

FlutterでOCRを活用した商品情報の取得を実装してみた



Press Space for next page →



# 自己紹介

- 飯野陽平 ([wheatandcat](#))
- 会社: 合同会社UNICORN 代表社員
- Blog: <https://www.wheatandcat.me/>
- 今までに作ったもの
  - [memoir](#)
  - [OOMAKA](#)
  - [MarkyLinky](#)

# OCR

- 光学式文字認識（Optical Character Recognition）の略
- 画像から文字を認識してテキストに変換する技術

# モチベーション

- 在庫管理アプリを作成中で、なるべく手間を省いて在庫商品を登録できるようにしたい
- 初めに実装したのはバーコードを読み込んで商品を登録する方式
- しかし、実際に運用してみると以下のケースがあることが分かった
  - バーコードから商品情報を取得できないケースがある
  - バーコードが外箱に記載されており、商品のみ手元にある
- 上記の課題を解決するために**OCR**を活用し、画像から商品情報を取得する方法を試した。

# 使用するライブラリ & サービス

- google\_mlkit\_text\_recognition
  - 画像から文字列を抽出
- Cloud Natural Language | Google Cloud
  - OCRで取得したテキストから商品名を抽出
- Yahoo!ショッピングAPI
  - 抽出したキーワードで商品情報を取得

実装

# OCRで画像から文字を抽出①

```
import 'package:google_mlkit_text_recognition/google_mlkit_text_recognition.dart';
import 'dart:io';

Future<List<String>> imageTextRecognizer(File image) async {
  final inputImage = InputImage.fromFile(image);
  final textRecognizer = TextRecognizer(script: TextRecognitionScript.japanese);
  final RecognizedText recognizedText =
    await textRecognizer.processImage(inputImage);
  final texts = recognizedText.blocks
    .map((block) => block.text.replaceAll('\n', ' '))
    .where((text) => text.length > 1 && !RegExp(r'^\d+$').hasMatch(text))
    .toList();

  return texts;
}
```

# OCRで画像から文字を抽出①

```
import 'package:google_mlkit_text_recognition/google_mlkit_text_recognition.dart';
import 'dart:io';

Future<List<String>> imageTextRecognizer(File image) async {
  final inputImage = InputImage.fromFile(image);
  final textRecognizer = TextRecognizer(script: TextRecognitionScript.japanese);
  final RecognizedText recognizedText =
    await textRecognizer.processImage(inputImage);
  final texts = recognizedText.blocks
    .map((block) => block.text.replaceAll('\n', ' '))
    .where((text) => text.length > 1 && !RegExp(r'^\d+$').hasMatch(text))
    .toList();

  return texts;
}
```

# OCRで画像から文字を抽出①

```
import 'package:google_mlkit_text_recognition/google_mlkit_text_recognition.dart';
import 'dart:io';

Future<List<String>> imageTextRecognizer(File image) async {
  final inputImage = InputImage.fromFile(image);
  final textRecognizer = TextRecognizer(script: TextRecognitionScript.japanese);
  final RecognizedText recognizedText =
    await textRecognizer.processImage(inputImage);
  final texts = recognizedText.blocks
    .map((block) => block.text.replaceAll('\n', ' '))
    .where((text) => text.length > 1 && !RegExp(r'^\d+$').hasMatch(text))
    .toList();

  return texts;
}
```

# 文字を抽出②

右の画像を読み込むと以下のテキストが抽出できる

```
[  
    "38%",  
    "\\"卖上/",  
    "No.",  
    "*インテージSRI+",  
    "一般用ウェットティッシュ",  
    "累計販売金額",  
    "207年\\月20B",  
    "▲ここまで引っぱってご使用ください。",  
    "シルコット",  
    "ウェットティッシュ",  
    "、純水99%",  
    "手・口まわりの汚れに",  
    "メイク時の手指の拭き取りに デ",  
    "コアウォーター",  
    "ノンアルコール"  
]
```



# 文字を抽出②

右の画像を読み込むと以下のテキストが抽出できる

```
[  
    "38%",  
    "\\"卖上/",  
    "No.",  
    "*インテージSRI+",  
    "一般用ウェットティッシュ",  
    "累計販売金額",  
    "207年\\月20B",  
    "▲ここまで引っぱってご使用ください。",  
    "シルコット",  
    "ウェットティッシュ",  
    "、純水99%",  
    "手・口まわりの汚れに",  
    "メイク時の手指の拭き取りに デ",  
    "コアウォーター",  
    "ノンアルコール"  
]
```



# 文字を抽出②

右の画像を読み込むと以下のテキストが抽出できる

```
[  
    "38%",  
    "\\"卖上/",  
    "No.",  
    "*インテージSRI+",  
    "一般用ウェットティッシュ",  
    "累計販売金額",  
    "207年\\月20B",  
    "▲ここまで引っぱってご使用ください。",  
    "シルコット",  
    "ウェットティッシュ",  
    "、純水99%",  
    "手・口まわりの汚れに",  
    "メイク時の手指の拭き取りに デ",  
    "コアウォーター",  
    "ノンアルコール"  
]
```



# 抽出文字列から商品名を特定①

Google Cloud Natural Languageの「エンティティ分析」を使ってテキストを解析  
商品名を自動抽出する

エンティティ分析可能なType

```
import { LanguageServiceClient } from '@google-cloud/language'

export const analyzeText = async (text: string) => {
  const client = new LanguageServiceClient()

  const document = {
    content: text,
    type: 'PLAIN_TEXT' as const,
  }

  const [result] = await client.analyzeEntities({ document })
  const { entities } = result

  const entityList = entities.filter(
    (entity) => entity.type === 'CONSUMER_GOOD'
  )
}
```

省略 ...

