

令和5年度 特別研究報告書

画像認識による冷蔵庫内食品管理と
レシピの提案についての検討

龍谷大学 先端理工学部 知能情報メディア課程

Y200175 畑中智貴

指導教員 三好力教授

内容梗概

現代社会ではインターネットが普及していて便利な生活ができる環境である。例えば、コンビニエンスストアやスーパーマーケットなどで、商品を買う時にはセルフレジを使って決済を行う事が増えてきている。今までは従業員が商品の一つ一つ商品のバーコードをスキャンしていたが現在では自分で買いたい商品のバーコードをスキャンして決済を完了するような事がスーパーマーケットやコンビニエンスストアなどで増えている。現在ではコンビニエンスストアやスーパーマーケットなどで行われているサービスにスマートフォンで商品のバーコードをスキャンし、専用レジで会計するレジがある。このようにインターネットが普及していて便利な生活をする事ができる一方で環境問題がたくさんある。例えば、食材を買って冷蔵庫に入れているが食材を残してしまい捨ててしまうことが増えている。このようなことを食品ロスと言って食料の多くを海外からの輸入に依存しているにも関わらず、これだけ多くの食品を廃棄することは、経済的にも大きな無駄であると考えている。このような問題を解決するために冷蔵庫に入っている食材を管理するシステムを開発していきたいと考えている。食品ロスを減らすためには効率的に料理ができるために料理を提案することだと考えている。冷蔵庫に入れて使用せずに捨てるようなことをしないために、アプリケーション内で冷蔵庫に入れてどのくらい日がたったかを把握するためにスマートフォンに通知する機能が必要である。コンビニエンスストアやスーパーマーケットで買った食材を自動的に管理できるシステムが必要だと考えている。自動的に商品を管理するために今回はカメラを使用して一つの商品をカメラに写すだけでアプリケーション内に追加できるシステムを開発する。商品を冷蔵庫に入れるだけで自動的にバーコードを読み取ることができるためにパソコンとスマートフォンのカメラを用いてアプリケーションに登録できるかを検証する。また、食材には野菜と果物のようにバーコードが貼られていない食材がたくさんある。バーコードが貼られていない商品を識別するために今回はインターネットで野菜と果物の画像を取ってきてその画像で学習データを作成してデジタルカメラで取ってきた画像を比較してどのくらいの確率で識別できるかを検証する。

目次

第1章 緒言	1
第2章 既存技術	2
2.1 バーコード.....	2
2.2 QRコード.....	3
2.3 レジゴー.....	4
2.4 冷蔵庫管理アプリケーション.....	5
第3章 提案手法	6
3.1 提案するシステムの概要.....	6
3.2 QRコードを簡単に作れるアプリケーション.....	7
3.3 ウェブ上で冷蔵庫の中身を管理するアプリケーションの作成.....	8
3.4 料理を提案する機能.....	10
第4章 実験内容・結果・考察	11
4.1 カメラを用いてバーコード、QRコードの読み取り実験.....	11
4.1.1 実験内容.....	11
4.1.2 実験結果・考察.....	12
4.2 野菜、果物の静止画像のマッチング実験.....	16
4.2.1 実験内容.....	16
4.2.2 実験結果・考察.....	17
4.3 レシピの料理データ.....	20
4.3.1 実験内容.....	20
4.3.2 実験結果.....	21
第5章 結論	23
謝辞	24
参考文献.....	25
付録	26

第1章 緒言

最近ではコンビニエンスストアやスーパーマーケットなどで、商品を買う時にはセルフレジを使って決済を行う事が増えてきている。今までは従業員が商品の一つ一つ商品のバーコードをスキャンしていたが現在では自分で買いたい商品のバーコードをスキャンして決済を完了するようなシステムが増えてきている。イオンモールの食料品売り場にはレジゴーというアプリケーションがある。[1]レジでの混み具合を減らすためにスマートフォンのカメラを使ってバーコードを読み取るようにできるシステムである。具体的には今までは商品をかごに入れていたが、商品をかごに入れる前にスマートフォンのカメラ機能を使ってバーコードを読み取ると買いたい商品を登録することができる。会計は登録しているアプリケーションがレジと連動しているのでQRコードを読み取ると会計ができる仕組みになっている。会計が終了した後はスマートフォンが提示しているQRコードを読み取ると全ての商品履歴のデータを削除することができる。このようにインターネットが普及していることによって便利なシステムが増えているため今回は冷蔵庫の中身の商品を自動的にアプリケーション内で登録することができるシステムを提案する。

一方、環境問題として食品ロスが増えている。食品ロス[2]とは、まだ食べられるのに廃棄される食品のことである。日本の食品ロスは643万トン。日本人1人当たり年間約51キログラムにもものぼる。日本の食料自給率(カロリーベース)は2017年度で38%(消費者庁)。食料の多くを海外からの輸入に依存しているにも関わらず、これだけ多くの食品を廃棄することは、経済的にも大きな無駄だ。家庭から出る食品ロスは平成29年に消費者庁が行った調査は、家庭系の食品ロスの原因を明らかにした。捨てた理由として、(1) 食べ残し 57%、(2) 傷んでいた 23%、(3) 期限切れ 11%(賞味期限切れ6%、消費期限切れ5%)の順で多いことが分かった。(図1.1) 食品ロスを減らすために冷蔵庫の中身を管理できるシステムの開発や食べ残しが無くなるために料理を提案することによって減らすことができるのではないかと考えている。食品ロスを減らすためにIoTシステムを活かして食品管理できるシステムと料理を提案できるシステムを提案して、気軽に冷蔵庫の中の商品を管理できるようなシステムを考察する。

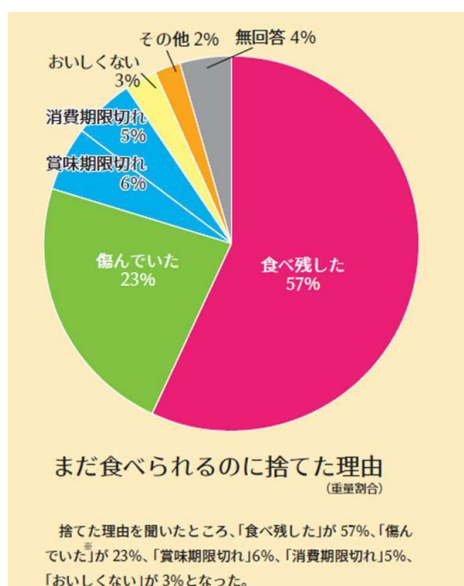


図1.1 家庭から出る食品ロスについての消費者庁が行った調査

第2章 既存技術

2.1 バーコード

バーコード[7]とは、バーとスペースの組合せにより、数字や文字などを機械が読み取れる形で表現したものである。今日、スーパーやコンビニエンスストアなどで販売されている商品のパッケージに、下の絵のような縞模様をよく見かける。バーコードは、太さの異なるバーとスペースの組合せにより構成されている。バーコードはJANコード、Interleaved 2 of 5、Codabar(NW-7)、CODE39、CODE128、GS1 DataBar(RSS)の6種類ある。その中でも共通商品コードは、UPCとEANに大きく分けられている。UPCとはアメリカで規格化された共通商品コードであり、EANとはUPCをもとに欧州で規格化された共通商品コードである。日本では、EANの1つとしてJAN(Japanese Article Number)が用いられている。EAN(JAN)は標準13桁、短縮8桁で構成され、UPCは12桁と7桁です(チェックデジット含む)EAN/UPCは国際標準化されており(ISO/IEC15420)JIS規格(JISX0507)にも制定されている。日本では、食料品、雑貨品などに付けられており、主にPOS(Point Of Sales:販売時点情報管理)に用いられている。共通商品コードの構成は万国共通で、各国それぞれに国コードが割り当てられていて例えば、日本は「49」と「45」が登録されている。図2.1はブラックコーヒー(ペットボトル)のバーコードの図を示す。



図2.1 ブラックコーヒーのバーコードの表示

バーコードを読み取るためにはバーコードリーダーと言う製品を使用している。バーコードリーダーとは、バーコードを光学的に検知しバー(黒)とスペース(白)を解析して、上位端末(コンピュータなど)が識別可能な電気信号あるいはキャラクタ(文字)に変換する装置である。バーコードリーダーは光学部品、使い方、形状などから複数の種類がある。現在ではバーコードリーダーを使用してバーコード決済ができる店頭レジが増えている。バーコード決済とは、近年普及が進んでいるキャッシュレス決済手段のひとつである。レジ端末やスマホ決済アプリでバーコードを読み取ることで、現金の代わりとして支払いを済ませることができる。ス

