Implementação do Jogo *Rock-Paper-Scissors-Lizard-Spock* Utilizando Reconhecimentos de Gestos via K-means e Redes Neurais Artificiais

Vitor Coimbra
Hugo Silva
UnB/CIC
Brasília-DF

Alexandre Zaghetto
Bruno Macchiavello
UnB/CIC/LISA
Brasília-DF

Marcus Vinicius Lamar
UnB/CIC
Brasília-DF

Resumo

Gestos manuais são um componente importante para comunicação não verbal. Como consequência, o reconhecimento automático de gestos tem sido o foco de pesquisas atuais em diversas áreas como visão computacional, processamento de imagens, reconhecimento de padrões, jogos computacionais, etc. O presente artigo propõe um método de reconhecimento de gestos aplicado ao jogo *rock-paper-scissors-lizard-spock*. A primeira etapa consiste na segmentação da mão utilizando o algoritmo *k-means*. Em seguida são extraídas algumas características da imagem que, então, são utilizadas na classificação dos gestos com o auxílio de uma rede neural artificial. Os resultados experimentais mostraram que o classificador proposto foi capaz de identificar corretamente os gestos do conjunto de teste em 93% dos casos.

Keywords: Jogo, K-means, Redes Neurais Artificia, Reconhecimento de Gestos, Rock-Paper-Scissors-Lizard-Spock.

Author's Contact:

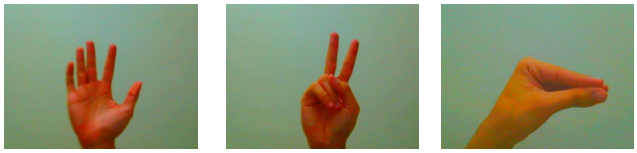
vitorc@aluno.unb.br
ha2385@columbia.edu
{zaghetto, bruno}@image.unb.br
lamar@cic.unb.br

1 Introdução

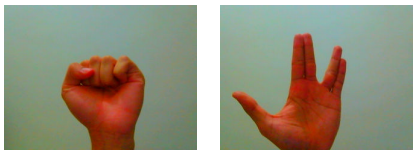
Dentre as diversas formas possíveis de interação homem-computador, como mouse, teclado, uso de *joysticks* ou comandos de voz, a comunicação através de gestos manuais é considerada uma das mais intuitivas [Panwar 2012]. Um indício disso é o fato de a maioria das pessoas utilizar esse tipo de recurso como acompanhamento da fala. Sendo assim, uma área que merece atenção dentro do campo de visão computacional [Szeliski 2010] é o reconhecimento automático gestos. Muitos outros trabalhos abordam o problema proposto [Suarez and Murphy 2012], [Rautaray and Agrawal 2011], [Min et al. 1997], [Ishikawa and Matsumura 1999], [LAMAR 1998]. No entanto a busca por um método universal exigiu ao longo do tempo um aumento da complexidade dos algoritmos propostos. Por isso, o presente trabalho busca desenvolver um método de baixa complexidade que resolve de forma robusta o problema do reconhecimento de gestos aplicado ao caso específico do jogo *rock-paper-scissors-lizard-spock*. Nesse jogo, o usuário deve escolher um dos cinco gestos possíveis e executá-lo na frente de uma *webcam*. Em seguida, o computador escolhe um gesto aleatoriamente e o resultado do jogo é mostrado na tela, segundo as regras abaixo:

- ***Rock*:** vence *scissors* e *lizard*, mas perde para *Spock* e *paper*;
- ***Scissors*:** vence *paper* e *lizard*, mas perde para *Spock* e *rock*;
- ***Paper*:** vence *rock* e *Spock*, mas perde para *lizard* e *scissors*;
- ***Lizard*:** vence *paper* e *Spock*, mas perde para *scissors* e *rock*;
- ***Spock*:** vence *rock* e *scissors*, mas perde para *paper* e *lizard*.