# 抽出文字列から商品名を特定①

抽出結果は以下の通り、この情報を元に商品名を特定する

```
[  
  "38%", → UNKNOWN  
  "\\売上/", → UNKNOWN  
  "No.", → UNKNOWN  
  "※インテージSRI+", → UNKNOWN  
  "一般用ウェッタティッシュ", → UNKNOWN  
  "累計販売金額", → UNKNOWN  
  "207年\\月20B", → UNKNOWN  
  "▲ここまで引っぱってご使用ください。", → UNKNOWN  
  "シルコット", → UNKNOWN  
  "ウェットティッシュ", → CONSUMER_GOOD  
  "、純水99%", → UNKNOWN  
  "手・口まわりの汚れに", → UNKNOWN  
  "メイク時の手指の拭き取りに デ",  
  "コアウォーター", → UNKNOWN  
  "ノンアルコール" → UNKNOWN  
]
```

# 抽出文字列から商品名を特定①

抽出結果は以下の通り、この情報を元に商品名を特定する

```
[  
  "38%", → UNKNOWN  
  "\\売上/", → UNKNOWN  
  "No.", → UNKNOWN  
  "※インテージSRI+", → UNKNOWN  
  "一般用ウェッタティッシュ", → UNKNOWN  
  "累計販売金額", → UNKNOWN  
  "207年\\月20B", → UNKNOWN  
  "▲ここまで引っぱってご使用ください。", → UNKNOWN  
  "シルコット", → UNKNOWN  
  "ウェットティッシュ", → CONSUMER_GOOD  
  "、純水99%", → UNKNOWN  
  "手・口まわりの汚れに", → UNKNOWN  
  "メイク時の手指の拭き取りに デ",  
  "コアウォーター", → UNKNOWN  
  "ノンアルコール" → UNKNOWN  
]
```

## 商品検索の自動取得①

先ほど抽出した商品名をYahoo!ショッピングAPIで検索して商品情報を取得

ヒット商品の中から、商品名以外のキーワードが一致する商品以外をフィルタリングする

## 商品検索の自動取得②

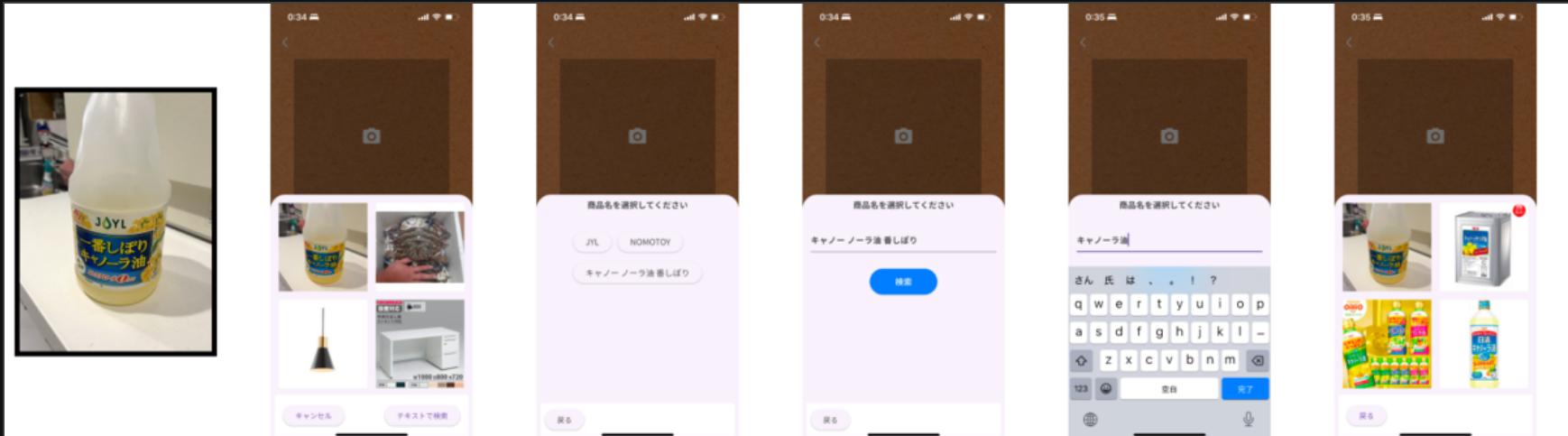
取得できた商品情報を表示させて、ユーザーに選択させるUIを表示させる

自動検索で商品が取得できなかった場合のために、手動検索の機能も用意している



# 商品の手動検索機能

- 1. 「テキストで検索」ボタンを押下
- 2. 抽出されたテキスト一覧から商品名を選択または修正
- 3. 検索ボタンをタッチ



# デモ

動画

# まとめ

- 今回の仕組みの実装で、8-9割の商品情報を自動取得できるようになった
- 実際に運用してみて以下の課題もあるので
  - フォントやデザインの影響でOCRの精度が低下する場合がある
  - ショッピングAPIでヒットしない商品がある
- OCR + AIの可能性は工夫次第で、色々活用できそう

ご清聴ありがとうございました 🎉