

Uma Ferramenta de Mobile Learning para Aprendizado e Treinamento de Escrita Japonesa

Ed Charles Nascimento Ferreira, Antonio Lucas de Almeida
Alcides Xavier Benicasa

¹ Departamento de Sistemas de Informação – Universidade Federal de Sergipe (UFS)
Itabaiana – SE – Brasil

charles.si.ufs@gmail.com, lucas.d1995@gmail.com, alcides@ufs.br

Abstract. *Given the great growth of mobile technologies in recent years and its application as a learning tool, this work aims to develop a mobile application for the Android platform aimed at learning and practicing Japanese writing, specifically the Hiragana and Katakana alphabets. The work presents a model with two Multilayer Perceptron neural networks, one for the classification of the pixels drawn by the user, and another to classify the directions of the strokes, considering the relevance of this matter in the Japanese writing system. The accuracy of the model is tested with samples collected in a production environment. In general, the model presented interesting results, showing great potential for future improvements and research.*

Resumo. *Dado o grande crescimento das tecnologias móveis nos últimos anos e sua aplicação como ferramenta de apoio ao ensino, este trabalho tem como objetivo o desenvolvimento de uma aplicação móvel para a plataforma Android voltada ao aprendizado e prática de escrita japonesa, mais especificamente os alfabetos Hiragana e Katakana. O trabalho apresenta um modelo com duas redes neurais Multilayer Perceptron, uma para classificação dos pixels desenhados pelo usuário, e outra para classificar as direções dos traços, tendo em vista a relevância deste quesito no sistema de escrita japonês. A acurácia do modelo é testada com amostras coletadas em ambiente de produção. No geral, o modelo apresentou resultados interessantes, demonstrando bastante potencial para futuros aperfeiçoamentos e pesquisas.*

1. Introdução

Nos últimos anos os *smartphones* e *tablets* têm se tornado dispositivos bastante comuns ao cotidiano das pessoas. De acordo com Statista (2018a), a quantidade de usuários de *smartphones* ao redor do mundo atingirá a marca de 2.53 bilhões em 2018, o que representará aproximadamente 33,3% da população mundial (STATISTA, 2018c), e ainda prevê que em 2020 este número chegará a aproximadamente 2,87 bilhões, o que representará aproximadamente 37,0% da população mundial. Com relação a *tablets*, (STATISTA, 2018b) cita que a quantidade de usuários deste tipo de dispositivo em todo o mundo vai ultrapassar a casa de 1,3 bilhão em 2018, número que representará aproximadamente 17,7% da população mundial (STATISTA, 2018d), e que em 2020 este número tende a alcançar a marca de aproximadamente 1.46 bilhões, cerca de 19,2% da população mundial.

Dentro do mundo dos aplicativos educacionais, há uma área relativamente nova que vem ganhando visibilidade e tendo vários trabalhos desenvolvidos, esta área é

chamada de *m-learning*, ou *mobile learning*. Lan e Sie (2010) descreve *mobile learning* como um modelo de aprendizado que permite ao aprendiz obter materiais de aprendizado em qualquer lugar e a qualquer hora utilizando tecnologias *mobile* e a *internet*. Atualmente, é possível encontrar vários aplicativos de *m-learning* focados em ensinar escrita japonesa, como é o caso de algumas ferramentas apresentadas na seção de revisão bibliográfica.

Apesar da existência de uma grande variedade de aplicativos, um importante ponto de investigação deste trabalho está relacionado à didática utilizada por estes. Por exemplo, questões pertinentes à avaliação da escrita do usuário, que muitas vezes não ocorre, o acompanhamento do usuário, indicando onde está o erro ou que haja necessidade de melhorias, ou ainda, permitir ao usuário liberdade para a escrita dos caracteres, de forma a possibilitar sua interação de maneira mais intuitiva.