A Figura 1 ilustra os cinco gestos possíveis. Para resolver o problema proposto, foi necessário utilizar algoritmos e técnicas de processamento de imagens [Gonzalez and Woods 2006] e aprendizagem de máquina, esta última uma área da Inteligência Artificial. Essa área preocupa-se com as tarefas relacionadas à identificação e reconhecimento de padrões dentro de um conjuntos de dados, que, por sua vez, podem vir dos mais diversos contextos, como por exemplo: *(i)* previsão de preços de imóveis a partir do tamanho, número de quartos, área ocupada e número de banheiros; *(ii)* reconhecimento de voz baseado em componentes de frequência; *(iii)* reconhecimento de assinatura baseado em *templates* e *(iv)* automatização da condução de carros seguindo exemplos fornecidos por condutores humanos.



(a) *Paper*. (b) *Scissors*. (c) *Lizard*.



(d) *Rock*. (e) *Spock*.

Figura 1: *Gestos permitidos pelo jogo.*

Os processos de aprendizagem de máquina podem ser classificados como supervisionados ou não-supervisionados. No primeiro caso sabe-se de antemão a classe a que cada vetor de entrada pertence. No segundo caso, não se tem acesso a essa informação durante o treinamento do classificador. No presente trabalho foram empregadas técnicas de ambos os tipos, o algoritmo *k-means* [Duda et al. 2000], não-supervisionado, usado para segmentação da mão; e redes neurais artificias [Haykin 1998], [Kasabov 1996] do tipo *feed-forward* de aprendizado supervisionado, aqui empregada na classificação os gestos. Na seção a seguir será detalhado método proposto.

2 Método proposto

O algoritmo proposto nesse trabalho pode ser resumido nos seguintes passos:

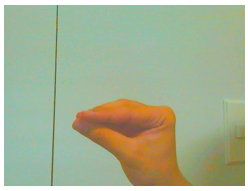
1. Captura da imagem;
2. Pré-processamento e segmentação por *k-means*;
3. Extração de características; e
4. Classificação por rede neural artificial.

2.1 Captura da Imagem

A imagem é capturada utilizando uma *webcam*. A captura é configurada de tal modo que a saturação seja a maior possível de forma a auxiliar no destaque da mão em relação ao fundo. As limitações impostas para o reconhecimento correto são: o fundo deve ser uniformemente branco e a mão deve estar na posição vertical. Trabalhos futuros podem ter como objetivo retirar tais limitações, atualmente o foco é o reconhecimento dos gestos para a interação homem-máquina.

2.2 Pré-processamento e segmentação

Após a captura, a imagem é convertida para o espaço de cores HSV e são tomados os componentes Matiz (*Hue*) e Saturação (*Saturation*) de cada *pixel*, sendo desprezado o outro componente (*Value*). A segmentação é feita utilizando o algoritmo *k-means* com 2 *clusters*. Esse algoritmo identifica os centroides de cada *cluster*, efetivamente particionando o conjunto de dados e classificando cada dado de acordo com o centroide mais próximo. Espera-se que os *pixels* contendo as cores da mão fiquem em um *cluster* enquanto os pixels de fundo fiquem no outro. Para que o algoritmo funcione,



(a) Gesto *Lizard* após captura pela *webcam*.

(b) Após pré-processamento e segmentação por *k-means*.

(c) Após remoção de ruídos.

Figura 2: *Processo de captura, pré-processamento e segmentação de um gesto.*

devem ser fornecidos valores iniciais para os centroides. Esses valores são normalmente aleatórios, mas, no caso deste experimento, um valor foi atribuídos tomado os componentes H e S típicos da cor da pele e o outro foi tomado como os componentes típicos da cor branca. Ambos os vetores foram determinados através da inspeção desses valores em fotos reais.

