

平成24年度 卒業研究論文

題目 タブレット端末における
アイコン入力方式の
有用性に関する研究

指導教員 服部 峻

提出者 室蘭工業大学 情報電子工学系学科

氏名 遠山 美優

学籍番号 2124111

提出年月日 平成25年2月13日

目次

第 1 章	序論	1
第 2 章	関連研究	3
2.1	アイコンを活用した従来研究	3
2.2	一般向け Web サービスにおける既存のアイコン入力	4
第 3 章	提案手法	6
3.1	アイコン入力方式の提案	6
3.2	実装機器	7
3.3	実行の手順	7
3.4	タッチ回数による文章変化	9
第 4 章	評価実験	12
4.1	実験の手順	12
4.2	入力速度結果	13
4.3	アンケート結果	15
第 5 章	考察	22
第 6 章	結論	27
	謝辞	28
	参考文献	29
	付録 A	30

目次

1.1	ソフトウェアキーボードの入力画面	2
2.1	顔アイコンによるファイル転送	3
2.2	ソーシャル顔アイコン	4
2.3	いいね!(Like) ボタン	5
2.4	気持玉 (BIGLOBE)	5
3.1	アイコン入力方式のイメージ	6
3.2	実行画面	7
3.3	手順 1 通常入力からアイコン入力へ画面遷移	7
3.4	手順 2 アイコン入力による文章入力	8
3.5	手順 3 複数回アイコンタッチによる入力文章の変化	8
3.6	手順 4 アイコン入力文章の選定と通常入力への画面遷移	8
4.1	手順 1 と手順 3 の入力速度結果のまとめ	14
5.1	ソフトウェアキーボードの入力画面	23
5.2	アイコンに登録されている文章を編集する機能	23
5.3	アイコンを追加できる機能	24
5.4	アイコンに相応しい文章を機械学習する機能	25

表目次

1.1	使用機器別 入力速度平均	2
3.1	感情アイコンの対応文章	9
3.2	行動アイコンの対応文章	10
3.3	天気アイコンの対応文章	11
4.1	タブレット端末における文章入力速度の比較	13
4.2	ソフトウェアキーボードの入力の使い易さ	15
4.3	アイコン入力の使い易さ	15
4.4	アイコン入力を利用することで簡単に文章を作成することが出来ると感じるか	16
4.5	アイコンの数	16
4.6	アイコンの大きさ	16
4.7	アイコンの絵柄と文章は適切か	17
4.8	使い易い, 難しいアイコン (感情)	17
4.9	使い易い, 難しいアイコン (行動)	18
4.10	使い易い, 難しいアイコン (天気)	18
4.11	アイコン入力の機能面について	19
4.12	所持している機器 (複数回答可)	20
4.13	今後のタブレット端末の使用について	20
4.14	利用しているアプリケーションや SNS	21
4.15	アイコン入力でメールや SNS を利用するか	21
4.16	コミュニケーションの向上について	21
5.1	アイコンに登録されている文章を編集する機能についての調査	24
5.2	アイコンを追加できる機能についての調査	25
5.3	アイコンに相応しい文章を機械学習する機能についての調査	26

第1章

序論

近年、スマートフォンやタブレット端末が急激に進歩を遂げている。その中でも、世界市場におけるタブレット端末の出荷台数は、2010年が2,000万台強だったのに対し、2011年には3倍の6,600万台、2012年には1億台に達する見通しである。MM総研の2012年度上期国内タブレット端末出荷概要 [1] より、国内の2012年度上期（12年4月～9月）の国内タブレット端末の出荷台数は前年比82.1%増の193万台となっている。また、今後は7インチ未満で片手での操作が簡単なモバイルタブレットが主流になるとみられており、今後のタブレット端末市場の底上げに貢献すると考えられている。

このようにタブレット端末が急激に普及してきた魅力の一つとして、ビジネスや日常など、利用シーンに左右されないという点が挙げられる。また、タッチパネルにより視覚的、直感的に把握し易く、持ち運びに便利であり画面が大きいため利用者を選ばないといったメリットがあるため、高齢者にも注目され始めている。シニア・高齢者のインターネット利用による調査 [2] から詳しくなりたい端末機において、パソコンに次いでタブレット端末が高いという調査結果が出ていることから、今後更にタブレット端末は普及していくと考えられる。

しかし、タブレット端末のデメリットの一つとして、画面表示が挙げられる。図 1.1 は Sony タブレット SGPT112JP/S を用いて、「Yahoo!JAPAN」のトップページでソフトウェアキーボードを表示した状態である。図 1.1 から分かるように、液晶画面の大半を使用するため、画面が見え難くなり表示できる情報量が減っている。サイトの閲覧や、文章の作成には若干使い難いのではないかと考えられる。また、使い難い理由の一つとして iOS, AndroidOS には標準で日本語入力システムが搭載されておらず、各メーカーによって搭載されているため、統一性が無いことも挙げられる。

一方で、近年 mixi や facebook 等の SNS(ソーシャルネットワーキングサービス) による繋がりが活性化している。若者や中高年がビジネスや友人との日常的な連絡に使うのに対し、高齢者の利用目的は趣味を深めること、孫との交流を図るなど第二の人生の生きがいを求める場合が多いのが特徴である。facebook を例にすると、日本国内では 2012 年 10 月の時点でユーザー数は約 1,600 万人であり、前年度よりも約 1,000 万人増加している。今後も増加していくと考えられ、利用者の大半は 20 代から 30 代であり、ビジネスにも用いられている。50 代以上は 8% と少ないが、疎遠になっていた友人や家族とのコミュニケーションとして今後利用さ



図 1.1 ソフトウェアキーボードの入力画面

れるのではないかと考えられる。また、東日本大震災後の安否確認に利用されたことにより、facebookをはじめとする SNS の関心度は高まっている。しかし、現状では興味関心を持っていても高齢者がメールや SNS 等を利用することは少ない。そこには文字入力の違いがあるのではないかと考える。そこで、実際に文字入力速度の調査を行った。表 1.1 は大学生、高齢者各 10 人を比較対象とし、100 文字程度の文章を PC、携帯電話のキー入力、フリック入力、及びタブレット端末のソフトウェアキーボードを用いて入力する速度をそれぞれ測り平均を出したものである。表 1.1 より、高齢者は 4 つの機器において大学生の倍以上掛かっていることが分かる。これは普段から機器に触れていないことや、文字入力に慣れていないことが影響していると考えられる。その中でもタブレット端末のソフトウェアキーボードの入力においては大学生、高齢者共に時間が掛かっている。これよりタブレット端末のメリットである視覚的、直感的に把握することが出来ていないことが分かる。

表 1.1 使用機器別 入力速度平均

使用機器	大学生平均	高齢者平均
PC	2分 13秒	6分 48秒
キー入力	2分 50秒	8分 17秒
フリック入力	1分 56秒	13分 33秒
タブレット端末	3分 55秒	10分 44秒

以上のことを踏まえ、家族や友人とのコミュニケーションを強化するために、タブレット端末で簡単に文章を作成できるような仕組みが必要なのではないかと考える。最近では音声入力や背面入力インターフェースといった新しい入力方式も挙げられるが、本研究では新たな入力方法として、簡単に文章を作成できるようなアイコン入力方式を考案し、その有用性を検証する。

第2章

関連研究

本研究では，次章以降，タブレット端末において SNS 等を通じたコミュニケーションを促進するため，誰もが簡単に文章を作成できるアイコン入力方式について提案し，その有用性を評価実験で検証して行く．本章では，関連研究として，顔アイコンによるファイル転送システム [3] とソーシャル顔アイコン [4] の 2 つのアイコンを活用した従来研究と，一般向け Web サービスにおける既存のアイコン入力について紹介する．

2.1 アイコンを活用した従来研究

顔アイコンによるファイル転送システム [3] では，図 2.1 のようにデスクトップ上に用意した転送したい相手の顔アイコンにドラッグ&ドロップするだけでファイルを送ることが出来る．また，顔アイコンはドラッグ&ドロップによるファイル転送の機能とともに，メッセージをメールとして送る機能も備えている．複雑な手順を必要とせずに手軽に文章やファイルを送ることが出来るものである．