Ao longo do tempo, muitas pesquisas vêm sendo feitas com o objetivo de solucionar problemas relacionados a idiomas, tal como a língua japonesa, como será visto no capítulo de trabalhos relacionados. Uma das áreas que têm dado grande contribuição neste sentido é a Inteligência Artificial (IA) e, de acordo com os objetivos deste trabalho, mais especificamente as Redes Neurais Artificiais (RNA). Para Braga, Carvalho e Ludermir (2000), a solução de problemas gerais utilizando redes neurais artificiais é bastante interessante. Isso ocorre principalmente devido ao paralelismo inerente a sua arquitetura que permite que ela apresente um desempenho superior aos demais modelos factícios. A solução de problemas ocorre principalmente devido a sua capacidade de aprender através de exemplos e a partir daí generalizar a informação aprendida.

Sendo assim, o trabalho proposto visa o desenvolvimento de um aplicativo móvel que avalie a escrita japonesa do usuário para caracteres dos alfabetos *Hiragana* e *Katakana*, dando liberdade ao mesmo para desenhar em qualquer posição e tamanho, levando em consideração a direção dos traços. A aplicação deverá avaliar a correção da escrita do usuário e fornecer *feedbacks* sobre seu desempenho. Para isso, será aplicado o uso de redes neurais artificiais, mais precisamente do tipo *Multi-Layer Perceptron*, tendo em vista que é um modelo já utilizado para solucionar problemas similares.

2. Trabalhos Relacionados

Desde a década de 80, muitos trabalhos focados no reconhecimento de escrita japonesa, tanto *off-line* quanto *on-line*, têm sido sugeridos. Alguns desses trabalhos, em ordem cronológica, serão apresentadas juntamente com suas principais características.

Zhu et al. (2010) descreve um modelo de integração de contexto robusto para reconhecimento *on-line* de texto japonês escrito à mão. Com base em aproximação de probabilidade de classe de cadeia, o método proposto avalia a probabilidade de caminhos candidatos a segmentação e reconhecimento através da combinação dos escores de reconhecimento de caracteres, características geométricas unárias e binárias, bem como contexto linguístico. O critério de avaliação de caminho pode combinar de forma flexível os escores dos vários contextos e é insensível à variabilidade no comprimento do caminho, e assim, o caminho de segmentação ótimo com sua classe de cadeia pode ser efetivamente encontrado com a busca de Viterbi.

Murai et al. (2010) apresenta um sistema com o objetivo de ensinar pessoas com deficiência visual a escrever *kanji* usando uma interface háptica. O sistema é inspirado

na aplicabilidade direta do modo *dual-stylus* de dispositivos táteis para imitar o método de prática entre professor-aluno de segurar a mão. O sistema assiste o controle do *stylus* pelo usuário ao fornecer forças táteis em sua caneta. O sistema destina-se a autoaprendizagem repetitiva da escrita *kanji* sem a ajuda de outras pessoas, e são disponibilizados os seguintes treinamentos de escrita: prática pelo método do professor de segurar a mão, e autoaprendizagem sob vários modos de controles táteis.

O Kanji Draw de Leafdigital (2011) foca no reconhecimento de um símbolo desenhado pelo usuário, sem nenhuma pista inicial do que será desenhado. O ponto forte deste aplicativo é que o mesmo leva em consideração a quantidade, ordem e direção dos traços no processo de reconhecimento. Como deficiência do aplicativo, pode-se citar o fato de que o reconhecimento é a única função do mesmo, não mostrando, por exemplo, informações sobre o símbolo reconhecido.

Na lista dos aplicativos mais populares do gênero, está o Kanji Recognizer de Elenkov (2015), que reconhece e avalia símbolos com um ótimo aproveitamento. O aplicativo mostra uma boa quantidade de informações sobre o símbolo reconhecido. Os três alfabetos japoneses podem ser reconhecidos pelo aplicativo, porém o aplicativo apenas mostra informações sobre os *kanjis*. O reconhecimento de escrita desta ferramenta leva em consideração a ordem e a direção dos traços, característica muito importante. Também é possível ao usuário da aplicação treinar sua escrita japonesa, “cobrindo” símbolos mostrados na tela.

