論　　文

スポーツ団体による大規模測定に伴うデータハンドリングのための テンプレート型データ抽出システムの開発

林　　勇　樹*
追手門学院大学

Development of a Template-Based Data Extraction System for Handling Data from Large-Scale Assessments by Sports Organizations

Yuki HAYASHI
Otemon Gakuin University

Abstract

Sports organizations increasingly conduct large-scale physical assessments to support athlete development and to build longitudinal datasets. While competition results are often captured digitally, newly designed assessments frequently rely on paper forms filled by hand at the venue, which leads to substantial post-event data entry workload and delayed feedback. This paper reports the development of a template-based data extraction system that supports end-to-end handling of such paper-based measurements, from reception and form issuance to OCR-assisted digitization, validation, and result return. The system consists of (i) an on-demand form generation and printing module that assigns an unpredictable GUID and prints it as a QR code, (ii) a browser-based OCR module that performs homography-based alignment using corner L-markers and a SafeZone-constrained Hough line detector implemented with OpenCV.js, (iii) cell-wise single-digit recognition using the GPT-4o vision API with configurable parallelism, and (iv) server-side validation, storage, and web-based feedback. We describe design choices made under field constraints, including a one-digit-per-cell form layout, a JSON-based region template with adjustable padding, and a generalized mechanism to reconstruct multi-cell values using group identifiers. Development was iterative and time-constrained because operational changes become difficult once large-scale measurement events start; large language models were used as a development aid for rapid integration across heterogeneous components, with human review at each iteration. Operational logs from multiple events were analyzed to characterize throughput and registration delays under different collection modes (on-site scanning vs. postal submission). The contribution of this work is not a new OCR algorithm but a practical integration pattern and operational knowledge for deploying template-based extraction in sports assessments.

*責任著者：林 勇樹，追手門学院大学，yu-hayashi@otemon.ac.jp
Corresponding Author：Hayashi Yuki, Otemon Gakuin University, yu-hayashi@otemon.ac.jp

キーワード：OCR，テンプレート，測定シート，データハンドリング，スポーツ団体，大規模言語モデル
Keywords：OCR, template-based extraction, measurement form, data handling, sports organization, large language model

緒言

スポーツ団体では，競技会における記録や順位のように，電子計時・電子掲示・計測機器の仕様によりデータ形式が規定されている情報は，過去のデジタル基盤整備により比較的管理しやすい．その結果，全国規模の横断データや複数年にわたる縦断データの収集が容易になっている例が多い．

一方で，国内競技団体（National　Federation：NF）が競技者の特性を把握する目的で新たな測定を企画する場合，既存システムを流用できないことがある．測定機器や測定会場の整備，測定員の育成に加え，測定値の取りまとめと管理，ならびに被測定者への即時性のあるフィードバックが運用上の課題となる．特に，測定会が開始された後に記入フォーマットや業務手順を変更することは困難であり，限られた準備期間で運用に耐えるワークフローを確立する必要がある．

本研究では，段階的に測定時の記入シートとデータ収集方法を更新しながら，大規模測定における受付から結果返却までのデータハンドリングを支える技術基盤を開発した．中心となる課題は，①測定現場での追加作業を最小化しつつ，②手書きで記入された測定値を安定的にデジタル化し，③エラー検出と確認を行い，④結果を迅速に返却することである．

研究目的

本研究の目的は，スポーツ団体による大規模測定において，紙ベースの手書き測定シートを前提としながら，データの一元収集と結果返却を可能にするテンプレート型データ抽出システムを設計・実装し，運用ログに基づいて実用性を検討することである．

なお，同日開催の複数会場での並列運用は現時点では少ないが，将来の運用形態や会場分散を想定し，複数会場での処理集中が起こり得る状況でも破綻しにくい構成を目指した．

関連研究

手書き帳票のデータ抽出は，帳票の幾何補正と項目領域の切り出し，ならびに文字認識を組み合わせて実現されることが多い．幾何補正では，線分検出の代表的手法である Hough 変換[1] や，射影変換（ホモグラフィ）[2] に基づく整列が用いられる．本研究でも，四隅のＬ字マークから基準点を推定し，ホモグラフィで座標系を正規化した上で，テンプレートで定義された領域を切り出す方式を採用した．

文字認識については，従来から Tesseract に代表される OCR エンジン[3] が広く利用されている．一方で，本研究の対象である測定シートは「1 マス 1 文字（数字）」という制約を設計段階で導入できるため，単語認識よりも単一文字分類に近い問題設定となる．本研究では，単一数字に特化したプロンプト設計により，汎用視覚モデル（GPT-4o）を OCR として利用した[4]．

本研究に先立ち，ブラウザ内で完結する軽量 OCR（例：Tesseract.js）や，TensorFlow.js 等を用いた小型 CNN による数字分類も検討対象とした．しかし，測定会場で得られるスキャン画像は，紙の汚れ，コピー機由来の影，傾き・台形歪みなどの変動が大きく，一般的な OCR エンジンでは予備検証の段階で十分な認識精度が得られなかった．また，小型モデルを採用する場合は，帳票・筆跡・スキャン条件に合わせた学習データの収集と再学習が必要となり，限られた準備期間で運用に投入するには負担が大きい．そこで本研究では，単一数字に特化した制約（1 マス 1 数字）と前処理を併用した上で，汎用視覚モデルを外部 API として呼び出す構成を採用した．

実装面では，PDF の表示・レンダリングに PDF.js[5] を用い，画像処理と Hough 変換にはブラウザ上で動作する OpenCV.js[6] を用いた．また，開発過程では複数の技術領域（PDF 生成，ブラウザ GUI，画像処理，サーバサイド実装）の統合が必要であったため，大規模言語モデルをコーディング支援として併用した[7]．ただし，生成物は逐次レビューし，運用要件に照らして仕様を確定した．

要件整理と設計方針

データ入力の自動化においては，マークシート

表 1：要求事項と主な設計要素

要求事項	設計要素（実装）
測定員の追加作業を最小化	受付でのオンデマンド発行（AutoPrint），紙への手書き記入を維持
スキャン品質のばらつきに対する頑健性	四隅 L 字マーク検出＋ホモグラフィ補正，SafeZone 制約付き Hough 線分検出
項目数が多い測定の入力負担軽減	1 マス 1 数字の帳票設計，セル単位 OCR，groupId による複数セル連結の一般化
現場での設定変更の必要性	領域テンプレートを JSON で管理，GUI で region 定義・padding・group 追加を編集可能
処理集中への耐性	セル OCR を並列実行（並列数を設定可能），バッチ実行と進捗表示
結果の信頼性と修正容易性	サーバ側の妥当性検証，確認 UI での修正と再送信

（表注）設計要素は本稿で述べる主要機能に対応する．

（OMR）による入力効率化も代替案として考えられる．一方で，本研究の運用では，測定直後に選手へ原本を返却しつつ，転記内容の控えを手元に残して確認・フィードバックに用いる必要があった．このため，数字として可読な転記欄を維持することが重要であり，マーク方式は控えの可読性や現場での目視チェック効率の点で適合しにくかった．また，測定現場では机上で集中して塗りつぶす環境が常に確保できるとは限らず，記入作業の追加が測定業務の妨げとなり得る．以上より，本研究では数字記入方式を採用し，後工程の入力・チェック負担をシステム側で低減する方針とした．

