

手書き数式認識を用いた算数・数学自動採点システム

小西渉^{†1} 佐々木進亮^{†2} 松下朋永^{†3} レ・ドゥック・アイン^{†2} 中川正樹^{†2}

概要: 本稿では、生徒が算数や数学の問題に数式を書いて解答し、システムが手書き解答を認識かつ採点し、問題や採点に疑問を持った生徒が教師に遠隔対話できるシステムを報告する。数学的に考える力を測るには、e-learningなどで利用されるマウス選択あるいはキーボード入力可能な問題ではなく、数式を筆記させる問題が有用である。しかし、教師による採点は労力を要し、返却に時間を要すると生徒の復習効果も薄れる。そこで、本稿では、手書きされた数式を認識し、それを即座に採点し、その後で、生徒と教師が遠隔対話できるシステムを提案する。そして、システムの実現を報告する。

キーワード: 記述式問題、手書き解答、数式認識、自動採点システム

Automatic Scoring System for Mathematics using Handwritten Mathematical Expression Recognition

WATARU KONISHI^{†1} SHINSUKE SASAKI^{†2} TOMOHISA MATSUSHITA^{†3}
ANH DUC LE^{†2} MASAKI NAKAGAWA^{†2}

Abstract: In order to assess mathematical thinking ability, questions requiring examinees to answer by writing mathematical expressions are effective rather than those answered by selection. However, examiners must spend large time for marking them and the review effect by the examinees decreases if the feedback of marking is delayed. Therefore, we propose to recognize and mark handwritten answers automatically and the examinees receive the feedback immediately and they are able to ask about the questions and marking through a remote communication system.

Keywords: Descriptive Questions, Handwritten Answers, Mathematical Expression Recognition, Automatic Scoring System

1. はじめに

試験は、学習成果や問題解決能力を測るために不可欠である。また、それは学習や能力を測るためだけでなく、更なる学習を促すためにも重要である。しかし、その採点に、教師は多大な時間を要している。また、教師の採点にも誤りがあり、教師ごとに採点のばらつきもある。そして、採点や答案の返却が遅れば遅れるほど、生徒の復習効果も低下していく。Open University では、そうした事態を回避すべきとの提言がある[1]。こうした課題を解決する手段として、大規模一斉試験のためにマークシート方式が採用され今に至っている。ただ、これだけでは学力の判定に不十分であるために、個別の筆記試験などが併用されてきている。最近では、コンピュータやインターネットを用いた CBT (Computer Based Testing)、あるいは、WBT (Web Based Testing) が普及し始めている。ところが、現状では受験者がマウス選択、あるいは、キーボード入力できる問題に限られ、考える力や深い理解を問える記述式問題はほとんど除

外されている。選択式問題では、答えを考えるのではなく「答えをあてる」ことを習慣づけ、実際の現場では解決策を創案したり工夫したりするのではなく、既存の方法から選択する行動様式を助長しているとの反省がある。これでは、教育の情報化が試験の質を下げることになりかねない。我が国の政策として、記述式問題の重要性が認識され、大学新入試に反映されようとしている。

また別の課題として、入学試験の誤採点が受験者の進路が決まっただいぶ後になってから明らかになるという不幸事が後を絶たない。学校内の試験では答案が返却されてから誤採点を指摘することもできるが、入学試験では正しい採点が行われたことを補償する、あるいは、確認する手段はない。この課題を解決するために、受験者が受験番号とパスワードを打ち込めば、答案と採点を確認できるようにすべきである。

我々は、これら2つの課題を解決する手段として、採点の正しさを保障あるいは確認できる採点確認機能を前提として、筆記答案の自動採点機能を開発し、これらを統合することを提案する。このうち、採点確認機能は主に開発コストなどの技術的課題の解決が必要であり、一方、筆記答案の自動採点には本質的課題が伴う。ただし、前者は後者の性能が良くなればなるほどそのコスト効果比は高まり、後者は筆記答案パターンの蓄積によって性能の向上が期待できる。これが実現できれば、受験生の考える力や深い理

^{†1} 東京農工大学工学部情報工学科
Department of Computer and Information Sciences
Tokyo University of Agriculture and Technology
^{†2} 東京農工大学大学院工学研究科電子情報工学専攻
Department of Electronic and Information Engineering,
Tokyo University of Agriculture and Technology
^{†3} 株式会社アイラボ
ILABO CO.,LTD