Apesar disso, ainda é possível ter objetos do fundo classificados como parte da mão. A remoção desses ruídos após a segmentação é feito utilizando o algoritmo de remoção de componentes conectados. Esperando-se que a mão seja o maior componente da imagem, o tamanho máximo que um componente deve ter para ser removido é calculado como metade dos *pixels* brancos da imagem. Também é feita uma remoção de buracos, mas apenas dos menores, já que se supõe que a presença de um buraco grande pode servir como indício de que o gesto é um dos desejados, como o *lizard*, por exemplo. O limiar tomado nesse caso corresponde a 1% do total de pixels brancos da imagem resultante da aplicação da remoção de componentes brancos. Um exemplo deste processo pode ser visto na Figura 2.

2.3 Extração de características

Diferentes algoritmos foram propostos com o objetivo de se calcular as principais características necessárias à classificação adequada dos gestos em questão. As características são:

- Razão entre altura e largura do retângulo de interesse;
- Número de dedos levantados (sem contar o polegar);
- Razão entre área da mão e área do retângulo;
- Razão entre perímetro da borda e perímetro da circunferência interna; e
- Presença ou não de buraco.

A seguir vamos detalhar cada um dos algoritmos propostos.

2.3.1 Definição do retângulo de interesse

O *retângulo de interesse* é aqui definido como o menor retângulo que contém a mão. Para se determinar tal retângulo, os seguintes passos são realizados:

1. Definição do *pixel* esquerdo p_e: varrer os *pixels* da imagem de cima para baixo, da esquerda para a direita até encontrar o primeiro *pixel* branco;
2. Definição do *pixel* superior p_s: varrer da esquerda para a direita, de cima para baixo;
3. Definição do *pixel* direito p_d: varrer de cima para baixo, da direita para a esquerda.

Se a imagem possui $M \times N$ *pixels*, as coordenadas do canto superior esquerdo do retângulo que envolve a mão são $(p_s.linha, p_e.coluna)$. Como consequência, a largura do retângulo é dada por $p_d.coluna - p_e.coluna$. Assume-se que o gesto esteja na posição vertical e com isso a parte inferior do retângulo coincide com a parte inferior da imagem. Sendo assim, a altura é dada por $M - p_s.linha$.

2.3.2 Presença do polegar

Para se determinar se há ou não polegar, dois sub-retângulos são definidos a partir do retângulo obtido por meio do processamento descrito na etapa anterior. Ambos tem a mesma altura, porém um é posicionada na parte esquerda do retângulo principal e o outro na parte direita. A razão disso é para se verificar a presença de polegares em cada lado da mão. Supõe-se de que, se o polegar estiver estendido, alguma dessas caixas conterá menos de 7% da área da mão. A largura da caixa é determinada empiricamente e seu valor é de 30 *pixels*.

2.3.3 Razão entre altura e largura do retângulo de interesse

Gestos como *scissors* tendem a ser envolvidos por caixas com maior altura e menor largura. Gestos do tipo *rock* serão normalmente envolvidos por caixas aproximadamente quadradas. Gestos como *paper*, por sua vez, serão envolvidos por caixas mais largas. Como esse efeito contribui na diferenciação entre os gestos, optou-se por incluir uma característica que corresponde à razão entre a altura e a largura do retângulo que envolve a mão.

2.3.4 Número de dedos levantados

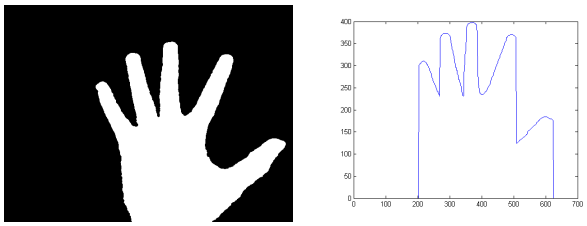
Uma das *características* mais essenciais para o reconhecimento do gesto manual é a classificação do número de dedos levantados. No presente trabalho, esta tarefa foi desenvolvida mediante os seguintes passos:

1. Detecção de mão fechada;
2. Detecção do contorno da mão;
3. Suavização do contorno;
4. Detecção de máximos no contorno; e
5. Remoção do máximo relativo ao polegar.