図 2.1 顔アイコンによるファイル転送

ソーシャル顔アイコン [4] は Twitter や facebook 上の友人をアイコンにしてデスクトップに置くクライアントソフトである。図 2.2 のようにソーシャルウェブサイト上にいる人をアイコンとしてデスクトップに置くと、アカウントの発言を時系列 (タイムライン) で確認する事が出来る。ブログや SNS 等による繋がりを持ち、コミュニケーションを図る事が増えたため、ソーシャル顔アイコンのシステムを用いることで複数のソーシャルウェブサイトを横断しながらもその人の発言が埋もれず、より手軽に人やサイトにアクセス出来る。

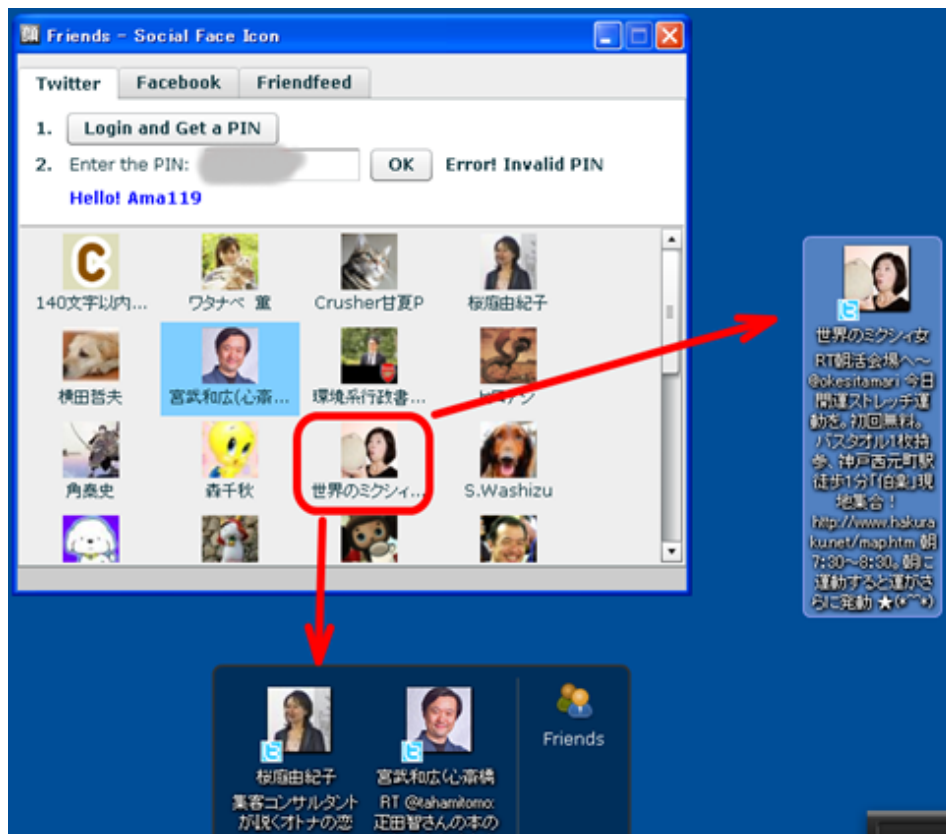


図 2.2 ソーシャル顔アイコン

この2つの研究に共通することはアイコンを活用することで手順を簡略化していることである。アイコンは手順を簡略化することに向いていることが分かる。

2.2 一般向け Web サービスにおける既存のアイコン入力

一般向け Web サービスにおける既存のアイコン入力として、図 2.3 の facebook における「いいね!(Like) ボタン」や、図 2.4 の BIGLOBE の「気持玉」などが挙げられる。図 2.3 の「いいね!ボタン」は facebook の機能であり、ユーザーはコメントや写真、友達とのリンク共有、広告といったコンテンツに好きという意味を示すために利用される。図 2.4 の BIGLOBE の「気持玉」は、ブログ記事への感想を『なるほど・驚いた・面白い・ナイス・ガッツ・かわいい』の6種類の感情アイコンを用いて、感じたことを6種類の気持玉で相手に伝えることが出来る。



図 2.3 いいね!(Like) ボタン



図 2.4 気持玉 (BIGLOBE)

例として挙げた既存のアイコン入力の特徴は、アイコン一つで出来る容易さが挙げられる。共感や、良いと思ったことを相手に伝えることが簡単に出来る。しかし、読者の言葉は直接的には伝わらず、また上記のようなアイコン入力ではマイナスな気持ちを伝えるものはあまりない。そのような場合はアイコンで示すのではなくコメント欄やメール等に文章を書き、相手に伝えることが多いのではないかと考えられる。

第3章

提案手法

本章では、タブレット端末においてメールや SNS 等を通じたコミュニケーションを促進するため、誰もが簡単に文章を作成できるアイコン入力方式について提案する。

3.1 アイコン入力方式の提案

前章で紹介したアイコンを活用した従来の研究、また一般向け Web サービスにおける既存のアイコン入力より、アイコン入力の機能を引き上げることで、ブログ記事やコメント欄等における文章の入力を更に補助することが可能ではないかと考えられる。

そこで本研究では、タブレット端末において視覚的、直感的に分かり易くするために図 3.1 のようなアイコン入力方式を提案する。ここで示すアイコン入力方式とは、対応するアイコンボタンをタッチすることで、予め登録された簡単な文章が入力される文章作成の補助機能である。文章を入力するのに慣れていない人でも簡単に文章を作成することが出来れば、メールや SNS などを通じてコミュニケーションの輪が広がっていくのではないかと考えられる。



図 3.1 アイコン入力方式のイメージ

3.2 実装機器

本研究では Android4.0.3 でアプリケーションを作成する。開発環境は eclipse を用い、使用するタブレット端末は Sony タブレット SGPT112JP/S である。

3.3 実行の手順

図 3.2 はアイコン入力方式を実装したアプリケーションの画面である。アプリケーションは図 3.2 の左側である通常のソフトウェアキーボードによる入力画面と、右側のアイコン入力を実装した 2 つの画面を遷移し、利用することで文章を作成する。以下にアプリケーションの実行手順を示す。



図 3.2 実行画面

1. アプリケーションを起動すると、始めにソフトウェアキーボードによる、通常の入力画面が現れる。図 3.3 のように、入力画面で「アイコン入力」ボタンを押すと、アイコン入力画面へと遷移する。



