

4.数学学習システムにおける手書き入力文字の認識

猪又 健輔

指導教員 佐々木 建

1. はじめに

現在では,小学生・中学生でもスマホを所持し,ゲームだけではなく勉強にも活用している.その様な背景の中で,私も何か手軽にかつ分かりやすい学習アプリを作成できるのではないかと考えたところ,過去に,中学の数学教師を目指していた時もあり,何か数学の学習の助けになるようなものを作りたいと考え,タブレット上で手書き入力文字により,解答させる数学学習システムの作成を卒業研究のテーマにすることにした.

2. 開発環境の準備

今回作成したアプリは、Androidver8.0 以上のタブレットに対応することを想定して,開発を行った.主な開発環境は以下に示す.

表 1 開発環境

開発環境	
OS	Windows10 Pro
IDE	Android Studio ver 3.2
言語	Java ver 10.0.2
SDK	Android SDK ver 3.2.1
JDK	JDK1.10
デバック用端末	Android ver 8.0.0

3. 文字認識について

3.1 使用ライブラリ

本研究の手書き文字認識に使用したのは,「tesseract-OCR」という光学文字認識の OCR ライブラリである.このライブラリはオープンソースが公開されており,AndroidStudio 内で手書き文字を認識させる部分にあたっては,今研究では画像から文字を認識する「tess-two」という技法を使用した.

3.2 導入方法

手書き文字を認識させるための OCR ライブラリを使うためには ,AndroidStudio 内にある SDKManager を開き,SDKTool から「CMake」・「LLDB」・「NDK」の 3 つをインストールする必要がある。



図 1 SDKManager

また,build.gradle に外部依存の設定として「tess-two」をアプリモジュールのファイルに編集し,追加する.

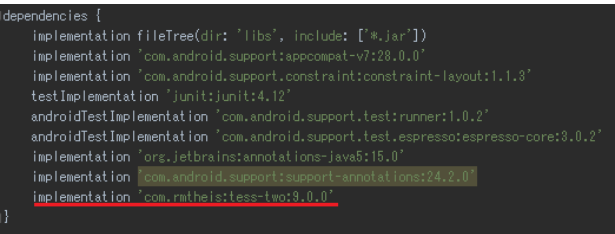


図 2 build.gradle

3.3 手書き文字認識用学習データ

今回の研究で活用する学習データは,「.traineddata」といったファイルを使用する.今研究では,既にある程度学習させているデータをダウンロードして活用した.ダウンロードしたファイルは OCR ライブラリに指定されたディレクトリ下に配置する.今回開発に使用したファイルは,「eng.traineddata」で英語を学習させたデータが入っているファイルである.日本語を学習させた「jpn.traineddata」もあった.が,数字を入力しても漢

字にしか認識されなかったため、数字を認識することができた英語版を採用した。

4.1.3 計算用紙

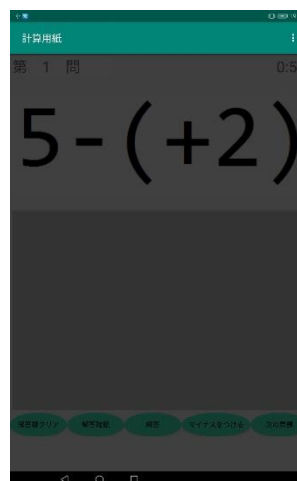


図 5 計算用紙

4.1.4 結果表示



図 6 結果表示

4.2 機能説明

4.2.1 トップページ

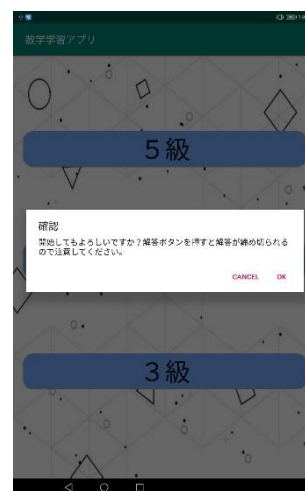
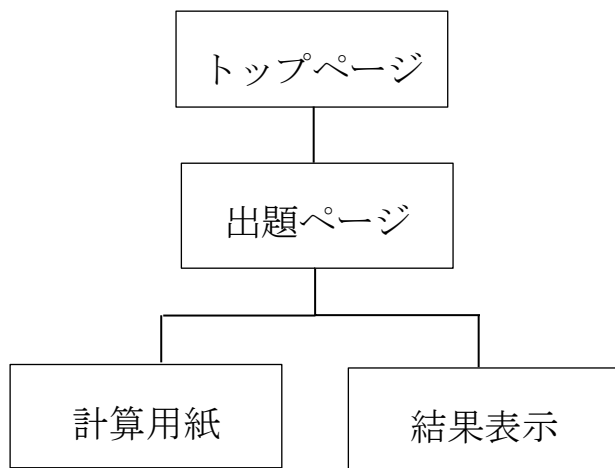


