OPTYM - AI SCIENTIST ASSESSMENT

Saipraneeth Devunuri, Ravi Ahuja
Optym, Inc.
3401 Olympus Blvd Suite 500,
Dallas, TX 75019
{praneeth.devunuri, ravi.ahuja}@optym.com

1 Problem Statement

Checkboxes may be tiny, but they encode binary choices that can decide the fate of legal, financial, and regulatory documents. A single misread box in a loan application, clinical consent form, or compliance report can void the filing, stall transactions, or trigger statutory penalties. Classical Optical Mark Recognition (OMR) works well on pristine, template-locked pages, yet it falters in modern workflows rife with varied layouts, low-resolution scans, and handwritten marks [1].

Large Vision–Language Models (LVLMs) such as GPT-4o (Vision) have raised the bar in document understanding, excelling at tasks like OCR [2]. Even so, top models still misclassify 30–50 % of checkbox queries after domain-specific fine-tuning [3]. The stubborn error rate reveals a blind spot shared by both open-source and proprietary multimodal LLMs from OpenAI, Anthropic, and Google—fueled by scarce annotated data, information-dense contexts, and the visual diversity of checkmarks (ticks, crosses, manual scribbles, machine-printed glyphs).

Recent research tries to tackle the problem with explicit checkbox pipelines: a lightweight detector that isolates candidate boxes, and a downstream model reasons over their states. YOLOv8 variants trained in synthetic and domain-specific data already show promising gains in the localization of boxes and the classification of their states in clinical and administrative forms [4].

Assessment objective: Build comparable pipelines (see Figure 1) that given a pdf (1) detects and crops checkbox fields of interest, (2) interprets checkboxes using existing or custom VLMs (general purpose or specialized), and (3) outputs predicted classes along with their confidence scores.

2 Data Sources

You are free to use any data sources as long as they are public. Here is a list of sources that you can start with:

- *CheckboxQA [3]:*
  A purpose-built benchmark for checkbox state understanding in documents. Includes bounding box annotations and state labels (checked/unchecked).
  `https://github.com/Snowflake-Labs/CheckboxQA`
- *FUNSD Dataset [5]:*
  A dataset for form understanding, covering form fields, key-value pairs, and spatial relationships. It is useful for pretraining or layout-aware modeling.
  `https://guillaumejaume.github.io/FUNSD/`
- *RVL-CDIP (forms subset) [6]:*
  A large collection of scanned documents labeled by type.
  `https://adamharley.com/rvl-cdip/`

Synthetic data may be used for augmentation or training, but must constitute no more than 50% of the evaluation set unless accompanied by a strong justification. You may focus on a specific domain (e.g., clinical forms, financial

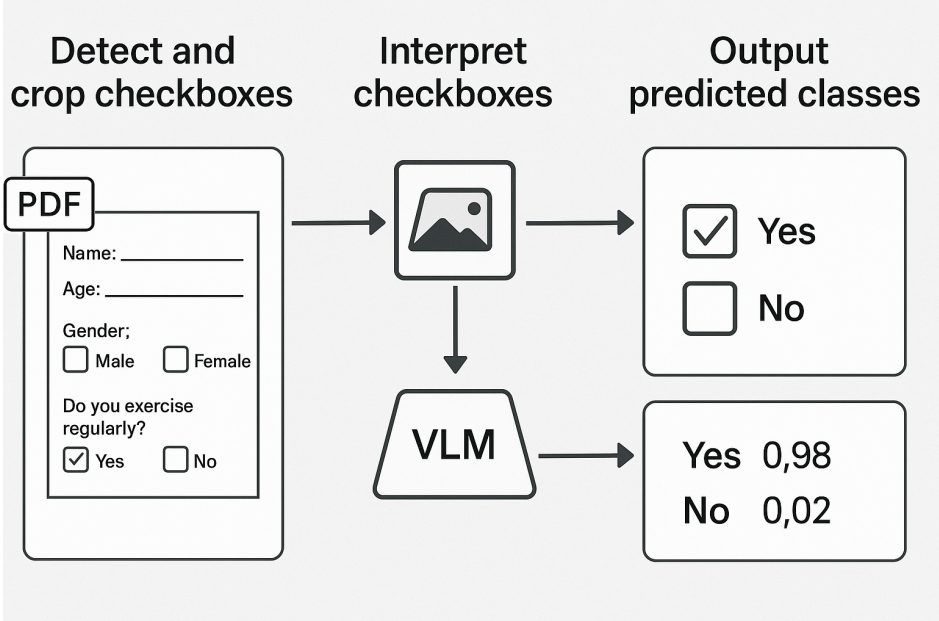

Figure 1: Illustration of the expected pipeline, including PDF preprocessing, checkbox localization, state classification, and output generation

checklists), but clearly state your assumptions. Your pipeline should be generalizable beyond a single template, and hardcoded templates are not allowed.