図 3.3 手順 1 通常入力からアイコン入力へ画面遷移

2. 図 3.4 のようにアイコンをタッチすると、アイコンに予め登録された文章が入力される。

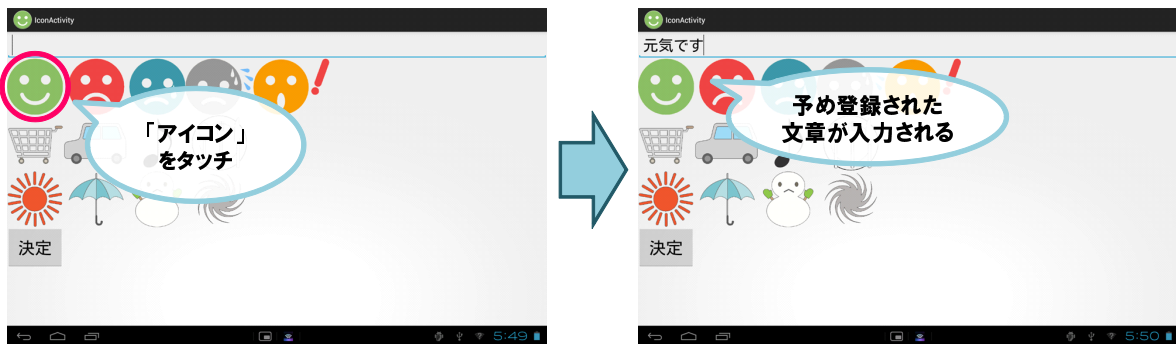


図 3.4 手順 2 アイコン入力による文章入力

3. 図 3.5 のように複数回アイコンをタッチすることで文章が変化する。

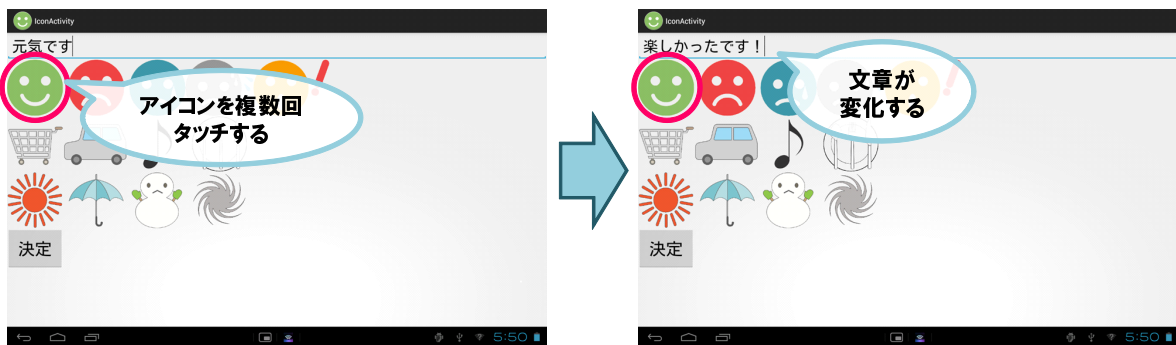


図 3.5 手順 3 複数回アイコンタッチによる入力文章の変化

4. 図 3.6 のように文章を選定後、「決定」ボタンを押すことで画面が入力画面に遷移し、文章が挿入される。



図 3.6 手順 4 アイコン入力文章の選定と通常入力への画面遷移

実装は叶わなかったが、上部のアクションバーに格納されている「同期」ボタンを押すことでメールや SNS 等に投稿できる仕様にする予定である。

また、文章を作成する際、相手に気持ちを伝えること、自身の行動を示すこと、周囲の様子を表すこと、この3つの要素は必要なのではないかと考える。そのため、アイコン入力画面に表示されているアイコンの上段は感情を伝える『感情アイコン』、中段は自身の行動を伝える『行動アイコン』、下段は天気を伝える『天気アイコン』の3種に分類し、計13種のアイコンを実装している。

3.4 タッチ回数による文章変化

アイコン1つに対し文章が1つでは実装しているアイコン13種のみで文章を作成することになる。これでは幅広い文章を作成できない。よって、アイコン入力には汎用性の問題があり、デメリットである。アイコン入力画面でのソフトウェアキーボードによる入力も行えるが、アイコン入力の機能を更に活用させるためにアイコンのタッチ回数によって文章が切り替わり、選択する方式を用いている。

3.4.1 感情アイコン

表3.1は感情を伝えるアイコン(感情アイコン)と対応する文章を表している。実装したアイコン入力では、感情アイコンは5種類である。喜怒哀楽を中心に文章を構成している。




表 3.1 感情アイコンの対応文章

感情アイコン	文章 1	文章 2	文章 3
	元気です。	楽しかったです！	よろしくお願いします。
	怒っています。	理不尽でした。	—
	とても悲しいです。	調子がよくありません。	とても残念です。
	焦りました・・・	怖かったです。	難しいですね。
	驚きました！	すごいですね！	びっくりしました！

3.4.2 行動アイコン

表 3.2 は自身の行動を伝えるアイコン (行動アイコン) と対応する文章を表している。実装したアイコン入力では、行動アイコンは『買い物』『車』『音楽』『食事』の4種類である。

表 3.2 行動アイコンの対応文章

行動アイコン	文章 1	文章 2	文章 3
	買い物に行きました。	今日はショッピングです。	買い物に行ってください。
	ドライブに行きました。	車に乗りました。	車を運転しました。
	音楽を聴きました。	演奏しました。	コンサートに行きました。
	料理をしました。	料理を食べました。	外食をしました。

3.4.3 天気アイコン

表 3.3 は天気を伝えるアイコン (天気アイコン) と対応する文章を表している。実装したアイコン入力では、天気アイコンは『晴れ』『雨』『雪』『風』の4種類である。

表 3.3 天気アイコンの対応文章

天気アイコン	文章 1	文章 2	文章 3
	今日は晴れでした。	天気が良いです。	暑いですね。
	今日は雨でした。	天気が悪いです。	肌寒いですね。
	今日は雪でした。	雪が積もりました。	寒いです・・・
	今日は風が強かったです。	台風でした。	—

第 4 章

評価実験

本章では、タブレット端末において SNS などを通じたコミュニケーションを促進するために実装したアイコン入力方式の有用性を検証する。大学生，高齢者各 10 人ずつを比較の対象として、タブレット端末においてソフトウェアキーボードとアイコン入力の入力速度を計測した。加えて、定性的評価のためアンケート調査を行った。

4.1 実験の手順

以下に評価実験の実験手順を示す。

1. 予め用意した 30 文字程度の文章をソフトウェアキーボード，アイコン入力でそれぞれ作成してもらいその入力速度を計る。その際アイコンにどの文章が対応しているかは説明しない。
2. アイコン入力について詳しく説明した上で，5 分間アプリケーションに自由に触れてもらう。
3. 5 分後，手順 1 とは違う文章を同じように作成してもらいその入力速度を計る。
4. 実験終了後にアンケートに答えてもらう。

手順 1，手順 3 で入力の順序によってどちらかが不公平にならないよう，大学生，高齢者 5 人ずつを A グループ，B グループに分けた。ソフトウェアキーボード，アイコン入力の順で入力を行う A グループ，逆の順序が B グループである。

以下に評価実験で使用した文章を載せる。この文章は実際にアイコン入力方式のみで入力可能な文章である。

説明前 (手順 1)	説明後 (手順 3)
今日は雪でした。 天気が悪いです。 音楽を聴きました。 楽しかったです。	今日は風が強かったです。 びっくりしました！ 買い物に行ってきて下さい。 よろしくお願ひします。

4.2 入力速度結果

表 4.1 は大学生，高齢者の手順 1 と手順 3 における入力速度の平均を比較したものである。手順 1 と手順 3 を比較すると入力速度が速くなっていることが表 4.1 から分かる。ソフトウェアキーボードの入力は使い難いと事前調査から分かっていたが，今回の評価実験でもアイコン入力に比べると時間が掛かっていることが分かる。また，手順 1 ではアイコン入力について詳しく説明はしていないため，手順 3 との時間の差が大きくなったのではないかと考えられる。

表 4.1 タブレット端末における文章入力速度の比較

	ソフトウェア キーボード	アイコン入力 (提案手法)
手 順 1	大学生	64.8 ± 12.4 秒
	高齢者	172.0 ± 74.0 秒
	平均	118.4 ± 75.5 秒
手 順 3	大学生	60.6 ± 18.1 秒
	高齢者	159.1 ± 72.8 秒
	平均	109.9 ± 72.3 秒

また，図 4.1 は手順 1 と手順 3 における入力速度結果の散布図である。縦軸はソフトウェアキーボードによる入力速度，横軸はアイコン入力の入力速度を表している。手順 1 の大学生は青，手順 3 の大学生は緑，手順 1 の高齢者は赤，手順 3 の高齢者は橙で示しており，図上の直線は $y=x$ である。手順 1 において，大学生のソフトウェアキーボードの入力速度で一番早い場合で 47 秒であり，一番遅い場合では 81 秒であった。アイコン入力の場合には一番早い場合は 21 秒，遅い場合は 53 秒であった。高齢者のソフトウェアキーボードの入力速度で一番早い場合で 89 秒，一番遅い場合で 312 秒，被験者によつての差が非常に大きい。アイコン入力の場合では一番早い場合は 21 秒と，大学生と同じであった。また一番遅い場合で 72 秒である。

