



# 切符をOCRするシステムの開発

東京都立多摩科学技術高等学校  
後藤 成都 高橋 巧 上野 哲



## 背景

- 改札と不正乗車の防止
    - 改札のない無人駅は不正乗車の温床になっている
    - 手動改札は、係員とのやり取りの時間が発生する
    - 自動改札機には高額な導入費用がかかる
      - 安価で自動改札できるシステムが必要
  - 特急列車等の自由席での車内改札
    - 混雑時には実施できない
    - 旅客の作業や睡眠等を妨害し、快適性を下げる要因になっている
      - 旅客の手で車内改札ができると問題が解決できる
- 現在はチケットレスサービスの普及が進んでいる  
しかし、
- チケットレスサービスは便利だが、物が残らず、趣がない
  - 切符は思い出の品として保管できる有形物  
**切符の価値を重視し、OCRによって利便性を高めるシステムの開発**

## 目的

- 切符を活用した、**低コスト**なOCRシステムを開発する
- 切符の価値を向上**させる

## 方針

- 従来の紙媒体の切符の様式を**変えず**に利用
- 第一段階として、**車内改札システム**の開発に着手
- 乗務員との接触を減らすため、**携帯電話端末**で操作を完結
- 管理を容易にするため、**各種データをAPIサーバー**で管理
- システムの名前は以下のようにしている
  - 本システム全体 → 「Comet」
  - サーバーサイド → サーバーサイド(Stellar)
  - クライアントサイド → クライアントサイド(Andromeda)

## 設計

- システム構成(図1)
  - クライアントサイドと、サーバーサイドからなる
  - HTTP通信によるJSONデータの送受信の時、相互に接続される
  - クライアントサイドでは、ユーザインタフェースとバックグラウンドプロセスに分かれ、ユーザインタフェースの操作によってバックグラウンドプロセスが呼び出される
  - サーバーサイドでは、JSONを受信すると、YAML及びORMライブラリを通じたデータベースの呼び出しでリクエスト処理を行う

- クライアントサイド(Andromeda)
  - UIをFlutterで、OpenCVをC++で実装
  - FlutterからFlutterでOpenCVを呼び出し、画像処理を行う
  - Google ML Kit Text Recognize APIを、Flutterで実装
  - 送信データとしてJSON形式を用いる
- サーバーサイド(Stellar)
  - Goで実装
  - テストケースにYAML、送信データにJSONを利用
  - データベースを使用するため、ORMライブラリとしてGORMを採用
- 処理の流れ(図2)

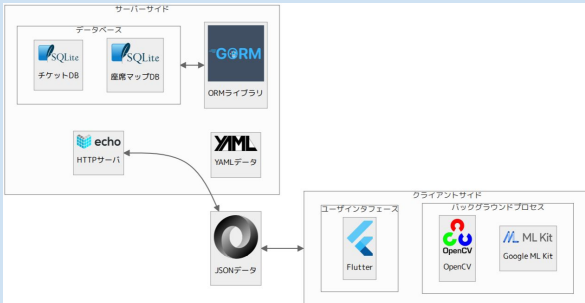


図1. システム構成図

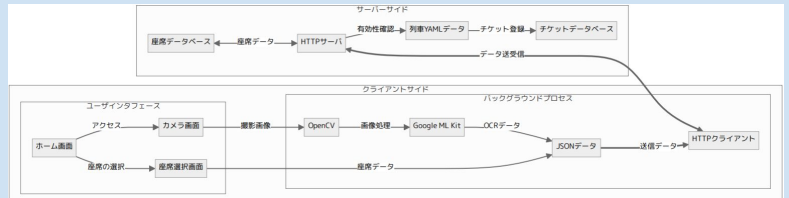


図2. 処理の流れ

## 現状・検証

### 現在完成した要素

- クライアントサイド(Andromeda)
  - カメラでのリアルタイムOCR
- サーバーサイド(Stellar)
  - 受信したJSONのパルス、YAMLのテストケースとの照合
  - チケットデータベース
  - 座席データベース
  - ORMを用いたプログラム上からのデータベースへの変更

### 検証

- 実機Android上で、OCRを普通列車グリーン車内にて行った結果(図3、図4、表1)

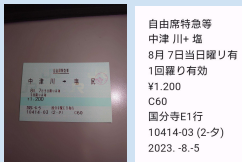


表1. OCR読み取り検証の結果

券の発行場所	○	△	×	計
8月7日当日曜日有	0	0	0	0
1回限り有効	2	5	2	9
¥1,200	0	2	1	3
C60	0	0	4	4
国分寺E1行	0	0	4	4
10414-03(2夕)				
2023-8-5				

(発着駅、日付、発券番号の読み取りで判定。正確数3…○, 1~2…△, 0…×)

図3. 特急券 図4. OCR結果

- Arch LinuxでStellar APIサーバーを使用して、curlでJSONデータを送信した結果(図5)

```
curl -X POST -H "Content-Type: application/json" -d '{"from_sta": "国分", "to_sta": "新秋津", "date": "2023/10/15", "ticket_number": "116"}' localhost:1323/ticket
```

```
{["is_verified": 1, "verify_number": "2234"]}
```

```
curl -X POST -H "Content-Type: application/json" -d '{"from_sta": "国分", "to_sta": "新秋津", "date": "2023/10/15", "ticket_number": "116"}' localhost:1323/ticket
```

```
{["is_verified": 0, "verify_number": ""]}
```

図5. curlコマンドによるJSONデータの送受信の様子

## 分析・展望

### 分析

- 図3・表1から、OCRの精度改善が必要  
→ OpenCVによる画像処理で余計な情報をカットすることが必要

### 展望

- クライアントサイド(Andromeda)
  - 利便性向上のため、より直感的で使いやすいUIを実装する
  - OCR精度改善のため、OpenCVを実装して画像処理を行う
  - 着席状況管理のため、**座席選択機能**を実装する
  - 乗り過ごし防止のため、降車駅近時の**プッシュ通知**を実装する
- サーバーサイド(Stellar)
  - 保守性の向上のため、**WebUIによるダッシュボード**を実装する
- 検証について
  - OCR精度が満足できるレベルに達するまで、検証を続ける
  - そのレベルに達したら、**クロスプラットフォーム化**を行う
  - 実際に地方私鉄で使ってもらうことも視野に入れる
  - 車内で不正乗車を見分けられるように**することを可能にしたい
- [+α]範囲の拡大に向けクライアントサイド(Andromeda)に以下の実装
  - 紙媒体チケットの**eチケット化**
  - 分散インスタンス**に向けたAPIサーバー追加機能

## アプリ

- リアルタイムOCR機能のみを実装した、簡易版です。
- 現在、Androidでの動作のみをサポートしています。
- 開発中のため、予期せぬ動作をする場合があります。
- まだ開発中の段階のものですが、興味があればダウンロードしてみてください。