解を問える記述式問題を出すことができ、採点者の負荷は軽減され、受験者は記憶が新しいうちに採点を見て復習することができる。自動採点には、ある程度の誤採点が予想されるが、受験生は採点確認機能を通して誤採点を報告し、採点者はこれを確認訂正するだけでなく、受験生の質問にも答えることもできる。ただし、自動採点の精度が低いと、この労力が過大になる。

筆記答案の媒体としては、従来は紙であり、近年ではタブレットもありえる。ここでは、手書き認識性能が高いタブレット入力を前提にする。タブレットの場合は、筆跡が時系列の筆点座標系列（電子インク）として採集される。

自動採点機能は、試験だけでなく、自習システムにも大きな用途である。児童生徒が漢字の書写学習や算数の勉強をするのに、その場で正誤を判定してくれれば、学習効果は高まる。携帯情報機器の進歩により、いつでも、どこでも学習が可能な環境は整っている。

我々は、小中学校の主要科目を対象に、タブレットで筆記された解答を認識し、採点する研究に着手している[2]。特に、手書き解答の必要性が高い算数や数学において2つのプロトタイプを発表した[3, 4]。本稿では、それらの問題を解決し、生徒が算数や数学の問題に数式を書いて解答し、システムが手書き解答を認識かつ採点し、問題や採点に疑問を持った生徒が教師に遠隔対話できるシステムを報告する。

2. 既存研究

記述式解答の自動採点に手書き認識を利用しようとする研究は多くない。初期は着想段階であった[5]。しかし、タブレットが普及し始め、その教育利用を討議する学会も開催された[6]。そこでは、自動採点の我々の初期の試み[7]や、手書き認識に誤りがあっても、全体の統計をすばやく取り、学習状況を即座に把握できることが報告されている[8]。そして、教育へのタブレット利用を後押しする書籍[9]も刊行された。これらの学会や書籍では、手書きの利点と情報技術の融合、即時性、共有、遠隔利用、学習者全体の動向把握などの研究提案が多い。

一方、記述式のエッセイなどの自動採点は、パターン認識と自然言語処理の両方の課題である。ただし、キーボード入力されたテキストに対しては、自然言語処理だけの課題になる。英文エッセイの自動採点では、1966年のPage[10]に始まり、1990年代後半から2000年初期にかけて、Bursteinらの発表[11]など多くの報告がある。和文エッセイでは石岡らによるJessがある[12]。構文解析や単語の共起モデル、限られた意味解析を用いることによって、人間による採点との差はあまりなく、むしろ人間より偏りやバイアスがないと報告している。さらに、Srihariらは、手書きエッセイをパターン認識し、自然言語解析で採点する試み

を報じている[13]。

3. 算数・数学自動採点システムの設計

本章では手書き数式認識を用いた算数・数学自動採点システムの設計について述べる。

3.1 要件定義

システムを実現するにあたり、必要な要件を次に定める。

- (1) 試験における問題の登録
- (2) 手書きによる数式入力及び訂正
- (3) 自由な順番での自由な記述
- (4) 入力された手書き数式の認識
- (5) 入力された解答の採点
- (6) 試験用紙に近い採点結果画像の生成
- (7) 試験結果についての生徒と教師の遠隔対話

以上の7つの要件定義について補足説明をする。

要件(1)では、教師側が自由に問題を登録でき、また、その問題登録では、配点及び許容される解答や部分点についても設定できるようにする。

要件(2)では、生徒が実際の試験を受ける時の実用性を考え、手書きによる入力を採用し、また、誤った記入や、考え直した時には、訂正が可能である必要がある。

要件(3)について、実際の試験を受ける際には、図1に示すような、認識に関係の無いメモ書きができることは、生徒にとって必要である。また、問題を飛ばしたり、簡単な問題から取り掛かったりするといった解答順の自由さも、生徒にとって必要なことである。

図1 自由な記述の例

要件(4)は、入力された手書き数式を認識し、正答かどうかを判定するために不可欠である。実現のために手書き数式認識エンジンは、当研究室で開発したものを利用する。

要件(5)については、登録された正答と照合して正解不正解を判定するだけでなく、予め許容された範囲の解答や、正答と同値の解答に対しての部分点をつける事で、より精度の高い採点を行えるようにする。