手順 3 において，大学生のソフトウェアキーボードの入力速度で一番早い場合で 34 秒であ

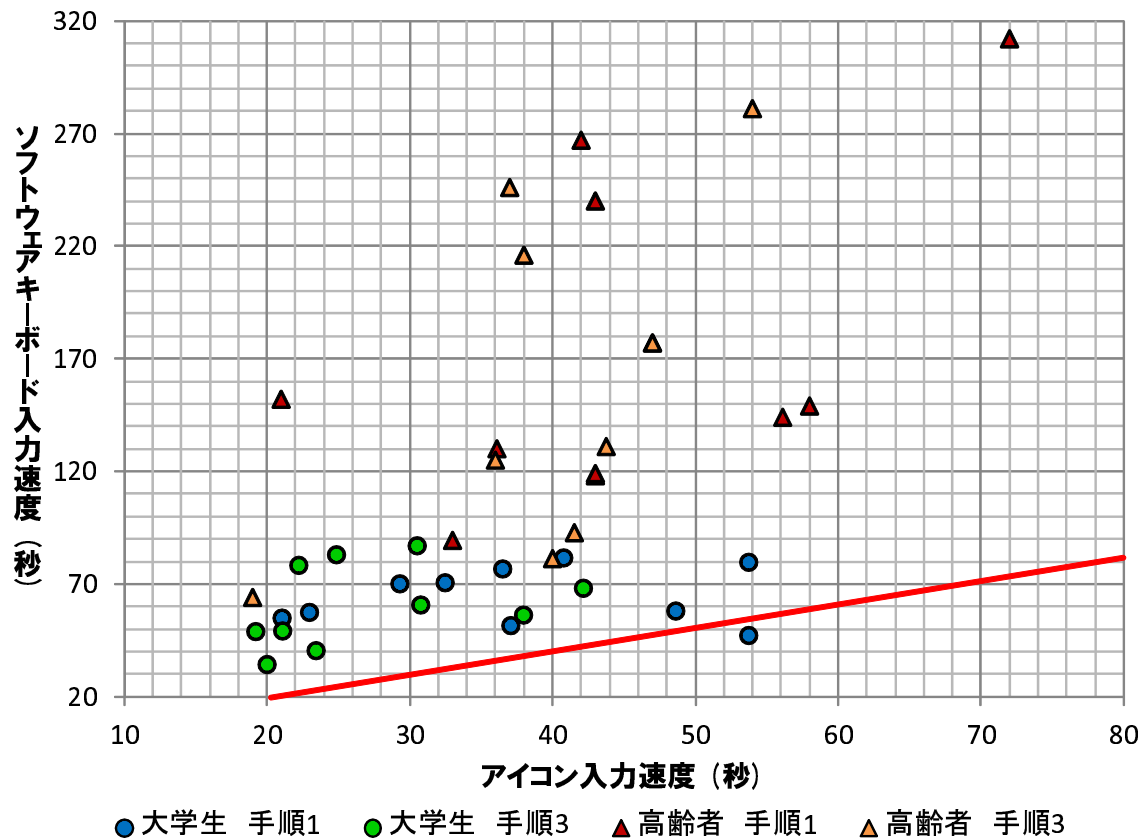


図 4.1 手順 1 と手順 3 の入力速度結果のまとめ

り、一番遅い場合では 86 秒であった。アイコン入力の場合には一番早い場合は 19 秒、遅い場合は 42 秒であった。高齢者のソフトウェアキーボードの入力速度で一番早い場合で 64 秒、一番遅い場合で 281 秒と、被験者によつての差は相変わらず大きいが速度は速くなっている。アイコン入力の場合では一番早い場合は手順 1 の 21 秒と同一人物が出した 19 秒となり、一番遅い場合で 54 秒となった。

図 4.1 より、手順 1 と手順 3 を比べると全体的に手順 3 の方が左下寄りになり、全体的に入力速度が早くなっていることが分かる。更に、直線 $y = x$ の $y > x$ の領域に結果が集中している。これはアイコン入力の方がソフトウェアキーボードよりも入力が早いことを示しており、直線から離れているほどこの事が顕著に現れている。このことから、高齢者の方がアイコン入力方式の効果が大きいと言える。

また、表 4.1 より、大学生のアイコン入力の入力速度の説明前と後の時間差は、アイコンの文章を評価実験の手順 3 の 5 分間で覚えてしまったためと考えられる。しかし説明前の文章を入力する際、対応するアイコンを最初に押していることが多く見受けられた。そのためアイコン入力方式の目的である直感的に把握するという部分に対しては、十分に実現出来ていると考えられる。

高齢者の場合はソフトウェアキーボードでは大学生に比べ高齢者は 2 倍以上掛かっているがアイコン入力では大学生と大差ないことが図 4.1、及び表 4.1 から分かる。時間は確かに高

年齢の方が掛かっているが大学生同様、直感的に把握するという目的は達成されたと考えられる。

4.3 アンケート結果

次より評価実験後に行ったアンケート結果を示す。

4.3.1 入力面

はじめに、入力面についてのアンケート結果を示す。表 4.2 はソフトウェアキーボードの入力、表 4.3 はアイコン入力の使い易さについてのアンケート結果である。ソフトウェアキーボードの入力についての回答は散けているが、アイコン入力にはソフトウェアキーボードに比べ使い易いと回答された被験者が多い。このことからアイコン入力はタブレット端末のメリットを活かしていると考えられる。

表 4.2 ソフトウェアキーボードの入力の使い易さ

	使い易い	やや使い易い	やや使い難い	やや使い難い
大学生 A	2	2	1	0
大学生 B	0	1	3	1
高齢者 A	2	1	2	0
高齢者 B	0	0	3	2
計	4	4	9	3
%	20	20	45	15

表 4.3 アイコン入力の使い易さ

	使い易い	やや使い易い	やや使い難い	やや使い難い
大学生 A	2	2	1	0
大学生 B	3	2	0	0
高齢者 A	4	1	0	0
高齢者 B	3	2	0	0
計	12	7	1	0
%	60	35	5	0

また、表 4.4 はアイコン入力を利用することで簡単に文章を作成することが出来ると感じるか、という問いに対する結果である。全体の 85% が出来たと回答しているため、アイコン入

力の目的を達成していると言える。

表 4.4 アイコン入力を利用することで簡単に文章を作成することが出来ると感じるか

	出来た	分からない	出来ない
大学生 A	4	1	0
大学生 B	5	0	0
高齢者 A	5	0	0
高齢者 B	3	2	0
計	17	3	0
%	85	15	0

4.3.2 機能面

次に機能面についてのアンケート結果を示す。表 4.5 はアイコンの数、表 4.6 はアイコンの大きさについて調べた結果である。表 4.5 より、高齢者の 70% はアイコンの数について丁度良いと答えているが、大学生は反対に少ないに多く回答し、汎用性の問題を指摘された。また、表 4.6 は 9 インチのタブレット端末の場合であり、それ以下の端末では結果は分からないが現状の大きさが良いと分かった。

表 4.5 アイコンの数

	多い	丁度良い	少ない
大学生 A	0	3	2
大学生 B	0	1	4
高齢者 A	0	5	0
高齢者 B	2	2	1
計	2	11	7
%	10	55	35

表 4.6 アイコンの大きさ

	大きい	丁度良い	少ない
大学生 A	0	5	0
大学生 B	0	5	0
高齢者 A	0	5	0
高齢者 B	0	5	0
計	0	20	0
%	0	100	0

表 4.7 はアイコンの絵柄と文章は適切か、という問いに対する結果である。現在実装されているアイコンでは全体の 65% が適切であると答えているが、絵柄と文章が合っていないものもあると考えられる。数が制限されている現在のアイコン入力で、実際にどれが適切であったか調べるためにその調査も行った。

表 4.8 は感情アイコン、表 4.9 は行動アイコン、表 4.10 は天気アイコンについて使い易い、使い難いアイコンを 3 つまで回答して頂いた結果である。該当がなければ回答しなくても良いとしている。

表 4.7 アイコンの絵柄と文章は適切か

	適切である	分からない	適切ではない
大学生 A	2	2	1
大学生 B	3	1	1
高齢者 A	5	0	0
高齢者 B	3	2	0
計	13	5	2
%	65	25	10

表 4.8 使い易い, 難いアイコン (感情)

感情 アイコン	使い易い		使い難い	
	大学生	高齢者	大学生	高齢者
	3	5	1	0
	1	2	0	0
	2	2	0	0
	1	4	0	1
	1	4	0	0

まず, 表 4.8 の感情アイコンから見ていくと, 比較的使い易いと多くのアイコンで回答されていることが分かる. しかし, 感情アイコンに対して, 日本語は繊細な表現が多く, 単純なアイコンでは感情を表現できないといった意見があった. また, 感情アイコンを押す順番によって文章が変化するやり方や, アイコンの顔を半分ずつにして繊細な表現を試みるなどの意見もあった. 今後の対応として参考に出来ればと考えている.

