | 0.877 | 0.868 | 0.928 |
| Fi-Fo Detector Deformable [20] | 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 | 0.914 | 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 |

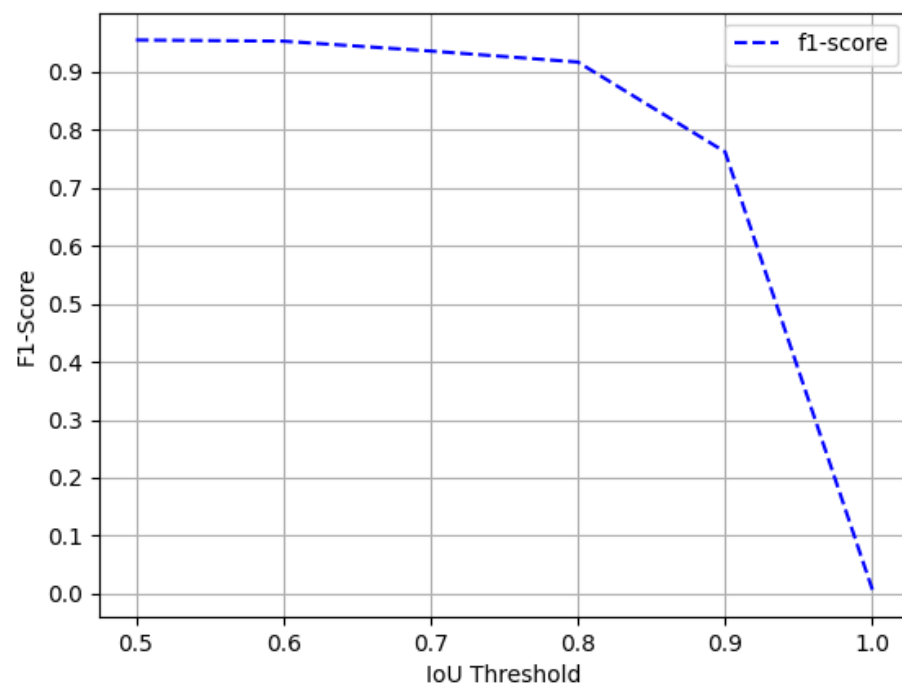


Figure 7. Performance evaluation in terms of f1-score over the varying IoU thresholds ranging from 0.5 to 1.0 on the ICDAR-2017-POD dataset.

Comparison with State-of-the-Art Methods

By looking at Table 3, 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 [20] 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 [20] depends on the heavy pre-processing pipeline consisting of distance transform and connected components analysis (CCA) applied on grayscale images. However, our generic data-driven method operates on the lightweight dilation technique to produce better results.

alternative x_i defeats (in pairwise comparison) alternative x_j ($h_{ij}^k = 1$) or not ($h_{ij}^k = 0$), i.e.,

$$h_{ij}^k = \begin{cases} 1 & \text{if } p_{ij}^k < 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

then, the degree of support to alternative x_j by individual e^k is calculated as follows,

$$v_j^k = \frac{1}{n-1} \sum_{i=1, i \neq j}^n h_{ij}^k \quad (19)$$

which is clearly the extent, from 0 to 1, to which individual e^k is not against alternative x_j ($h_{ij}^k = 0$: definitely not against, $h_{ij}^k = 1$: definitely against). Then we calculate the following averaging value

$$v_j = \frac{1}{m} \sum_{k=1}^m v_j^k \quad (20)$$

which expresses to what extent, from 0 to 1 as in (19), all the individuals are not against alternative x_j . Then a linguistic majority concept is used to approximate the final solutions,

$$v_Q = \mu_Q(h_j) \quad (21)$$

which is to what extent, from 0 to 1 as before, Q (say, *most*) individuals are not against alternative x_j , where Q is a linguistic quantifier “*most*” defined in Figure 4 (a). The final result (fuzzy- Q -core) is expressed as

$$\tilde{z}_Q = v_Q^1/x_1 + \dots + v_Q^n/x_n \quad (22)$$

which is interpreted as a fuzzy set of alternatives that are not defeated by Q (say, *most*) individuals. Similarly, fuzzy α/Q -core and fuzzy s/Q -core can also be obtained. The fuzzy α/Q -core is determined by updating (18) into

$$h_{ij}^k(\alpha) = \begin{cases} 1 & \text{if } p_{ij}^k < \alpha < 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

other steps remain the same. $(1 - \alpha)$ represents a degree of defeat to which x_i defeats x_j . The fuzzy α/Q -core obtained in this way can be interpreted as a fuzzy set of alternatives that are not sufficiently (at least to a degree $1 - \alpha$) defeated by Q (say, *most*) individuals. For fuzzy s/Q -core, only the equation (18) is updated into

$$h_{ij}^k = \begin{cases} 2(0.5 - p_{ij}^k) & \text{if } p_{ij}^k < 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

by introducing the strength of defeat. The obtained fuzzy s/Q -core can be interpreted as a fuzzy set of alternatives that are not strongly defeated by Q (say, *most*) individuals. In order to alleviate some “rigidness” of the conventional concept of consensus, i.e., full consensus occurs only when all the individual experts agree as to all the issues, a degree of consensus model was proposed by Kacprzyk [44],

then advanced by Fedrizzi and Kacprzyk [27], and Kacprzyk and Fedrizzi [45]. By introducing linguistic quantifiers into group decision making, in the new degree of consensus proposed in [47][46], full consensus may occur when most of the individual experts agree as to almost all (of the relevant) issues (alternatives, options).

