Towards Better Identification of Fact Tables in Statistical Spreadsheets

Paul Kronlund-Drouault [1,2,3] Ioana Manolescu[2,3]

[1]ENS de Lyon [2]INRIA [3]IP Paris

The StatCheck project

As a response to modern challenges posed by fact checking, and as a tool for combating misinformation, the StatCheck system represents an ambitious attempt to automatically gather, structure, and utilize statistical data from trusted institutional sources

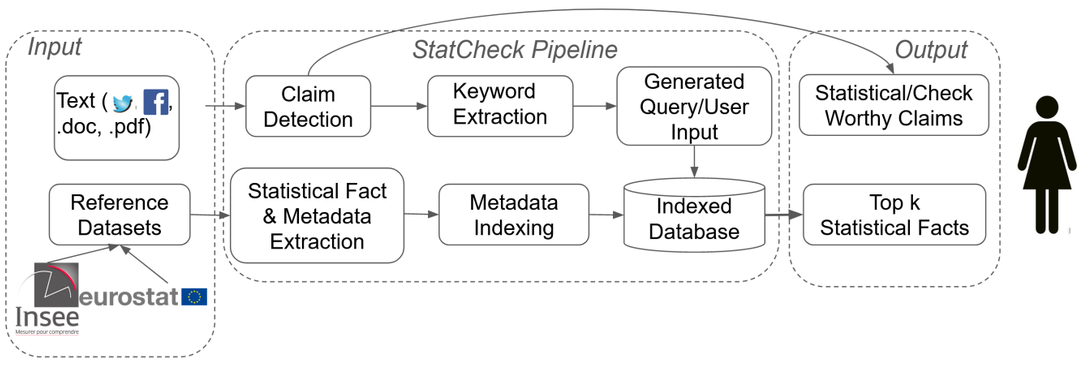

Figure 1. The StatCheck system.

The system targets a comprehensive pipeline that includes **web scraping** of official statistics, **automated data extraction** from complex spreadsheet formats, and **indexing** for rapid claim verification. StatCheck's architecture enables the ingestion of high-dimensional statistical data from sources such as INSEE and Eurostat, converting heterogeneous spreadsheet formats into structured, queryable databases.

Our extraction pipeline

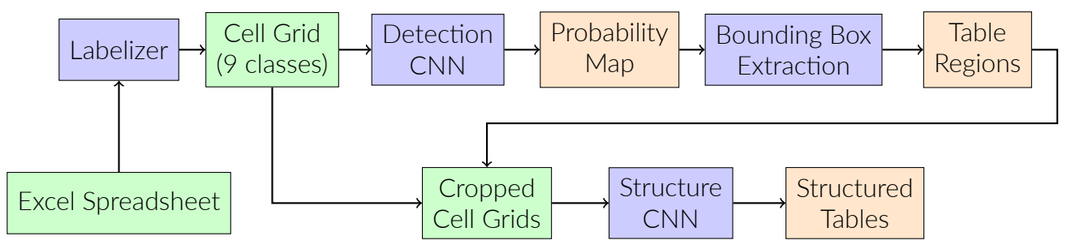

Figure 2. Architecture overview diagram

Core approach

Our approach employs a two-stage pipeline that progressively refines table detection and structure identification. The pipeline consists of two U-Net inspired Convolutional Neural Networks (CNNs) and intermediate processing steps:

- **Grid pre-processing**: A labeling system classifies the raw cell values from each sheet into 9 data categories as detailed in Table 1.
- **Table Detection**: A trained detection network is used to identify **table regions** from the entire cell grid, producing a probability segmentation map with two classes: `background` and `table`.
- **Intermediate post-processing**: An algorithm converts the probabilistic cell-level predictions into bounding boxes, localizing rectangular candidate table regions for further analysis.
- **Structure Classification**: A refinement network processes cell grids cropped to the detected bounding boxes, and produces a 4-class segmentation into `background`, `column headers`, `row headers`, and `data` regions.

Performance and results

We tested our approach on a manually annotated INSEE set of 164 tables. Our compact, grid-native CNN pipeline achieves strong performance for table detection on **challenging spreadsheets** and is ready to integrate as a fallback in the StatCheck extractor.

Model	Table Detection			Structure Classification				Pipeline
	Prec.	Rec.	IoU	Overall	CH	RH	Data	Overall
CNN	0.48	0.55	0.59	0.62	0.43	0.46	0.76	51.70%
CNN-large	0.56	**0.71**	0.74	**0.70**	0.57	0.70	0.75	**61.26 %**
CNN-large + Comp.	0.73	0.51	0.67	0.67	0.54	0.50	0.70	57.96%

Results suggested that:

- Compression greatly improves scalability
- Structure labeling remains the bottleneck due to class imbalance and noise

Next steps include:

- Using **bigger models**
- **Strengthening structure learning** with class-balanced/structured losses
- Better **synthetic data calibration**
- Replacing non-differentiable post-processing with an **end-to-end detection head**