表 4.9 使い易い, 難いアイコン (行動)

行動 アイコン	使い易い		使い難い	
	大学生	高齢者	大学生	高齢者
	2	3	0	2
	0	0	5	0
	4	1	0	0
	0	0	0	0

表 4.10 使い易い, 難いアイコン (天気)

天気 アイコン	使い易い		使い難い	
	大学生	高齢者	大学生	高齢者
	5	4	0	0
	2	0	0	0
	7	6	0	0
	1	1	7	5

次に表 4.9 の行動アイコンでは『車』のアイコンに使い難いと回答した大学生が多いことが分かる。これは、『ショッピングカート』のアイコンと同じ意味に見えてしまうといった意見のためであった。評価実験の手順 3 の文章で『買い物に行ってきて下さい。』という文を入力

する際に『車』アイコンに格納されていると勘違いしている被験者や、「買い物には車で行く」という個人的な思い込みがあったためかと考えられる。『ショッピングカート』に高齢者が使い難いと回答しているのは、アイコンが何を示しているのか最初分からなかったためである。そのためこのアイコンは直感的にという目的に反しているので今後変更した方がよいと考えられる。

表 4.10 の天気アイコンに関しては『風』のアイコンが大学生、高齢者共に使い難いと回答する被験者が多かった。それ以外のアイコンについては使い易いと多く回答している。これは天気予報等でよく見かけるマークであったためと考えられる。そのため『風』のアイコンも天気予報等で見られるようなマークにすれば解決できると考えられる。

以上のことから、改良すれば更に使い易いものになると推測できる。また、現状のアイコン入力の機能面に関しては表 4.11 のようになっている。満足、やや満足と答えた被験者が全体の 85% であったため、アイコン入力是有用であると考えられる。

表 4.11 アイコン入力の機能面について

	満足	やや満足	やや不満	不満
大学生 A	2	3	0	0
大学生 B	2	0	3	0
高齢者 A	1	4	0	0
高齢者 B	4	1	0	0
計	9	8	3	0
%	45	40	15	0

また、機能面の意見として、被験者の大学生からアイコンをタッチして文章を変化する仕組みではなく、フリック入力のようにアイコンを長押しすることで文章全てが表示されるような仕組みの方が良いのではないかと意見があった。直感的に把握して貰うことが目的の一つであるため、タッチすることで文章が切り替わる方式を採っていたが、文章をタッチし全ての文章を確認するだけで 2,3 回の手間が掛かってしまうため非効率的と被験者が感じたためこのような意見が出たと考えられる。また、フリック入力による入力速度は事前調査で行った表 1.1 から分かる通り大学生は普段からスマートフォンを利用している被験者が多かったことから 4 つの機器の中で入力速度が一番早いことが分かり、この事も起因していると考えられる。しかし高齢者の場合、普段使い慣れていないとはいえ、4 つの機器の中で一番時間が掛かっている機器がフリック入力であるため、あまり適しているとは言えない。今後、更に利用しやすいアプリケーションを目指すためには、アイコン入力の文章表示の仕方についても考えていく必要がある。

4.3.3 その他

実際に所持している機器について調査した結果を表 4.12 に示す。タブレット端末を使用しているのが、高齢者 1 人だけであった。大学生はスマートフォンの所持率が高く、またパソコンに関しては大学生活に必需品であるため、現時点ではタブレット端末を所持する予定はないといった意見が多かった。

表 4.12 所持している機器 (複数回答可)

	携帯電話	スマートフォン	パソコン	タブレット端末
大学生	3	7	10	0
高齢者	6	2	5	1
計	9	9	15	1

表 4.13 は今後タブレット端末を使用するか、という問いに対する結果である。表 4.12 の所持している機器のタブレット端末の割合とは違い、大学生、高齢者共に多くの被験者が今後タブレット端末を使用したいと答えている。このことから、タブレット端末への興味関心は高いと考えられ、普及していくのではないかと考えられる。

表 4.13 今後のタブレット端末の使用について

	使用したい	分からない	使用しない
大学生 A	3	2	0
大学生 B	4	1	0
高齢者 A	4	1	0
高齢者 B	4	1	0
計	15	5	0
%	75	25	0

表 4.14 は利用しているアプリケーション、SNS の調査結果である。携帯電話、スマートフォンの所持率からメールは高齢者も利用していることが分かる。SNS では大学生、高齢者共に facebook が多かった。

表 4.14 利用しているアプリケーションや SNS

	メール	ブログ	Twitter	mixi	facebook
大学生	10	0	3	3	5
高齢者	8	2	1	0	2
計	18	2	4	3	7

表 4.15 はアイコン入力でメールや SNS を利用するか、という問いに対する結果である。全体の 55% が利用すると答えており、高齢者のみでは 80% と割合が高い。しかし、文章の汎用性の問題があり幅広い文章を作成できないため分からない、全体で利用しないと答えた割合も同程度である。

だが、アイコン入力を利用することでコミュニケーションの向上に繋がるかという問いに対しては、表 4.16 のようになっており、全体の 85% が向上すると答えている。家族や友人とのコミュニケーションを強化するという目的はこの結果から達成できたと考えられる。

よって、汎用性を始めとする問題を解決できればメールや SNS に利用され易くなり、コミュニケーションの向上も図ることが出来ると推測できる。

表 4.15 アイコン入力でメールや SNS を利用するか

	利用する	分からない	利用しない
大学生 A	1	4	0
大学生 B	2	1	2
高齢者 A	3	1	1
高齢者 B	5	0	0
計	11	6	3
%	55	30	15

表 4.16 コミュニケーションの向上について

	向上する	分からない	向上しない
大学生 A	4	1	0
大学生 B	4	1	0
高齢者 A	5	0	0
高齢者 B	4	1	0
計	17	3	0
%	85	15	0

第5章

考察

今回の評価実験では大学生、高齢者5人ずつをAグループ、Bグループに分けた。ソフトウェアキーボード、アイコン入力の順で入力を行うAグループ、逆の順序がBグループである。初めてタブレットを使用する被験者が多く見受けられたが、Aグループ、Bグループでの入力速度、アンケート結果に大差はなかった。

入力速度の調査より、ソフトウェアキーボードの入力は大学生に比べ高齢者の速度が2倍以上掛かっているのに対し、アイコン入力では殆ど変わらないことが分かる。また、図4.1より手順1と手順3を比べると、全体的に手順3の方が全体的に入力速度が早くなっており、特に高齢者に対してアイコン入力方式の効果が顕著に現れている。

アイコン入力はタブレット端末のメリットを活かすために視覚的、直感的に把握し易く文章を作成出来る入力を目指し作成した。一方、ソフトウェアキーボードは機能上、図5.1のように画面の大半を使い、キーの1つ1つが小さいため入力し難い。今回使用した9インチのタブレット端末よりも小さいものになればなるほど、この結果は顕著になっていくのではないかと考えられる。また、アンケート調査からもソフトウェアキーボードの入力は使い難い、やや使い難いに答えた方が全体の60%いることから、入力に慣れていないことも影響しているとも考えられる。

アイコン入力を利用することで、メールやSNSを利用すると答えた被験者は全体の55%であるが、高齢者の回答率がとても高い。また、アイコン入力を利用することでコミュニケーションの向上に繋がると答えた被験者は全体の85%であり、目的を達成したと言える。以上のことから、アイコン入力はソフトウェアキーボードより簡単に文章を作成することが出来、今後のコミュニケーション方法として期待できる。よって、提案したアイコン入力方式は有用であると言える。

しかしアイコンの数や文章の汎用性は問題が多く挙げられ、現在のアイコン入力では幅広い文章を作成することに向いていない。特に、行動アイコンにおいては今回実装した数が少なかったため、アイコンの数を多くして欲しいといった要望や、文章を変更したいといった意見があった。そのため今後の課題として汎用性の問題を改善するために、アイコンに登録されている文章を編集する機能(図5.2)、アイコンを追加できる機能(図5.3)、アイコンに相応しい文章を機械学習する機能(図5.4)等が必要であると考えた。これら3つの機能が実際に必要であ