O Kanji Draw de Lusil (2016) é mais um aplicativo que foca na avaliação da escrita japonesa do usuário, avaliando como certo ou errado. A ferramenta fornece o modo de treinamento e o modo de teste, além de fornecer bastante informações sobre os símbolos e estatísticas de aproveitamento do usuário. Em termos de avaliação/reconhecimento, a ferramenta leva em consideração a quantidade, ordem e direção dos traços, e além disso, possui uma característica muito interessante, que é a possibilidade de o usuário escolher o nível de tolerância da avaliação, tornando-a mais rígida ou tolerável. O Kanji Draw suporta apenas o alfabeto *kanji*, mas é importante observar que também existe o Kana Draw, que também é de (LUSIL, 2014), e que tem as mesmas características do Kanji Draw, porém dá suporte aos alfabetos *hiragana* e *katakana*.

Sendo assim, o trabalho aqui proposto visa o desenvolvimento de um aplicativo móvel que avalie a escrita japonesa do usuário, especificamente os alfabetos *hiragana* e *katakana*. Com relação às suas características, pretende-se dar liberdade ao usuário para desenhar em qualquer posição e tamanho, levar em consideração a quantidade, ordem e direção dos traços, atribuir uma nota para a escrita do usuário, e fornecer dicas de como melhorar seu desenho. As notas dadas pelo aplicativo formarão um *ranking*, que servirá como referência para o usuário.

3. Modelo

Nesta seção serão apresentadas as estruturas e passos que compõem o modelo proposto para a solução do problema em questão. Primeiramente, serão apresentadas as redes neurais utilizadas e os passos necessários para que as redes avaliem o caractere desenhado pelo usuário no aplicativo *mobile*. Por último, o aplicativo proposto será apresentado com mais detalhes.

3.1. Redes Neurais

Este modelo propõe duas redes neurais *Perceptron* Multicamada com três camadas (entrada, intermediária e saída) para a solução deste problema, onde a primeira é responsável pelo reconhecimento dos *pixels* que representam o caractere desenhado, e a segunda rede é encarregada de avaliar os traços feitos pelo usuário no processo de desenho.

A rede neural de *pixels* é responsável por avaliar os *pixels* do desenho feito pelo usuário. Para a entrada da rede, cada padrão provém da normalização de uma imagem de dimensões 24x24, o que gera um total de 576 valores, que portanto é a quantidade total de neurônios na camada de saída. Para a camada intermediária da rede neural, foi definida uma quantidade de 512 neurônios, pelo fato de que este valor alcança uma boa taxa de convergência e ao mesmo tempo gera um arquivo de pesos com um tamanho aceitável de cerca de 2.6 MegaBytes.

A rede neural de traços é responsável por avaliar os traços feitos pelo usuário no processo de desenho do caractere. Foi realizada uma análise a respeito dos traços de todos os 92 caracteres dos dois alfabetos abordados por este trabalho e foi constatado que a maior quantidade possível de traços de um caractere é 4 e que existem 37 padrões de direção dentro deste conjunto de símbolos. Padrões de direção são sequências de traços que se repetem ou não entre os 92 caracteres analisados. Sendo possível projetar uma rede neural para a solução desta questão dos traços. Para a camada de entrada, temos 32 neurônios.

3.1.1. Treinamento e Reconhecimento

Para o treinamento das redes neurais, foi desenvolvida uma aplicação *desktop* Java de linha de comando. O algoritmo de treinamento da rede neural de *pixels* itera sobre todas as imagens de caracteres na pasta especificada no arquivo de configuração do treinamento, primeiramente normalizando o conteúdo da mesma, ou seja, transformando todos os valores de *pixels* em -1 ou 1. Logo após, é gerado um *array* que representa as saídas esperadas para o caractere atual na rede neural. O *array* contém um tamanho de 92 posições e é constituído de 91 valores “-1” e apenas 1 valor “1”.

O algoritmo de treinamento da rede neural de traços se assemelha bastante com o da rede de *pixels*, com a diferença de que, ao invés de iterar pelas imagens de uma pasta, ele itera sobre os padrões de direção em um arquivo JSON especificado no arquivo de configuração do treinamento de traços.

O processo de reconhecimento é feito pela aplicação *mobile* desenvolvida, que utiliza os arquivos de peso gerados durante o treinamento. Porém, antes de qualquer reconhecimento, se faz necessária a captura dos dados que serão as entradas das redes neurais. Para os pixels, a aplicação *mobile* disponibiliza uma tela de pintura onde o usuário desenhará o caractere. A identificação das direções dos traços desenhados se dá por meio da análise individual de cada segmento desenhado. Ao posicionar o dedo ou a caneta sobre a tela de pintura, o ponto inicial do traço é armazenado.