図 7 トップページ

ボタンを押すとアラートが出て、「OK」で出題画面へ、「Cancel」で選択画面に戻る。級の選択をすると、出題が始まる。

4. 実装機能

4.1 画面構成



4.1.1 トップページ

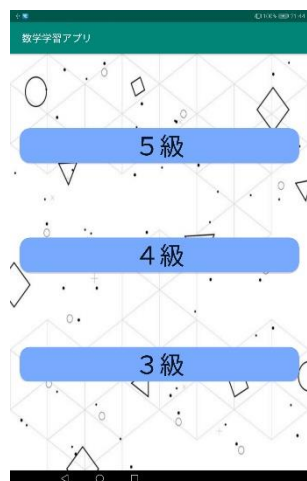


図 3 トップページ

4.1.2 出題ページ



図 4 出題ページ

4.2.2 出題ページ

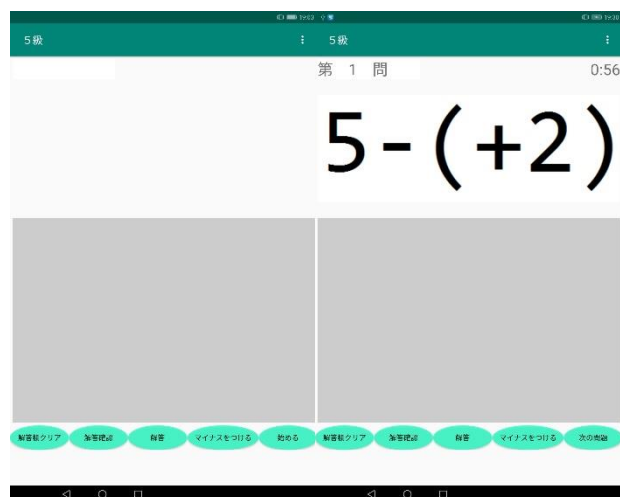


図 8 出題ページの変化

最初の画面には問題は出しておらず、「始める」ボタンを押すことで出題されるようになっている。次の問題に移行する際に、解答欄をリセットするため、自分でリセットすることは不要にしている。10 問解くことにより、結果表示画面に移行する。また、出題と同時にカウントが始まる。解答するとカウントが止まり、次の問題に進むことでカウントがリセットされる。0 秒になると解答ボタンを押しても解答ができなくなり、不正解判定になる。前の問題に戻ることはできない。

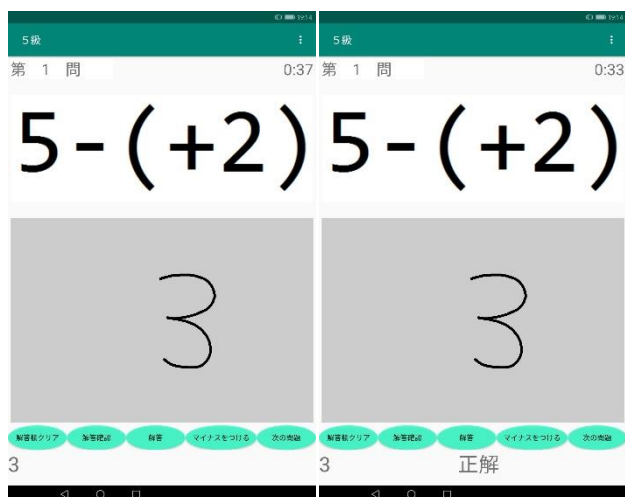


図 9 解答確認と解答後

1 問に対する解答は 1 度だけに設定している。文字認識を英語にしている都合上、書いた数字が英語に認識されてしまう可能性もある。そのため、自分が書いた数字に何と認識されているかを確認で