大規模測定では，受付，測定（複数ステーション），集約，入力，確認，登録，結果返却という一連の工程が存在する．測定員は測定業務に専念するため，データ入力のための追加作業や機器操作は最小化される必要がある．また，計測後の入力・確認作業は測定担当者の負担となり，移動時間などを用いて長時間の入力・チェックを行う状況も生じ得る．

本研究では，紙での記入という現場制約を維持しつつ，後工程（入力・確認・登録・結果返却）の負担を軽減することを主眼に，次の設計方針を置いた．(i) 受付で測定シートをオンデマンド発行し，識別子を QR コードで付与する．(ii) 記入シートは 1 マス 1 数字の形式とし，テンプレートに基づいてセル単位で切り出して OCR する．(iii) スキャン品質のばらつき（傾き，台形歪み，縮尺差，影，汚れ）に対して，四隅基準点と SafeZone を用いた頑健なキャリブレーションを行う．(iv) OCR 結果はサーバ側で妥当性検証し，確認画面で修正可能にする．(v) 同日複数会場の並列運用は少ないが，処理集中に備えて OCR の並列数を調整可能とする．

システム構成

システムは，①受付時に測定シートを発行する PDF 生成・印刷モジュール（AutoPrint），②スキャン PDF をブラウザ上で読み込み，③テンプレートに基づいて OCR する抽出モジュール（OCR テンプレート GUI），④抽出結果を検証・保存し，結果を返却するサーバサイド機能から構成される（図 1）．

従来は，測定終了後に紙シートを持ち帰り，手入力と目視チェックを行うことが多く，データ登録とフィードバックの遅延が課題であった．本研究では，スキャンと OCR を介在させ，確認 UI を通じて入力を短縮するフローを設計した（図 2）．

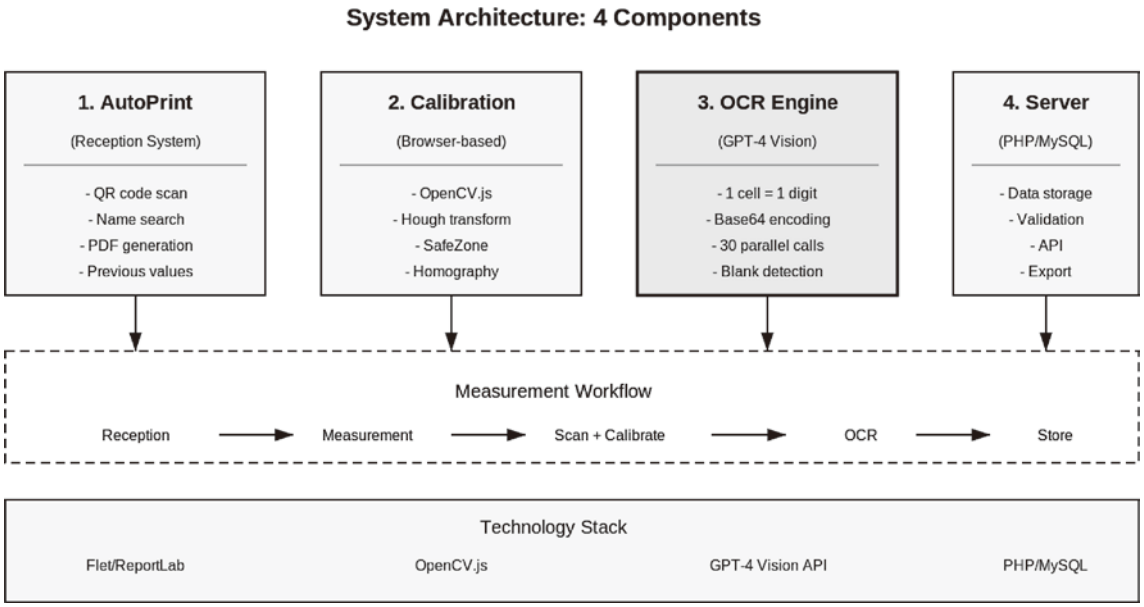


図 1：システム全体アーキテクチャ

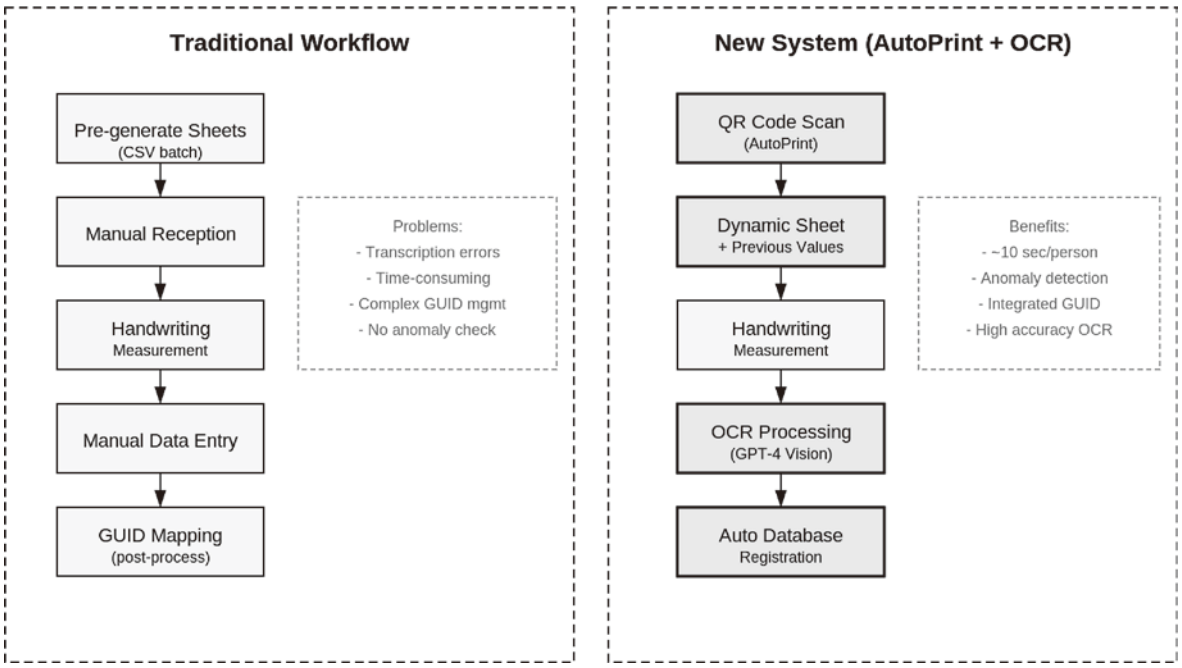


図 2：従来の入力中心フローと提案フローの比較

測定シート設計と受付システム（AutoPrint）

測定シートは，手書き記入を前提としつつ，データ抽出を容易にするために「1 マス 1 数字」の入力枠を採用した．この制約により，文字列認識ではなく単一数字の分類として OCR を設計でき，桁数が固定された測定項目に適合する．また，記入ガイドとして「1」と「8」の合字を薄色で印字し，等幅フォント Courier を採用した．