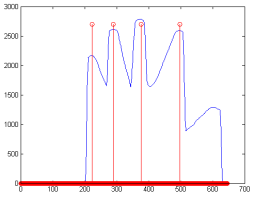
Primeiro, é necessário saber se há dedos a serem detectados ou não, ou seja, saber se a mão está fechada ou não. A verificação é feita fazendo-se uma varredura linha a linha da imagem. As linhas em que os *pixels* passarem de zero para um e retornarem a zero e permanecerem ali são consideradas linhas com *pulsos únicos*. Também são possíveis linhas com múltiplas transições de zero para um e retornos ao zero, nesse caso, também acontecem pulsos, mas não são pulsos simples. Nesse caso, serão chamados de *pulsos múltiplos*. É calculada, então, a razão entre o números de pulsos únicos e o número total de pulsos. Se a mão estiver fechada, a maioria dos pulsos será simples e a razão tende a 1. Se a mão estiver aberta, esse valor é tende a ser bem menor. Sendo assim, após de alguns experimentos, determinou-se um limiar de 0.85% para a tomada de decisão. É importante ressalter que antes de se computar a razão proposta, deve-se ser realizada uma remoção de todos os buracos internos da mão, uma vez que tais buracos podem gerar falsos pulsos. Essa remoção é feita por meio da eliminação de componentes pretos menores que 50% do tamanho total da imagem. O efeito dessa operação é a remoção temporária dos buracos que não foram removidos após a segmentação. A remoção é dita temporária, pois a presença ou não de buracos será mais tarde utilizada como uma *caraterísitca*. Se após essas operações a mão tiver sido classificada como aberta, o cálculo da quantidade de dedos levantados prossegue. Senão, a extração de características passa para a próxima etapa.

Como passo seguinte, determina-se o contorno máximo da mão, varrendo-se todas as colunas da imagem, de cima pra baixo, e armazenando a altura do primeiro *pixel* branco que aparece. O resultado disso pode ser visto na Figura 3(b). Realiza-se, então, a suavização do sinal resultante a partir de um filtro de média com 7 coeficientes. Aplica-se um algoritmo para se achar máximo locais. No contexto do algoritmo proposto, um efeito indesejado é o polegar ser detectado como um máximo. Para evitar esse artefato, todos os máximo que tem menos de 70% da altura do máximo de maior valor pico são removidos. O resultado final pode ser visto na Figura 3(c). Com isso, tem-se o número de dedos levantados.

A imagem sem buracos gerada nesta etapa será usada posteriormente na extrações de outras *características*. A partir daqui será referida como *imagem binária sem buracos*.

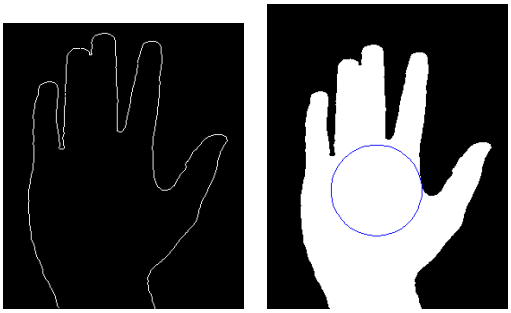


(a) Gesto *Paper* após segmentação. (b) Contorno calculado do gesto.



(c) Detecção de máximo.

Figura 3: *Processo de detecção dos picos que indicam dedos.*



(a) Contorno de *Paper*. (b) Circunferência de *Rock*.

Figura 4: *Contornos e circunferências internas*

2.3.5 Razão entre a área da mão e a área do retângulo de interesse

A porcentagem do retângulo que corresponde à mão também ajuda a diferenciar os gestos, já que, para *rock*, a porcentagem tende a (100%), enquanto, para gestos que envolvem maior abertura da mão, ou mesmo o próprio *lizard*, a porcentagem será bem menor. A área da mão é obtida simplesmente como o somatório dos *pixels* brancos da imagem sem buracos gerada anteriormente, enquanto a área do retângulo corresponde a sua largura vezes a sua altura.