Basically, the new degree of consensus is derived in three steps [47][46]:

- For each pair of individuals, a degree of agreement as to their preferences between all the pairs of alternatives is derived;
- These degrees are aggregated to obtain a degree of agreement of each pair of individuals as to their preferences between Q_1 (a linguistic quantifier, such as, “*most*”, “*almost all*”) pairs of relevant alternatives;
- The degrees of agreement resulting from step (2) are aggregated to obtain a degree of agreement of Q_2 (a linguistic quantifier similar to Q_1) pairs of important individuals as to their preferences between Q_1 pairs of relevant alternatives. This is the degree of consensus we seek.

Herrera-Viedma et al [38] proposed a group decision making framework which is different from the Kacprzyk-Fedrizzi-Numi’s scheme. Basically, this framework consists of four main processes:

1. *Resolution process* The goal of resolution process is to obtain a uniform representation of the preferences. Once the information is uniformed, a set of m individual fuzzy preference relations $\{P^1, \dots, P^m\}$ are available.
2. *Selection process* Selection process consists of two phases-*aggregation* and *exploitation*. The aggregation phase is to obtain a group collective preference relation $P^c = (p_{ij}^c)$ by means of the operator to aggregate all individual fuzzy preference relations $\{P^1, \dots, P^m\}$ and indicates the global preference between every ordered pair of alternatives according to the majority of experts’ opinions. Here, the aggregation operation is carried out by the linguistic quantifier guided OWA operator $\phi_Q(\cdot)$ [97][96].

In the exploitation phase, the group collective preference about the alternatives is transformed into a collective ranking of them, then a set of solution alternatives is obtained. The collective ranking is based on two choice degrees of alternatives: the *quantifier guided dominance degree (QGDD)* and the *quantifier guided nondominance degree (QGND)*, in which

$$QGDD_i = \phi_Q(p_{ij}^c | j = 1, \dots, n) \quad (25)$$

(a) True positives on a two-column document image.

Also, the functions I , J , and μ are multiplicative (verify). A useful property of the Möbius function is the following:

Theorem 2.17. For any multiplicative function f , if $n = p_1^{e_1} \dots p_r^{e_r}$ is the prime factorization of n , we have

$$\sum_{d|n} \mu(d)f(d) = (1 - f(p_1)) \dots (1 - f(p_r)). \quad (2.7)$$

In case $r = 0$ (i.e., $n = 1$), the product on the right-hand side of (2.7) is interpreted (as usual) as 1.

Proof. The non-zero terms in the sum on the left-hand side of (2.7) are those corresponding to divisors d of the form $p_{i_1} \dots p_{i_t}$, where p_{i_1}, \dots, p_{i_t} are distinct; the value contributed to the sum by such a term is $(-1)^t f(p_{i_1} \dots p_{i_t}) = (-1)^t f(p_{i_1}) \dots f(p_{i_t})$. These are the same as the terms in the expansion of the product on the right-hand side of (2.7). \square

For example, suppose $f(d) = 1/d$ in the above theorem, and let $n = p_1^{e_1} \dots p_r^{e_r}$ be the prime factorization of n . Then we obtain:

$$\sum_{d|n} \mu(d)/d = (1 - 1/p_1) \dots (1 - 1/p_r). \quad (2.8)$$

As another example, suppose $f = J$. Then we obtain

$$\mu \star J(n) = \sum_{d|n} \mu(d) = \prod_{i=1}^r (1 - 1)$$

which is 1 if $n = 1$, and is zero if $n > 1$. Thus, we have

$$\mu \star J = I. \quad (2.9)$$

Theorem 2.18 (Möbius Inversion Formula). Let f and F be arithmetic functions. Then we have $F = J \star f$ if and only if $f = \mu \star F$.

Proof. If $F = J \star f$, then

$$\mu \star F = \mu \star (J \star f) = (\mu \star J) \star f = I \star f = f$$

and conversely, if $f = \mu \star F$, then

$$J \star f = J \star (\mu \star F) = (J \star \mu) \star F = I \star F = F$$

\square

(b) True positives on a single column document image.

Figure 8. Performance evaluation of the proposed method on the ICDAR-2017-POD dataset. The green colour depicts the ground truth, while red denotes the predicted bounding boxes. (a,b) exhibit true positives on a two-column and a single column document image, respectively.

4.3.2. 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 4 summarizes the performance of our method on the Marmot dataset. As explained in Section 4.2.6, 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.

Besides calculating detection accuracy, we compute AP at an IoU threshold of 0.5 and 0.75 for both the isolated and embedded formula detection on the Marmot dataset. The achieved results are highlighted in Figure 10. Moreover, in Figure 11, we present the performance of correct detection accuracy over various IoU thresholds ranging from 0.5 to 1.0.

