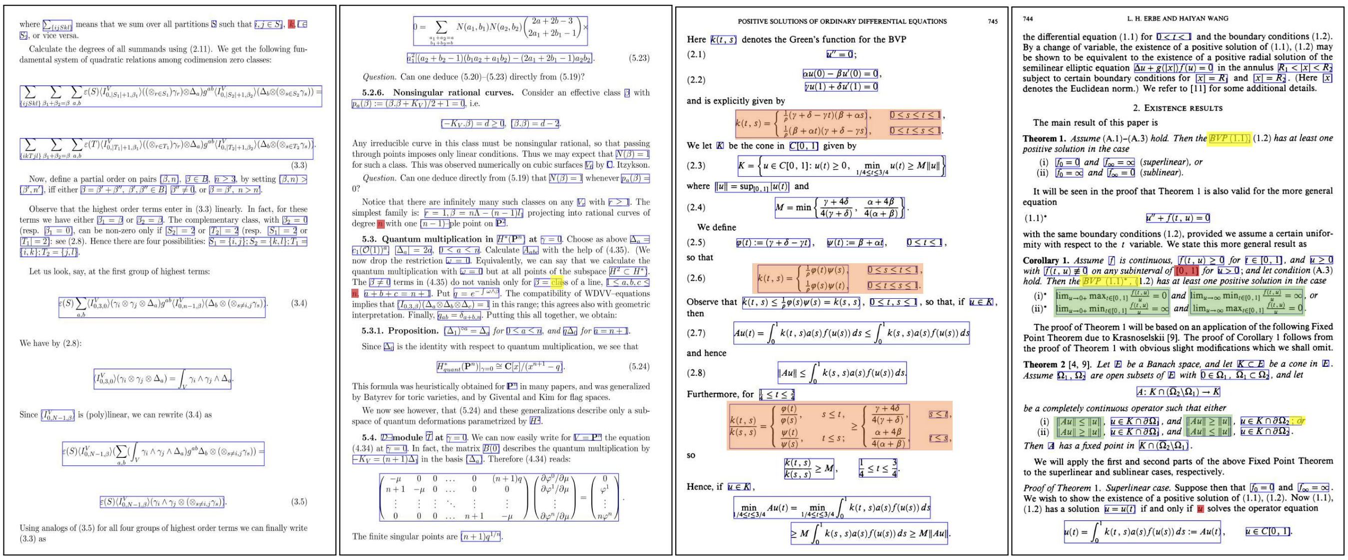

Fig. 6: Detection results. Detected formulas are shown as blue bounding boxes. Split formulas are highlighted in pink (3rd panel), and merged formulas are highlighted in green (4th panel). A small number of false negatives (red) and false positives (yellow) are produced.

with SSD512 performs far better (+5% f-score) than SSD300, [31] and cross-entropy loss with hard-negative mining performs better than focal-loss [32] (with or without hard-negative mining). Focal-loss reshapes the standard cross entropy loss such that it down-weights the loss for well-classified examples.

We evaluated SSD models with different parameters[7] and found that our HBOXES512 model, which introduces additional default box aspect ratios (see Section IV-B) performs better than SSD512, and MATH512 performs better than HBOXES512. For HBOXES512 we used default boxes with aspect ratios $\{1, 2, 3, 5, 7, 10\}$ instead of default boxes with aspect ratios $\{1, 2, 3, 1/2, 1/3\}$ for SSD512. MATH512 uses default boxes with aspect ratios $\{1, 2, 3, 5, 7, 10\}$ as well as rectangular kernels of size $1 \times 5$ rather than the square $3 \times 3$ kernel used in SSD512. From our experiments on the validation set, we observed that the MATH512 model consistently obtained the best detection results for the $512 \times 512$ inputs (by 0.5% to 1.0% f-score). So we use MATH512 for our evaluation. We then re-trained MATH512 using all TFD-ICDAR2019v2 training data.

ScanSSD was built starting from an existing PyTorch SSD implementation.[8] The VGG16 sub-network was pre-trained on ImageNet [33].

### *B. Quantitative Results*

We used two evaluation methods, based on the ICDAR 2019 Typeset Formula Detection competition [5] (Table II), and the character-level detection metrics used by Ohyama et al. [4] (Table III).

[7]Details are available in [31]

[8]https://github.com/amdegroot/ssd.pytorch

TABLE II: Results for TFD-ICDAR2019

| | IOU $\geq 0.75$ | | | IOU $\geq 0.5$ | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F-score | Precision | Recall | F-score |
| ScanSSD* | **0.781** | **0.690** | **0.733** | **0.848** | **0.749** | **0.796** |
| RIT 2† | 0.753 | 0.625 | 0.683 | 0.831 | 0.670 | 0.754 |
| RIT 1 | 0.632 | 0.582 | 0.606 | 0.744 | 0.685 | 0.713 |
| Mitchiking | 0.191 | 0.139 | 0.161 | 0.369 | 0.270 | 0.312 |
| Samsung‡ | 0.941 | 0.927 | 0.934 | 0.944 | 0.929 | 0.936 |

* Used TFD-ICDAR2019v2 dataset
† Earlier ScanSSD, placed $2^{nd}$ in TFD-ICDAR 2019 competition [5]
‡ Used character information

**Formula detection.** An earlier version of ScanSSD placed second in the ICDAR 2019 competition on Typeset Formula Detection (TFD) [5].[9] The new ScanSSD system outperforms the other systems from the competition that did not use character locations and labels from ground truth.

Figure 7 gives the document-level f-scores for each of the 10 testing documents, for matching constraints $IOU \geq 0.5$ and $IOU \geq 0.75$. The highest and lowest f-scores for $IOU \geq 0.75$ are 0.8518 for Erbe94, and 0.5898 for Emden76. We think this variance is due to document styles: we have more training documents with a style similar to Erbe94 than Emden76. With more diverse training data we expect better results.

Examining the effect of the IOU matching threshold on results demonstrates that the detection regions found by ScanSSD are highly precise: 70.9% of the ground-truth formulas are found at their exact location (i.e., IOU threshold of 1.0). Requiring this exact matching of detected and ground truth formulas also yields a precision of 62.67%, and an f-score of 66.5%. To obtain a more complete picture, we next look at the detection of math symbols.

[9]The first place system used provided character information.