知的障がい児に対する算数・数学の学習を支援する Web アプリケーションの開発

Development of a web application to support math learning for children with intellectual disabilities

山品 壱平[*1]，杉谷 賢一[*1]，中野 裕司[*1]，久保田 真一郎[*1]，市原 大裕[*1]，田村祐貴[*1]
Ippei YAMASHINA[*1], Kenichi SUGITANI[*1], Yuji NAKANO[*1], Shinichiro KUBOTA[*1], Daisuke ICHIHARA[*1], Yuki TAMURA[*1]
[*1] 熊本大学
[*1]Kumamoto University
Email: 244d8746@st.kumamoto-u.ac.jp

あらまし：障がいを持つ児童に対する学習支援は「特別支援教育」として，2007 年から行われるようになった．知的障がい児を含む障がいを持つ児童への学習支援として，始めは生活上の困難を克服するための支援が行われていたが，近年では特別支援教育にも普通教育と同様に国語，算数といった教科教育を行うことが求められるようになった．また，平成 29 年度の学習指導要領[1]の改訂により，ICT を活用した教育を行うことが求められるようになった．こうした背景から，我々は教科教育と ICT の活用を同時に実現できる Web アプリケーションの開発をを目指す．我々は熊本大学教育学部附属特別支援学校の教員らと打ち合わせの中で，知的障がい児の算数・数学教育における課題である「硬貨の計算が難しい」という点に着目した．現在はタブレットを使用したカメラに映る硬貨の合計を自動で計算する Web アプリケーションの開発を行っている．
キーワード：特別支援教育，Web アプリケーション，画像処理

1. 背景

日本では知的障がい，身体障がい，視覚障がい，聴覚障がいといった様々な障害を持つ児童に対して，学習支援が行われている．その中でも，知的障がい児は障がいの程度に個人差があり，学習の進捗も変化するため，特徴合わせた支援が困難となる．

また，平成 29 年度の学習指導要領(1)の改訂により，「障害による学習上または生活上の困難さを改善するために ICT を活用すること」，「教科指導の推進と情報活用能力の育成」が教員に求められるようになった．

現在，知的障がいに対する学習支援の問題点として，教科教育の教材が不足しているという点が挙げられる．そのため，ICT を活用した教科教育における学習支援を行う必要がある．

2. 研究目的

我々は熊本大学教育学部および附属特別支援学校と打ち合わせを行い，知的障がい児が抱える課題を調査した．知的障がいをもつ児童の抱える課題として，算数・数学では「硬貨の計算が難しい」ということが挙げられた．これは硬貨の種類ごとに分類はできても，合計をすぐに計算できていないからである．我々はこの課題を解決するためのソリューションとして Web アプリケーションの開発を行う．Web アプリケーションはデバイスや OS を問わず利用可能であり，インターネットが接続できる環境であればどこでも利用可能である．また，個別の端末で更新作業を行う必要がないという利点もある．

3. 研究方法

本研究では Web アプリケーションの開発を行い，完成したものを熊本大学教育学部附属特別支援学校の児童に利用してもらい，学習効果があるかを検証する．現在の進捗としては Web アプリケーションの開発途中であり，学習効果の検証は行っていない．

3.1 Web アプリケーションの概要

本研究で開発する Web アプリケーションはカメラがとらえた複数の硬貨の合計金額を計算する仕様である．これは以下のような環境で使用する事を想定している．

(1)使用機器はタブレット(iPad 第 7 世代)を想定する．これは熊本大学教育学部附属特別支援学校で児童が使用しているためである．
(2)カメラで撮影する際は硬貨を正面から撮影する．
(3)背景は単一色にする．

使用機器の画面にはカメラ映像，硬貨の種類ごとの枚数，合計金額を表示する．これは今後の開発によっては変更になる可能性がある．

3.2 開発環境

統合開発環境(IDE)には Visual Studio Code を使用する．使用する理由としては 2 つある．1 つ目は，拡張機能が充実していることである．Visual Studio Code は言語サポート，バージョン管理システム，デバッグツールなど，非常に多岐に渡る拡張機能が用意してあり，効率よく開発を進めることができる．2 つ目は軽量で起動が速く，リソースの消費が少ないため，開発環境としてストレスなく利用できるという点である．

また，使用言語は HTML，JavaScript，Rust を使用し，ライブラリとして OpenCV.js を使用する．OpenCV.js は OpenCV を JavaScript 上で使用する事を可能にしたものであり，Web アプリケーション上での画像処理が容易となる．

画像処理のアルゴリズムやコードは以下のサイト(2)を参照する．

3.3 WebAssembly

WebAssembly(3)とは Web ブラウザ上で実行可能なバイナリフォーマットである．WebAssembly はサイズとロード時間の効率的なバイナリ形式でエンコードされるように設計されており，ネイティブに近い速度で実行することが可能になる．今回は JavaScript の関数を一部 WebAssembly で実装するこ

とで，画像処理の高速化を実現する．また，開発言語は Rust を使用する．

3.4　Web アプリケーションの開発過程

まずは，画像処理部分の実装を行う．これは Web カメラで捉えた画像 1 枚が入力され，硬貨の判別結果が表示された画像が出力されるまでの処理を指す．その後，Web カメラによる動画処理部分の実装を行う．