4.1 The Basic Euclidean Algorithm

57

$$\begin{aligned}
 a &= r_0 \\
 b &= r_1 \\
 r_0 &= r_1 q_1 + r_2 \quad (0 < r_2 < r_1) \\
 &\vdots \\
 r_{i-1} &= r_i q_i + r_{i+1} \quad (0 < r_{i+1} < r_i) \\
 &\vdots \\
 r_{\ell-2} &= r_{\ell-1} q_{\ell-1} + r_\ell \quad (0 < r_\ell < r_{\ell-1}) \\
 r_{\ell-1} &= r_\ell q_\ell \quad (r_{\ell+1} = 0)
 \end{aligned}$$

Note that by definition, $\ell = 0$ if $b = 0$, and $\ell > 0$, otherwise. Then we have $r_\ell = \gcd(a, b)$. Moreover, if $b > 0$, then $\ell \leq \log b / \log \phi + 1$, where $\phi := (1 + \sqrt{5})/2 \approx 1.62$.

Proof. For the first statement, one sees that for $i = 1, \dots, \ell$, we have $r_{i-1} = r_i q_i + r_{i+1}$, from which it follows that the common divisors of r_{i-1} and r_i are the same as the common divisors of r_i and r_{i+1} , and hence $\gcd(r_{i-1}, r_i) = \gcd(r_i, r_{i+1})$. From this, it follows that

$$\gcd(a, b) = \gcd(r_0, r_1) = \gcd(r_1, r_{i+1}) = \gcd(r_i, 0) = r_i$$

To prove the second statement, assume that $b > 0$, and hence $\ell > 0$. If $\ell = 1$, the statement is obviously true, so assume $\ell > 1$. We claim that for $i = 0, \dots, \ell - 1$, we have $r_{\ell-i} \geq \phi^i$. The statement will then follow by setting $i = \ell - 1$ and taking logarithms.

We now prove the above claim. For $i = 0$ and $i = 1$, we have

$$r_i > 1 = \phi^0 \quad \text{and} \quad r_{i-1} > r_i + 1 > 2 > \phi^1$$

For $i = 2, \dots, \ell - 1$, using induction and applying the fact the $\phi^2 = \phi + 1$, we have

$$r_{\ell-i} \geq r_{\ell-(i-1)} + r_{\ell-(i-2)} \geq \phi^{i-1} + \phi^{i-2} = \phi^{i-2}(1 + \phi) = \phi^i$$

which proves the claim. \square

Example 4.1. Suppose $a = 100$ and $b = 35$. Then the numbers appearing in Theorem 4.1 are easily computed as follows:

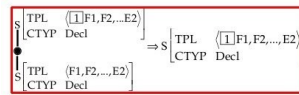
| | | | | | |
|-------|-----|----|----|---|---|
| i | 0 | 1 | 2 | 3 | 4 |
| r_i | 100 | 35 | 30 | 5 | 0 |
| q_i | | | 2 | 1 | 6 |

So we have $\gcd(a, b) = r_3 = 5$.

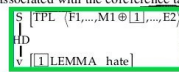
(a) True positives and false positives.

(b) False positives and false negatives.

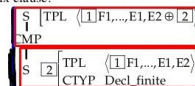
Figure 9. Performance evaluation of the proposed method on the ICDAR-2017-POD dataset. The green colour depicts the ground truth, while red denotes the predicted bounding boxes. (a) represents an example of both true positives and a single false positive case, whereas (b) shows three false positives with one false negative.



Filling a slot also involves coreference tags. For example, the HeadS of English verb frames obtain their position in the local topology by looking up the slot associated with the coreference tag:



The information associated with the foot node of the HeadS segment will now be appended to the current content, if any, of slot M1. The same mechanism serves to allocate the finite complement clause (or rather its root S-node) to slot E2 of the matrix clause:



Other clause constituents receive their landing site (cf. Table 1) in a similar manner. Figure 2 depicts the configuration after Fronting of NP Kim.

Figure 3 below includes a paraphrase where the focus on Kim is stressed prosodically rather than by Fronting. This is indicated by the disjunctive set carrying the tag [4]. In sentence generation, the Read-out module selects one alternative, presumably in response to pragmatic or other context factors. In parsing mode, one or the alternatives is ruled out because it does not match word order in the input string.

The formalism defined so far yields unordered hierarchical structures. However, the values of the TPL features enable the derivation of ordered output strings of lexical items. As indicated above in connection with Figure 2, we assume that this task can be delegated to a simple Read-out module that traverses the clause hierarchy in a depth-first manner and processes the topologies from left to right. If a slot is empty, the Reader jumps to the next

² A slot may contain more than one phrase (e.g., Direct and Indirect OBJECT in slot M3; cf. Table 1). We assume they have been ordered as part of the append operation, according to the sorting rule associated with the slot.

slot. If a slot contains a lexical item, it is appended to the current output string and tagged as already processed. It follows that, if a slot happens to be shared with a lower topology, its contents are only processed at the higher clause level, i.e., undergo promotion.