受付では，測定者の識別と後工程の紐付けのため，予測困難な GUID を発行し，QR コードとして印字した測定シートをオンデマンドで生成・印刷する（図 3）．

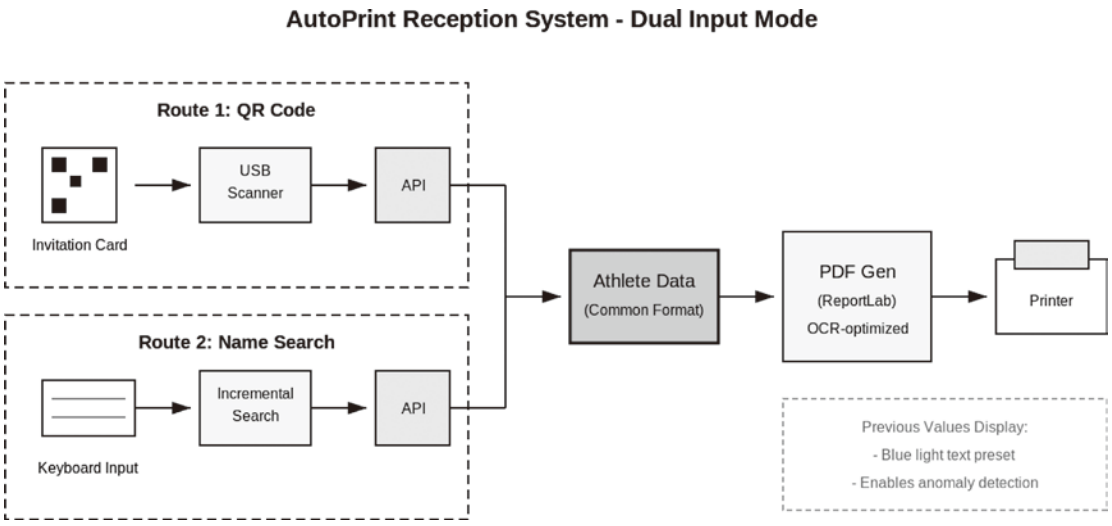


図 3：受付でのオンデマンド発行（AutoPrint）のシステム構成

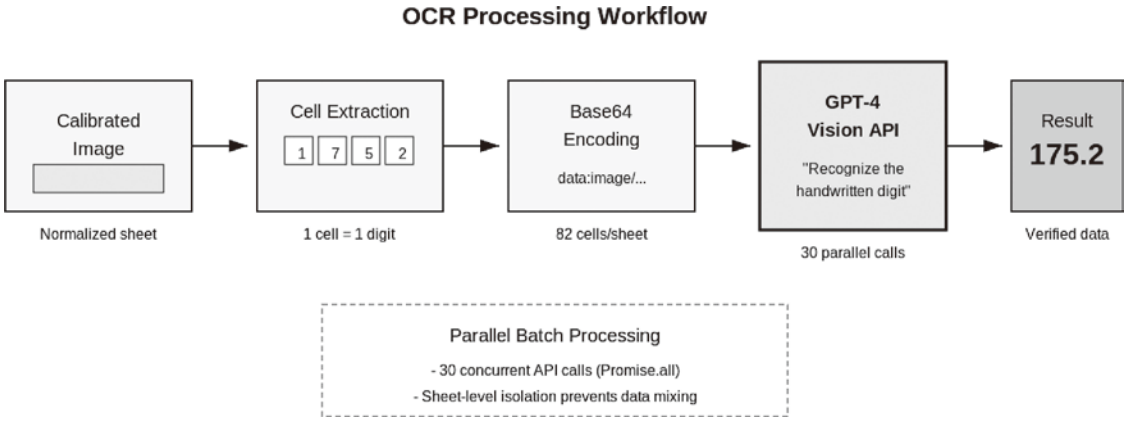


図 4：テンプレート型 OCR の処理フロー

GUID は推測攻撃が成立しにくい長さ・形式で生成し，URL に含めることで入力・確認画面への到達を制御する．ただし，本研究の主眼はアクセス制御の研究ではないため，運用上は追加の認証や配布方法を組み合わせることを前提とする．

大規模な測定会が開始された後は，帳票フォーマットや受付手順の変更が現場運用に与える影響が大きく，変更が困難になる．そのため，AutoPrint では帳票の版管理と出力ログを残し，帳票の更新を段階的に行えるようにした．

テンプレート型データ抽出（ブラウザ OCR モジュール）

OCR モジュールは単一の HTML ファイルと領域テンプレート（JSON）を中心に構成され，API キーを設定することでブラウザ上で動作する．環境依存のビルド工程を避けることで，測定拠点や持ち込み端末が変わっても実行環境を揃えやすい設計とした．

処理フロー

スキャン済み PDF を読み込み，PDF.js でページをレンダリングしてキャンバス画像を得る．次に，四隅の L 字マークを検出して射影変換（ホモグラフィ）行列を推定し，ページの傾き・台形歪み・縮尺差を補正する．補正後の画像座標系において，テンプレートで定義された region（セル）を切り出し，セル単位で OCR を実行する（図 4）．

OCR 結果（各セルの数字）は groupId で束ねられ，複数セルを連結して 1 項目の値に復元する．復元値はサーバへ送信され，妥当性検証と保存，および結果返却に用いられる．

領域テンプレート（JSON）と groupId による値復元

領域テンプレートは JSON で管理し，ページ座標に対する相対座標（0 ～ 1）として region（セル）の位置と大きさを定義する．region は，①セル画像の切り出し領域（x，y，width，height），②所属する測定項目を表す groupId，③ OCR 形式（single　digit，QR など），④値の制約（例：0/1 のみ）を持つ．group は測定項目のメタ情報（名称，小数点位置，送信パラメータなど）を持ち，region 群と組み合わせて帳票全体を記述する．テンプレートの一部を付録 A に示す．

本研究では，フィールドを groupId で束ねることで「複数マスを連結して 1 項目の数値に復元する」処理を一般化した．具体的には，groupId ごとにセルを並び順（左から右）で連結して数字列を構成し，group が持つ小数点桁数に基づいて小数点を挿入する．これにより，身長のように複数桁かつ小数を持つ項目や，試行回数のように桁数が異なる項目を同一の集約ロジックで扱える．

また，OCR 切り出し領域の上下左右 padding を GUI のサイドメニューから調整できるようにし，罫線や記入位置の揺らぎによる数字の欠損を抑える設計とした．region 定義の追加・修正，group の追加・編集も GUI 上で行い，JSON としてエクスポートして版管理できる．

キャリブレーション—四隅 L 字マークと SafeZone 制約

スキャン PDF には，傾き，台形歪み，縮尺差，影，汚れが含まれ得る（図 6）．テンプレートに基づく領域切り出しを成立させるためには，帳票画像をテンプレート座標系へ正規化するキャリブレーションが必須である．

初期実装では，各隅の近傍 ROI を切り出して二値化し，隅に最も近い暗画素（L 字マークの頂点候補）を基準点として採用した．しかし，ページ端の汚れや影，かすれ等により，頂点以外の画素が先に検出されると基準点が大きくずれ，射影変換後に認識領域が系統的にずれる問題が生じた．

この課題に対し，後期実装では，隅近傍の SafeZone（外周帯）に限定した前処理（照明むら補正，適応的閾値処理，モルフォロジー処理）を行い，Canny エッジ検出後に Hough 線分検出を適用して水平・垂直の代表線分を推定し，その交点を L 字頂点として採用した（図 5）．SafeZone により，コピー機由来の影や外周部の濃度変動の影響を抑制し，線分検出の安定性を高めた．