要件(6)について、採点結果は、生徒によって分かりやすい形でフィードバックする事が大切なため、問題と自分の解答、及びその採点が視覚的にわかるように、試験用紙の様な画像を生成することとする。

要件(7)について、自動採点において、手書き数式認識を扱っている上で避けられない誤認識等による誤採点を補完

出来る対話システムが不可欠である。誤採点の補完だけでなく、対話による振り返りの学習効果も期待できる。対話できると同時に、試験結果を確実に共有できることが必要である。

3.2 採点手法

本システムでは、生徒の解答に対して、採点し得点を与える。この採点の手法を述べる。

数式各行での採点のフローチャートは図2に示す通りである。

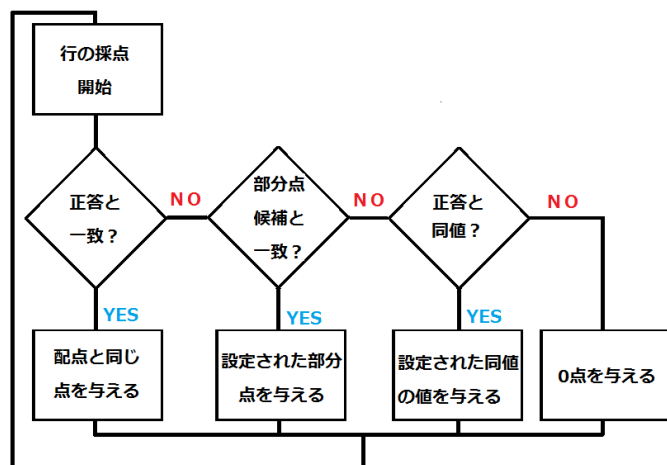


図2 各行の採点のフローチャート

採点は数式一行毎で行う。問題登録時に、その問題の配点、及び想定された部分点候補とその得点、正解と同値の場合の得点を登録しておき、それらによって場合分けをして、採点する。

まずは正解と完全一致をしているかを判別する。これは空白部分を正規化した上で、得られた解答の数式が正解と完全一致をしているかで判別を行う。この結果正解と一致していた場合には、正解であるため、配点分の得点全てを与える。ここで一致しなかった場合は、同様に、部分点候補の数式との判別を行う。この結果部分点候補と一致していた場合には、対応する部分点を与える。ここで一致しなかった場合には、正解との同値判定を行う。同値判定の方法は、2.2.1節にて述べる。ここで正解との同値であると判別できた場合には、同値の場合の得点を与え、この判別で同値ではなかった場合、解答は誤答と見なし、0点を与える。

これらの判別を経て、各行での得点が決定されるとその行での採点を終え、次の行の採点を始める。

こうして各行での得点が得られた後は、その問題に対しての得点を決定する処理をして、問題の採点を完了する。

問題に対しての得点の決定方法は、「問題内の全ての行の中での得点の最大値」とする。この方法により、計算過程の良かった部分が評価される採点となっている。

3.3 同値判定に用いる Web API

図2のフローチャート内であった、解答の数式と、正答の数式が同値であるか判定には、Web APIである Wolfram Alpha Web Service API[14]を利用している。Wolfram Alphaは、質問に対して、構造化されたデータを使って計算し、答えを返すWebサービスであり、このAPIの「Equal(lhs,rhs)関数」という、引数 lhs と引数 rhs の等価性を評価する関数を利用して、引数に正答の数式と、生徒によって得られた解答の数式を入力して評価している。

3.4 遠隔対話機能設計

遠隔対話を達成するものとして、独自で開発することも考えられるが、開発や保守の工数を考慮し、マイクロソフト社の Skype[15]を利用することを考える。Skypeは、暗号化通信が可能で、広く流通し、かつ導入の敷居も低く、本システムの対象者である小中学生にも適したソフトウェアである。

また、3.1節の要件(7)にあるように、試験結果を確実に共有できることが求められることから、Skypeのシステムの導入が容易であり、本システムで生成された画像の通信が容易な、ブラウザ上で機能するウェブアプリケーションを設計する。

4. システムの実現

本章では、システムの実現について述べる。

4.1 開発環境

本システムの実現のために開発に使用した環境を述べる。

使用 OS は Windows 7 Professional で、自動採点部分の開発環境は Microsoft Visual Studio 2008 とし、言語は C++を用い、対話部分の開発環境は Atom とし、言語は Python 及び html を用いた。