4. Linearization of complement clauses in English, Dutch and German

The PG formalism developed above provides a simple quantitative linearization method capturing both within-clause and between-clause phenomena. The assignment of constituents to topology slots (including, e.g., scrambling in Dutch and German) has been dealt with in Kempen & Harbusch (in press; forthcoming). In the present paper we focus on promotion in complement constructions—a domain where the three target languages exhibit rather dissimilar ordering patterns. We highlight the fact that PG enables highly similar treatments of them, differing only with respect to the settings of some quantitative parameters.

The movement (promotion) phenomena at issue here depend primarily on the values assigned to sharing parameters LS and RS in five different types of complement clauses. These settings are shown Table 2. They are imported from the lexicon and control the instantiation of the TPL feature of the root S-node of the complement. We begin with some illustrations from English.

Table 2. Size of the left- and right-peripheral shared topology areas (LS and RS) in diverse complement constructions.

| Clause type | English | Dutch | German |
|--|--------------|----------------|----------------|
| Interrogative | LS=0 RS=0 | LS=0 RS=1 | LS=0 RS=1 |
| Declarative & Finite | LS=1 RS=0 | LS=1 RS=1 | LS=1 RS=1 |
| Decl. & Non-Finite, VP Extraposition | LS=3 RS=0 | LS=1 RS=1 | LS=1 RS=1 |
| Decl. & Non-Finite, Verb Raising | LS=3 RS=0 | LS=3 RS=1 | LS=5 RS=1 |
| Decl. & Non-Finite, Third Construction | n.a. | LS=1:6 RS=1 | LS=1:6 RS=1 |

The non-finite complements of do and have in sentence (1) below are both declarative. (Cf. the paraphrase "For which person x is it the case that I have to call x", which highlights the scope of who.) It follows that LS=3 in both complements. Notice that do is treated as a "Verb Raiser", have (in have to) as a VP Extraposition verb.

Table 4. 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. [11] | Isolated | 50.37 | 39.14 | 91.18 |
| | Embedded | 22.9 | 58.45 | 81.35 |
| Phong et al. [18] | 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 |

The qualitative performance analysis of the presented method on the Marmot dataset is exhibited in Figures 12–14. 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.

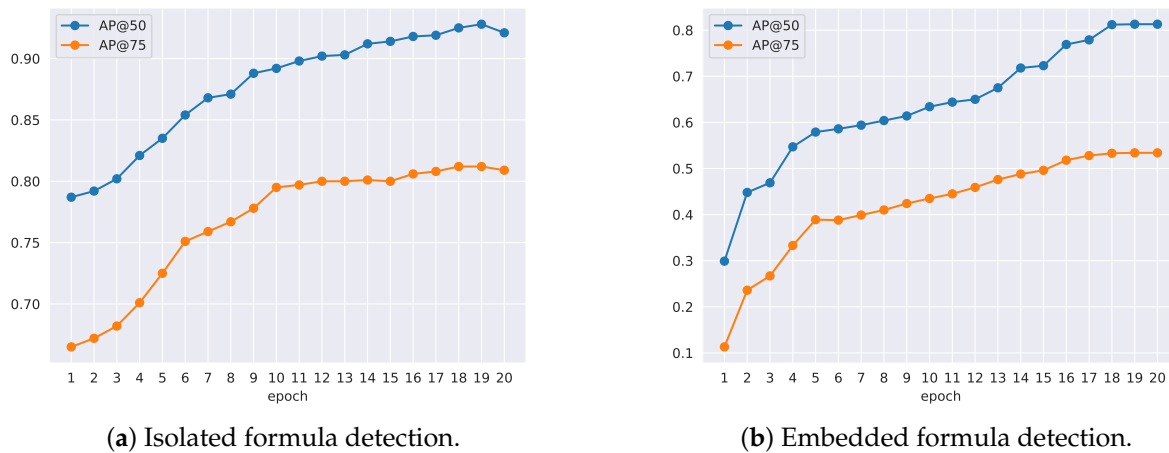


Figure 10. Performance evaluation of the proposed method on the Marmot dataset in terms of average precision (AP) at an IoU threshold of 0.5 and 0.75. (a) represents the the evolution of AP on isolated formulas, whereas (b) exhibits the the evolution of AP on embedded formulas.