3.5　画像処理

画像 1 枚が入力され，判別結果が出力されるまでの処理は大きく分けて 5 ステップある．(1)二値化 (2)閾値処理 (3)輪郭抽出 (4)形・色の特徴抽出 (5)硬貨の分類である．

4.　実験

本研究では実験として (1)画像 1 枚にかかる処理時間の検証，(2)Web カメラによる動画処理部分の動作検証を行う．実験 1，2 では (1) 画像 1 枚にかかる処理時間の検証，実験 3 では (2)Web カメラ実装時の精度評価を行う．

4.1　実験 1

実験 1 では，WebAssembly が未実装の場合に，複数枚の硬貨を写した画像が入力され，処理後の画像が出力されるまでの処理時間を確認する．画像サイズは 1050×1400，実行回数は 30 回とする．全体の処理時間と各ステップの処理時間の平均を算出する．

処理時間の結果を表 1 に示す．画像処理 1 枚あたりの処理時間は 6272ms となっており，約 6 秒かかっていた．また，各ステップで処理時間が約 1 秒以上かかっていたのは，閾値処理の 985ms，色の特徴抽出の 4910ms である．

4.2　実験 2

実験 2 では，WebAssembly を実装した場合，図 1(1)の画像が入力され，(2)の画像が出力されるまでの処理時間を確認する．画像サイズは 1050×1400，実行回数は 30 回とする．全体の処理時間と各ステップの処理時間の平均を算出する．

処理時間の結果を表 1 に示す．画像処理 1 枚あたりの処理時間は 622ms となっており，約 0.6 秒かかっていた．また，実験 1 において各ステップで処理時間が約 1 秒以上かかっていた閾値処理は 87ms，色の特徴抽出は 158ms となった．

表 1．実験 1, 2:各ステップごとのの処理時間(ms)

画像処理の手順	実験 1	実験 2
二値化	273	273
閾値処理	985	87
輪郭抽出	14	14
形の特徴抽出	17	17
色の特徴抽出	4910	158
硬貨の分類	67	67

4.3　実験 3

実験 3 では Web カメラを実装した際の精度評価を行う．環境の明るさを 4 段階に分け，計 40 枚の静止画を使用した検証を行う．Web カメラの解像度は 1050×1400 に設定し，背景には黒色のマットを使用する．また，照明は白色光となっている．照明環境ごとの結果を表 2 に示す．

表 2．実験 3 の検出結果

	明るい ⟷ 暗い							
	Precison	Recall	Precison	Recall	Precison	Recall	Precison	Recall
1	0.6667	0.5000	0.7500	0.5625	0.7500	0.5625	0.8000	0.5000
5		0.0000		0.0000		0.0000		0.0000
10	0.5789	0.7333	0.4762	0.6667	0.5500	0.7333	0.7273	0.5333
50	1.0000	0.5294	1.0000	0.5882	1.0000	0.5882	1.0000	0.5294
100	0.2727	0.2000	0.2727	0.2000	0.3333	0.6000	0.2667	0.5333
500	0.4117	0.5384	0.3158	0.4615	0.4000	0.1538	0.5000	0.1538

5.　考察

5.1　考察-実験 1，2

表 1 から，実験 1 で処理に多くの時間を費やしていたのは閾値処理と色の特徴抽出である．これらの処理を記述した JavaScript コードを確認したところ 2 重ループ処理にかなりの時間を要していることが分かった．

実験 2 で WebAssembly を実装した場合だと，閾値処理は 985ms から 87ms と約 10 分の 1，色の特徴抽出は 4910ms から 158ms と約 30 分の 1 まで短縮した．今回，WebAssembly はループ処理部分のみに実装したが，全体の処理時間も 6272ms から 622ms と約 10 分の 1 まで短縮したため，かなり有効な手法であると考えられる．

5.2　考察-実験 3

Web カメラを使用した場合にはいくつかの問題点が見られた．1 つ目は硬貨の誤分類である．これは硬貨が金属であるため，照明により RGB 値が変化することが理由として挙げられる．最適な照明条件も硬貨によって変化することが分かった．また，面積が小さい硬貨に関しては未検出が多々あった．

6.　結論

本研究では，知的障がい児の教科学習上の課題として挙げられた「硬貨の計算が難しい」という課題を解決するために Web アプリケーションの開発を行っている．現在は WebAssembly の実装により，1 フレームの処理の高速化を実現し，Web カメラを導入することで，
リアルタイムの動画処理まで開発を行った．しかし，硬貨の分類精度に問題があり，実際に使用するまでには改善すべき点が多い．

今後の方針としては Tensorflow.js をはじめとする機械学習を取り入れ，Web アプリケーションの開発を進める．

参考文献

(1) https://www.mext.go.jp/content/20200407-mxt_tokubetu01-100002983_1.pdf, (参照：2025 年 1 月 10 日)
(2) https://qiita.com/spc_ehara/items/b0c913a89877615d30b2, (参照：2024 年 12 月 20 日)
(3) https://webassembly.org/, (参照：2024 年 11 月 9 日)