CNN-based table detection

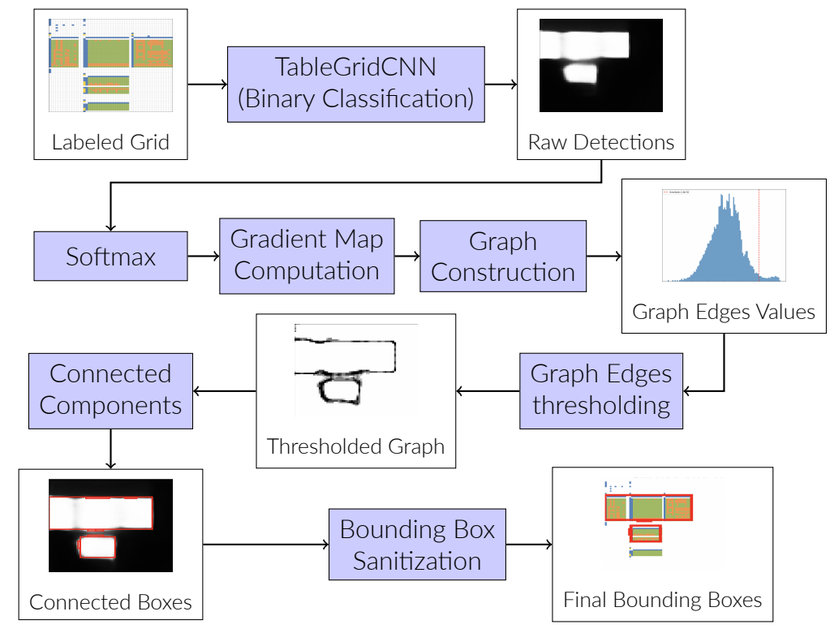

Figure 3. Table Detection Pipeline: from labeled grid to bounding boxes through CNN classification and graph-based postprocessing.

Cell grid labels

0	`EMPTY`	The cell is truly empty or blank (`""` |)
1	`TEXT`	Non-numeric, non-date string
2	`FLOAT`	Non-integer IEEE-754 value (e.g. 3.14, 1.2e−3)
3	`INTEGER`	Signed or zero integer that is likely *not* a year/date, e.g., 12, or 120 321
4	`MIXED`	Alphanumeric value with a high digit ratio (e.g. "A12b")
5	`DATE`	Cell already typed as DATE by the spreadsheet editor *or* a date-like string
6	`AMBIGUOUS DATE`	Cell that contains a number and string that **could** be a date "2020 Q4"
7	`WEIRD`	Anything that falls through the above rules
8	`MISSING VALUE`	A set of predefined values used by the data provider to represent missing data.

Grid compression strategy

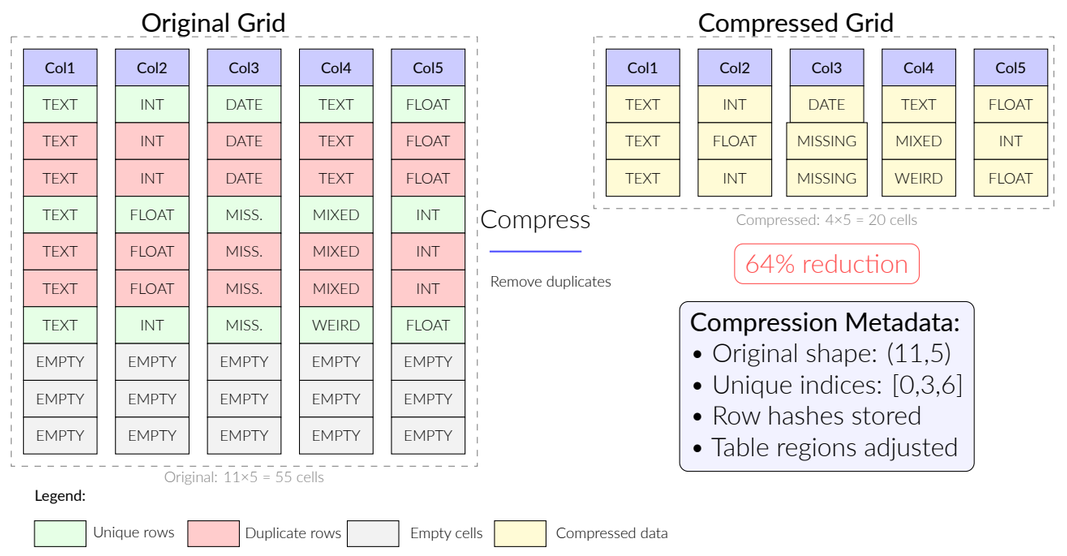

Figure 4. Grid compression algorithm inspired by SpreadsheetLLM. The method detects and removes consecutive duplicate rows (highlighted in red) using SHA1 hash comparison.

Training methodology

We split training into two supervised learning stages due to non-differentiable post-processing steps.

Detection Model Training

The detection model uses a composite loss combining focal loss with spatial regularization:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{focal}} + \lambda_w \mathcal{L}_{TV}$$

- $\mathcal{L}_{\text{TV}} = \frac{1}{H \times W} \times \left[\sum_{i,j}^{H,W} \left(p_{i,j-1} - p_{i,j} \right) + \sum_{i,j}^{H,W} \left(p_{i-1,j} - p_{i,j} \right) \right]$
- $\mathcal{L}_{\text{focal}} = -\alpha(1-p)^{\gamma} \log(p) \quad \text{if } y = 1$

Structure Model Training

Structure training uses ground-truth bounding boxes with random offset noise to mimic real conditions. Loss computed as standard cross-entropy:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$$

paul.kronlund-drouault@ens-lyon.fr

orcid.org/0009-0001-9326-4878