3.2. Classificação ponderada entre pixels e traço

Após as duas redes neurais efetuarem a classificação de suas respectivas entradas, ambas as saídas são combinadas em uma única saída. O cálculo desta nova saída é feito apli-

cando uma média ponderada entre a saída da rede de *pixels* e a saída correspondente da rede de traços. Este cálculo é feito sobre as saídas de todos os caracteres abordados por este trabalho, o que resultará no mesmo número de 92 saídas da rede de *pixels*, porém ponderadas pela saída correspondente ao caractere na rede de traços. Por exemplo, caso um caractere com ID 5 tenha sido definido como tendo o padrão de direção de ID 10, a saída 5 da rede de *pixels* é ponderada pela saída 10 da rede de traços, o que dará o valor da nova saída.

Um ponto importante nesta etapa da avaliação é a definição de um valor *lambda* (λ) que representará os pesos de ambas as redes neurais. Diante do fato de que a direção dos traços é muito importante para a escrita japonesa, porém não sendo vital para o entendimento do leitor final, foi definido que os *pixels* desenhados pelo usuário terão um peso de 70% na saída ponderada, o que deixa a rede de traços com um peso de 30%. Estes valores se resumem em um valor de *lambda* igual a 0.3. A Equação 1 mostra como é feito o cálculo da saída final.

$$S_f = (1 - \lambda)S_p + \lambda S_t \quad (1)$$

Para decidir se o desenho do usuário é de fato o caractere treinado, as saídas ponderadas são analisadas em busca do maior valor. É checado então se a maior saída corresponde à saída do caractere treinado. Caso negativo, o usuário é informado que seu desenho apresenta falhas e também recebe uma dica de como melhorar. Caso positivo, uma nota é atribuída ao seu desenho. O cálculo desta nota é explicado na próxima subseção.

3.3. Aplicativo Mobile

O aplicativo *mobile* é uma aplicação específica para a plataforma *Android*, e conta com alguns recursos extras além da avaliação da escrita *Hiragana* e *Katakana* do usuário. O diagrama mostrado na Figura 1 apresenta uma visão geral dos principais componentes da sua arquitetura.

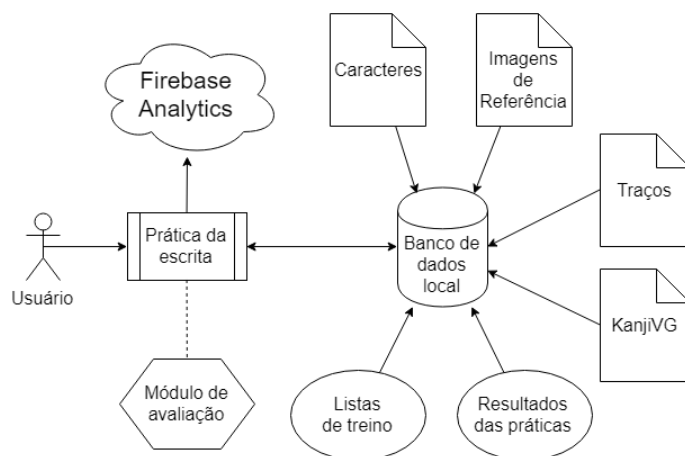


Figura 1. Diagrama da arquitetura da aplicação

Para funcionar, a aplicação precisa obter informações sobre os caracteres tratados neste trabalho, como metadados (símbolo, leitura, alfabeto, id do padrão de direção, etc),

sua imagem de referência para ser utilizada no cálculo da similaridade cosseno, os traços corretos para comparação com a os traços desenhados pelo usuário ou os gráficos vetoriais do *KanjiVG* utilizados na animação da maneira correta de se desenhar o caractere. Estas informações estão inicialmente contidas em arquivos no formato *JSON* e são adicionadas ao banco de dados local no primeiro uso do aplicativo após a instalação, com o intuito de facilitar a utilização destes dados posteriormente.