SafeZone 設定とキャリブレーション調整 UI

実運用では，帳票の版や印刷状態により，外周マークの太さや余白が変化することがある．本研究では，SafeZone の幅や検出パラメータを GUI で調整できるようにし，帳票ごとの適用範囲を視覚的に確認しながら設定できるようにした（図 7，図 8）．設定は JSON に保存されるため，帳票版の更新と合わせて再現可能である．

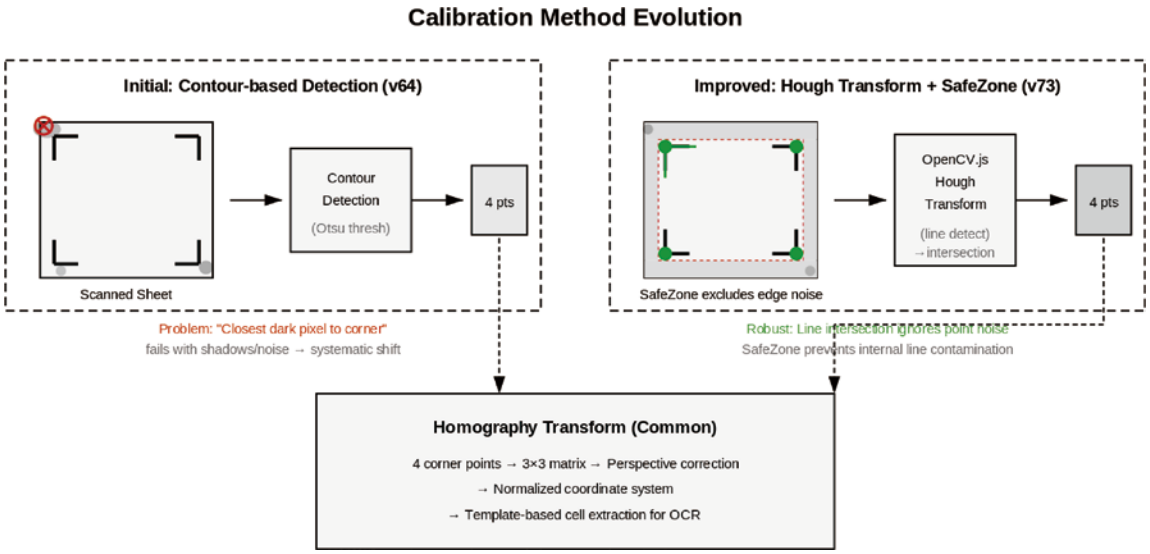


図 5：キャリブレーション手法の変更点



図６：キャリブレーション枠に対する実スキャン例（赤：校正前枠，青：検出枠）

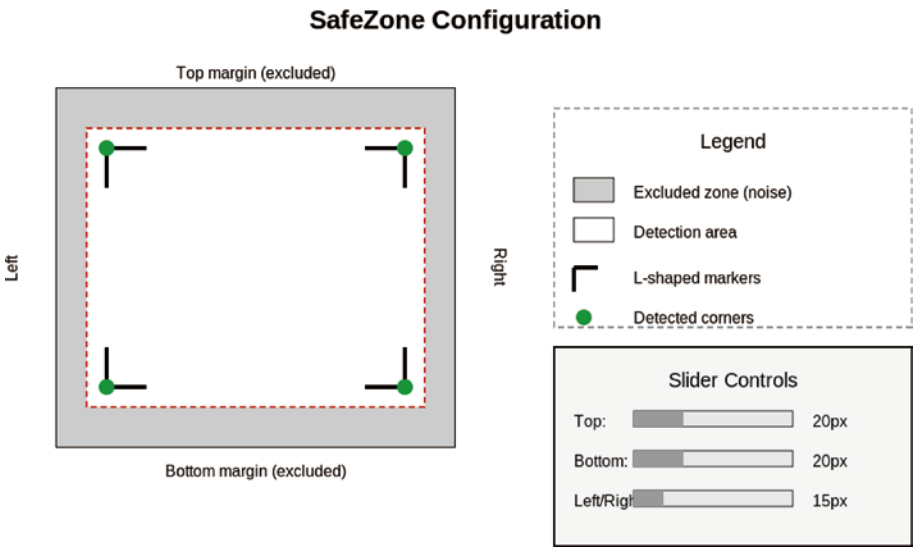


図 7：SafeZone 設定 UI

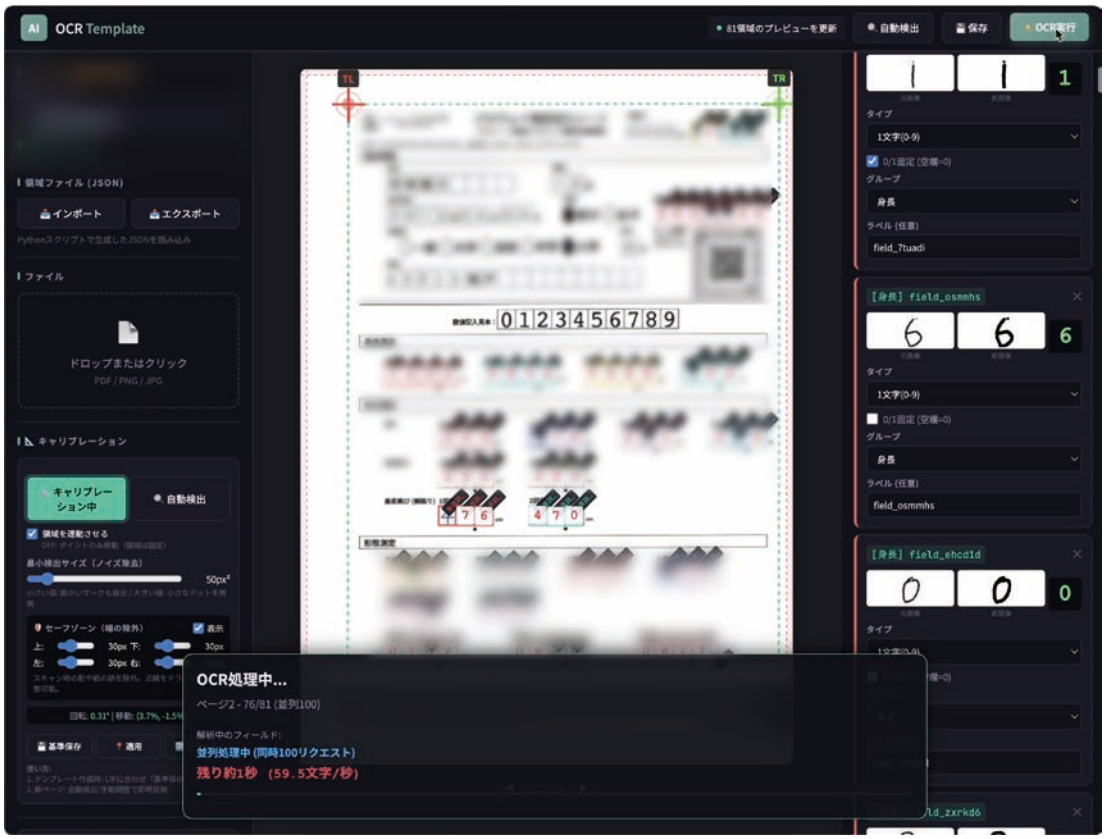


図 8：OCR テンプレート GUI のスクリーンショット

セル画像前処理と単一数字 OCR

セル画像は，ホモグラフィ補正後のキャンバスから region ごとに切り出し，必要に応じて最小サイズ（例：200px 以上）へアップスケールする．また，region の上下左右 padding を設定して罫線成分を避け，1 と枠線の混同などのエラーを避ける．