4.2 教師側画面

教師側画面では、問題と解答の入力を行う。

教師側画面で入力する項目は次の通りである。

- (1) 問題となる数式
- (2) 正解となる正答
- (3) 問題の配点
- (4) 解答と同値の場合に与える得点
- (5) 部分点を与える解答
- (6) 部分点の得点

以上の6つの項目について補足説明をする。

項目(1)については、試験における問題となる数式であり、手書きにより入力する。手書き入力は教師にとっても使いやすい。

項目(2)については、最終的に正解と採点する正答の数式を入力する。

項目(3)については、入力する問題の、試験における配点を入力する。たとえば、5つの問題で、100点満点の試験を

作成する場合に、この項目の配点を全て「20」に設定すれば、各問題の配点が等しく 20 点の試験を作成できる。

項目 (4) については、解答と同値であるが、正答では無い場合に与える得点を入力する。この同値であるが、正答ではない場合とは、主に解答の途中段階が挙げられ、途中まで導けたが、最終的な解答にまで辿り着けなかった場合等の部分点としての得点の意味合いとして使用する。たとえば、分数を約分していない解答などである。この同値を部分点として認めない場合は、この数字を「0」と入力すれば良い。

項目 (5) 及び (6) については、(4) の同値以外の場合においての部分点を想定する場合に、その想定した解答と、その解答に与える部分点の得点を入力する。例として、符号の間違い等が挙げられる。また、この項目では同値の場合であっても特別に点数を設定したい解答も入力できる。

これらの項目を入力し、問題を作成できる実際の画面を図 3 に示す。

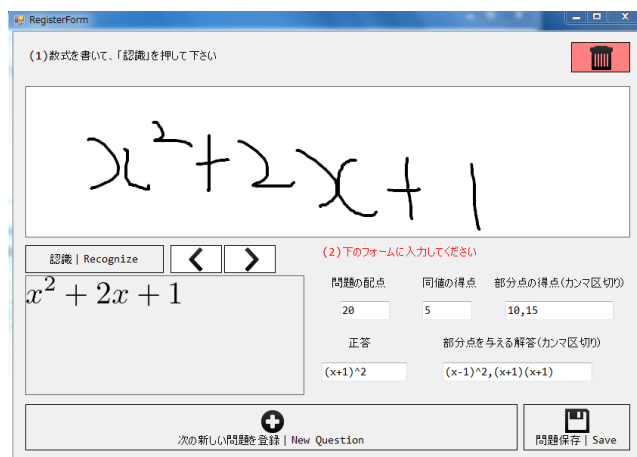


図 3 教師側画面

図 3 において、画面内入力部にて (1) から (6) までの項目を入力した上で、画面左下部のボタンを押すと、次の問題が登録できるので、試験の問題数だけ問題を登録して、最後に画面右下のボタンを押すことにより、(1) から (6) までの項目を CSV 形式で保存する。実際に保存される CSV の例を図 4 に示す。

```
1,2+3,5,20,5,,
2,(x+1)(x+2),x^2+3x+2,20,5,,
3,8x-2x+6x,12x,20,5,,
4,1/6+1/6,1/6,20,5,,
5,x^2+2x+1,(x+1)^2,20,5,10,15,(x-1)^2,(x+1)(x+1)
```