2.3.6 Razão entre o perímetro da mão e o perímetro da circunferência interna

Como o perímetro difere bastante de um gesto para outro, ele também foi usado como *característica*. Deve-se atentar, porém, para as diferenças de tamanho entre as mãos. Por isso, essa *característica* deve corresponder, na verdade, à razão entre o perímetro de mão e o perímetro da circunferência interna à mão. A circunferência foi obtida aplicando o filtro de detecção de contornos *LoG* (laplaciano da gaussiana) e definindo como raio a distância entre o ponto mais próximo desse contorno e o centro de massa da mão. A Figura 4 ilustra esse processo para o gesto *paper*.

2.3.7 Presença ou não de buraco

A presença de um buraco interno na imagem, caso a segmentação tenha sido realizada de forma adequada, é indício de que o gesto é um *lizard*. Assim, essa característica também foi levada em consideração. Para isso, é feita a subtração entre a imagem binária sem buracos e a imagem binária original e, em seguida, tomado-se o módulo da matriz resultante. Os *pixels* binários são somados. Se o resultado for zero, a imagem não possui buraco, caso contrário, possui.

2.4 Classificação dos Gestos

As características enumeradas na seção anterior são calculadas para cada imagem de um conjunto de treinamento com 275 imagens divididas igualmente entre cada gesto. Essas características são utilizadas como entradas para a rede neural. Como são 6 características e 275 exemplos, a rede recebe uma matriz de tamanho 275×6 como entrada para seu treinamento. Na operação, porém, apresenta-se apenas uma amostra ao classificador já treinado.

Foram utilizadas três camadas: a camada de entrada, com 6 elementos que não realizam computação, pois correspondendo às seis características da mão; a camada intermediária, composta por 13 neurônios com função ativação log-sigmoide; e a camada de saída, com cinco neurônios e funções de ativação também log-sigmoide. Cada neurônios da camada de saída é responsável pode identificar um dos gestos. Assim, idealmente a saída do primeiro neurônio deve ser 1, caso o gesto identifica seja o gesto 1, e os demais neurônios devem ter como saída o valor 0. O segundo neurônio deve responder como 1, caso o gesto identifica seja o gesto 2, e os demais neurônios devem responder como 0. E assim por diante. Desta forma são construídos os alvos que identificam cada classe no processo de treinamento. Na prática, porém, os neurônios de saída não respondem necessariamente com zeros e uns, mas qualquer valor entre esses dois extremos. Logo, considera-se que o neurônio que responder com o maior valor é o que identifica o gesto.

A Figura 5 ilustra a arquitetura da rede proposta.

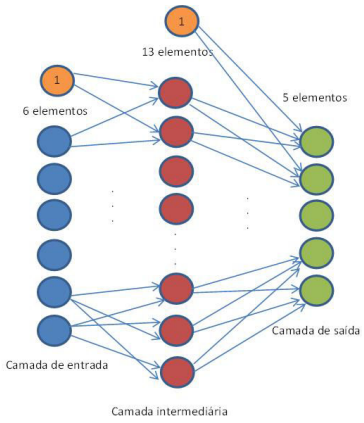


Figura 5: *Representação da rede neural utilizada*

3 Resultados Experimentais

Nos testes dois voluntários ficaram responsáveis por realizar 20 amostras de cada gesto, divididos igualmente entre mão esquerda e direita. Isso resultou em um total de 100 amostras. O gráfico que detalha os acertos e erros pode ser visto na Figura 6. A taxa total de acerto é de 93%, sendo que os gestos com menor e menor índices de acerto são o *paper* (85%) e o *spock* (100%), respectivamente.

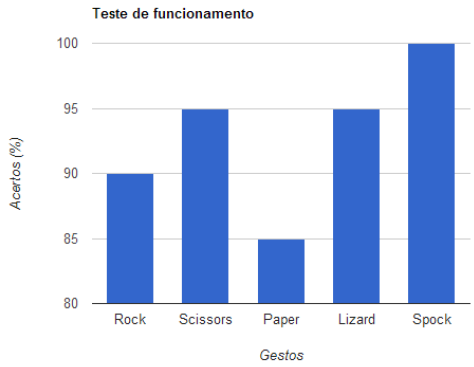


Figura 6: *Gráfico mostrando a porcentagem de acertos de cada gesto para o conjunto de teste. A taxa total de acertos é de* 93%.