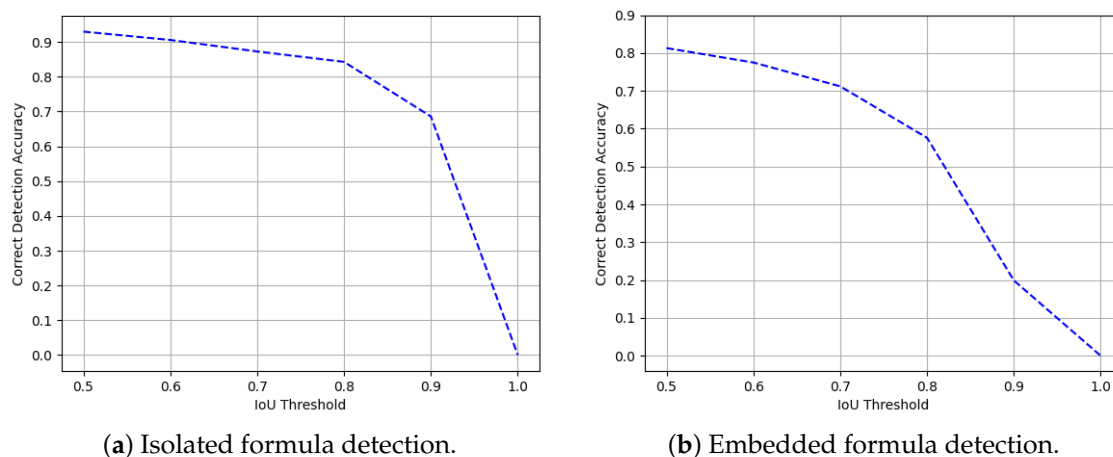


Figure 11. Correct detection accuracy achieved on varying IoU threshold ranging from 0.5 to 1.0 on the Marmot dataset. (a) depicts the correct detection accuracy on isolated formulas, whereas (b) demonstrates the correct detection accuracy on embedded formulas.

Comparison with State-of-the-Art Methods

We compare our results with earlier approaches on the Marmot dataset in Table 4. From the table, It is evident that our cascade network with a deformable composite backbone has clearly outsmarted the prior state-of-the-art method [18] in detecting embedded formulas while accomplishing identical results in the case of isolated formulas. We reduce the relative error of 30% by achieving a 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 97.86%. For embedded formulas, we achieve a partial detection accuracy of 6.77%, which adds up to an 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 missed detections in isolated and embedded formulas demonstrates the superiority of the proposed method.

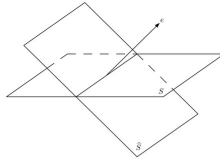


Figure 3: Illustration of \hat{S} , the orthogonal complement of e in $S \oplus e$, i.e., $\hat{S} = (S \oplus e) \cap e^\perp$.

With the TD(λ) method, we solve a projected form of the multistep Bellman equation

$$I - P^{(\alpha, \lambda)} = (1 - \lambda) \sum_{j=1}^m \lambda^j (\alpha P^j)^{j-1}, \quad b = \sum_{j=1}^m \lambda^j (\alpha P^j) g$$

where the matrix A and the vector b are defined for a pair of values (α, λ) by

respectively, with either $\alpha \in [0, 1], \lambda \in [0, 1]$, or $\alpha = 1, \lambda \in [0, 1]$. Notice that the case $\lambda = 0$ corresponds to $A = \alpha P, b = g$.

We note that for TD(λ) with $\lambda > 0$, we do not yet have an efficient simulation-based method for estimating the bound of Theorem 2; we have calculated the bound using common matrix algebra, and we plot it just for comparison.

Discounted Problems
Consider the discounted case: $\alpha < 1$. For $\lambda \in [0, 1]$, with ξ being the invariant distribution of the Markov chain, the modulus of contraction of $P^{(\alpha, \lambda)}$ with respect to $\|\cdot\|_\xi$ is

$$\rho^{(\alpha, \lambda)}_\xi = \frac{(1 - \lambda)\alpha}{1 - \lambda\alpha}$$

Let e denote the constant vector of all ones. Like P , the matrix $P^{(\alpha, \lambda)}$ has e as an eigenvector associated with the dominant eigenvalue $\frac{1 - \lambda\alpha}{1 - \lambda\alpha}$.

If the approximation subspace S contains or nearly contains e , the bound of Theorem 1 can degrade to the worst case error bound given by (2), as remarked in Section 2.2. In such a case, in order to have a sharper bound for the approximation of Πx^* , we can estimate separately the projection of x^* on e and the projection of x^* on another subspace $\hat{S} = (S \oplus e) \cap e^\perp$, which is the orthogonal complement of e in $S \oplus e$ (see Figure 3), and redefine \hat{x} as the sum of the two estimates. When the first projection can be estimated with no bias, the error bound for the second projection carries over to the combined estimate \hat{x} . This is true generally, not only for e , but for any eigenspace of P replacing e , as discussed in Section 2.2, Prop. 2 and Remark 4. In the case here, with ξ being the invariant distribution of the Markov chain, the projection of x^* on e can be calculated asymptotically exactly through simulation. It can be seen that the projection of x^* on e equals

$$\xi^* x^* = \xi^* b + \xi^* P^{(\alpha, \lambda)} x^* = \xi^* b + \frac{(1 - \lambda)\alpha}{1 - \lambda\alpha} \xi^* x^*, \quad \Rightarrow \quad \xi^* x^* = \frac{1 - \lambda\alpha}{1 - \lambda\alpha} \xi^* b$$

(a) Couple of samples illustrating correct detections.