3 Deliverables

1. **Detailed Report** (PDF, limit to 8 pages)
   - *Approach* — Clearly describe the model architecture(s) you selected, including both detection and classification components. Justify your model choices based on the problem constraints (e.g., accuracy, latency, robustness). If using off-the-shelf models (e.g., GPT-4o, YOLOv8), explain why they are appropriate and how they are integrated into your pipeline. Mention any alternative models considered and trade-offs involved.
   - *Workflow* — Provide a step-by-step explanation of the full pipeline, from input PDF to final output. Include diagrams or flowcharts where helpful. Describe all key stages: preprocessing (e.g., deskewing, denoising), detection, checkbox state classification, postprocessing, and output formatting.
   - *Tools & Frameworks* — List the key libraries, models, toolkits, and cloud services you used. Specify the compute resources (e.g., GPU specs, RAM, disk) and software stack (e.g., PyTorch, OpenCV, Docker, Hugging Face Transformers). Mention any version constraints or special setup requirements.
   - *Benchmark* — Explain how you split or curated your datasets (train, validation, test). Define your baseline (if any) and explain the cost/latency assumptions for your comparisons. Clearly describe the evaluation setup and methodology to ensure reproducibility.
   - *Results* — Report your results using the graded metrics (detection mAP, classification accuracy, JSON-F1, efficiency) and any additional metrics you consider useful. Include:
     (a) A metric table summarizing your system performance.
     (b) A plot showing latency vs cost or accuracy trade-offs.
     (c) Qualitative examples and error analysis.
     (d) A brief discussion of limitations and potential future improvements.
2. **Source Code & Assets**
   - Submit all scripts (train.py, infer.py, evaluate.py), utility functions, and configuration files required to run your pipeline end-to-end.
   - Include a Dockerfile or environment.yml that fully specifies the runtime environment.

- If using pretrained or fine-tuned models, include weights or provide links to publicly hosted checkpoints on huggingface.

3. **Reproducibility Instructions**
   - A `README.md` that lists system requirements, setup steps, and one-line commands for: `bash run.sh train`, `bash run.sh infer <pdf>`, `bash run.sh evaluate`.
   - Expected output paths (e.g. `results.json`) and how to regenerate the figures and tables in the report.

4 Evaluation Metrics

Table 1 lists the evaluation metrics that will be used to assess submissions, in addition to evaluating the overall approach and quality of the report and documentation. Detection mAP is measured at IoU ≥ 0.5 [1] for checkbox bounding boxes. Classification accuracy is computed per checkbox instance. JSON-F1 is based on structured output correctness at the form level. See the example output format below:

```
[
  {
    "page": 1,
    "checkbox_id": "cbx_001",
    "coordinates": [100, 150, 130, 180],
    "state": "checked",
    "confidence": 0.98
  },
  ...
]
```

Efficiency reflects the practical usability of your pipeline. It is measured in terms of wall-clock time and peak VRAM usage during inference. Solutions that are excessively slow or resource-heavy may be impractical in production settings. Your complete pipeline should process a representative PDF document in under **2 minutes** on a single GPU (GeForce RTX 4060 or similar) . Submissions exceeding this threshold will be penalized unless strongly justified. Benchmark quality is evaluated based on how clearly trade-offs between cost, latency, and accuracy are analyzed and justified.

Stage	Metric	Weight
Detection	mAP	30%
State Classification	Accuracy	30%
End-to-End	JSON-F1	15%
Efficiency	Wall-clock & peak VRAM	10%
Benchmark Quality	Cost/Latency/Accuracy analysis	15%

Table 1: Graded evaluation metrics.

Feel free to report additional metrics if they provide useful insights.

5 FAQ

Common FAQs:

- **Can I use external data?** Yes, as long as it is public OSI-approved. Additionally, feel free to create augmented datasets
- **Can I use external models?** Can I use external models? Yes — as long as the model license is OSI-approved and weights are publicly downloadable. You are free to choose the detection and classification models, but you must explicitly describe, justify, and benchmark your choices in the report.
- **What do I do in case of Skewed scans?** Pre-processing is allowed; template hard-coding is not.

In case of any questions, reach out via email to clarify!

[1] Use 10 IoU thresholds from 0.5 to 0.95

6 License

Liability: Candidates are responsible for ensuring that any external models or datasets used are compliant with the licensing terms and may not submit proprietary or closed-weight models unless usage is explicitly permitted under their license.

IP Assignment: Upon submission, all code and assets become the property of Optym, Inc. in accordance with this assessment agreement.

References

[1] Erik Miguel de Elias, Paulo Marcelo Tasinaffo, and R Hirata Jr. Optical mark recognition: Advances, difficulties, and limitations. *SN Computer Science*, 2(5):367, 2021.

[2] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

[3] Michał Turski, Mateusz Chiliński, and Łukasz Borchmann. Unchecked and overlooked: Addressing the checkbox blind spot in large language models with checkboxqa. *arXiv preprint arXiv:2504.10419*, 2025.

[4] Henning Schäfer, Cynthia S. Schmidt, Johannes Wutzkowsky, Kamil Lorek, Lea Reinartz, Johannes Rückert, Christian Temme, Britta Böckmann, Peter A. Horn, and Christoph M. Friedrich. A multimodal pipeline for clinical data extraction: Applying vision-language models to scans of transfusion reaction reports, 2025.

[5] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. Funsd: A dataset for form understanding in noisy scanned documents, 2019.

[6] Adam W Harley, Alex Ufkes, and Konstantinos G Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *International Conference on Document Analysis and Recognition (ICDAR)*.