OCR には OpenAI の視覚モデル（GPT-4o）を用い，system prompt で「0 ～ 9 のいずれか 1 文字のみを返す」

「空欄は 0」「印字されたガイド文字は無視する」等の制約を明示し，temperature=0 で実行する．この設計により，セル単位で一定の出力形式を確保し，後段の集約処理を単純化した．0/1 の二値項目については，出力が 0/1 以外の場合に 0 へフォールバックし，確認画面で補正できるようにした．

並列実行とスループット

OCR はセル単位の独立なリクエストとして実行し，ブラウザ側で並列数を設定できるようにした．実運用では，ブラウザ処理負荷の軽減を主目的として並列数を 30 ～ 50 程度に設定し，端末性能やネットワーク状況に応じて調整した．並列数自体はより大きく設定することも可能である．

対象帳票では 1 枚あたりの region 数が 81（数字セル 80 + QR セル 1）である．セルごとの API 応答が 1 ～ 2s 程度で返る条件では，並列数 30 ～ 50 の場合，概ね 2 ～ 3 バッチで処理が進み，OCR 部は 6 ～ 8s ／枚程度で完了する．並列数は端末性能に応じて調整可能であり，実行例では並列 100 設定で約 60 文字／秒の進捗表示を示し，81region の処理が数秒以内で完了した（図 8）．ただし，実測時間は端末性能，通信遅延，API 負荷に依存する．

サーバサイド統合と結果返却

サーバ側では，GUID に紐付く測定データを受信し，測定項目ごとの妥当性検証（範囲，桁数，必須項目など）を行った上で保存する．妥当性検証は，OCR の誤認識や記入ミスを早期に検出し，確認画面での修正を促すためのものである．

本研究では，GUID を URL の主要キーとして扱い，予測困難な識別子である点を利用して，無差別な URL 推測によるアクセスが困難になるよう設計した．ただし，個人情報を扱う運用では，GUID の秘匿に加えて，通信経路の保護やアクセス権限の管理が必要である．

結果返却は，測定者が自身の結果を閲覧できる Web 画面として実装し，運用上は測定終了後できるだけ早いタイミングで返却することを想定した．本稿では，結果返却の心理的・行動的効果を評価することまでは扱わず，データハンドリング基盤としての実装に焦点を当てる．

開発プロセスとバージョン更新

本開発では，OCR（画像処理・LLM　API），PDF 生成，GUI，サーバサイド実装という異なる技術領域を短期間で統合する必要があった．特に，大規模測定会が開始されてしまうと，その後に帳票フォーマットやオペレーションを変更することは極めて困難である．この制約の下で，段階的なプロトタイピングと運用フィードバックに基づく改修を繰り返した．

開発効率化のため，コーディング支援として大規模言語モデル（Claude）を利用した．具体的には，仕様の言語化，既存コードの理解補助，GUI 実装や例外処理の追加，デバッグ観点の洗い出しなどに用い，人手によるレビューと実機検証を前提として採用した．

主要な機能追加と改修はバージョン単位で管理し，

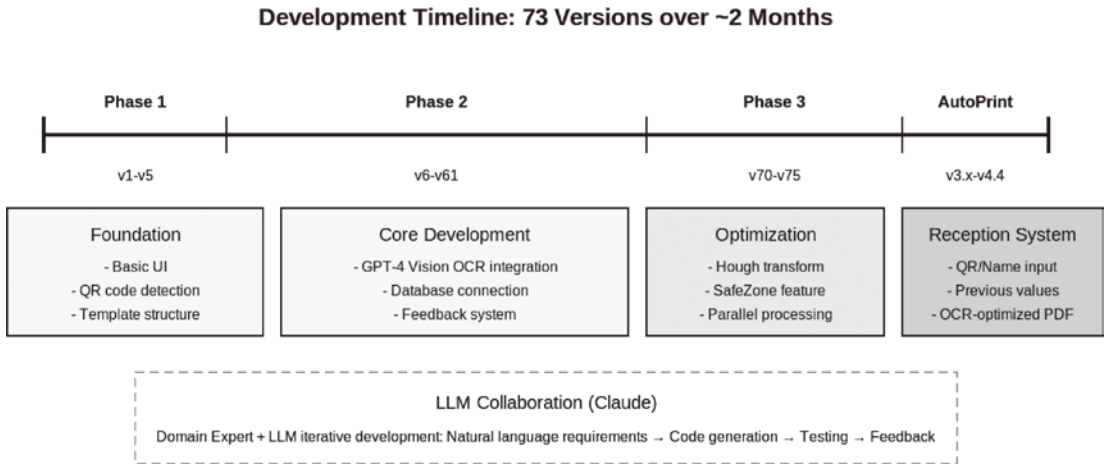


図 9：開発タイムライン

キャリブレーション頑健化（v64 → v73），SafeZone 導入（v70 系），groupId による値復元の一般化（v60 系），GUI からの region 編集（v60 系）などを段階的に実装した（図 9）．

開発上の課題と対応

（a）キャリブレーション不安定．初期は隅近傍 ROI で二値化輪郭から角に最も近い点を選択していたが，ページ端の汚れ・影・かすれで誤検出し，射影変換後に region が系統的にずれることがあった．SafeZone 制約付きの前処理＋ Hough 線分検出に変更し，頑健性を改善した．

（b）設定データの肥大化．テンプレート編集と試行を繰り返す過程で，ブラウザ側に保存する設定が肥大化し，保存・復元が不安定になることがあった．設定の粒度を整理し，JSON の入出力と版管理に移行して再現性を確保した．

（c）region 定義の微調整．スキャン環境や印刷状態により最適な切り出しが変化するため，padding や region 位置の微調整が必要であった．GUI 上で region 定義と group 追加を編集できるようにし，現場調整と再利用を可能にした．

（d）OCR 誤認識と確認負荷．セル単位 OCR では誤認識が完全には避けられない．出力形式を 1 桁に制約し，空欄時の規定値を設け，さらにサーバ側の範囲チェックと確認画面での修正導線を整備した．

（e）並列処理時の負荷．並列数を過大にするとブラウザが高負荷になり，UI 応答が低下する．並列数を設定可能とし，バッチ単位での進捗表示とエラーハンドリングを整備した．

精度検証（OCR 認識）

本章では，本研究により開発された全システムの導入によるフルスペック運用を行った測定イベント（以下，フルスペック大会）で取得した測定シートを対象に，目視確認済みの正解データと OCR 出力を照合し，文字認識精度を評価した．なお，本章の評価は，運用で採用している SafeZone=30px（初期設定値）で実行した結果に基づく．

データと評価方法

評価対象は，フルスペック大会で回収した測定シート 246 件である（1 人 1 枚）．各シートは 24 項目（groups）から構成され，全体で 5,904 グループ（=24 × 246）を評価対象とした．また，帳票は「1 マス 1 数字」の設計であり，各シートは 80 セル（regions）から構成されるため，全体の単一数字認識は 19,680 セル（=80 × 246）である．