From the above relation, the following inequalities can be established

$$\lambda_m(\Sigma_t) \leq \lambda_{m-1}(\Sigma_t) < \lambda_{m-2}(\Sigma_t) < \dots < \lambda_2(\Sigma_t^{m-2}) \leq \lambda_1(\Sigma_t^{m-1}) \quad (25)$$

$$\lambda_{m+1}(\Sigma_t) \leq \lambda_m(\Sigma_t) \leq \lambda_{m-1}(\Sigma_t) \leq \dots \leq \lambda_1(\Sigma_t^{m-1}) \leq \lambda_1(\Sigma_t^m) \quad (26)$$

$$\lambda_1(\Sigma_t^{m-1}) \leq \dots \leq \lambda_1(\Sigma_t) \leq \lambda_1(\Sigma_t) \leq \lambda_1(\Sigma_t) \quad (27)$$

Also, recall that

$$\lambda_1(\Sigma_t) \leq \|\Sigma_t\| \quad (28)$$

Suppose first that $\lambda_m(\Sigma_t) = O(N)$. Then from (25) $\lambda_1(\Sigma_t^{m-1})$ is also unbounded, implying from (28) that $\|\Sigma_t^{m-1}\| = O(N)$. Hence, Σ_t has at least m dominant units, which proves (i). Vice versa, suppose that Σ_t has m dominant units. Then we know from (24) that Σ_t has at least one eigenvalue unbounded in N . Further, note that, by the definition of dominant units, $\|\Sigma_t^m\|_\infty$ is bounded, and hence, from (28), $\lambda_1(\Sigma_t^m)$ is also bounded. From (26) it follows that $\lambda_{m+1}(\Sigma_t)$ is bounded, which proves (ii). ■

Note that in several cases the number of column vectors in Σ_t having unbounded sums largely exceeds the number of unbounded characteristic roots. In the extreme case, Σ_t could have N dominant units, with only one eigenvalue exploding to infinity, as in the following example of equiproportion

$$\Sigma_t = \sigma^2 \begin{pmatrix} 1 & \theta & \dots & \theta & \theta \\ \theta & 1 & \dots & \theta & \theta \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \theta & \theta & \dots & 1 & \theta \\ \theta & \theta & \dots & \theta & 1 \end{pmatrix}, \quad (29)$$

where $|\theta| < 1$. In this case all column sums are unbounded. However, the characteristic roots of the above matrix are

$$\lambda_1(\Sigma_t) = \sigma^2 [1 + (N-1)\theta]$$

$$\lambda_j(\Sigma_t) = \sigma^2 (1 - \theta) \quad \text{for } j = 2, \dots, N$$

namely only the largest eigenvalue is unbounded in N . In the next section we will see that processes with a covariance matrix like (29) can be well represented by the means of common factor models.

5 Common factor models

Originally proposed in the psychometric literature (Spearman, 1904), factor models are extensively used in macroeconomics and finance to represent the evolution of large cross sectional samples with strong co-movements. Panels with common factors have been applied to characterize the dynamic of stock and bond returns (Chamberlain and Rothschild 1983; Connor and Korajczyk, 1993; Kapetanios and Pesaran, 2007), and in macroeconomics to summarize the empirical content

(b) Couple of samples exhibiting partial detections.

Figure 12. Instances of correct and partial detection of isolated formulas on the Marmot dataset. The green color represents the correct detections, whereas the partial and missed detections are highlighted with red and blue colors, respectively. (a) depicts a couple of samples of correct detection in which an IoU score between ground truth and predicted region is greater than or equal to 0.5, whereas (b) illustrates a few cases of partial and missed detection.

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 $w_i = \frac{1}{N}, i = 1, \dots, N$. We further considered a misspecified structure that ignores the presence of common factors and/or spatial correlations, i.e. the fixed effects estimator

$$P_{FE} = \left(\sum_{i=1}^N X_i' M_i X_i \right)^{-1} \sum_{i=1}^N X_i' M_i y_i \quad (44)$$

where $M_i = I_T - \tau(\tau'\tau)^{-1}\tau'$, and τ 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

$$\hat{r}_{ij} = \frac{\sum_{t=1}^T \hat{u}_{it} \hat{u}_{jt}}{\left(\sum_{t=1}^T \hat{u}_{it}^2 \right)^{1/2} \left(\sum_{t=1}^T \hat{u}_{jt}^2 \right)^{1/2}}$$

and \hat{u}_{it} is an estimate of the regression residuals $u_{it} = y_{it} - \alpha_i d_{it} - \beta' X_{it}$, using the pooled estimator $\hat{\beta}_p$ of β . Pesaran (2004) has shown that the CD test is suitable under global alternatives such as the multi-factor residual models. However, when the cross section units can be ordered, it is more appropriate to compute the following CD(p) test statistic

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

where p is the order of the spatial weight matrix. Finally, the average of pair-wise cross section correlation coefficients is

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

This Monte Carlo study is intended to investigate the relationship between the small sample properties of a number of estimators and the source of cross section dependence. In addition, this analysis provides interesting results for a number of issues. First, the performance of the fixed effects estimator

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

Proof: This follows by observing that for $\kappa \leq \kappa_L$, 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 $\kappa \in (\kappa_L, \kappa_H)$, $\underline{v} < v_D^* < \bar{v} = v_R^*$.