マホだけで支払いを完結させることができるので、財布を出す必要がなくスピーディーに会計を終えられるのが特徴である。バーコード決済とはユーザーがスマートフォンのアプリ上にQRコードやバーコードを表示して、店側がバーコードリーダーで読み取って支払う、あるいは店頭レジに掲げられている専用QRコード(二次元コード)をスマートフォンのカメラで読み取って支払いをするというものである。このほかにも、水道料金や通信費などのコンビニ払込票が送られてきた場合に、アプリで払込票に掲載されているバーコードを読み込んで支払う方法もある。バーコードを読み取る場合、バーコードリーダーを使用して読み取ることができる。

2.2 QRコード

QRコード(キューアールコード)[6]は、1994年(平成6年)に日本・愛知県の自動車部品メーカーであるデンソーの開発部門(現在は分社化してデンソーウェーブ)が発明したマトリックス型二次元コードである。データ読み取りや店頭決済用コードとして世界中で多用されている。QRは Quick Response の頭字語であり、高速読み取りを目的の一つとしている名称である。QRコードはデンソーウェーブの登録商標である。例えば、インターネットのURLを参照するためにQRコードを作成してそのQRコードをスマートフォンのカメラなどで読み取ると作成したURLに飛ぶことやライブやスポーツ観戦などスタジアムでの席をインターネットでの予約するとQRコードを発行してそのQRコードを会場で見せると席の予約が完了するためにQRコードが使われています。下の図2.2はQRコードの図を示す。



図2.3.1 QRコードの表示

QRコードを読み取るためにはスマートフォンの機能がある。iPhoneでは設定を開いてコントロールセンサーを開くとコードスキャナーというアプリケーションを追加する必要がある。追加するとホーム画面に戻り右上にある電波状態やiPhoneの充電のパーセントのところを下にスライドすると図2.3の画面になります。図2.3の画面の右下のアイコンをタップするとQRコードを読み取るアプリケーションを開くことができる。その他にもappleストアからQRコードを読み取る既存のアプリケーションがある。



図2.3 コントロールセンサーの画面表示

2.3 レジゴー

レジゴー[1]とはお買い物しながらセルフレジができるシステムである。一人で商品のバーコードを読み取って簡単にアプリケーション内で登録することができる。また、バーコードがない商品はアプリケーション内で入力することで登録することができる。買いたい商品を登録することで合計金額が分かるため、事前に計画を立てて買い物することができるため便利である。(図2.4)

1. スマートフォンの位置情報の取得を許可にしてレジゴーを使いたいお店を選択。
2. 買い物する商品のバーコードをスマートフォンのカメラを用いてバーコードをスキャンすることでアプリケーション内で買った商品を登録することができる。商品を登録したがいない商品を買ってしまった場合も登録した商品を削除する機能がある。
3. 買いたい商品をアプリケーション内で全て登録することができれば支払いのボタンを押すと支払いコード(QRコード)が出力される。
4. 出力されたコードをセルフレジ専用のレジの前にあるコードを読み取るシステムに出力されたコードを読み取ると会計するレジとつながり一人で会計をすることができる。
5. 会計が終了するとスマートフォンに表示されるコードをゲートにあるシステムにスマートフォンの画面をかざすと買い物が完了する。



図2.4 合計金額を知ることができる画面

2.4 冷蔵庫管理アプリケーション

冷蔵庫管理アプリケーションで一番人気があるアプリケーションは現在では献立や料理を提案することができるpecco(ぺっこ)というアプリケーションである。機能としては主に三つある。

1. 今日食べたものをレポートして摂取カロリーを把握することができる
2. 家にある食材を記入することができる
3. ある食材を用いて料理を提案することができる

食材を管理するために野菜、肉類、魚類、主食・粉物、調味料、スパイス、卵・乳製品・豆類、デザート、その他に分けて記入することができる。料理を提案することや食材を探すために検索エンジンを使用することができる。pecco(ぺっこ)と言う冷蔵庫管理アプリケーションのホーム画面を示す。(図2.5.1)



図2.4 peccoのホーム画面

第3章 提案手法

3.1 提案するシステムの概要

第2章で述べた既存技術の問題を解決するために三つのシステムを提案する。

1. 冷蔵庫の中身を管理するアプリケーション内で入力せずに商品を登録するために冷蔵庫に内蔵しているカメラを使用して自動的に登録できるシステムを提案する。
2. 自宅以外にいても冷蔵庫の中身を把握するためにインターネット上で管理するシステムを提案する。また、家族で暮らしている人がアプリケーションを共有することができるシステムも提供する。
3. 食品ロスを解決するために計画的に買い物できるように料理を提案するシステムを提案する。

具体的な全体のシステム概要として以下のシステムを提案する。

カメラを冷蔵庫の扉の部分に設置し、冷蔵庫内のカメラとスマートフォンの通信接続を行い、食品を認識すると以下のようにアプリケーションを起動する。

- 商品を購入すると冷蔵庫に内蔵しているカメラを使用して商品を識別する。
- ➔ カメラ機能を用いて商品を登録することができるが料理を提案する機能がない場合や料理を提案する事ができるがカメラ機能を用いて商品を登録する機能が無いアプリケーションがたくさんある。カメラ機能があるアプリケーションはバーコードの読み取り専用として使用されています。既存技術では商品を記入しなければならないがカメラを使用することで便利に管理できるようにする。
 - バーコードとQRコードの場合、カメラに写すと自動的に商品を登録する。
 - 野菜と果物(バーコードが記載していない商品)の場合、カメラに写すだけで商品を登録する。
 - 料理をして余ってしまった場合、QRコードを作成ページを開き何を作ったかどのくらい量が入っているかを記入する。
 - 記入することができれば自動的にQRコードが書かれたシールを発行する。
 - ➔ 料理をして出来上がった食品が残る可能性を考えて商品を管理する機能が存在していません。例えば、カレーやシチューなど鍋で作ったものはほとんど余ってしまった冷蔵庫に保存することができる。また、商品の袋やプラスチックなどを捨ててしまうとバーコードの存在が無くなるため再び冷蔵庫に入れてしまうと食品の質が落ちてしまい次第には長く保存してしまい食えることができなくなってしまう問題が起こってしまう可能性がある。
 - スマートフォンのQRコードを読み取るアプリケーションを使用すると食べていたものといつ冷蔵庫に入ったかを把握する。
 - 賞味期限を記入すると賞味期限が2日前に近づくと通知することと賞味期限が切れていることを通知する。
 - 簡単にできる料理を作るために検索できるようにする。

- 既存のアプリケーションでは作るレシピが限られていて既存の商品を使ってレシピを提案することができない。
- 楽天レシピ検索サイトを利用して簡単にできる料理を検索できるようにする。
- 期限が迫ると通知できるようにする
- 既存のアプリケーションでは賞味期限を記入できないし通知する機能がついていません。このようなことがあると冷蔵庫の中に放置している状態になってしまうためその食品は食べれるか把握できないので危険である。
- 賞味期限を記入していると二日前に賞味期限が迫っていることを通知して賞味期限が過ぎてしまうと賞味期限が切れていることを通知する。

3.2 QRコードを簡単に作れるアプリケーション

冷蔵庫の中身を共有するためのQRコードを作成する。Googleのスプレッドシート[8]を使って冷蔵庫管理システムを作成していつ入荷しているのかを把握する。第3章で問題としていた作った料理が余ってしまうことやバーコードが書かれていない野菜、果物などを管理するためにQRコードを作成。そのQRコードの中身はどのくらいの期間保存することができるのかを把握するために、使い捨てのQRコードを作成する。具体的にはGoogleColaboratoryを使用して始めに何の具材が入っているかを入力すると次にどのくらいの量が残っているかを入力する。(図3.1、図3.2)最終的にできたQRコードは商品と作った日にちを記録しているため商品だけではなくいつQRコードが作成されたかを知ることができる。

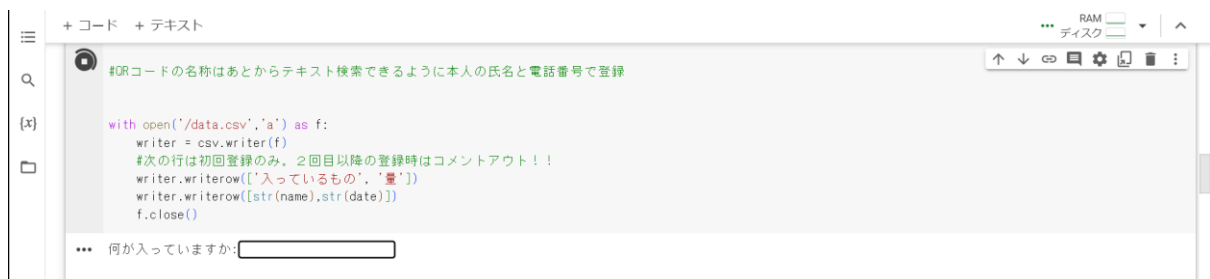


図3.1 QRコードを作成するために具材を入力する場合



図3.2 QRコードを作成するために量を入力する場合

実際にQRコードをスマートフォンで読み取った場合、図3.3のように読み取ることができる。ウェブ上で作成できたQRコードはコピー機を使用してQRコードの画像をシールに印刷する。