Quando o usuário realiza a prática de um caractere, o módulo de avaliação entra em ação fazendo uso das redes neurais para a checagem da corretude do desenho do usuário, e a similaridade cosseno para a atribuição de uma nota ao mesmo. O resultado é salvo no banco de dados local para futura consulta e também é enviado ao *Firebase Analytics* para possivelmente ser analisado pelo desenvolvedor.

3.4. Funcionalidades

O aprendizado ou prática da escrita japonesa é a principal funcionalidade da aplicação móvel desenvolvida, e esta ação se dá na tela de prática. Nesta tela o usuário irá praticar a sua escrita desenhando o caractere na tela utilizando a ponta do dedo ou uma caneta especial para telas com tecnologia *touch*. A tela de prática conta com, além do componente de pintura, um menu flutuante que disponibiliza as seguintes opções: desfazer o último traço, refazer último traço desfeito, mostrar animação da forma correta de desenhar (melhor explicado mais à frente), e avaliar o desenho, como pode ser visto na Figura 2.

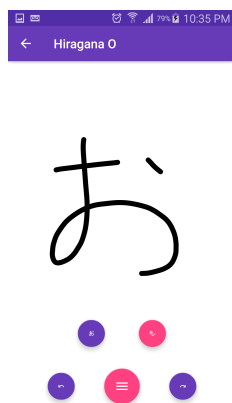


Figura 2. Tela de prática da escrita

Após o usuário desenhar o caractere e selecionar a opção de avaliar, o processo de reconhecimento e avaliação é executado, o resultado final é armazenado em um banco de dados local e é apresentado na tela de resultado. Esta tela apresenta a nota atribuída ao desenho, além de uma mensagem de *feedback* acerca do desempenho do aprendiz. Esta mensagem é baseada nas saídas das redes neurais e na nota recebida pelo usuário.

4. Experimentos e Resultados

Os resultados foram coletados a partir do uso da aplicação desenvolvida em uma versão alfa, liberada inicialmente para 7 pessoas com pouco ou nenhum conhecimento sobre a língua japonesa.

Os dados foram coletados através do uso do *Firestore Analytics* e do *BigQuery*, ambos serviços oferecidos pela *Google* e que se integram de maneira efetiva. O primeiro trata-se de um serviço de *analytics* que permite ao desenvolvedor analisar o comportamento dos usuários dentro de sua aplicação, enquanto que o segundo consiste em um serviço de *Data Warehouse* utilizado para análise de grandes volumes de dados. Mais objetivamente, o *BigQuery* é utilizado para a visualização mais detalhada dos eventos ocorridos dentro da aplicação registrados pelo *Firestore Analytics*.

5. Avaliação do Modelo

Para analisar a capacidade de classificação da rede neural, foram selecionados 8 caracteres que, de acordo com observações, apresentam uma boa chance de confundir a mesma. Para cada um destes caracteres, foram coletadas aproximadamente 20 amostras de desenhos feitos pelos usuários alfa consideradas próximas ao ideal. Após a rede neural classificar estas amostras, os resultados foram utilizados para montar a matriz de confusão que pode ser vista na Tabela 1. É importante salientar que esta análise levou em consideração apenas o desenho do usuário, tendo em vista que a rede neural de *pixels* possui maior relevância.

	H-O	H-MO	H-CHI	K-KU	K-RA	K-U	K-HU	K-WA
H-O	18	6	0	0	0	0	0	0
H-MO	0	14	4	0	0	0	0	0
H-CHI	0	2	12	0	0	0	0	0
K-KU	0	0	0	10	8	0	0	0
K-RA	0	0	0	0	12	0	0	0
K-U	0	0	0	0	0	16	0	0
K-HU	0	0	0	10	0	0	8	0
K-WA	0	0	0	0	0	5	11	17

Tabela 1. Matriz de confusão com os resultados dos caracteres selecionados. As colunas representam as classes reais enquanto as linhas representam os valores classificados pela rede neural

Em termos de precisão, que é taxa com que todos os exemplos classificados como positivos são realmente positivos, os destaques positivos são o *Katakana Ra* e o *Katakana U* nas linhas 5 e 6 respectivamente. Ambos receberam a pontuação máxima, o que mostra que a rede não confundiu nenhum outro caractere com estes dois. O destaque negativo é novamente o *Katakana Hu* na linha 7, já que 10 amostras do *Katakana Ku* foram confundidas com ele. Ao analisar o quesito Taxa de FN (falsos negativos), os destaques são os mesmos do quesito sensibilidade, tendo em vista que estas são medidas inversamente proporcionais.