きるように「解答確認」ボタンを作成した。ボタンを押すことで、画面左下に認識された文字を確認することができる。自分が書いた数字と違った際は、「解答欄クリア」ボタンから解答欄をクリアして、書くことができる。解答確認と解答欄クリアに回数制限はない。次の問題に行く際に、解答確認表示エリアと正誤表示エリアがリセットされる。

認識の都合上「-」が上手く認識されなかった。そのため、答えに「-」がつく場合は「マイナスをつける」ボタンを押すことにより、自分の解答にマイナスが付与される。このボタンは、もう一度押すことで「-」を外すことができる。

画面右上のメニューからは、計算用紙の表示、またはトップページに戻ることができる。最終問題まで問題を解かずにトップページに戻った場合でも、途中の問題から再開することはできず、最初の問題から解くことになる。計算用紙を表示している際は、解答欄に解答を書くことができないため、閉じる必要がある。

4.2.3 計算用紙ページ

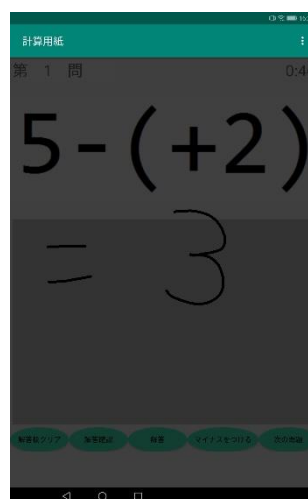


図 10 計算用紙

背景を半透明にすることで、問題を見えるようにしている。計算用紙は、ページ全体に書くことができる。

ペンの色は黒のみで、消すときは画面をクリアするようにしている。

4.2.4 結果ページ

10 問中何問正解したか、どこの問題を間違えたかが分かるようにしている。ボタンでトップページに戻る、またはこのページにとどまることができる。(4-2 ページの図 6 を参照)

5. おわりに

計画の段階では、中間発表までにエミュレータ内での実装完了、最終発表では、実機での実装、追加機能の実装を予定していたが、中間発表の段階では、エミュレータ内での実装完了以前に、学習アプリに関する機能や認識など、今研究に必要な機能のほとんどが中途半端なできになってしまった。最終発表の段階では、文字認識機能の実装完了やカウントダウンなどの学習アプリに必要な最低限の機能は実装することができた。しかし、当初の計画では、実装しようと考えていた「続きから再開できる機能」と「ジャンル別に出題する機能」が間に合わなかった。

また、修正しなければならない点としては、問題の出題の仕方である。開発した段階では、問題が 10 種類しか存在せず、ランダムで出題されることもない。本来の目標では、問題は CSV ファイルに格納し、問題の登録機能を追加する予定だった。だが、CSV ファイルの管理や、複雑な問題の登録・出題方法が分からず、今回は実装することができなかった。

これらを踏まえると、現段階での完成度は、7 割くらいだと考えている。自分で使ってみた感想だが、初見の人には分かりにくくなってしまったと思っている。出題ページに遷移した際に「始める」ボタンを押さないと出題されないこと、次の問題に行く時アラートがでることや、解答確認・解答・マイナスの際にもボタンを押さなければなく、何かと不便が多いと考えられた。友人にもこの学習アプリを使ってもらったこともあり、その時点では解答確認ボタンは無かった。自分が書いた文字が何と認識されているかが分かるようにした方がいいのではないかと、言った意見をもらって実装した機能である。

今回の認識に使用した「traineddata」は、既存の学習データをそのまま使用している。しかし、まだ精度が高いというわけではない。「tess-two」関連を調べていた段階では、そのまま使用するという情報しかなかった。が、精度を上げられないか調べたところ、「tesseract-OCR」の学習の仕方が 3 通りあることが分かった。認識の実装だけで手一杯になってしまい、今研究では既存データに学習させることはできなかった。提供されている学習ツール自体が不安定らしく、試してみないと分からないが向上が見込めるのではないかと考えている。

6. 参考文献・ダウンロードサイト

- ・ OCR ライブラリの導入

https://qiita.com/noboru_i/items/d08694c6ab9585bd561c

- ・ 「Traineddata」ダウンロードサイト

<https://github.com/tesseract-ocr/tessdata>

- ・ 背景が半透明の Activity を作成

<https://akira-watson.com/android/thme-translucent.html>