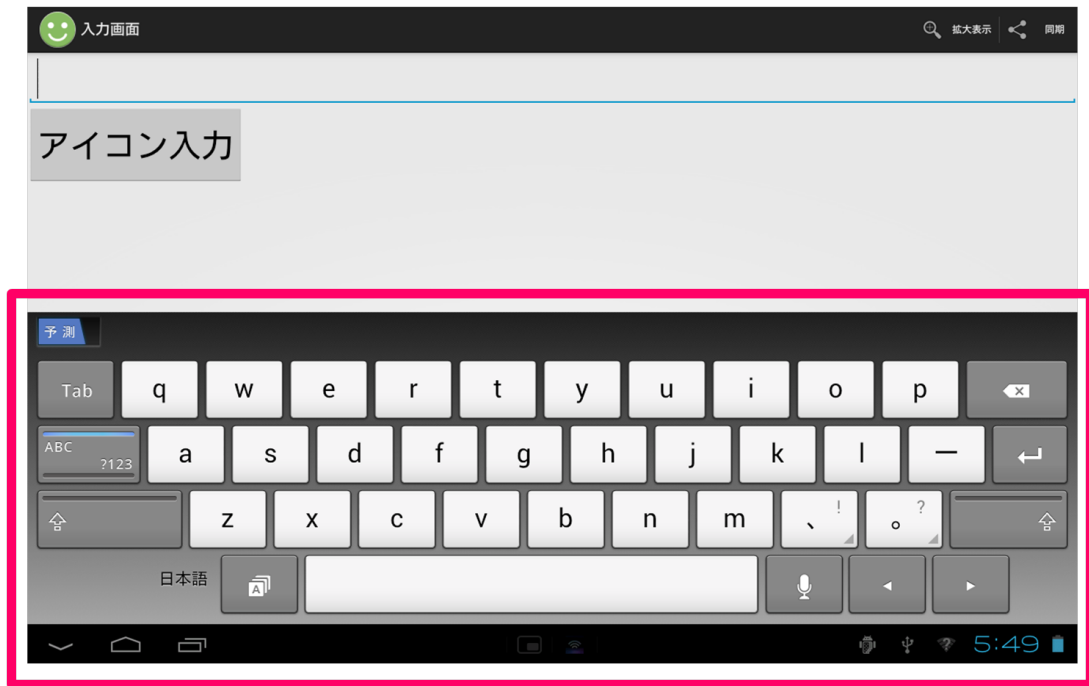


図 5.1 ソフトウェアキーボードの入力画面

るのかどうか、評価実験の際に調査を行った。以下からその結果内容を示していく。

アイコンに登録されている文章を編集する機能

図 5.2 はアイコンに登録されている文章を編集する機能のイメージである。例として感情アイコンの1つの文章を変更させている。既の実装されているアイコンの文章を変更させることで、使用者にとって使い易い文章をカスタマイズすることが出来ると考えている。

また、表 5.1 から、実装された場合使用する可能性が高いことが分かる。

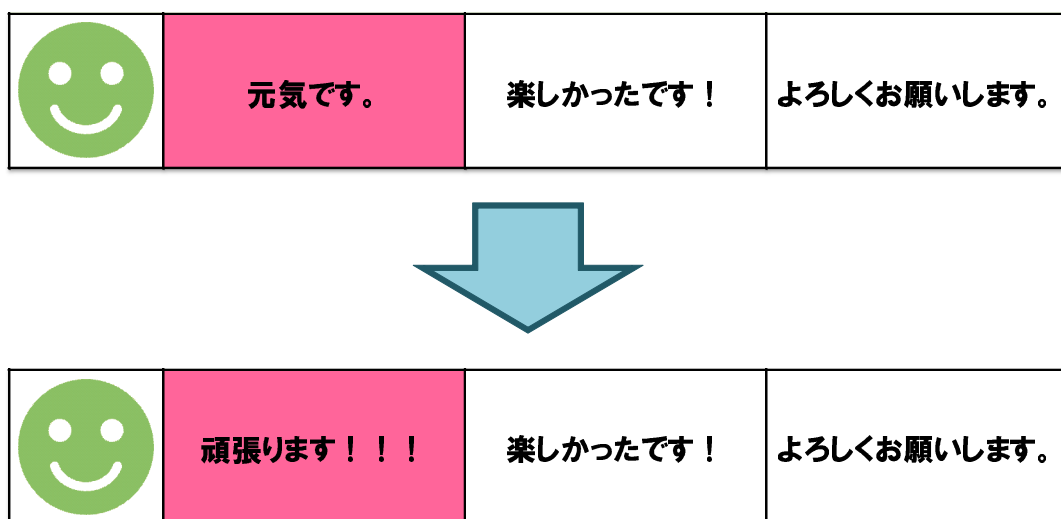


図 5.2 アイコンに登録されている文章を編集する機能

表 5.1 アイコンに登録されている文章を編集する機能についての調査

	使用する	分からない	使用しない
大学生 A	5	0	0
大学生 B	5	0	0
高齢者 A	4	1	0
高齢者 B	4	1	0
計	18	2	0
%	90	10	0

アイコンを追加できる機能

図 5.3 はアイコンを追加できる機能のイメージである。例として『星』のアイコンを追加している。使用者が作成したい文章に合わせて新しく登録することで汎用性の問題を解決できるのではないかと考えている。

また、表 5.2 から、使用者はこの機能を求めていることが分かる。



図 5.3 アイコンを追加できる機能

表 5.2 アイコンを追加できる機能についての調査

	使用する	分からない	使用しない
大学生 A	5	0	0
大学生 B	5	0	0
高齢者 A	5	0	0
高齢者 B	5	0	0
計	20	0	0
%	100	0	0

アイコンに相応しい文章を機械学習する機能

図 5.4 はアイコンに相応しい文章を機械学習する機能のイメージである。アイコン入力画面でも文章を変更させることが出来る。同じように変更していくことで機械学習していけば使用者にとって更に使い易いものになると考えている。また、表 5.3 から、使用者はこの機能を求めていることが分かる。



図 5.4 アイコンに相応しい文章を機械学習する機能

表 5.3 アイコンに相応しい文章を機械学習する機能についての調査

	使用する	分からない	使用しない
大学生 A	5	0	0
大学生 B	5	0	0
高齢者 A	5	0	0
高齢者 B	5	0	0
計	20	0	0
%	100	0	0

3つの機能を実装してもアイコンだけで文章を作成することには限界がある。しかしながら、現状のアプリケーションでも、表 4.15 より被験者から利用したいといった意見が多数あった。よってアイコン入力を用いる事で入力の仕方や文章作成に慣れるためのきっかけになり、コミュニケーションの輪が広がれば良いと考えている。

また、現状のアイコン入力はタブレット端末の画面の大半をアイコンで使用するため、大きさや数によって画面の体裁が崩れてしまう。今回使用した Sony タブレット端末は 9 インチであり、今後主流となるのは 7 インチ未満のモバイルタブレットである。そのため今回のアンケート調査での数や大きさで満足していた方はタブレット端末によっては意見が変わる場合もある。これらを解消するためにソフトウェアキーボードにアイコン入力方式を組み込むことや、アイコン毎にタブで切り替えて利用することが考えられる。

第6章

結論

本研究では、近年進歩を遂げているタブレット端末と SNS に注目した。タブレット端末のソフトウェアキーボードの使い難さと、一人暮らしの高齢者等がメールや SNS での家族や友人とのコミュニケーションを強化するために必要な文章作成において、タブレット端末で簡単に文章を作成できるような仕組みが必要と考えた。そこで本研究では、新たな入力方法として、直感的に把握し易く簡単に文章を作成できるようなアイコン入力方式を考案し、有用性を検証した。

作成したアプリケーションを実際に大学生、高齢者に利用してもらったと、高齢者の場合ソフトウェアキーボードの入力は大学生の2倍以上掛かっているが、アイコン入力では大学生と大差ない。このことからアイコン入力は視覚的、直感的に把握し易い文章作成の補助機能として成り立ち有用であると言える。また、今後アイコン入力を利用することでコミュニケーションは向上していくと捉えられており、アイコン入力の目的の一つであるコミュニケーションの向上が図れると分かった。

しかし、文章の汎用性の問題や、アイコン入力画面の情報量など課題は多く残っている。特に汎用性の問題はアイコン入力方式の最大のデメリットである。そのため、アイコンに登録されている文章を編集する機能、アイコンを追加できる機能、アイコンに相応しい文章を機械学習する機能などが必要であると考えられる。

また、本研究で使用したタブレット端末は9インチであり、今後主流となる7インチ未満のモバイルタブレットではアイコンの数や大きさが現状のものでは満足できるアプリケーションにはならない。アイコン入力をソフトウェアキーボードに組み込むことや、タブを利用して画面の情報量を改善するなどで解消できると考えられる。