図 4 教師画面で保存された CSV ファイルの例

図 4 に示す様に、各問題に対して、その問題の問題番号と、(1) から (6) の項目がカンマ区切りで保存される。

4.3 生徒側画面

生徒側画面は、4.2 節の教師側画面で登録された試験を実際に受ける画面となる。

実際の動作を追って、記述する。

登録された CSV ファイルを読み込んだ後の生徒側画面を図 5 に示す。

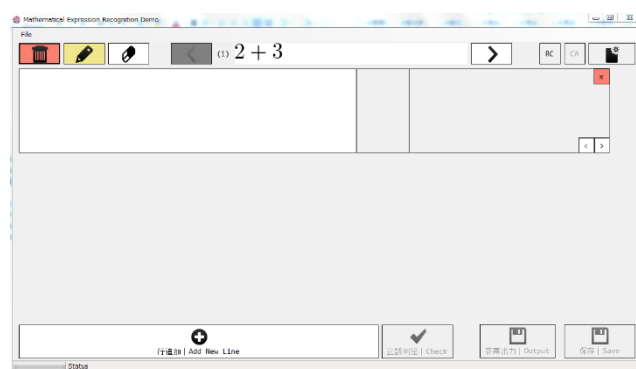


図 5 生徒側画面

画面上部に書かれた問題に対して、手書きで解答を記述する。その様子を図 6 に示す。

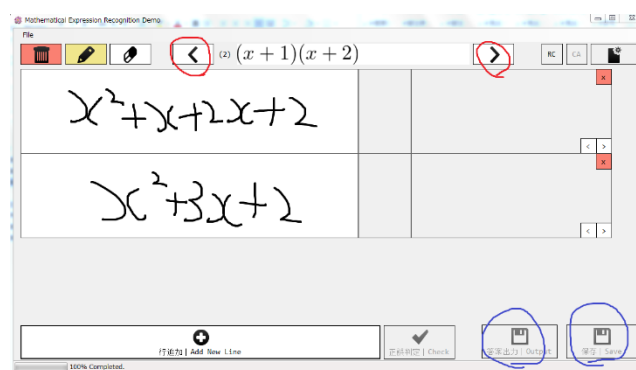


図 6 解答中の生徒画面

図 6 に示すように、各問題に対して、手書きで解答を記述していき、図 6 内の赤丸で示した矢印で、問題及び解答を移動し、全ての問題に対して解答を終えたら、図 6 内の青丸で示した保存及び答案出力ボタンを押す。

保存ボタンを押すと、各問題に対する数式認識した結果の数式と、各行の採点結果、そして各問題の採点結果と、試験全体の採点結果をカンマ区切りで CSV ファイルとして保存する。実際に保存される CSV の例を図 7 に示す。

```
1,0,5,20
2,0,x^2+x+2x+2,5
2,1,x^2+3x+2,20
3,0,7x+6x,0
```

```
3,1,13 x,0
4,0,\frac{3}{6},5
4,1,\frac{1}{2},20
5,0,(x+1)(x+1),15
1,max,20
2,max,20
3,max,0
4,max,20
5,max,15
sum,75
```

図 7 生徒画面で保存された CSV ファイルの例

図 7 で示した CSV ファイルの構成では、前半部分は、「問題番号, 0 から始まる行番号, その行で認識された数式, その行の得点」となっており、後半部分は、各問題番号に対して、全ての行から最大値を取った値を格納しており、構成としては「問題番号, "max", その問題の得点」となっている。最後には、その試験の合計点が「sum, その試験の合計点」という構成で記述される。

保存ボタンが押されてからは、隣の答案出力ボタンが押下可能になる。これを押すと、答案となる PNG ファイルが出力される。実際に出力される答案を図 8(a), (b) に示す。

合計: 75点

20 (1): $2 + 3$

5

○

20 (2): $(x+1)(x+2)$

$x^2 + x + 2x + 2$

△

$x^2 + 3x + 2$

○

0 (3): $8x - 2x + 6x$

$7x + 6x$

×

$13x$

×

図 8(a) 出力される答案用紙 1 枚目

図 8 で示したように、答案に出力される内容は、各問題とそれに対して試験時に記述した際の手書き数式である。それに関して、各行で、「正解, 部分的に正解, 間違い」の三段階の判定がなされ、問題の得点が左部に出力される。そして、1 枚目の右上には試験全体の合計得点が出力される。

20 (4): $\frac{1}{6} + \frac{2}{6}$

$\frac{3}{6}$

△

$\frac{1}{2}$

○

15 (5): $x^2 + 2x + 1$

$(x+1)(x+1)$

△

図 8(b) 出力される答案用紙 2 枚目

PNG ファイルは、幅 630×高さ 891 ピクセルの A4 サイズの比となっており、答案は、各問題で使用した行数から、問題と解答がその答案用紙に収まりきるサイズか計算し、収まるならばそのまま出力、収まらないサイズなら次以降の答案用紙を生成する。これは、問題数や筆記過程の量に対応するためである。

4.3.1 即時認識と即時採点

本システムでは、基本的に試験の場での実用を想定しているため、解答の途中等の採点や、文字認識は行わない。ただし、通常無効にしているが、オプションで書いた数式の認識が可能な即時認識のボタンと、各行ごとに、書いた数式を認識し、採点が可能な即時採点のボタンを有効にできる。これらを有効にして、即時認識を行った際の画面を図 9 に、即時採点を行った際の画面を図 10 に示す。