将来的には自動的に商品とどのくらい冷蔵庫に保存できるかを出力できるシステムができると便利になると考えている。



図3.3 作ったQRコードを読み取った場合

3.3 ウェブ上で冷蔵庫の中身を管理するアプリケーションの作成

一人暮らしの人対象ではなく家族がいる人のためのアプリケーションを作成する。一人暮らしの人も同様に簡単にアプリケーションを使用できるようにする。Googleのスプレッドシート[8]を使用してアプリケーションを作成する。GoogleのスプレッドシートはMicrosoft 365に入っているExcelと同じように一つ一つのセルを使って数字の表計算や分類などができる仕様になっている。GoogleスプレッドシートがExcelに比べて圧倒的に有利な点は、同時編集が容易である。Excelは共同編集機能が限定的で、ファイル共有の手間もかかる。Googleスプレッドシートは共有・同時編集を前提としており、招待リンクからアクセスした複数名のユーザーで快適なオンライン同時編集が可能。編集内容はリアルタイムで全ての共同編集者に反映され、他のユーザーがどのセルで作業しているかもひと目で分かる。セル単位でコメントを作成・追記・返信する機能もあるため、共同作業に最適。

Googleのスプレッドシートを使って賞味期限や商品名、入荷日、在庫数、前回買った値段を把握する。スマートフォンでも編集可能にするためにiPhoneであればappleストアで無料のアプリケーションをダウンロードが必要。管理することだけではなく料理を提案する機能が便利だと思い楽天レシピ検索サイトのURLを貼り付けていつでもURLに移動できるように作成。また、料理を提案できる機能だけではなかなか使用する目的が少ないため、買い物する前や休日の時に冷蔵庫の中身と比較して一週間の献立を記入できる欄を作成。一人暮らしの人以外にも家族で暮らしている人も便利に使用するためにアプリケーションを共有することで自分以外の方が買い物している場合、連絡せずに効率的に買い物をすることができる。食材を使用したことを把握するためにチェックリストを作成。チェックマークをつけると使用済みだと分かるようにした。将来的にはバーコードを読み取ることで入荷日からどのくらい冷蔵庫に保存できるかを自動的に登録できるようなシステムを開発をすることが必要だ。

Googleのスプレッドシート[8]の拡張機能を使用する。Google Apps Script[9]とは、Googleが提供しているプログラミング言語である。それぞれの頭文字をとってGAS(ガス)とも呼ばれている。Google Apps Scriptの基本的な文法は、JavaScriptの文法を踏襲しており、JavaScriptを学習済みなら、ほとんどそのまま利用できる。それに加えて、GmailやGoogleカレンダー、Googleドライブ、Google翻訳など、Googleのサービスを扱うための機能を提供しており、それらのGoogleサービスを連携、自動化させて活用することができる。さらにTwitterやLINEなど、Google以外のサービスとも連携が可能で、このような自動化や連携により、業務の効率化を図れる。今回はGASと言う拡張機能を使用してGmailやemailに賞味期限が近づいていることと賞味期限が切れていることを通知するようにする。賞味期限をスプレッドシートに記入し、本日の日付と賞味期限を見比べて2日前であれば通知できるようにしました。また、賞味期限が過ぎている時も同じように通知するように設定。スマートフォンの場合、次の図3.4のように通知がきて、パソコンの場合は図3.5に通知される。



図3.4 スマートフォンに通知が来た場合

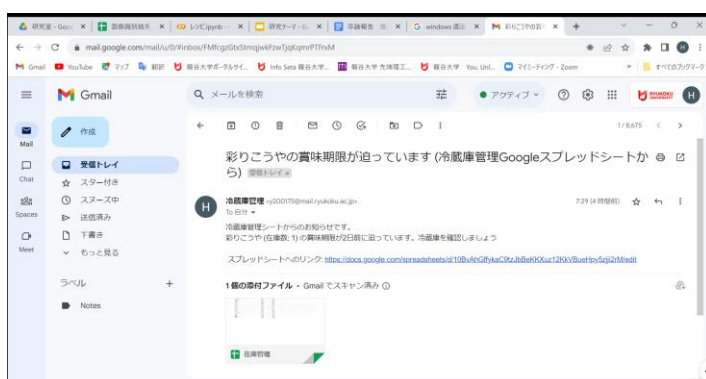


図3.5 パソコンに通知が来た場合

3.4 料理を提案する機能

緒言で話した食品ロス問題を解決するためには管理している冷蔵庫の中の商品を利用するために料理を提案するシステムを提案する。冷蔵庫の中身を管理することだけではなく、既存のアプリケーション(ぺっこ)のように料理を提案できるように楽天レシピ検索サイトのURLを添付。既存のアプリケーション(ぺっこ)では既存の商品を使ったレシピを提案することができないため今回は楽天レシピ検索サイトを利用する。楽天レシピ検索サイトでは20万以上のレシピが登録されているため簡単なレシピから難しいレシピまで幅広い料理を提案することができる。作りたいレシピの画像以外にも作り方や材料費までサイトに載っているため(図3.6、図3.7)冷蔵庫の中身の商品と比べて何が足りないかを知ることができるため間違えて同じ商品を買ってしまうリスクがありません。例えば、ハンバーグを作ってみようと思って買い物をしているとオンライン上でレシピを見ているため買い物中でも効率的に買い物をすることができる。

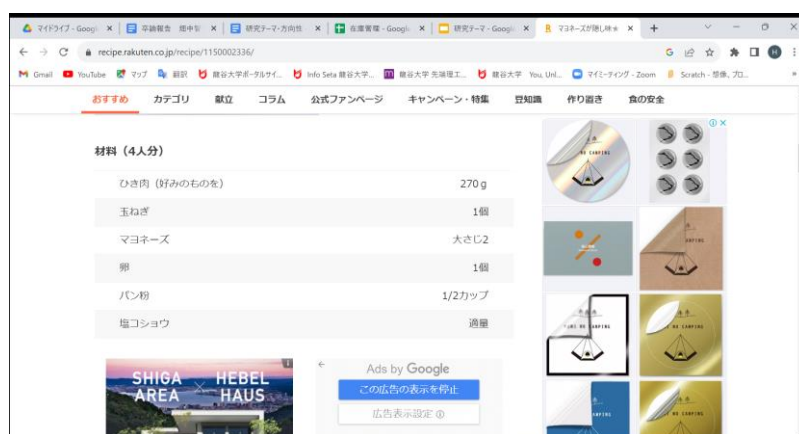


図3.6 楽天レシピ検索サイト ハンバーグの材料



図3.7 楽天レシピ検索サイト ハンバーグの作り方

第4章 実験内容・結果・考察

4.1 カメラを用いてバーコード、QRコードの読み取り実験

冷蔵庫の中身に入れるだけで自動的にバーコードを認識してアプリケーション内に入れた商品を把握することができるアプリケーションを提案するために実験を行う。そのためにはパソコンとスマートフォンのカメラを使って簡単にバーコードを読み取ることができるかを検証する。また、読み取ったバーコードの番号が商品と連携しているかを検証する。現在でもQRコードを読み取るためにスマートフォンではカメラを使用していますがバーコードを読み取る機能は既存のアプリケーションを使用しなければ読み取ることができません。今回はバーコードで商品検索という既存のアプリケーションを使ってどのくらいの確率で読み取ることができるのかを検証する。

4.1.1 実験内容

バーコードやQRコードを読み取るためにはubuntu環境を使ってvirtual studio code[9]というアプリケーションを使用してPythonを記述、実行を行う。Visual Studio Code は、マイクロソフトが開発したオープンソースのコードエディタで、軽量かつ高速、高機能が特徴である。多くのプログラミング言語に対応し、Git との統合やデバッグ機能なども備えている。また、クロスプラットフォーム対応であり、Windows、Mac、Linux で使用できるのが魅力。Visual Studio Code はシンプルで直感的な操作性を提供し、豊富な拡張機能やコミュニティのサポートにより、初心者から上級者まで幅広い開発者が利用できるエディタとしてアプリ開発現場で広く利用されている。パソコンの方ではvirtualstudiocode[9]を使用してバーコードとQRコードを読み取ることができるようにPythonで記述して作成する。パソコンでは数字を出力できるが商品を出力することができないため別のPythonファイルで商品検索できるアプリケーションを作成する。具体的には楽天APIを利用して数字を入力すると商品が出力できるように作成する。また、楽天API以外でもウェブ上で商品検索できるサイトがあるので使用する。パソコンのカメラで実際に付いている商品のバーコードを読み取るために、図4.1のようにカメラをかざすだけで番号が出力できるように実験を行う。QRコードも同様にカメラ機能を使用してQRコードの中身を出力できるかを検証する。