以上のことから、現状のアイコン入力方式のアプリケーションにカスタマイズ性を持たせることで課題を解決できると考えられるが、高齢者向けのアプリケーションであるため複雑になってしまうと本末転倒になってしまうことが懸念される。しかしながら、現状のアプリケーションでも表 4.15 より、ターゲットである高齢者の80%がアイコン入力をメールや SNS で利用したいと回答している。そのため、アイコン入力方式を用いる事で入力の仕方や文章作成に慣れるためのきっかけになり、メールや SNS を通じてコミュニケーションの輪が広がればと考えている。

謝辞

本研究に際して、様々なご指導を頂きました服部峻先生に深謝いたします。研究室の皆様には、研究の内容から評価実験方法など多くのご指摘を頂きました。ありがとうございました。また、事前調査、評価実験の際に本研究の趣旨を理解し、時間を割いて協力して頂いた皆様に心から感謝します。

参考文献

- [1] 株式会社 MM 総研, “2012 年度上期国内タブレット端末出荷概要,” <http://www.m2ri.jp/newsreleases/main.php?id=010120121205500> (2012).
- [2] 株式会社ジー・エフ, “シニア・高齢者のインターネット利用に関する調査,” <http://reposes.jp/3530/13/83.html> (2012).
- [3] 高林 哲, 塚田 浩二, 増井 俊之, “顔アイコン: 手軽なファイル転送システム,” インタラクシオン 2003, 情報処理学会シンポジウム論文集, Vol.2003, No.7, pp.33–34 (2003).
- [4] 神原 啓介, 塚田 浩二, “ソーシャル顔アイコン,” 日本ソフトウェア科学会論文誌 (コンピュータソフトウェア), Vol.28, No.2, pp.172–182 (2011).
- [5] 遠山 美優, 服部 峻, “タブレット端末のためのアイコンに基づく入力方式,” 情報処理学会第 75 回全国大会, 3ZA-3 (2013). [口頭発表予定]

付録 A

以下より本研究で作成したアイコン入力方式アプリケーションのソースコードを載せる。

MainActivity

MainActivity.java

```
package com.example.icon_input_ver_3_5;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.text.Editable;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity {
    private EditText editText;
    private String resultData;
    private Button button;
    float size;
```

MainActivity.java 続き

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    getWindow().requestFeature(Window.FEATURE_ACTION_BAR);
    setContentView(R.layout.main);
    button = (Button) findViewById(R.id.result_button);
    button.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this,
                                       IconActivity.class);

            int requestCode = 123;
            startActivityForResult(intent, requestCode);
        }
    });
}

protected void onActivityResult
(int requestCode, int resultCode, Intent intent) {
    super.onActivityResult(requestCode, resultCode, intent);
    @SuppressWarnings("unused")
    Bundle bundle = intent.getExtras();
    switch (requestCode) {
    case 123:

        resultData = intent.getStringExtra("INPUTDATA");
        EditText edit = (EditText)findViewById(R.id.editText1);
        int start = edit.getSelectionStart(); //カーソル位置の取得
        int end = edit.getSelectionEnd();
        edit.getText().replace( Math.min( start, end ),
                               Math.max( start, end ), resultData );

        break;
    }
}
```

MainActivity.java 続き

```
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);

    //インスタンスの保存
    outState.putString("EDITTEXT_KEY",
        ((EditText)findViewById(R.id.editText1)).getText().toString() );
}

protected void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
    //インスタンスの復帰
    String str = savedInstanceState.getString
        ("EDITTEXT_KEY"((EditText)findViewById
        (R.id.editText1)).Text(str);

}

public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

// メニューボタン押下時の処理
public boolean onOptionsItemSelected(MenuItem item) {
    Toast.makeText(this, "Selected Item: " + item.getTitle(),
        Toast.LENGTH_SHORT).show();

    switch(item.getItemId()){
        case R.id.appzoom: /*文字サイズ拡大*/
            editText.setTextSize(editText.getTextSize()+6);
            break;

    } return super.onOptionsItemSelected(item);
}
```

IconActivity

IconActivity.java

```
package com.example.icon_input_ver_3_5;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager.LayoutParams;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;

public class IconActivity extends Activity {
    private EditText inputData;
    private Button Mbutton;
    //イメージボタン用変数
    //感情アイコン
    private ImageButton imagButton1_1;//『楽』
    private ImageButton imagButton1_2;//『怒』
    private ImageButton imagButton1_3;//『悲』
    private ImageButton imagButton1_4;//『焦り』
    private ImageButton imagButton1_5;//『驚き』
    //行動アイコン
    private ImageButton imagButton2_1;//『買い物』
    private ImageButton imagButton2_2;//『車』
    private ImageButton imagButton2_3;//『音楽』
    private ImageButton imagButton2_4;//『食事』
    //天気アイコン
    private ImageButton imagButton3_1;//『晴れ』
    private ImageButton imagButton3_2;//『雨』
    private ImageButton imagButton3_3;//『雪』
    private ImageButton imagButton3_4;//『風』
```

MainActivity.java 続き

```
//カウント変数
int clickcount[] = new int[12];

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    this.getWindow().setSoftInputMode(
        LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDDEN);
    setContentView(R.layout.icon1);
    inputData=(EditText) findViewById(R.id.editText2);
    Mbutton = (Button) findViewById(R.id.move_button1);

    //Intent,Bundle 制御
    Mbutton.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        Intent intent = new Intent(IconActivity.this, MainActivity.class);
        Bundle bundle = new Bundle();
        bundle.putString("INPUTDATA", inputData.getText().toString());
        intent.putExtras(bundle);
        // setResult() で bundle を載せ、送る Intent の data(intent) をセットする
        setResult(RESULT_OK, intent);
        // finish() で終わらせて Intent data を送る
        finish();
    }
    });
};
```

MainActivity.java 続き

```
//感情アイコン
//楽
imageView1_1=(ImageButton) findViewById(R.id.imageView1);
//変数とXMLで作ったイメージボタンを紐付け
imageView1_1.setOnClickListener(new View.OnClickListener(){
public void onClick(View v) {
switch(clickcount[0]){//値によって分岐

case 0://1回目
    inputData.setText("元気です。");
//カーソル処理
    CharSequence str = inputData.getText();
    inputData.setSelection(str.length());
    clickcount[0]++;
    break;

case 1://2回目
    inputData.setText("楽しかったです!");
    CharSequence str1 = inputData.getText();
    inputData.setSelection(str1.length());
    clickcount[0]++;
    break;

case 2://3回目
    inputData.setText("よろしくお願いします。");
    CharSequence str11 = inputData.getText();
    inputData.setSelection(str11.length());
    clickcount[0]=0;
    break;

default:
    clickcount[0]=0;
    break;
} } } );
```

MainActivity.java 続き

```
//怒
imagButton1_2=(ImageButton) findViewById(R.id.imageButton3);
imagButton1_2.setOnClickListener(new View.OnClickListener(){
public void onClick(View v) {
switch(clickcount[1]){//値によって分岐

case 0://1回目
    inputData.setText("怒っています。");
    CharSequence str = inputData.getText();
    inputData.setSelection(str.length());
    clickcount[1]++;
    break;

case 1://2回目
    inputData.setText("理不尽でした。");
    CharSequence str1 = inputData.getText();
    inputData.setSelection(str1.length());
    clickcount[1]=0;
    break;

default:
    clickcount[1]=0;
    break;
} } } );
```


MainActivity.java 続き

```
//悲
imagButton1_3=(ImageButton) findViewById(R.id.imageButton4);
imagButton1_3.setOnClickListener(new View.OnClickListener(){
public void onClick(View v) {
switch(clickcount[2]){//値によって分岐

case 0://1回目
    inputData.setText("とても悲しいです。");
    CharSequence str = inputData.getText();
    inputData.setSelection(str.length());
    clickcount[2]++;
    break;

case 1://2回目
    inputData.setText("調子がよくありません。");
    CharSequence str1 = inputData.getText();
    inputData.setSelection(str1.length());
    clickcount[2]++;
    break;

case 2://3回目
    inputData.setText("とても残念です。");
    CharSequence str11 = inputData.getText();
    inputData.setSelection(str11.length());
    clickcount[2]=0;
    break;

default:
    clickcount[2]=0;
    break;
} } } );
```

IconActivity.java 続き

```
//焦り
imagButton1_4=(ImageButton) findViewById(R.id.imageButton5);
imagButton1_4.setOnClickListener(new View.OnClickListener(){
public void onClick(View v) {
switch(clickcount[3]){//値によって分岐