正解データは，運用で生成されたデータを人手で目視確認し，誤りを修正したものを用いた．OCR の出力は，テンプレート（JSON）で定義された region を切り出し，セル単位に 0–9 の 1 文字を推定して復元した数値列である．認識モデルには OpenAI　API の GPT-4o（Vision）を用いた．

評価指標は以下の 2 水準とした．第一にグループ正解率（group　accuracy）は，項目単位（groupId で束ねた複数セルの集合）で復元された数値が正解データと完全一致した割合とした（小数点を含む）．第二に数字正解率（digit　accuracy）は，セル単位での 0–9 認識が一致した割合とした．さらに，数字の誤認傾向を把握するため，0–9 の混同行列（confusion matrix）を作成した．

結果

表 2 に精度指標の要約を示す．グループ正解率は 94.92%（5,604/5,904）であり，数字正解率は 98.05%（19,297/19,680）であった．グループ誤りは 300 件で，平均すると 1 枚あたり 1.22 項目の不一致に相当する（平均正解 22.78/24）．また，数字誤りは 383 件で，平均すると 1 枚あたり 1.56 セルの誤認に相当する（平均正解 78.44/80）．

表 2：OCR 認識精度の要約（n=246）

指標	値
評価シート数	246
評価グループ数（24 項目× n）	5,904
グループ正解率	94.92%
グループ誤り数	300
評価セル数（80 セル× n）	19,680
数字正解率	98.05%
数字誤り数	383
平均グループ誤り / 枚	1.22
平均数字誤り / 枚	1.56

（表注）グループ正解率は項目単位の完全一致，数字正解率はセル単位の一致である．

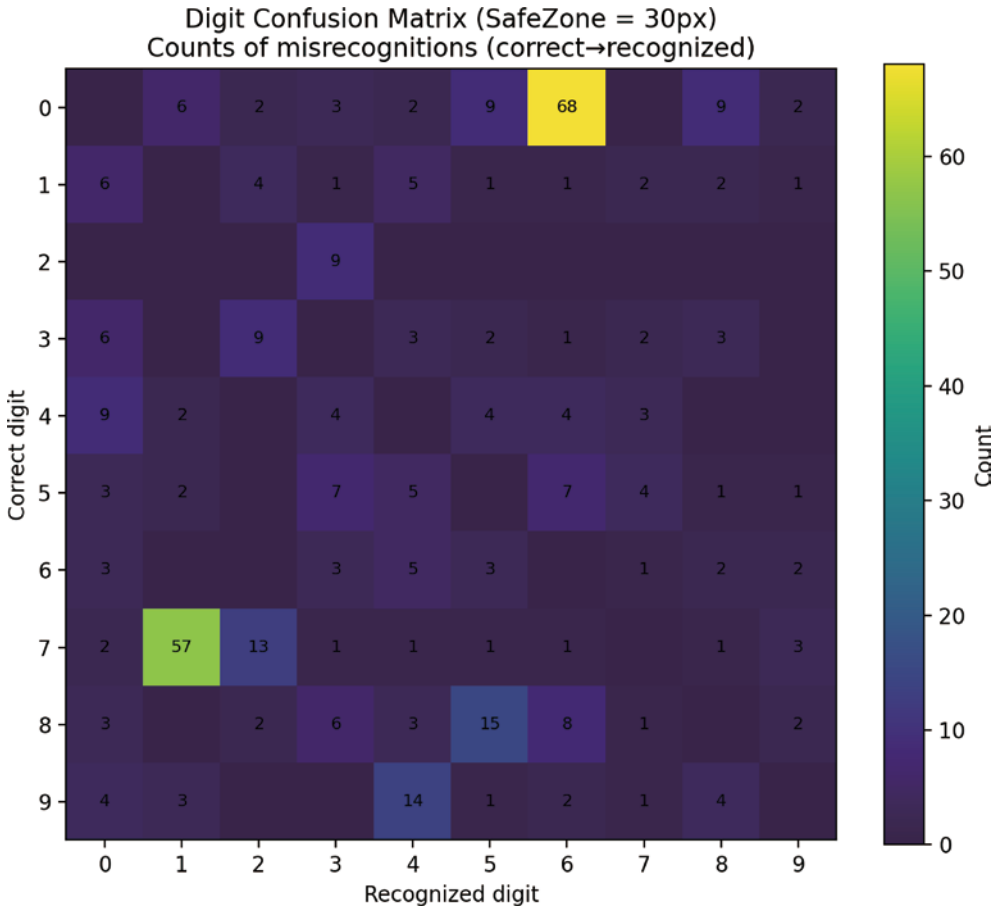


図 10：数字混同行列

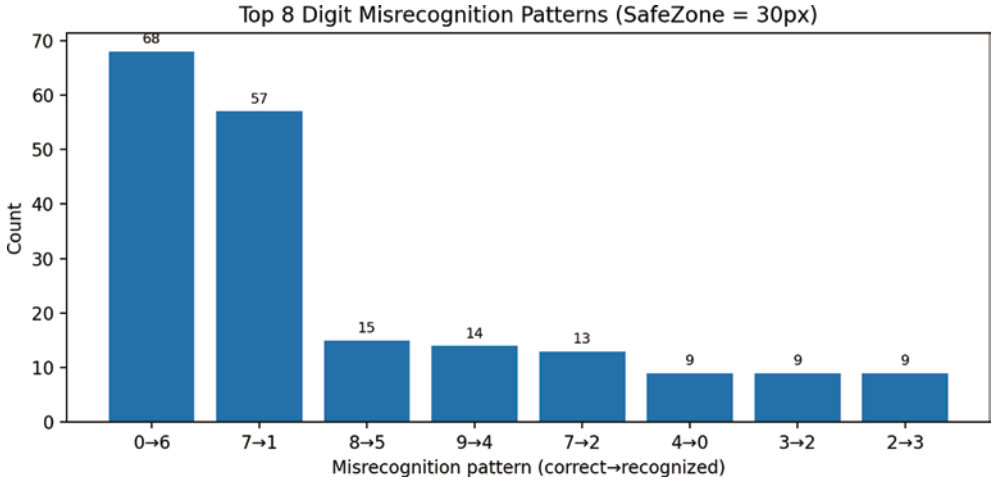


図 11：誤認パターン上位

誤認パターンの分析

図 10 に数字レベルの混同行列を示す．誤認の多くは，形状が近い数字間で発生していた．例えば，0 → 6 が 68 件，7 → 1 が 57 件と多く，「閉ループの欠落」や「上部横線の弱さ」に起因する誤認が残存した（図 11）．