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

$$\frac{1}{b} - \xi [-\kappa + \underline{v} - \bar{v}] (1 + T_v(\bar{v})) + \xi [\Delta + \bar{v} - \underline{v}]$$

$$\frac{1}{2} - \xi [-\kappa_H + \underline{v} - \bar{v}] (1 + T_v(\bar{v})) + \xi \Delta = 0$$

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

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

$$\frac{1}{2} - \xi \kappa (1 + T_v(\bar{v})) + \xi \Delta < \frac{(1 + T_v(\bar{v}))}{2} + \xi \Delta < 0$$

where the last inequality follows from Assumption 2. Thus, the best response for the Democrats must be $v_D < \bar{v}$. To see that $v_D > \underline{v}$, observe that $1 + T_v(\underline{v}) = 0$. To prove the last statement, observe that $v_D(\bar{v})$ is defined from:

$$-\frac{1}{2} + \xi [\kappa + v_D(\bar{v}, \kappa) - \bar{v}] ((1 + T_v(v_D(\bar{v}, \kappa))))$$

$$= \xi [\Delta + v_D(\bar{v}, \kappa) + T(v_D(\bar{v}, \kappa)) - \bar{v}] \quad (8)$$

At any point where this equality holds, $((1 + T_v(v_D(\bar{v}, \kappa))) < 0$. Moreover, a maximum exists on $[\underline{v}, \bar{v}]$. Elementary arguments now show that, at any point satisfying (8), $v_D(\bar{v}, \kappa)$ is increasing in κ . ■

Lemma A4: There exists $\kappa > \kappa_H$, for which we have an interior equilibrium with $v_p^* \in (\underline{v}, \bar{v})$ for $p \in \{D, R\}$.

Lemma A2: If $\kappa < \kappa_T$ the Democratic party wins for sure and picks $\tau = v_D = v$

Proof: This follows by observing that for $\kappa < \kappa_T$ 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 $\kappa \in (\kappa_L, \kappa_H)$, $v < v_D^* < \bar{v} = v_R^*$

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

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

from the definition of κ_T . 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 $v_D = v$. Then, a small increase in v_T alters the Democratic payoff by:

Figure 13. 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.

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 $w_i = \frac{1}{N}$, $i = 1, \dots, N$. 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 $\mathbf{M}_\tau = \mathbf{I}_T - \tau(\tau'\tau)^{-1}\tau'$ and $\mathbf{1}$ 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 14. 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. Conclusions 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 into a 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 a 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 such as figures, charts, titles, and headings in document images.

Author Contributions: Writing—original draft preparation, K.A.H.; writing—review and editing, K.A.H. and M.Z.A.; supervision, editing, and project administration, M.L., D.S. and 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.

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

References

1. Kieninger, T.; Dengel, A. The t-recs table recognition and analysis system. In Proceedings of the International Workshop on Document Analysis Systems, Nagano, Japan, 4–6 November 1998; Springer: Berlin/Heidelberg, Germany, 1998; pp. 255–270.
2. Inoue, K.; Miyazaki, R.; Suzuki, M. Optical recognition of printed mathematical documents. *Proc. Third Asian Technol. Conf. Math* **1998**, *3*, 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. In Proceedings of the 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), Sydney, Australia, 20–25 September 2019; Volume 5, pp. 116–121.
4. Smith, R. An overview of the Tesseract OCR engine. In Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, Brazil, 23–26 September 2007; Volume 2, pp. 629–633.
5. Ahmad, R.; Afzal, M.Z.; Rashid, S.F.; Liwicki, M.; Breuel, T. Scale and rotation invariant OCR for Pashto cursive script using MDLSTM network. In Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR), Nancy, France, 23–26 August 2015; pp. 1101–1105.
6. Mokhtar, K.; Bukhari, S.S.; Dengel, A. OCR Error Correction: State-of-the-Art vs an NMT-based Approach. In Proceedings of the 13th IAPR International Workshop on Document Analysis Systems (DAS), Vienna, Austria, 24–27 April 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. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 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. *IEEE Access* **2021**, *9*, 87663–87685. [[CrossRef](#)]
9. Fang, J.; Tao, X.; Tang, Z.; Qiu, R.; Liu, Y. Dataset, ground-truth and performance metrics for table detection evaluation. In Proceedings of the 2012 10th IAPR International Workshop on Document Analysis Systems, Gold Coast, QLD, Australia, 27–29 March 2012; pp. 445–449.
10. Chen, K.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Shi, J.; Ouyang, W.; et al. Hybrid task cascade for instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4974–4983.
11. 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. [[CrossRef](#)]
12. Kacem, A.; Belaïd, A.; Ahmed, M.B. Automatic extraction of printed mathematical formulas using fuzzy logic and propagation of context. *Int. J. Doc. Anal. Recognit.* **2001**, *4*, 97–108. [[CrossRef](#)]
13. Baker, J.B.; Sexton, A.P.; Sorge, V. Towards Reverse Engineering of PDF Documents. *DML Towards Digit. Math. Libr.* **2011**, *4*, 65–75.