No quesito acurácia, o *Hiragana O* e o *Katakana U* nas linhas 1 e 6 são os caracteres que apresentam maior destaque, superando a faixa de 70% de aproveitamento. O *Katakana Hu* na linha 7 é mais uma vez o destaque negativo com menos que 30% de aproveitamento, dada sua relevante quantidade de falsos positivos e negativos.

5.1. Avaliação do Aprendizado

Para avaliar o potencial de ensino da aplicação, foi feito um teste simples onde um caractere foi selecionado para ser praticado 15 vezes por 5 pessoas. As práticas se deram uma

Usuário 1	Usuário 2	Usuário 3	Usuário 4	Usuário 5
-	-	-	-	6.7
-	7.2	-	-	-
-	6.7	6.8	7.8	-
7.4	-	-	6.7	6.9
6.6	8.1	-	-	-
-	-	7.0	-	7.7
7.0	-	6.7	7.2	-
-	-	-	7.3	6.9
7.2	7.9	-	-	7.1
7.2	7.5	7.2	-	7.0
7.5	-	7.4	7.8	-
6.9.	8.0	-	7.7	8.1
8.2	7.8	7.1	-	7.2
-	7.7	7.8	7.4	7.5
7.9	8.2	7.7	7.8	7.4

Tabela 2. Tabela de resultados obtidos pelos usuários em ordem cronológica

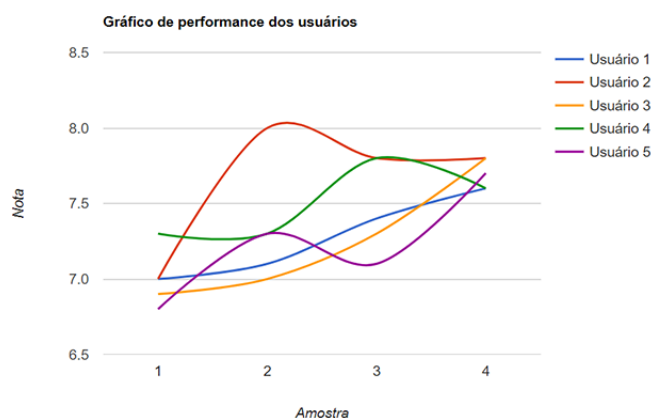


Figura 3. Gráfico de performance dos usuários

após a outra, sem muito tempo entre cada uma, apenas o suficiente para registrar o resultado. Não foi permitido aos usuários visualizar as práticas dos demais, com o intuito de obter o resultado mais fiel possível. O caractere escolhido para este teste foi o *Hiragana Na*, que é um caractere que possui 4 traços. A Tabela 2 apresenta todos os 15 resultados dos 5 usuários avaliados em ordem cronológica.

Para melhor visualização da performance dos usuários, a Figura 3 apresenta esta informação em um gráfico de linhas. Para a construção deste gráfico foram utilizadas as 8 primeiras notas válidas de cada usuário. Com o propósito de diminuir a quantidade de amostras a serem exibidas, as oito notas foram agrupadas (média aritmética) em 4 duplas, seguindo a mesma ordem cronológica. Estes valores são mostrados na Tabela 2.

Analisando o gráfico de performance dos usuários, é possível perceber que apesar de algumas instabilidades, os 5 usuários mostram uma perceptível evolução ao fi-

nal do processo. É importante mencionar que estas instabilidades, como no caso do usuário 5, possivelmente são uma consequência natural de ser o primeiro contato destes usuários com este caractere, e portanto, eles ainda estão numa fase de aprendizado ou experimentação, e não de aperfeiçoamento.

6. Conclusão

Diante dos resultados obtidos, a solução apresentada por este trabalho se mostrou válida para a avaliação da escrita japonesa do usuário, o que mostra que possui potencial de ser utilizado em outros sistemas de escrita. O aplicativo desenvolvido se mostra uma boa ferramenta para pessoas interessadas em aprender e praticar a escrita dos caracteres dos alfabetos *Hiragana* e *Katakana*.