図4.1 パソコンでバーコードを読み取る画像

スマートフォンの方は読み取ることができる商品を入力する既存のアプリケーションを使用してパソコンで出力された結果とスマートフォンのアプリケーションで出力された結果が同じであるかを検証します。また、楽天APIとウェブ上の商品検索結果と実際の商品を比較して合っているのかを検証する。

4.1.2 実験結果・考察

始めにパソコンとスマートフォンのカメラの性能の比較の結果は次の表4.1になった。

バーコード読み取り結果			
実際の商品	パソコンのカメラ	スマートフォンのカメラ	商品の形
伊藤園 おーいお茶	○	○	ペットボトル
大粒ラムネ強炭酸シュワコーラ	○	○	袋
ごま昆布	×	○	小袋
マルちゃん 焼きそば	○	○	袋
ガトーショコラ お菓子	○	○	箱
シーザードレッシング	○	○	ボトル
チキンナゲット	×	○	袋
あらびきポークウインナー	×	○	袋
ロースハム	×	○	袋
CCレモン	○	○	ペットボトル
ブラックコーヒー	×	○	缶

表4.1 パソコンとスマートフォンのカメラの性能の実験結果

考察

表4.1を見てスマートフォンのカメラは全ての商品を読み取ることができましたがパソコンのカメラでは全ての商品を読み取ることができなかった。箱やペットボトル(図4.2、図4.3)は読み取ることができたが凸凹な缶や開封済みの小袋などを読み取ることができなかった。(図4.4、図4.5)また、角度を変えて読み取ることができるのかを検証した結果、スマートフォンのカメラでは角度関係なく読み取ることができたがパソコンのカメラでは読み取ることができなかった。実験結果としてはパソコンのカメラよりスマートフォンのカメラの方がバーコードをしっかりと読み取ることができるとして出た。確率としてはスマートフォンのカメラの性能は100%読み取ることができて、パソコンのカメラでは55%読み取ることができるという結果が出た。今後、画像処理技術を用いる場合はスマートフォンのカメラや小型カメラを使用することで新しい画像処理技術を用いたIoT製品を開発することができると考えている。

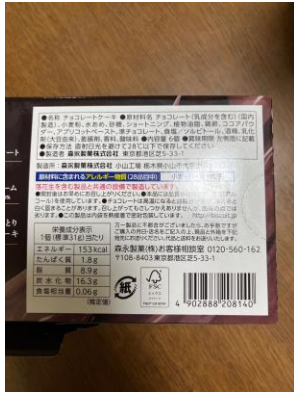


図4.2 ガトーショコラの箱のバーコードの画像



図4.3 ペットボトルのバーコードの画像



図4.4 凸凹な缶のバーコードの画像



図4.5 開封済みの小袋のバーコードの画像

次に実際の商品とウェブ上の商品検索結果とバーコードで読み取ったバーコードの番号を楽天APIで商品検索結果を比較した実験結果は次の表4.2、表4.3のような結果になった。

バーコード読み取り結果			
バーコード番号	pythonでの商品検索	ウェブブラウザでの商品検索	実際の商品名
4902102119443	爽健美茶 PET 600ml48本	爽健美茶 600ml	爽健美茶
4901777333994	◆サントリー ペプシ生ゼロ 1.5L 【8個セット】	ペプシ ジャパンコーラ ゼロ	ペプシコーラゼ ロ
4549414516852	なし	なし	さつまいも煮 トップバリュ
4970045022304	さば へしこ 鯖のへしこ へしこ鯖 半身 1切れ 珍味 福井 鯖へしこ	なし	鯖のへしこ
4902560171014	はごろもフーズ 朝からフルーツ パイミン 190g 1セット (3缶)	朝からフルーツ パイミ ン	朝からフルーツ
4902220092215	なし	なし	ごぼうとこんに ゃく煮
4902553036559	【チルド(冷蔵)商品】フジッコ お かず畑 おばんざい小鉢 彩りこう や	おばんざい小鉢 彩りこうや	彩りこうや
49212877	冷蔵 雪印メグミルク ナチュレ恵 megumi 400g×8個	ナチュレ 恵 megumi 400g	プレーンヨーグ ルト
4902016034641	ピククルス ご飯がススムキムチ 2 00g	ご飯がススム キムチ	キムチ
4975698004400	なし	なし	山椒ちりめん
4903050110841	冷蔵 雪印 北海道バター ミニパッ ク	雪印北海道バター ミニパック	雪印北海道バタ ー
4978288800079	なし	なし	たまり漬
4902777028910	ベストスリー袋 184 g 1袋	meiji ベストスリー (33枚入)	チョコレート
4902713127738	マルコメ 丸の内タニタ食堂の減 塩みそ 650g	丸の内 タニタ食堂の減塩みそ	味噌
4978033702344	なし	新食感 崩れにくい麻婆豆腐	麻婆豆腐
4902106012061	Mizkan 金のつぶ たまご醤油たれ	金のつぶ たれたっぷり! たま	納豆

		ご醤油たれ	
4549414485929	なし	なし	味付け油揚げ
4902553036719	【チルド(冷蔵)商品】フジッコ おかず畑 おぼんざい小鉢 なめらか白和え	フジッコ白和え. (おぼんざい小鉢)	白和え
4902553024020	フジッコ おまめさん 豆小鉢 やさい豆	おまめさん 豆小鉢 やさい豆 65g×2	やさい豆
4904566582184	なし	なし	白あえベースねりごま
4902220125142	なし	なし	焼鮭ほぐし
4977890125716	なし	なし	天理ラーメン
4548780560599	日清食品 日清のナポリタン 2人前	日清のナポリタン 135g×2人前	日清ナポリタン
4902402869499	ハウス こくまるハヤシ	こくまるハヤシ	こくまるハヤシ
4902402853887	ハウス食品 バーモントカレー 中辛	バーモントカレー 中辛 12皿分 (6皿×2) 230g	バーモントカレー 中辛
4901259500418	なし	なし	ほそめパン粉
4520799320173	なし	なし	すりごま
4520799241973	なし	北海道産片栗粉	片栗粉
4979250152479	なし	フランクリンアソート4種	ウインナー
4902586342894	なし	なし	ロースハム
4901231304881	なし	ポークビッツ	ポークビッツ
4902106860655	ミツカン ごましゃぶ 250ml	ごましゃぶ	ごましゃぶ
4901306097526	カゴメ 醸熟ソースとんかつ 500ml	カゴメ 醸熟 とんかつソース 500ml	とんかつソース
4971666488623	江崎グリコ アーモンド効果 砂糖不使用 200ml	アーモンド効果 砂糖不使用	アーモンド効果
4979618102436	なし	なし	うなぎのたれ
4903001922066	ヤマサ醤油 ヤマサ さしみしょうゆ 200ml	さしみしょうゆ	さしみしょうゆ
4902106291947	ミツカン すし酢 昆布だし入り	すし酢 昆布だし入り 360ml	すし酢昆布だし
4901577073540	キューピー コールスロードレッシング	コールスロードレッシング	コールスロードレッシング
4905132159878	なし	なし	胡麻ドレッシング
4903065241004	ヤマキ 割烹白だし500ml	割烹白だし 500ml	ヤマキ 割烹白だし500ml
4901577047275	キューピー パスタを手作りオイルソースしょうゆ&ペパー150ml	パスタを手作りオイルソースしょうゆ&ペパー	オイルソースしょうゆ&ペパー150ml
4903001941524	ヤマサ醤油 昆布ぼん酢 500ml	ヤマサ昆布ぼん酢 500ml	昆布ボン酢
4981099602206	なし	ごちそう生活 白玉	卵
2080003004012	なし	なし	レタス
4945167001032	なし	ほぐしめじ	しめじ