case 0://1回目
    inputData.setText("焦りました・・・");
    CharSequence str = inputData.getText();
    inputData.setSelection(str.length());
    clickcount[3]++;
    break;

case 1://2回目
    inputData.setText("怖かったです。");
    CharSequence str1 = inputData.getText();
    inputData.setSelection(str1.length());
    clickcount[3]++;
    break;

case 2://3回目
    inputData.setText("難しいですね。");
    CharSequence str11 = inputData.getText();
    inputData.setSelection(str11.length());
    clickcount[3]=0;
    break;

default:
    clickcount[3]=0;
    break;
} } } );
```

IconActivity.java 続き

```
//驚き
imageView1_5=(ImageButton) findViewById(R.id.imageView6);
imageView1_5.setOnClickListener(new View.OnClickListener(){
public void onClick(View v) {
switch(clickcount[4]){//値によって分岐

case 0://1回目
    inputData.setText("驚きました!");
    CharSequence str = inputData.getText();
    inputData.setSelection(str.length());
    clickcount[4]++;
    break;

case 1://2回目
    inputData.setText("すごいですね!");
    CharSequence str1 = inputData.getText();
    inputData.setSelection(str1.length());
    clickcount[4]++;
    break;

case 2://3回目
    inputData.setText("びっくりしました!");
    CharSequence str11 = inputData.getText();
    inputData.setSelection(str11.length());
    clickcount[4]=0;
    break;

default:
    clickcount[4]=0;
    break;
} } } );
```

IconActivity.java 続き

```
//行動アイコン
//買い物
imagButton2_1=(ImageButton) findViewById(R.id.imageButton2);
imagButton2_1.setOnClickListener(new View.OnClickListener(){
    public void onClick(View v) {
        switch(clickcount[5]){//値によって分岐

            case 0://1回目
                inputData.setText("買い物に行きました。");
                CharSequence str = inputData.getText();
                inputData.setSelection(str.length());
                clickcount[5]++;
                break;

            case 1://2回目
                inputData.setText("今日はショッピングです。");
                CharSequence str1 = inputData.getText();
                inputData.setSelection(str1.length());
                clickcount[5]++;
                break;

            case 2://3回目
                inputData.setText("買い物に行ってきて下さい。");
                CharSequence str11 = inputData.getText();
                inputData.setSelection(str11.length());
                clickcount[5]=0;
                break;

            default:
                clickcount[5]=0;
                break;
        } } } );
```

IconActivity.java 続き

```
//車
imagButton2_2=(ImageButton) findViewById(R.id.imageButton7);
imagButton2_2.setOnClickListener(new View.OnClickListener(){
public void onClick(View v) {
switch(clickcount[6]){//値によって分岐

case 0://1回目
    inputData.setText("ドライブに行きました。");
    CharSequence str = inputData.getText();
    inputData.setSelection(str.length());
    clickcount[6]++;
    break;

case 1://2回目
    inputData.setText("車に乗りました。");
    CharSequence str1 = inputData.getText();
    inputData.setSelection(str1.length());
    clickcount[6]++;
    break;

case 2://3回目
    inputData.setText("車を運転しました。");
    CharSequence str11 = inputData.getText();
    inputData.setSelection(str11.length());
    clickcount[6]=0;
    break;

default:
    clickcount[6]=0;
    break;
} } } );
```

IconActivity.java 続き

```
//音楽
imagButton2_3=(ImageButton) findViewById(R.id.imageButton8);
imagButton2_3.setOnClickListener(new View.OnClickListener(){
public void onClick(View v) {
switch(clickcount[7]){//値によって分岐

case 0://1回目
    inputData.setText("音楽を聴きました。");
    CharSequence str = inputData.getText();
    inputData.setSelection(str.length());
    clickcount[7]++;
    break;

case 1://2回目
    inputData.setText("演奏しました。");
    CharSequence str1 = inputData.getText();
    inputData.setSelection(str1.length());
    clickcount[7]++;
    break;

case 2://3回目
    inputData.setText("コンサートに行きました。");
    CharSequence str11 = inputData.getText();
    inputData.setSelection(str11.length());
    clickcount[7]=0;
    break;

default:
    clickcount[7]=0;
    break;
} } } );
```

IconActivity.java 続き

```
//食事
imageView2_4=(ImageButton) findViewById(R.id.imageView9);
imageView2_4.setOnClickListener(new View.OnClickListener(){
public void onClick(View v) {
switch(clickcount[8]){//値によって分岐
case 0://1回目
    inputData.setText("料理を食べました。");
//カーソル処理
    CharSequence str = inputData.getText();
    inputData.setSelection(str.length());
    clickcount[8]++;
    break;

case 1://2回目
    inputData.setText("料理をしました。");
    CharSequence str1 = inputData.getText();
    inputData.setSelection(str1.length());
    clickcount[8]++;
    break;

case 2://3回目
    inputData.setText("外食に行きました。");
    CharSequence str11 = inputData.getText();
    inputData.setSelection(str11.length());
    clickcount[8]=0;
    break;

default:
    clickcount[8]=0;
    break;
} } } );
```

IconActivity.java 続き

```
//天気アイコン
//晴れ
imagButton3_1=(ImageButton) findViewById(R.id.imageButton10);
imagButton3_1.setOnClickListener(new View.OnClickListener(){
public void onClick(View v) {
switch(clickcount[9]){//値によって分岐

case 0://1回目
    inputData.setText("今日は晴れでした。");
    CharSequence str = inputData.getText();
    inputData.setSelection(str.length());
    clickcount[9]++;
    break;

case 1://2回目
    inputData.setText("天気が良いです。");
    CharSequence str1 = inputData.getText();
    inputData.setSelection(str1.length());
    clickcount[9]++;
    break;

case 2://3回目
    inputData.setText("暑いですね。");
    CharSequence str11 = inputData.getText();
    inputData.setSelection(str11.length());
    clickcount[9]=0;
    break;

default:
    clickcount[9]=0;
    break;
} } } );
```


IconActivity.java 続き

```
//雨
imagButton3_2=(ImageButton) findViewById(R.id.imageButton11);
imagButton3_2.setOnClickListener(new View.OnClickListener(){
public void onClick(View v) {
switch(clickcount[10]){//値によって分岐

case 0://1回目
    inputData.setText("今日は雨でした。");
//カーソル処理
    CharSequence str = inputData.getText();
    inputData.setSelection(str.length());
    clickcount[10]++;
    break;

case 1://2回目
    inputData.setText("天気が悪いです。");
    CharSequence str1 = inputData.getText();
    inputData.setSelection(str1.length());
    clickcount[10]++;
    break;

case 2://3回目
    inputData.setText("肌寒いですね。");
    CharSequence str11 = inputData.getText();
    inputData.setSelection(str11.length());
    clickcount[10]=0;
    break;

default:
    clickcount[10]=0;
    break;
} } } );
```

IconActivity.java 続き

```
//雪
imageView3_3=(ImageButton) findViewById(R.id.imageView12);
imageView3_3.setOnClickListener(new View.OnClickListener(){
public void onClick(View v) {
switch(clickcount[11]){//値によって分岐

case 0://1回目
    inputData.setText("今日は雪でした。");
    CharSequence str = inputData.getText();
    inputData.setSelection(str.length());
    clickcount[11]++;
    break;

case 1://2回目
    inputData.setText("雪が積もりました。");
    CharSequence str1 = inputData.getText();
    inputData.setSelection(str1.length());
    clickcount[11]++;
    break;

case 2://3回目
    inputData.setText("寒いです。。。");
    CharSequence str11 = inputData.getText();
    inputData.setSelection(str11.length());
    clickcount[11]=0;
    break;

default:
    clickcount[11]=0;
    break;
} } } );
```

IconActivity.java 続き

```
//台風
imagButton3_4=(ImageButton) findViewById(R.id.imageButton13);
imagButton3_4.setOnClickListener(new View.OnClickListener(){
public void onClick(View v) {
switch(clickcount[12]){//値によって分岐
case 0://1回目
    inputData.setText("今日は風が強かったです。");
    CharSequence str = inputData.getText();
    inputData.setSelection(str.length());
    clickcount[12]++;
    break;

case 1://2回目
    inputData.setText("台風でした。");
    CharSequence str1 = inputData.getText();
    inputData.setSelection(str1.length());
    clickcount[12]=0;
    break;

default:
    clickcount[12]=0;
    break;
} } } );
} }
```