ScanSSD Paper: Detecting Math Formulas with the Scanning Single Shot Detector

Abstract

- SSD - effective for small size images (300x300, 512x512), reducing document to that size makes small formulas hard to detect
- Modified SSD, sliding window, voting scheme to combine detections.
- only visual features, simple, but effective

Intro

- The first step in making formulas more easily used for search is identifying their locations in documents.
- characters and formula locations required for high accuracy to recover formula structure.
- indentation and vertical gaps, font difference (italics). (However, italicized words make it difficult).

Contributions

1. ScanSSD system for detecting formulas in document images using visual features. SSD to locate formulas within windows at multiple scales and detections pooled to produce detected formula regions using a voting procedure. Wider detected rectangles than SSD to improve performance.
2. new benchmark for formula detection comprised of a dataset and evaluation tools.

Challenges

- Embedded (normal text bata na chutteko) mathematical expressions are more challenging.
- distinguishing dictionary words that appear in italics and embedded mathematical expressions is a non-trivial (not easy/obvious) task.

Related Works

OCR-based Formula Detection:

Each text line in a file is scanned: if the line contains one of the 25 most frequent mathematical symbols, the leftmost word containing a mathematical symbol is located and a formula region is then grown to the left and right using a set

of rules. Computationally intensive and non robust (due to manually created rules).

Formula Detection in Born-Digital Documents

use typesetting information encoded in born-digital PDFs, such as page layout, character labels, character locations, font sizes, etc.

i. 4 step detection process
ii. CRF using both layout features (font types) and linguistic features (n-grams). Start, middle, end of math formula - annotations
iii. Visual features + Typesetting information. CNN + RNNs. Top-down and bottom up layout analysis.

Image-Based Segmentation of Formulas

U-net to detect characters in formulas. The U-net acts as a pixel-level image filter, and does not produce explicit regions for symbols or formulas.

Deep Learning Object detection models: - R-CNN, Fast R-CNN, Faster R-CNN, Yolo, SSD. - Yolo and SSD - single stage network. - SSD - multiple grids with different scales. So, detect objects of varying sizes easily. - Predicts translation and scale modifications. - VGG16 architecture for feature extraction. - ScanSSD - Modified SSD in a manner similar to TextBoxes as a basis for formula detector. - Sliding window framework. (600 dpi document page images)

ScanSSD

- Sliding window (overlapping page image regions).
- Each window passed to SSD.
- NMS (greedy strategy) to select window level detections.
- Stitch window level detection together on the page.
- Voting based pooling method to obtain final detection results.

A. Sliding window

- 1200 x 1200 window , stride = 120 pixels. (10%)
- roughly 10 text lines in height
- Advantages:
  - Data augmentation
  - Prevent loss of visual information
  - Repetition of same formulas increases recall
  - Can detect small formulas as well

- Disadvantages:
  - Increased computational cost (prevent by parallelization, process each window in parallel)
  - Large formulas get cut into sub images (solve by detecting formulas across windows)
  - Stitching required

B. Region Matching and Default Boxes in SSD

- 32 x 32 grid of default boxes. Different sized and aspect ratios of default boxes used.
- Confidence score represented by color.
- Each GT box matched to a default box with highest IOU, and also with boxes with an IOU > 0.5.
- Added aspect ratios for wider formulas : $\{5, 7, 10\}$ in addition to those in the Original SSD.

C. Post Processing

- Expand and/or shrink initial formula detections so that they are cropped around the connected components they contain and touch at their border.
- Done at 2 stages.

Voting Based Detection Pooling

- As the SSD network sees the same page region multiple times, multiple bounding boxes are often predicted for a single formula.
- Detections within windows are stitched together on the page, and then each detection region votes at the pixel level.
- Voting strategy (see in paper).
- Uniform weighting (Detection count with threshold 30) - best detection results (76.8% F - score for IOU 0.75).

ICDAR2019v2 Dataset

Results

Quantitative Results

- detected formulas are often close to their ideal locations.
- ScanSSD's math symbol and formula detection f-scores are primarily due to merging and splitting formula regions.

- 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.

Qualitative Results

- can detect math regions of arbitrary size (single char to 100 chars).
- detects matrices, rejects page numbers, eq numbers, etc.
- detection error - When large space between charcters within a formula - splitting occurs (eg variable constraints) .
- Error: Merging when formulas too close to each other.
- Wide embedded graphs detected as math formulas.
- most detection 'failures' appear to be because of valid detections being merged or split, and are not true spurious detections or false negatives.

Conclusion

- First contribution: ScanSSD Architecture - SSD using sliding window, voting based pooling across windows and scales. The SSD is modified to improve formula detection by changing aspect ratios for local detection and changing the convolution kernel size (from 3x3 to 1x5).
- The second contribution is TFD-ICDAR2019V2, a modified version of the GTDB datasets that corrects differences in scale and translation for the publicly available versions of PDFs used in the collection.

Future Works:

- improve the non-maximal suppression algorithm and low-level detection merging to avoid under-segmentation of adjacent formulas, and over-segmentation of formulas with large whitespace gaps (e.g., for variable constraints to the right of a formula).
- Ideally, we would like to find a way to have an end-to-end trainable system able to learn pooling parameters to avoid these errors.