Como limitação do trabalho, o método de avaliação das direções do traços ainda possui algumas falhas como, por exemplo, caso um caractere tenha dois traços para a direção leste, o modelo não consegue identificar qual dos dois traços foi feito primeiro, permitindo assim que o usuário desenhe os traços na ordem incorreta. Além disso, a detecção da direção do traço não é flexível, uma vez ela não tolera diferenças pequenas de graus. Assim, dado um traço que a direção correta é sudoeste, se o usuário desenhar um traço com direção muito próxima ao sudoeste, porém que está na faixa de grau da direção sul, o modelo irá considerar o traço como incorreto.

Como continuação deste trabalho, pretende-se dar suporte também ao alfabeto *kanji*, que possui um nível de complexidade maior que os dois alfabetos abordados neste trabalho. Há também a possibilidade de fazer da aplicação um Sistema Tutorial Inteligente (STI), o que significa que o usuário estará apto a receber *feedback* e instruções do software de maneira personalizada, de acordo com o seu perfil de aprendizado. Uma outra possibilidade de melhoria a ser analisada é a implementação de um Sistema de Repetição Espaçada (SRS) para conduzir os treinamentos do usuário, permitindo ao mesmo revisar com maior frequência os caracteres que possui maior dificuldade e com menor frequência os que possui maior facilidade. Naturalmente, também está nos planos a experimentação de outros tipos de redes neurais, além de outras medidas de similaridade, visando sempre buscar a evolução do modelo.

Referências

BRAGA, A. d. P.; CARVALHO, A.; LUDERMIR, T. B. *Redes neurais artificiais: teoria e aplicações*. [S.l.]: Livros Técnicos e Científicos, 2000.

ELENKOV, N. *Kanji Recognizer*. 2015. Disponível em: <<https://play.google.com/store/apps/details?id=org.nick.kanjirecognizer>>.

LAN, Y.-F.; SIE, Y.-S. Using rss to support mobile learning based on media richness theory. *Computers & Education*, Elsevier, v. 55, n. 2, p. 723–732, 2010.

LEAFDIGITAL. *Kanji Draw*. 2011. Disponível em: <<https://play.google.com/store/apps/details?id=com.leafdigital.kanji.android>>.

LUSIL. *Kana Draw*. 2014. Disponível em: <<https://play.google.com/store/apps/details?id=net.lusil.android.kanjidraw.kanaclassic>>.

LUSIL. *Kanji Draw*. 2016. Disponível em: [⟨https://play.google.com/store/apps/details?id=net.lusil.android.kanjidraw.jlpt5⟩](https://play.google.com/store/apps/details?id=net.lusil.android.kanjidraw.jlpt5).

MURAI, Y. et al. Kanji writing training with haptic interface for the visually impaired. In: *World Automation Congress (WAC), 2010*. [S.l.: s.n.], 2010. p. 1–6. ISSN 2154-4824.

STATISTA. *Number of smartphone users worldwide from 2014 to 2020*. 2018. Disponível em: [⟨https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/⟩](https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/).

STATISTA. *Number of tablet users worldwide from 2013 to 2020*. 2018. Disponível em: [⟨https://www.statista.com/statistics/377977/tablet-users-worldwide-forecast/⟩](https://www.statista.com/statistics/377977/tablet-users-worldwide-forecast/).

STATISTA. *Smartphone user penetration as percentage of total global population from 2014 to 2020*. 2018. Disponível em: [⟨https://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005/⟩](https://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005/).

STATISTA. *Tablet user penetration worldwide as share of population from 2014 to 2020*. 2018. Disponível em: [⟨https://www.statista.com/statistics/219909/global-tablet-penetration-forecast/⟩](https://www.statista.com/statistics/219909/global-tablet-penetration-forecast/).

ZHU, B. et al. A robust model for on-line handwritten japanese text recognition. *International Journal on Document Analysis and Recognition (IJ DAR)*, v. 13, n. 2, p. 121–131, 2010. ISSN 1433-2825. Disponível em: [⟨http://dx.doi.org/10.1007/s10032-009-0111-y⟩](http://dx.doi.org/10.1007/s10032-009-0111-y).