A Figura 7 mostra a tela do jogo em operação. Também é possível ver uma gravação do jogo em operação no endereço indicado na nota de rodapé.[1]

[1] https://www.youtube.com/watch?v=EzIH5rb2Q1Q

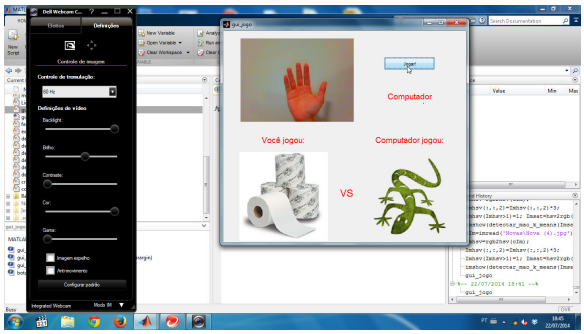


Figura 7: *Tela do sistema em operação.*

4 Conclusão

Foi proposto um classificador automático de gestos aplicado ao jogo *rock-paper-scissors-lizard-spock*. O processo de classificação contou com um segmentador de cor de pele, implementado por meio do algoritmos *k-means* e destinado à localização do objeto de interesse, a mão do jogador. Foi modelada uma rede neural artificial, que a partir de características extraídas da imagem da mão é capaz de identificar um dos gestos necessários à operação do jogo. Além do classificador em si, foi implementada um interface por meio da qual o usuário pode jogar com o computador. Os resultados experimentais mostraram que o método proposto apresenta um desempenho bastante promissor, alcançando um índice de 93% de acerto para o conjunto de teste. Pretende-se pra trabalhos futuros investigar um conjunto de características que seja capaz de elevar o índice de acerto do classificador e remover as limitações do jogo, como a necessidade do fundo branco.

Referências

DUDA, R. O., HART, P. E., AND STORK, D. G. 2000. *Pattern Classification (2Nd Edition)*. Wiley-Interscience.

GONZALEZ, R. C., AND WOODS, R. E. 2006. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

HAYKIN, S. 1998. *Neural Networks: A Comprehensive Foundation*, 2nd ed. Prentice Hall PTR, Upper Saddle River, NJ, USA.

ISHIKAWA, M., AND MATSUMURA, H. 1999. Recognition of a hand-gesture based on self-organization using a dataglove. In *Neural Information Processing, 1999. Proceedings. ICONIP '99. 6th International Conference on*, vol. 2, 739–745 vol.2.

KASABOV, N. K. 1996. *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, 1st ed. MIT Press, Cambridge, MA, USA.

LAMAR, MARCUS VINICIUS ; BHUIYAN, M. S. . I. A. 1998. From hand sign to japanese hiragana alphabet recognition using principal component analysis and neural networks. In *Proceedings of 12th Conference of the Japan Biomedical Society*, 162–165.

MIN, B.-W., YOON, H.-S., SOH, J., YANG, Y.-M., AND EJIMA, T. 1997. Hand gesture recognition using hidden markov models. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, vol. 5, 4232–4235 vol.5.

PANWAR, M. 2012. Hand gesture recognition based on shape parameters. In *Computing, Communication and Applications (ICCCA), 2012 International Conference on*, IEEE, 1–6.

RAUTARAY, S., AND AGRAWAL, A. 2011. Interaction with virtual game through hand gesture recognition. In *Multimedia, Signal Processing and Communication Technologies (IMPACT), 2011 International Conference on*, 244–247.

SUAREZ, J., AND MURPHY, R. 2012. Hand gesture recognition with depth images: A review. In *RO-MAN, 2012 IEEE*, 411–417.

SZELISKI, R. 2010. *Computer Vision: Algorithms and Applications*, 1st ed. Springer-Verlag New York, Inc., New York, NY, USA.