訂正書き（書き換え）による誤認

残存誤りの一部は，文字形状の曖昧さだけでなく，測

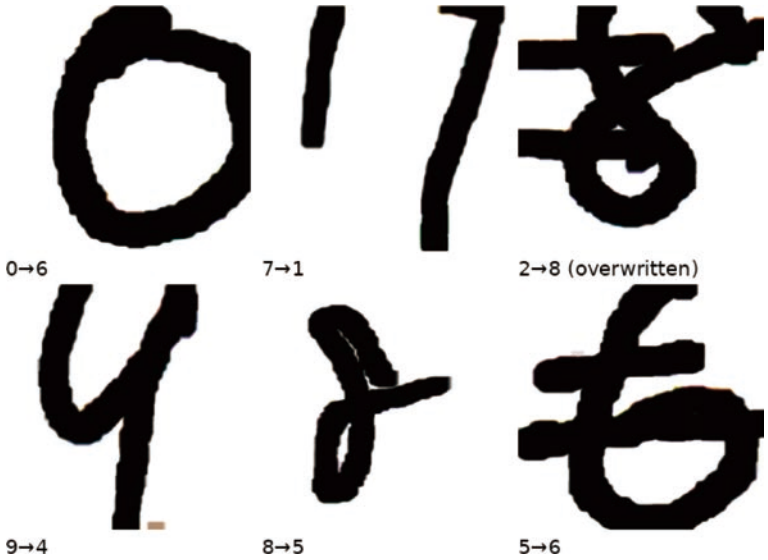


図 12：誤認識例

定現場での訂正書き（取り消し線，二重線，書き直し）に起因していた．本システムは 1 セル 1 文字の前提でセル画像を切り出すため，同一セル内に複数ストローク（取消線＋書き直し）が混在すると，単一数字としての判別が困難になり，誤認が発生しやすい．図 12 に代表例を示す．この種の誤りは，画像処理の改善だけでは完全な解消が難しく，運用上は「訂正が生じた場合は別セル・別行に再記入する」等のルール設計と併用する必要がある．

処理時間

初期設定値での処理時間は，セル単位の API 応答が平均 976.7ms（中央値 895.9ms）であった．本帳票は 1 枚あたり 80 セルであるため，セル処理を 30–50 並列で実行すると，API 待ち時間としては概ね 2–3 バッチ（約 3s 前後）で完了する．実運用では，PDF レンダリング，キャリブレーション，切り出し，サーバ送信等の周辺処理が加わるため，OCR 開始から結果表示までのエンドツーエンドは数秒オーダ（概ね 6–8s/ 枚）となった．ただし，ネットワーク遅延や API 混雑により外れ値（最大 13s 程度）が生じる場合があるため，現場運用ではバッチ処理と進捗表示により待ち時間を吸収する設計とした．

運用評価

運用評価として，複数の測定イベントで本システムを運用し，サーバに記録された登録ログ（時刻情報）を集計した．本稿では個人情報保護と運用秘匿のため，イベント名，団体名，会場名は匿名化し，集計値のみを示す．また，測定・スキャン・OCR・確認の各作業時間を厳密に分離して計測したわけではなく，ログ解析は主に登録時刻の系列に基づく．

ここでは，(i) 直近のフルスペック運用を行った測定イベント（イベント F）における登録の連続性，(ii) 複数地域ブロックで実施され，スキャン提出方法が地域により異なったイベント（イベント B）における登録遅延の分布，の 2 点を示す．

イベント F では，登録時刻差分の中央値が 45 秒であり，連続登録（60 秒以内）が約 61% であった（表 3）．これは，OCR 処理と確認・登録が一定期間に集中して連続的に実行されたことを示す指標であり，セル単位 OCR をバッチ処理する設計が運用上機能していたことを示唆する．

イベント B では，地域により登録遅延の中央値が大きく異なった（表 4）．遅延が短い地域では，会場または会場近傍でスキャンしてオンライン提出する運用が成立していた．一方で遅延が長い地域では，紙シートを郵送し，到着後に集約拠点で OCR 処理を行う運用が含まれていた．この結果は，OCR システム単体の処理能力だけでなく，スキャン提出の運用設計が全体のスループットに影響することを示す．

表 3：イベント F における登録時刻差分の要約

指標	値
連続登録間隔の中央値	45 秒
連続登録間隔の 25–75 百分位	34–150 秒
60 秒以内の割合	60.9 %
120 秒以内の割合	72.0 %

（表注）登録時刻の差分はサーバ保存時刻に基づく．測定開始時刻からの経過時間ではない点に留意する．

表 4：イベント B における地域別の登録遅延（測定日から登録日までの近似）

地域（匿名）	遅延の中央値（時間）	遅延範囲（最小 – 最大，時間）
地域 A	3.4	2.5–4.5
地域 B	5.6	5.5–5.8
地域 C	6.1	6.0–6.2
地域 D	6.9	6.3–7.1
地域 E	8.7	6.9–8.8
地域 F	11.4	11.3–12.1
地域 G	13.1	12.9–13.3
地域 H	32.2	32.2–32.2
地域 I	36.6	32.9–36.8
地域 J	38.8	28.7–53.4
地域 K	53.2	51.4–54.4
地域 L	87.5	76.2–87.8

（表注）遅延は測定日の 0 時を基準に算出したもので，測定開始時刻を厳密に反映しない．

考察

本研究の特徴は，測定現場の制約（紙・手書き・測定員負担最小化）を前提に，帳票設計（1 マス 1 数字）とテンプレート（JSON）を固定点として，キャリブレーション，セル単位 OCR，確認 UI，サーバ統合を一体として実装した点にある．特に，region を groupId で束ね，複数セル連結と小数点挿入を一般化したことで，項目の増減や桁数差に対する実装コストを抑えつつ，GUI 上での設定変更で運用に追随できる構造を得た．また，フルスペック大会の照合では，初期設定値でグループ正解率 94.92%，数字正解率 98.05% を確認した．

残存誤りは，(i) 0 と 6，7 と 1 等の形状類似，(ii) 線が薄い，途切れる，傾く等の筆跡差，(iii) 訂正書き（取り消し線，二重線，書き直し）による複数ストロークの混在，に整理できる．特に訂正書きは，セル単位に単一数字が存在するという前提を破るため，画像処理やモデル選択だけで完全に解消することは難しい．このため，誤認低減には，帳票設計（訂正の余地を設ける），運用ルール（訂正時は別セルに再記入し，取消線を避ける），確認 UI（疑義セルの優先提示）を組み合わせた全体最適が重要である．

本システムは外部の Vision API を利用するため，通信環境や API 提供条件（料金，モデル更新，利用制限）の影響を受ける．運用コストは，1 枚あたりの認識領域数に比例して増加するため，帳票設計（領域数の抑制）と並列数制御が重要となる．本研究の帳票では 81region であり，初期設定値では並列数を 30 ～ 50 程度に設定することで，ブラウザの応答性と処理時間のバランスを取りやすかった．また，フルスペック環境では並列 100 でも処理が成立しており（図 8），端末性能に応じて並列数を調整できることを確認した．

レート制限については，本研究で使用したアカウント環境（Usage tier 表示）において，1 セル（約 200 × 200px）を高詳細で送信する場合の入力トークンを概算し，1 枚あたりの消費量を見積もった．その結果，1 枚あたり約 81 リクエスト，約 35,000 トークン程度となり，ダッシュボード上で表示される RPM/TPM 上限に対して十分な余裕があった（付録 B）．したがって，本研究の運用規模では，レート制限よりもネットワーク遅延，ブラウザ側の描画・前処理負荷，および確認作業が律速となりやすい．料金については契約・単価の変動が大きいため，本稿ではトークン消費量を示し，必要に応じて単価を掛け合わせて概算できる形とした[9-11]．

最後に，個人情報を扱う運用では，帳票・ログ・スキャンデータの保護が不可欠である．本システムでは，外部 API へ送信する画像をセル単位（1 桁）に切り出し，氏名等の直接識別子を含めない設計とした．登録番号等の数値項目は数字列として復元され得るが，外部