これらのオプションは、使用用途によって使い分けられるべきであり、即時認識はより自動採点による誤採点を防ぎたい際に用いられるべきであるが、同時に、認識候補が見られることにより、認識候補に正答が含まれる場合など、本来試験の意図しないところから試験結果に影響を与えてしまうデメリットも考えられる。

即時採点は、主に試験の場ではなく、すぐに採点をフィードバックしたい、自習の場での用途が考えられる。

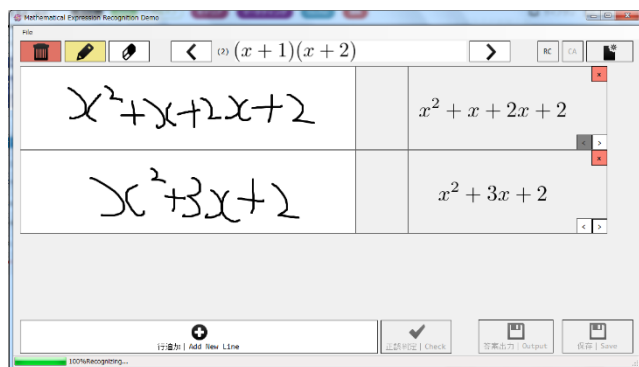


図 9 即時認識の画面

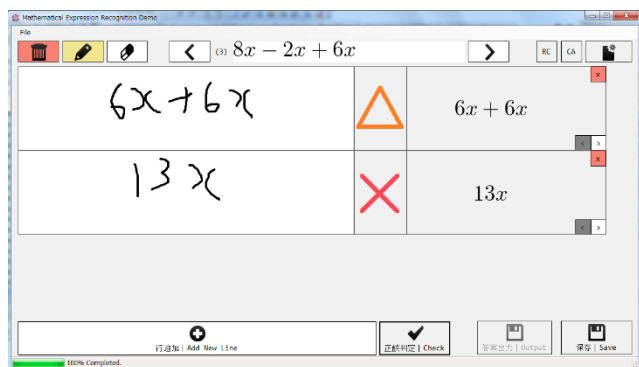


図 10 即時採点の画面

4.3.2 数式入力部インターフェース

図 5 に示す生徒側画面の左上部は、生徒の入力の補助を行うツールとなっており、3.1 節の要件定義を満たす実現をしている。

たとえば、ペンのマークのボタンをクリックすると、ペンの性質が変わり、認識に影響しないメモ記入のためのペンとして機能させることができる。これを用いれば図 11 に示すように、解答に必要なメモを残しながら試験を受けることができる。図 8(b)の(4)に示すように、このペンの入力も、答案用紙にそのまま出力される。

また、消しゴムマークのボタンをクリックすると、触れたストロークの消去を行うことができ、ゴミ箱のマークのボタンで入力のクリアも可能であり、自由な入力及び訂正の機能を備えている。

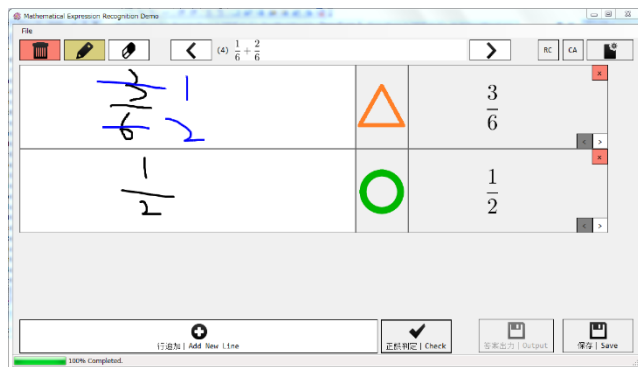


図 11 二種類のペンを使った解答画面

4.4 遠隔対話機能

Python 用のウェブアプリケーションフレームワークである Flask[16]を利用し、画像ファイルの共有が行えて、教師への対話が可能なウェブアプリケーションを実現した。