表4.2 商品検索結果の比較実験の結果

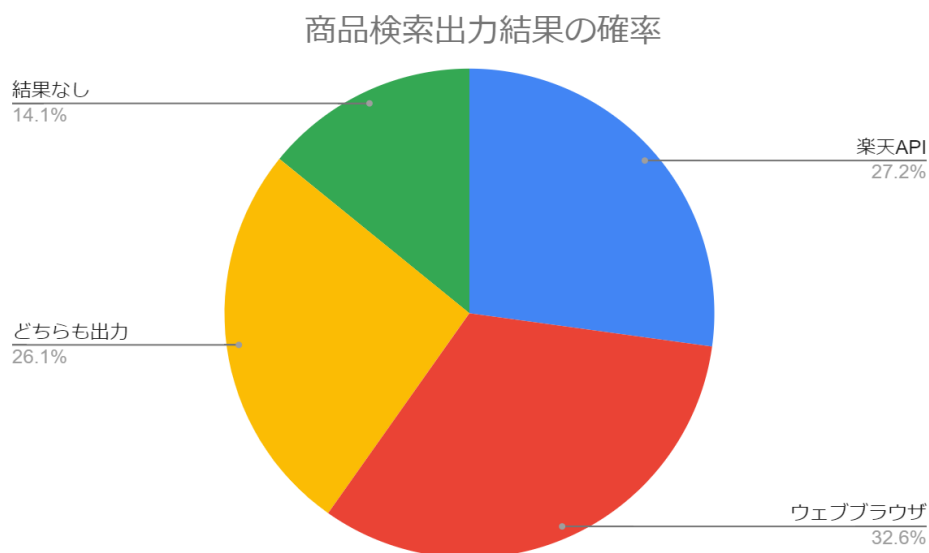


表4.3 商品検索出力結果の確率

考察

数字が出力されていても全国で商品が登録されていない商品がたくさんあるため商品検索をしても出力されない結果もあることが分かった。また、地元のお土産の商品は楽天APIの方が出力されることがある。新商品の場合はウェブ上の商品の方が出力されていることがある。表4.3を見てウェブブラウザの方が番号を入力したら出力される確率が高いため楽天APIよりウェブブラウザで商品登録している商品が多いことが分かった。

4.2 野菜、果物の静止画像のマッチング実験

4.1でバーコードを読み取る実験を行ったが、冷蔵庫に入れる商品の中にはバーコードが記載していない商品がある。例えば、袋に入っていない野菜や果物などがある。実験の目的は野菜や果物を自動的に識別できるようにすることで将来的にはリアルタイムで冷蔵庫の中に内蔵しているカメラの機能として自動的に野菜や果物を識別することができて識別したデータを登録できるようにしていきたいと考えている。

4.2.1 実験内容

事前に野菜や果物の画像をデジタルカメラを使用して野菜や果物の静止画像を撮る。撮った静止画像はGoogleドライブで画像を保存する。画像を読み取るためのオープンソースとしてopencvをインストールしてPythonを記述する。技術としてはGoogleColaboratoryと言うPythonを記述、実行できるサービスを使用する。野菜や果物を識別するためにGoogleColaboratoryを使用してオンライン上の画像を取得して画像を保存する。野菜や果物を識別するためにCNN(畳み込みニューラルネットワーク)を使用して画像を入力すると事前に学習していた

画像と比較して同じ結果になるか検証する。CNN(畳み込みニューラルネットワーク)を完成するためには膨大な量のデータを取得しなければなりません。実際には一つの野菜に対して約100枚程度の画像を取得して学習して静止画像と比較する。

GoogleColaboratoryのノートブック作成時には、GPUではないため、下記操作にてGPUを使えるようにする。上部メニューのランタイムから ランタイムのタイプを変更を選択。ハードウェアアクセラレータを None から GPU に変更して保存。野菜の25種類の学習データを用いて野菜の識別を行う。野菜の種類はトマト、アスパラガス、ブロッコリー、大根、枝豆、えのき、ごぼう、ゴーヤ、白菜、じゃがいも、かぼちゃ、キャベツ、きゅうり、水菜、なすび、長ネギ、人参、ピーマン、れんこん、レタス、さつまいも、椎茸、玉ねぎで事前に学習データを作成。また、人気の野菜ランキングのトップ5に入る野菜だけで識別を行う。野菜の人気ランキングのトップ5はキャベツ、玉ねぎ、じゃがいも、大根、トマトの5種類である。果物は10種類の学習データを用いて果物を識別する。果物の種類はバナナ、イチゴ、みかん、ブドウ、桃、キウイフルーツ、スイカ、りんご、パイナップル、メロンで学習データを作成。

4.2.2 実験結果・考察

野菜25種類の実験結果は次の表4.4のようになる。

野菜画像認識結果 (25種類の場合)				
画像の種類	正解	失敗	結果表示	人間での認識
IMG_3196.JPG	×	○	水菜	枝豆
IMG_3195.JPG	○	×	キャベツ	キャベツ
IMG_3194.JPG	○	×	じゃがいも	じゃがいも
IMG_3192.JPG	○	×	玉ねぎ	玉ねぎ
IMG_3193.JPG	○	×	じゃがいも	じゃがいも
IMG_3189.JPG	○	×	トマト	トマト
IMG_3186.JPG	×	○	ブロッコリー	レタス
IMG_3177.JPG	×	○	ネギ	玉ねぎ
IMG_3176.JPG	○	×	きゅうり	きゅうり
IMG_3175.JPG	×	○	なすび	椎茸
IMG_3165.JPG	×	○	えのき	玉ねぎ
IMG_3200.JPG	×	○	キャベツ	ピーマン
IMG_3216.JPG	×	○	水菜	ピーマン
IMG_3215.JPG	×	○	キャベツ	ピーマン
IMG_3214.JPG	○	×	トマト	トマト
IMG_3213.JPG	×	○	じゃがいも	玉ねぎ
IMG_3212.JPG	×	○	アスパラ	玉ねぎ
IMG_3211.JPG	○	×	人参	人参
IMG_3210.JPG	×	○	じゃがいも	キャベツ
IMG_3209.JPG	○	×	キャベツ	キャベツ
IMG_3208.JPG	×	○	キャベツ	大根

IMG_3207.JPG	×	○	かぼちゃ	レタス
IMG_3206.JPG	×	○	キャベツ	レタス
IMG_3205.JPG	×	○	ネギ	白菜
IMG_3204.JPG	×	○	キャベツ	白菜
IMG_3203.JPG	×	○	トマト	人参
IMG_3202.JPG	×	○	じゃがいも	玉ねぎ
IMG_3201.JPG	×	○	れんこん	じゃがいも

表4.4 野菜の25種類の識別結果

野菜5種類の実験結果は次のように表4.5のようになる。

野菜画像認識結果（5種類の場合）				
画像の種類	正解	失敗	結果表示	人間での認識
IMG_3195.JPG	○	×	キャベツ	キャベツ
IMG_3194.JPG	○	×	じゃがいも	じゃがいも
IMG_3192.JPG	○	×	玉ねぎ	玉ねぎ
IMG_3193.JPG	○	×	じゃがいも	じゃがいも
IMG_3189.JPG	○	×	トマト	トマト
IMG_3177.JPG	×	○	じゃがいも	玉ねぎ
IMG_3165.JPG	○	×	玉ねぎ	玉ねぎ
IMG_3214.JPG	○	×	トマト	トマト
IMG_3213.JPG	×	○	じゃがいも	玉ねぎ
IMG_3212.JPG	×	○	じゃがいも	玉ねぎ
IMG_3210.JPG	×	○	じゃがいも	キャベツ
IMG_3209.JPG	×	○	じゃがいも	キャベツ
IMG_3208.JPG	×	○	じゃがいも	大根
IMG_3203.JPG	×	○	じゃがいも	人参
IMG_3202.JPG	×	○	じゃがいも	玉ねぎ
IMG_3201.JPG	○	×	じゃがいも	じゃがいも

表4.5 野菜の5種類の識別結果

果物10種類の実験結果は表4.6のようになった。

果物画像認識結果（10種類の場合）				
画像の種類	成功	失敗	結果表示	人間の認識
IMG_3171.JPG	×	○	もも	りんご
IMG_3172.JPG	○	×	バナナ	バナナ
IMG_3191.JPG	×	○	みかん	パイナップル
IMG_3185.JPG	×	○	イチゴ	メロン
IMG_3187.JPG	○	×	ブドウ	ブドウ
IMG_3188.JPG	×	○	イチゴ	桃

IMG_3190.JPG	○	×	バナナ	バナナ
IMG_3197.JPG	○	×	りんご	りんご
IMG_3219.JPG	○	×	ブドウ	ブドウ
IMG_3220.JPG	×	○	バナナ	パイナップル

表4.6 果物10種類の識別結果

果物人気のある5種類の識別結果は表4.7のようになった。

果物画像認識結果（5種類の場合）				
IMG_3171.JPG	○	×	りんご	りんご
IMG_3172.JPG	○	×	バナナ	バナナ
IMG_3187.JPG	○	×	ブドウ	ブドウ
IMG_3190.JPG	○	×	バナナ	バナナ
IMG_3197.JPG	○	×	りんご	りんご
IMG_3219.JPG	×	○	バナナ	ブドウ