API へは 1 桁単位で送信されるシステム構造上，API 側に個人情報が一切渡らない．OpenAI の API では，デフォルトでは入力・出力がモデル学習に用いられない旨が示されている[8, 12]．また，API 提供者のデータ取り扱い方針に依存するため，運用時にはログ保持期間，および Zero Data Retention 等の選択肢を確認し，必要に応じてアクセス制御と保管期間設計を組み合わせる必要がある[8, 12]．

結論

本研究では，スポーツ団体の大規模測定における紙ベースの手書き測定シートを対象に，テンプレート型データ抽出システムを設計・実装した．オンデマンド発行（AutoPrint）による識別子付与，四隅 L 字マークと SafeZone 制約付き Hough 検出による頑健なキャリブレーション，JSON テンプレートと groupId による汎用的な値復元，セル単位 OCR の並列実行，およびサーバ側検証と結果返却を統合した．

運用ログの集計から，登録が連続して行われる局面ではバッチ処理が成立していること，また地域ごとの登録遅延はスキャン提出の運用設計に依存することが示された．さらに，フルスペック大会の照合では，初期設定値でグループ正解率 94.92%，数字正解率 98.05% を確認した．今後は，訂正書き等に起因する残存誤りを低減するための帳票・運用ルールの改善と，疑義セルの提示・再 OCR 等を含む確認プロセスの効率化を通じて，測定終了後の入力・チェック負担をさらに低減することが課題である．

謝辞

本システムの運用に協力した測定スタッフ，および試行運用に参加した関係者に感謝する．

また，本研究の論文執筆および解析システムの開発において，Anthropic 社の大規模言語モデル Claude Opus 4.5 およびコーディングツール Claude Code を補助的に使用した．最終的な内容の確認・修正は著者が行った．

利益相反

本研究において申告すべき利益相反は存在しない．

参考文献

1） Duda RO, Hart PE（1972）Use of the Hough Transformation to Detect Lines and Curves in Pictures. Communications of the ACM, 15（1）: 11-15.
2） Hartley R, Zisserman A（2004）Multiple View Geometry in Computer Vision, 2nd ed., Cambridge University Press, Cambridge.
3） Smith R（2007）An Overview of the Tesseract OCR Engine. Proceedings of the 9th International Conference on Document Analysis and Recognition（ICDAR）, pp：629-633.
4） OpenAI（2026）OpenAI API Documentation. https://platform.openai.com/docs/（参照日：2026 年 1 月 29 日）
5） Mozilla（2026）PDF.js. https://mozilla.github.io/pdf.js/（参照日：2026 年 1 月 29 日）
6） Bradski G（2000）The OpenCV Library. Dr. Dobb's Journal of Software Tools.
7） Peng S, Kalliamvakou E, Cihon P, Demirer M（2023）The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. arXiv preprint, arXiv:2302.06590.
8） OpenAI（2026）Data controls in the OpenAI platform（Your data）. https://platform.openai.com/docs/guides/your-data（参照日：2026 年 1 月 29 日）
9） OpenAI（2026）Rate limits | OpenAI API. https://platform.openai.com/docs/guides/rate-limits（参照日：2026 年 1 月 29 日）
10） OpenAI（2026）Images and vision | OpenAI API. https://platform.openai.com/docs/guides/images-vision（参照日：2026 年 1 月 29 日）
11） OpenAI（2026）Pricing | OpenAI API. https://platform.openai.com/docs/pricing（参照日：2026 年 1 月 29 日）
12） OpenAI（2026）How your data is used to improve model performance. https://openai.com/policies/how-your-data-is-used-to-improve-model-performance/（参照日：2026 年 1 月 29 日）

付録 A：領域テンプレート JSON の設計

領域テンプレートは，帳票の版ごとに JSON ファイルとして管理する．ocrSettings に前処理・切り出し・並列実行に関する設定をまとめ，groups と regions で帳票構造を記述する．また，キャリブレーションの参照点（四隅）と，元の region 定義を保持し，ホモグラフィ推定後の座標変換に用いる．

表 A1 に主要キーを示す．

表 A1：領域テンプレート JSON の主要キー

キー	内容
metadata	テンプレートの識別情報（版，シート種別など）
ocrSettings	PDF レンダリング倍率，最小切り出しサイズ，padding，前処理パラメータ，並列数など
groups	測定項目の定義（名称，小数点桁数，送信パラメータ等）
regions	セル（region）の定義（相対座標，groupId，形式，制約）
calibration	参照点と，補正前の region 定義（ホモグラフィ推定に利用）

以下に JSON の最小例（匿名化）を示す．

```
{
	“metadata”: {
		“version”: “vX”,
		“sheetType”: “measurement_sheet”
	},
	“ocrSettings”: {
		“pdfScale”: 4,
		“minRegionSize”: 200,
		“padding”: {
			“top”: 17,
			“bottom”: 11,
			“left”: 13,
			“right”: 10
		}
	},
	“groups”: [
		{
			“id”: “g_height”,
			“name”: “ 身長 ”,
			“decimalPlaces”: 1
		},
		{
			“id”: “g_weight”,
			“name”: “ 体重 ”,
			“decimalPlaces”: 1
		}
	],
	“regions”: [
		{
			“id”: “r001”,
			“label”: “ 身長（百）”,
			“groupId”: “g_height”,
			“format”: “single_digit”,
			“x”: 0.145,
			“y”: 0.58,
			“width”: 0.0375,
			“height”: 0.0321
		},
		{
			“id”: “r002”,
			“label”: “ 身長（十）”,
			“groupId”: “g_height”,
			“format”: “single_digit”,
			“x”: 0.183,
			“y”: 0.58,
			“width”: 0.0375,
			“height”: 0.0321
		}
	]
}
```

付録 B：並列数設定と API 制限に関する補足

セル単位 OCR では，1 枚あたりのリクエスト数が region 数に比例する．本研究の帳票では，セル OCR の対象が 80 セルであり，テンプレート自動検出（detectFields）を併用する場合でも，1 枚あたりのリクエスト数は概ね 81 回である．並列数を P とすると，理想化すれば約 ⌈80 ／ P⌉ バッチで処理が進む．

画像入力のトークン消費は解像度に依存するため，本研究ではセル画像を概ね 200 × 200px 程度に正規化して送信した．OpenAI のガイドでは，高詳細（detail=”high”）では 512px タイル単位でトークンが計上されるため[10]，本研究のセル画像は 1 タイル相当となり，画像トークンは約 255 tokens/ 枚（概算）であっ

た．これにプロンプト等（約 150　tokens）を加えると，1 セルあたり約 405　input　tokens，出力は 1 桁のため最大 5 tokens となる．したがって，1 枚あたりの総消費は約 35,000　tokens（概算）となる．料金の概算は，価格表に示される入力・出力単価（$/1M　tokens）を用い，（Tin ／ 10^6）× Pin ＋（Tout ／ 10^6）× Pout　として計算できる[11]．

本研究で使用したアカウント環境では，ダッシュボード上で　GPT-4o　の制限として　TPM=30,000,000，RPM=10,000，TPD=15,000,000,000　が表示されていた．上記の概算に基づけば，RPM 上限からは約 123 枚 / 分，TPM 上限からは約 857 枚 / 分が理論上の上限となり，本研究で想定する処理量ではレート制限がボトルネックになりにくいことを確認した[9]．もっとも，API 制限値や単価は契約形態・モデル更新で変化し得るため，運用時には公式情報を参照し再評価する必要がある[9, 11]．