Python のプログラムを実行し、ブラウザでアクセスし、画像ファイルのアップロードを行う画面を図 12 に示す。

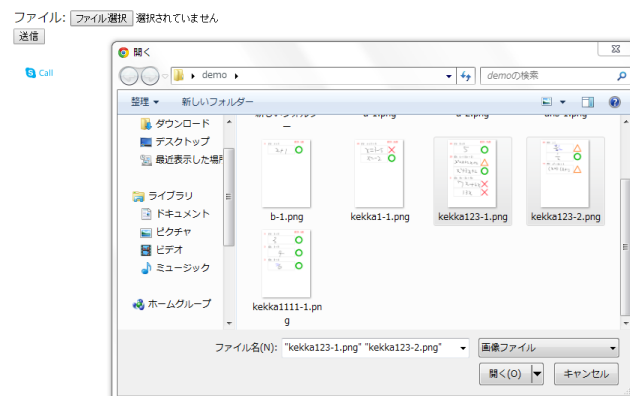


図 12 ファイルアップロードの画面

図 12 に示すように、トップページのファイル選択ボタンから、画像を選んでアップロードし、upload.html に遷移する。遷移先には教師の Skype ID へ通話をかけるリンクがあり、アップロードで画像を共有した上での対話が可能になる。

5. おわりに

本稿では、生徒が手書きした数式を自動認識、および採点し、生徒に即座にフィードバックし、生徒は採点や問題について遠隔対話できるシステムを提案し、そのプロトタイプを報告した。

謝辞

本研究は、一部、NEDO による課題設定型産業技術開発費助成金の補助による。

参考文献

- 1) Thorpe, M.: Assessment for Retention and Learning: Design, Feedback and Quality, Proc. International Symposium 2012 Student Assessment in Distance Learning and e-Learning, 2012.
- 2) 中川正樹, 内田吉郎, 佐々木進亮, 三田和広: タブレット上で筆記された回答の自動採点, 電子情報通信学会技術研究報告, Vol. 114, No. 513, pp.157-162 (2015.3).
- 3) 森重湧太, 中川正樹: 手書き数式認識を用いた計算過程の正誤フィードバック, 情報処理学会研究報告, Vol. 2014-CE-126, No. 3, pp.1-6 (2014.10).
- 4) 千葉智史, 中川正樹: 手書き数式認識を利用したタブレット PC 上での数学 e ラーニングシステムの試作, 情報処理学会研究報告, Vol. 2014-CE-127, No. 10, pp.1-5 (2014.12).
- 5) Dejan, G. et al. : A simple system for automatic exam scoring

- using optical mark reader, Proc. 2nd Balkan IFAC Int'l Symp.,
APPLIED AUTOMATIC SYSTEMS AAS'2000, Ohrid, Macedonia,
pp.149-153, 2000.
- 6) Proc. 1st Int'l Workshop on Pen-Based Learning Technologies,
Catania, Italy, <http://dl.acm.org/citation.cfm?id=1338440&picked=prox>,
2007
- 7) Nakagawa, M., Lozano, N., and Oda, H.: Paper Architecture and
an Exam Scoring Application, Proc. 1st Int'l Workshop on Pen-based
Learning Technologies (PLT), pp.113-118, 2007
- 8) Koile, K. et al.: Supporting Pen-Based Classroom Interaction: New
Findings and Functionality for Classroom Learning Partner, Proc. 1st
Int'l Workshop on Pen-based Learning Technologies (PLT), pp. 1-7,
2007.
- 9) Prey, J.C. et al. : The Impact of Tablet PCs and Pen-based
Technology on Education 2007: Beyond the Tipping Point, Purdue Univ.
Press, 2007.
- 10) Page, E.B.: The imminence of Grading Essays by Computer, Phi
Delta Kappa, pp.238-243. (1966)
- 11) Burstein, J. et al.: Automated Scoring Using A Hybrid Feature
Identification Technique, the Annual Meeting of the Association of
Computational Linguistics, pp.206-210 (1998).
- 12) Ishioka, T. et al. : Automated Japanese Essay Scoring System
based on Articles Written by Experts, Proc. 21st ICCL and 44th Annual
Meeting of the ACL, pp. 233-240, Sydney, July 2006.
- 13) Srihari, S. et al.: On the Automatic Scoring of Handwritten Essays.
Proc. IJCAI 07, Hyderabad, India, pp. 2880-2884. 2007.
- 14) Wolfram|Alpha: For Developers—API
<http://products.wolframalpha.com/developers/>
- 15) Skype
<https://www.skype.com/ja/>
- 16) Flask (A Python Microframework)
<http://flask.pocoo.org/>