表4.7 果物5種類の識別結果

野菜と果物の識別結果の正解率をまとめると表4.8のようになった。

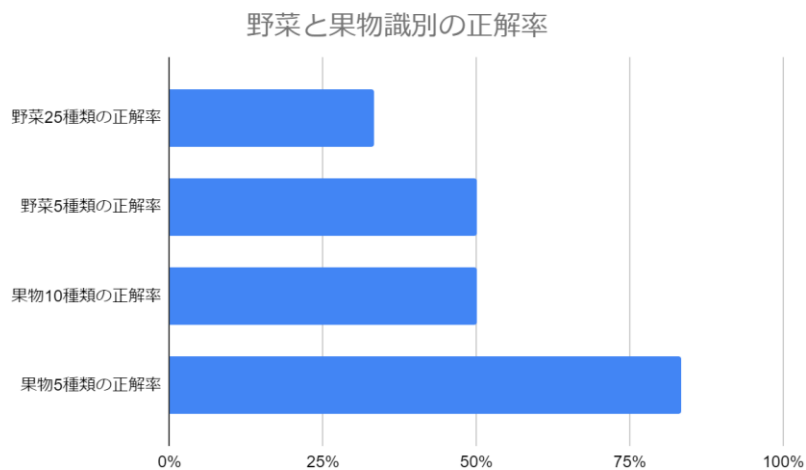


表4.8 識別結果の正解率のグラフ

考察

全体の結果を見て比較するデータが多いほど識別することが困難なことが分かった。表4.4を見て人間での識別と結果表示を比較してみるとしっかり識別できている野菜データがあればないデータも存在している。野菜のデータの種類を減らすことで表4.4と比べて正解の確率が上がっているが識別できない画像もたくさんあった。特に玉ねぎやキャベツの識別ができていないことが分かるがトマトはしっかり識別することができている。表4.6を見て野菜同様に人間での識別と結果表示を比較してみるとしっかり識別できている野菜データがあればないデータも存在している。また、袋に入っている画像(図4.6)やラップで巻いている画像

(図4.7)や使用している野菜や果物を識別することが難しいことが分かった。野菜や果物の識別が難しい場合、QRコードを作成して管理する方法で野菜や果物を自動的に識別せずに管理することができるのではないかと考えている。



図4.6 袋に入っている野菜の画像



図4.7 ラップを巻いている野菜の画像

4.3 レシピの料理データ

食品ロスとして問題となっている食べ残しを減らすために計画的に料理ができるために料理を提案するシステムを開発するためにどのようにしてレシピのデータを取得して提案しているかを調べる。インターネットが普及しているため料理を提案するアプリケーションが増えてきている中、今回は楽天レシピ検索サイトを利用する。現在、インターネット上にたくさんの情報がある中、料理を簡単に提案するためにどのようにしてデータを管理しているのかを把握する。

4.3.1 実験内容

楽天APIを使用するために無料でユーザーを登録してアプリIDを作成。作成後Google Colaboratoryを使用して新規ファイルを作成。データを自動的に取得できるようにプログラミング言語としてはPythonを使用して楽天APIを利用する。始めに楽天APIを使用して全ての料理レシピのデータを取得する。全ての料理データから調べたい料理データを取得するためにキーワードから料理を抽出するように作成する。

4.3.2 実験結果

全体の料理データから抽出した結果からキーワードを入力して出た料理データは次の表4.7のようになる。今回は全体の料理レシピのデータからご飯ものを抽出して炊き込みご飯の料理レシピのデータを取得する。

index	category1	category2	category3	categoryId	categoryName
4	14			14	ご飯もの
120	14	132		14-132	炊き込みご飯
858	14	132	1321	14-132-1321	栗ご飯
860	14	132	1322	14-132-1322	たけのご飯
862	14	132	2144	14-132-2144	とうもろこしご飯
864	14	132	1325	14-132-1325	松茸ご飯
867	14	132	2099	14-132-2099	ひじきご飯
868	14	132	1328	14-132-1328	かやくご飯
869	14	132	138	14-132-138	混ぜご飯
870	14	132	137	14-132-137	その他の炊き込みご飯
880	14	135	2146	14-135-2146	とろろ・麦とろご飯
1256	24	238	1047	24-238-1047	バーベキューのご飯もの

表4.7 ご飯ものを抽出した結果

次にご飯ものから炊き込みご飯のレシピを取得した結果を表4.8のように抽出したデータを取得する。表4.7ではcategoryIdが14-132とデータが抽出しているのでプログラムで14-132と入力して炊き込みご飯のデータを抽出する。

```
{'result': [{'foodImageUrl': 'https://image.space.rakuten.co.jp/d/strg/ctrl/3/89bc73e0b122856e5b1c0751a2be5209fe8712f2.84.9.3.3.jpg',
             'mediumImageUrl': 'https://image.space.rakuten.co.jp/d/strg/ctrl/3/89bc73e0b122856e5b1c0751a2be5209fe8712f2.84.9.3.3.jpg?thum=54',
             'nickname': 'momomo_',
             'pickup': 0,
             'rank': '1',
             'recipeCost': '300円前後',
             'recipeDescription': '鶏肉なしですが、食べごたえ抜群！！\n'
                                   '箸が止まらなくなる炊き込みご飯です。'\n
                                   'なんだかおばあちゃんを思い出すような昔ながらの間違いない味です。',
             'recipeId': 1850018095,
             'recipeIndication': '約1時間',
```



```
'recipeMaterial': ['米',
                  'ごぼう',
                  '人参',
                  'しいたけ',
                  '油揚げ',
                  '糸こんにゃく',
                  '●しょうゆ',
                  '●酒',
                  '●砂糖',
                  '●顆粒だし',
                  '水'],
'recipePublishday': '2020/12/30 09:19:44',
'recipeTitle': '3合用★おかわり確実！昔ながらの炊き込みご飯',
'recipeUrl': 'https://recipe.rakuten.co.jp/recipe/1850018095/,
```

表4.8 炊き込みご飯の料理データを抽出した結果

考察

楽天レシピのデータを取得できた。データを取得して複雑な仕組みになっていることから簡単なレシピデータを作成することができるのではないかと考えている。料理名と材料と一つの料理する画像を出力することで見栄えも良くなるのではないかと考えている。膨大なデータを管理するためにはカテゴリー別に識別することによってデータを絞っていることが分かった。冷蔵庫管理アプリケーションの機能として料理を提案するシステムを搭載しているが、楽天レシピのデータや他のレシピのデータを取得することで難しいレシピから簡単なレシピを提案するシステムができるのではないかと考えている。

第5章 結論

本研究では食品ロスの問題を解決するための冷蔵庫の中身を管理するためにカメラを用いて商品を識別できるシステムができるのではないかと考察してきた。

今回の実験ではカメラを使用して自動登録できるかを確認するためにバーコードを読み取る実験を行った。カメラを用いてバーコードを数字で読み取ることができたが商品名は出力されない商品もあった。バーコードを読み取ることができたが全国で商品が登録できていないことから今後はバーコードの商品登録したデータが簡単に取得できる機能が必要だと考えている。

バーコードがない野菜と果物はニューラルネットワークを用いて画像認識する実験を行った。今回の実験で識別できなかったラップを巻いている画像や使用済みの画像を識別できるようになると冷蔵庫内の管理方法として実用的であると考えている。

QRコードを作成するシステムを提案して上手く作成することができた。使用済みの商品や作ってしまった料理を管理するためにQRコードを作成してそのQRコードを貼り付けて使用するアプリケーションがあるとQRコードを読み取るだけで簡単に把握することができ、今後のシステムの機能として活用することができるのではないかと考えている。野菜と果物の管理方法も同様にQRコードを作成して野菜や果物を入れている袋に貼り付けると活用できるのではないかと考えている。

賞味期限を通知するシステムを提案して賞味期限を入力すると通知するシステムを作成することができた。既存の冷蔵庫管理アプリケーションでは賞味期限が近づくと通知する機能が存在していないことから、冷蔵庫に入れるだけでカメラが商品を読み取り、自動に賞味期限を設定する機能があると実用的であると考えている。

ホームページからレシピデータを取得することができたがレシピを提案することができなかった。食品ロス問題を解決するために料理を提案するシステムを検討して実際に楽天レシピのデータを取得して膨大なデータベースが存在していることが分かり、今後の機能としては冷蔵庫内の商品から簡単に料理を提案する機能が必要となる。この機能が実現できれば食品ロス問題を解決することに役立つのではないかと考えている。

謝辞

本研究を進めるにあたり、様々なご指導を頂きました三好力教授に深く感謝いたします。また、お忙しい中研究の過程で様々な助言を頂きました同研究室の皆様にも深く感謝いたします。