14. Jin, J.; Han, X.; Wang, Q. *Mathematical Formulas Extraction*; Icdar. Citeseer: Edinburgh, UK, 2003; pp. 1138–1141.
15. Chang, T.Y.; Takiguchi, Y.; Okada, M. Physical structure segmentation with projection profile for mathematic formulae and graphics in academic paper images. In Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), Curitiba, State of Paraná, Brazil, 23–26 September 2007; Volume 2, pp. 1193–1197.
16. Liu, Y.; Bai, K.; Gao, L. An efficient pre-processing method to identify logical components from pdf documents. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 500–511.
17. 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. [[CrossRef](#)]
18. 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. In Proceedings of the 12th International Conference on Knowledge and Systems Engineering (KSE), Can Tho City, Vietnam, 12–14 November 2020; pp. 85–90.
19. Mali, P.; Kukkadapu, P.; Mahdavi, M.; Zanibbi, R. ScanSSD: Scanning Single Shot Detector for Mathematical Formulas in PDF Document Images. *arXiv* **2020**, arXiv:2003.08005.
20. 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. *Appl. Sci.* **2020**, *10*, 6460. [[CrossRef](#)]
21. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497.
22. Huang, Y.; Yan, Q.; Li, Y.; Chen, Y.; Wang, X.; Gao, L.; Tang, Z. A YOLO-based table detection method. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 813–818.
23. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
24. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. *arXiv* **2017**, arXiv:1612.03144.
25. Agarwal, M.; Mondal, A.; Jawahar, C. CDeC-Net: Composite Deformable Cascade Network for Table Detection in Document Images. *arXiv* **2020**, arXiv:2008.10831.
26. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah, USA, 18–22 June 2018; pp. 6154–6162.
27. Gao, L.; Yi, X.; Jiang, Z.; Hao, L.; Tang, Z. ICDAR2017 competition on page object detection. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 10–15 November 2017; Volume 1, pp. 1417–1422.
28. Lin, X.; Gao, L.; Tang, Z.; Baker, J.; Sorge, V. Mathematical formula identification and performance evaluation in PDF documents. *Int. J. Doc. Anal. Recognit.* **2014**, *17*, 239–255. [[CrossRef](#)]
29. Fateman, R.J.; Tokuyasu, T.; Berman, B.P.; Mitchell, N. Optical character recognition and parsing of typeset mathematics1. *J. Vis. Commun. Image Represent.* **1996**, *7*, 2–15. [[CrossRef](#)]
30. Lee, H.J.; Wang, J.S. Design of a mathematical expression understanding system. *Pattern Recognit. Lett.* **1997**, *18*, 289–298. [[CrossRef](#)]
31. Toumit, J.Y.; Garcia-Salicetti, S.; Emptoz, H. A hierarchical and recursive model of mathematical expressions for automatic reading of mathematical documents. In Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318), Bangalore, India, 20–22 September 1999; pp. 119–122.
32. Garain, U.; Chaudhuri, B. A syntactic approach for processing mathematical expressions in printed documents. In Proceedings of the 15th International Conference on Pattern Recognition. ICPR-2000, Barcelona, Spain, 3–7 September 2000; Volume 4, pp. 523–526.
33. Chowdhury, S.; Mandal, S.; Das, A.K.; Chanda, B. Automated segmentation of math-zones from document images. In Proceedings of the Seventh International Conference on Document Analysis and Recognition, Edinburgh, UK, 3–6 August 2003; Citeseer: Princeton, NJ, USA, 2003; pp. 755–759.
34. Drake, D.M.; Baird, H.S. Distinguishing mathematics notation from English text using computational geometry. In Proceedings of the Eighth International Conference on Document Analysis and Recognition (ICDAR'05), Seoul, Korea, 29 September–1 August 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. *Appl. Sci.* **2021**, *11*, 5344. [[CrossRef](#)]
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. In Proceedings of the 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 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. In Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 10–15 November 2017; Volume 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. In Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 10–15 November 2017; Volume 1, pp. 230–235.
39. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Strasbourg, France, 27 September–1 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
40. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
41. Li, X.H.; Yin, F.; Liu, C.L. Page object detection from pdf document images by deep structured prediction and supervised clustering. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018; pp. 3627–3632.
42. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. *arXiv* **2017**, arXiv:1703.06211.
43. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
44. Zhao, Z.Q.; Zheng, P.; Xu, S.T.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Networks Learn. Syst.* **2019**, *30*, 3212–3232. [[CrossRef](#)]
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. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 11653–11660.
46. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. *arXiv* **2016**, arXiv:1611.05431.
47. Yosinski, J.; Clune, J.; Nguyen, A.; Fuchs, T.; Lipson, H. Understanding neural networks through deep visualization. *arXiv* **2015**, arXiv:1506.06579.
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. [[CrossRef](#)]
49. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–21 June 2019; pp. 9308–9316.
50. Lin, X.; Gao, L.; Tang, Z.; Lin, X.; Hu, X. Performance evaluation of mathematical formula identification. In Proceedings of the 10th IAPR International Workshop on Document Analysis Systems, Queensland, Australia, 27–29 March 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.; et al. MMDetection: Open MMLab Detection Toolbox and Benchmark. *arXiv* **2019**, arXiv:1906.07155.
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. *arxiv* **2014**, arxiv:1405.0312.
53. Powers, D.M. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2020**, arXiv:2010.16061.
54. Blaschko, M.B.; Lampert, C.H. Learning to localize objects with structured output regression. In Proceedings of the European Conference on Computer Vision, Marseille, France, 12–16 October 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 2–15.
55. Chu, W.T.; Liu, F. Mathematical formula detection in heterogeneous document images. In Proceedings of the 2013 Conference on Technologies and Applications of Artificial Intelligence, Taipei, Taiwan, 6–8 December 2013; pp. 140–145.