参考文献

[1]レジゴー | お買い物しながらセルフレジ

<https://www.regigo.jp/>

[2]エネルギーと環境 Vol.15 食品ロス問題 私たちに出来ること

https://ene-fro.com/article/ef120_a1/

[3]【基本を解説】画像認識とは？主な活用例5つ | 何ができる？その仕組みは？

<https://business.ntt-east.co.jp/content/cloudsolution/column-391.html>

[4]【Python×OpenCV】初めてのOpen CV(画像処理ライブラリ)ガイド

<https://www.klv.co.jp/corner/python-opencv-what-is-opencv.html>

[5]QRコード決済

<https://ja.wikipedia.org/wiki/QR%E3%82%B3%E3%83%BC%E3%83%89%E6%B1%BA%E6%B8%88>

[6]QRコード

<https://ja.wikipedia.org/wiki/QR%E3%82%B3%E3%83%BC%E3%83%89>

[7]バーコードとは

<https://www.denso-wave.com/ja/adcd/fundamental/barcode/barcode/index.html>

[8]Googleスプレッドシートの使い方は？

<https://www.itscom.co.jp/forbiz/column/cloud/9979/#:~:text=Google%E3%82%B9%E3%83%97%E3%83%AC%E3%83%83%E3%83%89%E3%82%B7%E3%83%BC%E3%83%88%E3%81%8CExcel%E3%81%AB%E6%AF%94%E3%81%B9%E3%81%A6%E5%9C%A7%E5%80%92%E7%9A%84,%E5%90%8C%E6%99%82%E7%B7%A8%E9%9B%86%E3%81%8C%E5%8F%AF%E8%83%BD%E3%81%A7%E3%81%99%E3%80%82>

[8]【Google Apps Script入門】GASでできることや連携でできることや連携できるツール、活用事例を解説！

<https://udemy.benesse.co.jp/development/system/intro-gas.html>

[9]1. Visual Studio Codeとは

<https://www.rworks.jp/cloud/azure/azure-column/azure-entry/28456/#:~:text=Visual%20Studio%20Code%20%E3%81%AF%E3%80%81%E3%83%9E%E3%82%A4%E3%82%AF%E3%83%AD%E3%82%BD%E3%83%95%E3%83%88,%E3%81%AA%E3%81%A9%E3%82%82%E5%82%99%E3%81%88%E3%81%A6%E3%81%84%E3%81%BE%E3%81%99%E3%80%82>

付録

1. QRコードとバーコードの読み取り

```
qrcode_barcode.py
import cv2
import sys
import json
from pyzbar.pyzbar import decode, ZBarSymbol

cap = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_SIMPLEX
BUF_FILE_PATH = 'stock.csv'
barcodes = []
qr_codes = []

while cap.isOpened():
    ret, frame = cap.read()

    if ret:
        d = decode(frame)
        # デコード
        value = decode(frame, symbols=[ZBarSymbol.QR_CODE])

        if value:
            for qr_code in value:
                # QRコード座標
                x, y, w, h = qr_code.rect

                # QRコードデータ
                QRcodeData = qr_code.data.decode('utf-8')

                if QRcodeData not in qr_codes:
                    qr_codes.append(QRcodeData)
                    font_color = (0, 0, 255)
                    with open(BUF_FILE_PATH, mode='a') as buf:
                        buf.write(QRcodeData + '\n')
                else:
                    font_color = (0, 154, 87)
                    frame = cv2.putText(frame, QRcodeData, (x, y-6), font, .3, (255, 0, 0), 1, cv2.LINE_AA)

                # バウンディングボックス
                cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 1)

            if d:
                for barcode in d:
                    barcodeData = barcode.data.decode('utf-8')

                    if barcodeData not in barcodes:
                        barcodes.append(barcodeData)
                        font_color = (0, 0, 255)
                        with open(BUF_FILE_PATH, mode='a') as buf:
                            buf.write(barcodeData + '\n')
                    else:
                        font_color = (0, 154, 87)

                    x, y, w, h = barcode.rect
                    cv2.rectangle(frame, (x, y), (x + w, y + h), font_color, 2)

                    frame = cv2.putText(frame, barcodeData, (x, y - 10), font, .5, font_color, 2, cv2.LINE_AA)

                    cv2.imshow('BARCODE READER', frame)

                    if cv2.waitKey(1) & 0xFF == ord('q'):
                        break

cap.release()
```

2. バーコードの商品検索

```
python search.py バーコードの番号
import sys
import json
from urllib.request import urlopen
from urllib.error import URLError
import csv
import pandas as pd

APPID = "<APPID>"

BUF_FILE_PATH = 'search.csv'
```

```
def jancode_to_name(code):
    product_name = None
    url = f"https://shopping.yahooapis.jp/ShoppingWebService/V3/itemSearch?appid={APPID}&jan_code={code}&results=1"

    try:
        with urlopen(url) as resp:
            res = json.load(resp)
    except URLError as err:
        print(err.reason, file=sys.stderr)
        return None
    if "hits" in res and res["hits"]:
        product_name = res["hits"][0]["name"]
    return product_name

if len(sys.argv) != 2:
    sys.exit("example: search.py <jan_code>")

name = jancode_to_name(sys.argv[1])
if name is None:
    sys.exit("Not found")
```

```
print(name)
```

3. QRコードを作成

```
!pip install qrcode
```

```
import qrcode
import datetime
import csv
import pytz

name = input('何が入っていますか:')
date = input('大きさは?')

#日時、時刻がデータベースに登録されるよう設定
dt_now=datetime.datetime.now(pytz.timezone('Asia/Tokyo'))
dt=dt_now.strftime('%Y年%m月%d日 %H時%M分')

img=qrcode.make('氏名:'+ str(name)+' 作成日時:'+ str(dt))
img.save(str(name)+str(dt))
#img.save(qrcode.png)
```

```
with open('/home/mjiusr01/streamlit/text.csv','a') as f:
    writer = csv.writer(f)
    #次の行は初回登録のみ。2回目以降の登録時はコメントアウト!!
    writer.writerow(['入っているもの','量'])
    writer.writerow([str(name),str(date)])
    f.close()
```

```
img
```

4. メールを通知する機能

```
function mailSend() {
    var mySs = SpreadsheetApp.getActiveSpreadsheet(); //スプレッドシートを取得
    var mySheet = SpreadsheetApp.getActiveSheet(); //シートを取得
    var url = mySs.getUrl(); //スプレッドシートのURL
    var today = new Date(); //今日の日付を取得
    var strTo = mySheet.getRange(2,8).getValue(); //To

    var strSender = "冷蔵庫管理"; //差出人
    var dateBuffer = mySheet.getRange(1,8).getValue(); //賞味期限の何日前に通知するか
```

```
/* 繰り返し計算 */
```

```
for (var i = 2; i <= 200; i++) {
    var goodsName = mySheet.getRange(i, 1).getValue(); //Ai列を取得
    var exDt = mySheet.getRange(i, 3).getValue(); //Ci列を取得
    var zaiko = mySheet.getRange(i, 4).getValue(); //Di列を取得
    var exDate = new Date(exDt.getValue()); //Ci列のDateを作成
    var dt = exDate.getTime() - today.getTime(); //exDateから今日の日付を引く
    var day = Math.ceil(dt / 1000 / 60 / 60 / 24); //日付に変換

    if (0 <= day && day <= dateBuffer) { //通知する期日を過ぎたか
        /* メール本文を準備 */
        var strSubject2 = goodsName + "の賞味期限が迫っています (冷蔵庫管理Googleスプレッドシートから)";
```

```

var strBody2 = "冷蔵庫管理シートからのお知らせです。Yn" + goodsName + " (在庫数: " + zaiko + ") の賞味期限が" + String(dateBuffer) + "日前に迫っています。冷蔵庫を確認しましょうYnYn スプレッドシートへのリンク: " + url;
/* メールを送信 */
var send = GmailApp.sendEmail(
  strTo,
  strSubject2,
  strBody2, {
    name: strSender
  }
);
} else if (day <= 0) { //今日の日付と締切日と比較
/* メール本文を準備 */
var strSubject1 = goodsName + "の賞味期限が切れています (冷蔵庫管理Googleスプレッドシートから)";
var strBody1 = "冷蔵庫管理シートからのお知らせです。Yn" + goodsName + " (在庫数: " + zaiko + ") の賞味期限が切れました。今すぐ冷蔵庫を見てください。YnYnスプレッドシートへのリンク: " + url;
/* メールを送信 */
var send = GmailApp.sendEmail(
  strTo,
  strSubject1,
  strBody1, {
    name: strSender
  }
);
}
}
}

```

5. 野菜と果物の識別

```

!pip install icrawler
!pip install keras
!pip install tensorflow
!pip install h5py

```

```

from icrawler.builtin import BingImageCrawler

```

```

# 人参の画像を100枚取得
crawler = BingImageCrawler(storage={"root_dir": "/content/drive/MyDrive/fruits/2"})
crawler.crawl(keyword="人参", max_num=100)

```

```

from PIL import Image
import os, glob
import numpy as np
from PIL import ImageFile
# IOError: image file is truncated (0 bytes not processed)回避のため
ImageFile.LOAD_TRUNCATED_IMAGES = True

```

```

# indexを教師ラベルとして割り当てるため、0にはbananaを指定し、1にはichigoを指定
classes = ["/drive/MyDrive/fruits/banana", "/drive/MyDrive/fruits/ichigo",]

```

```

num_classes = len(classes)
image_size = 64
num_testdata = 25

```

```

X_train = []
X_test = []
y_train = []
y_test = []

```

```

for index, classlabel in enumerate(classes):
  photos_dir = "." + classlabel
  files = glob.glob(photos_dir + "/*.jpg")
  for i, file in enumerate(files):
    image = Image.open(file)
    image = image.convert("RGB")
    image = image.resize((image_size, image_size))
    data = np.asarray(image)
    if i < num_testdata:
      X_test.append(data)
      y_test.append(index)
    else:
      # angleに代入される値
      # -20
      # -15
      # -10
      # -5
      # 0
      # 5
      # 10

```

```

# 15
# 画像を5度ずつ回転
for angle in range(-20, 20, 5):

```

```

  img_r = image.rotate(angle)
  data = np.asarray(img_r)
  X_train.append(data)
  y_train.append(index)
  # FLIP_LEFT_RIGHT は 左右反転
  img_trains = img_r.transpose(Image.FLIP_LEFT_RIGHT)
  data = np.asarray(img_trains)
  X_train.append(data)
  y_train.append(index) # indexを教師ラベルとして割り当てるため、0にはバナナを指定し、1にはイチゴを指定

```

```

X_train = np.array(X_train)
X_test = np.array(X_test)
y_train = np.array(y_train)
y_test = np.array(y_test)

```

```

xy = (X_train, X_test, y_train, y_test)
np.save("./fruits.npy", xy)

```

```

from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
from keras.optimizers import RMSprop # TensorFlow系
# from keras.optimizers import RMSprop # エラー (ImportError: cannot import name 'RMSprop' from 'keras.optimizers' (/usr/local/lib/python3.7/dist-packages/keras/optimizers.py)) が発生
# from tensorflow.keras.optimizers import RMSprop # TensorFlow2系

```

```

from keras.utils import np_utils
import keras
import numpy as np

```

```

# indexを教師ラベルとして割り当てるため、0にはbananaを指定し、1にはichigoを指定
classes = ["/drive/MyDrive/fruits/banana", "/drive/MyDrive/fruits/ichigo"]

```

```

num_classes = len(classes)
image_size = 64

```

```

"""
データを読み込む関数
"""

```

```

def load_data():
  X_train, X_test, y_train, y_test = np.load("./fruits.npy", allow_pickle=True)
  # 入力データの各画素値を0-1の範囲で正規化(学習コストを下げるため)
  X_train = X_train.astype("float") / 255
  X_test = X_test.astype("float") / 255
  # to_categorical()にてラベルをone hot vector化
  y_train = np_utils.to_categorical(y_train, num_classes)
  y_test = np_utils.to_categorical(y_test, num_classes)

```

```

  return X_train, y_train, X_test, y_test

```

```

"""
モデルを学習する関数
"""

```

```

def train(X, y, X_test, y_test):
  model = Sequential()

  # Xは(1200, 64, 64, 3)
  # X.shape[1:]とすることで、(64, 64, 3)となり、入力にすることが可能です。
  model.add(Conv2D(32, (3, 3), padding='same', input_shape=X.shape[1:]))
  model.add(Activation('relu'))
  model.add(Conv2D(32, (3, 3)))
  model.add(Activation('relu'))
  model.add(MaxPooling2D(pool_size=(2, 2)))
  model.add(Dropout(0.1))

  model.add(Conv2D(64, (3, 3), padding='same'))
  model.add(Activation('relu'))
  model.add(Conv2D(64, (3, 3)))
  model.add(Activation('relu'))
  model.add(MaxPooling2D(pool_size=(2, 2)))
  model.add(Dropout(0.25))

  model.add(Flatten())
  model.add(Dense(512))
  model.add(Activation('relu'))
  model.add(Dropout(0.45))
  model.add(Dense(2)) # バナナとイチゴを識別するため、2クラス分類のため2を指定

```

```

model.add(Activation('softmax'))

# https://keras.io/ja/optimizers/
# 今回は、最適化アルゴリズムにRMSpropを利用
opt = RMSprop(lr=0.00005, decay=1e-6)
# https://keras.io/ja/models/sequential/
model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
cs=[ 'accuracy' ])
model.fit(X, y, batch_size=28, epochs=40)
# HDF5ファイルにKerasのモデルを保存
model.save('./cnn.h5')

return model

"""
メイン関数
データの読み込みとモデルの学習を行います。
"""
def main():
    # データの読み込み
    X_train, y_train, X_test, y_test = load_data()

    # モデルの学習
    model = train(X_train, y_train, X_test, y_test)

main()

import keras
import sys, os
import numpy as np
from PIL import Image
from keras.models import load_model

imsize = (64, 64)

"""
dog1.jpgというファイル名の画像をGoogle Colab上にアップロードする方法は2通りあります。
1つが、下記のコードを実行し画像をアップロードする方法
from google.colab import files
uploaded = files.upload()
2つが、Colab左メニューの>>アイコンを押して、目次、コード スニペット、ファイルの3つ表示されますが、右のファイルタブから画像をアップロードする方法です。
このファイルタブをクリックするとアップロードと更新の2つがありますが、アップロードを押すと画像をアップロードすることが可能です。
"""

testpic = "./drive/MyDrive/camera/IMG_3220.JPG"
keras_param = "./cnn.h5"

def load_image(path):
    img = Image.open(path)
    img = img.convert('RGB')
    # 学習時に、(64, 64, 3)で学習したので、画像の縦・横は今回 変数imsizeの(64, 64)にリサイズします。
    img = img.resize(imsize)
    # 画像データをnumpy配列の形式に変更
    img = np.asarray(img)
    img = img / 255.0
    return img

model = load_model(keras_param)
img = load_image(testpic)
prd = model.predict(np.array([img]))
print(prd) # 精度の表示
prelabel = np.argmax(prd, axis=1)
if prelabel == 0:
    print(">>> バナナ 保存期間15日間")
elif prelabel == 1:
    print(">>> いちご 保存期間3日間")

6. レンビデータの取得
import requests
import json
from pprint import pprint

res = requests.get('https://app.rakuten.co.jp/services/api/Recipe/Cat
egoryList/20170426?applicationId=楽天で作成したID')

json_data = json.loads(res.text)
pprint(json_data)

import pandas as pd

# mediumカテゴリの親カテゴリの辞書

```

```

parent_dict = {}

df = pd.DataFrame(columns=['category1', 'category2', 'category3', 'categoryId', 'categoryName'])

# 大カテゴリ
for category in json_data['result']['large']:
    df = df.append({'category1':category['categoryId'], 'category2':"", 'category3':"", 'categoryId':str(category['categoryId']), 'categoryName':category['categoryName']}, ignore_index=True)

# 中カテゴリ
for category in json_data['result']['medium']:
    df = df.append({'category1':category['parentCategoryId'], 'category2':category['categoryId'], 'category3':str(category['parentCategoryId'])+"-"+str(category['categoryId']), 'categoryName':category['categoryName']}, ignore_index=True)
    parent_dict[str(category['categoryId'])] = category['parentCategoryId']

# 小カテゴリ
for category in json_data['result']['small']:
    df = df.append({'category1':parent_dict[category['parentCategoryId']], 'category2':category['parentCategoryId'], 'category3':category['categoryId'], 'categoryId':parent_dict[category['parentCategoryId']]+"-"+str(category['parentCategoryId'])+"-"+str(category['categoryId']), 'categoryName':category['categoryName']}, ignore_index=True)

# 2. キーワードからカテゴリを抽出する
df_keyword = df.query('categoryName.str.contains("食べる料理を入力")', engine='python')
df_keyword

# カテゴリの人気レシピを取得
res = requests.get('https://app.rakuten.co.jp/services/api/Recipe/Cat
egoryRanking/20170426?applicationId=1070364094894326330&categoryId=
食べる料理のcategoryIdを入力')

json_data = json.loads(res.text)
pprint(json_data)

```