# Reconhecimento de Gestos

1st Raphael Ramos
*Ciência da Computação*
*Universidade de Brasília*
Brasília, Brasil
raphael.soares.1996@gmail.com

2nd André Luis Souto
*Engenharia de Computação*
*Universidade de Brasília*
Brasília, Brasil
andresouto.as@gmail.com

3th Rafaela Sinhoroto
*Engenharia Mecatrônica*
*Universidade de Brasília*
Brasília, Brasil
rsinhoroto@hotmail.com

***Resumo*—Gestos são parte da comunicação não verbal diária, dessa forma, provendo um jeito inovador e natural de interação com o computador. Logo, reconhecimento de gestos tem aplicações largas na área de computação, como interação humano-computador, linguagem de sinais e jogos. Neste trabalho, duas abordagens de reconhecimentos de gestos, uma baseada em redes neurais convolucionais e outra baseada em parâmetros do formato da mão, são testadas com o *Marcel dataset* e têm sua acurácia comparadas.**

***Index Terms*—Processamento de Imagem, Redes Neurais Convolucionais, Reconhecimento de Gestos, Parâmetros de Forma.**

## I. Introdução

Neste trabalho foram comparados dois modelos diferentes para realizar reconhecimento de gestos: redes neurais convolucionais (CNN) e reconhecimento baseado em parâmetros de forma [9]. Para fazer o treinamento e o teste dos nossos modelos, nós usamos o *Marcel dataset* [7] que consiste de 6 sinais de mão (A, B, C, FIVE, POINT, V) executados por 24 pessoas em três tipos diferentes de *backgrounds*. Pessoas e *backgrounds* diferentes foram usados para aumentar a diversidade e a informação contida no *dataset*. Em termos de *background*, as imagens no *Marcel dataset* foram capturadas em um *background* uniforme com luz, um uniforme escuro e um *background* complexo variado. Devido ao número diferente de pessoas incluídas na criação desse dataset há também variação no formato e tamanho da mão, além de rotações e mudanças na forma que os sinais são executados. Este dataset contém 4872 imagens de treino e 658 de teste.

Por fim, a acurácia dos dois métodos utilizados é comparada a fim de determinar o mais eficiente para o dataset utilizado, considerando fundo da imagem e gestos reconhecidos.

## II. Trabalhos Relacionados

Redes neurais profundas são a base dos resultados do estado da arte para reconhecimento de imagens [10], detecção de objetos [4], reconstrução tridimensional de objetos [2], reconhecimento de faces [16], reconhecimento de discurso [5], *machine translation* [13], geração de legendas de imagens [17], tecnologia de carros autônomos [6], entre outros. Entretanto, treinar uma rede neural profunda é uma problema de otimização global difícil. Por isso, para o presente trabalho foi utilizado o método de *machine learning* conhecido como *transfer learning* [8]. *Transfer learning* é um metodo onde um modelo desenvolvido para uma tarefa é reusado como ponto de partida para um modelo em outra tarefa. Esse método foi utilizado neste projeto para a inicialização dos pesos visto que ele permite progresso rápido e performance melhorada para modelar a tarefa requerida.

O outro método utilizado para comparação, reconhecimento baseado em parâmetros de forma, utiliza como referência o trabalho [9] com algumas correções e adaptções para o *dataset* utilizado. Para reconhecimento de gestos, são utilizados apenas parâmetros baseados no formato da mão, como centro de massa, orientação, e posicionamento dos dedos. Logo, cor da pele e textura não são considerados. No trabalho anterior, são analisadas apenas gestos em fundo branco e sem a presença de partes da roupa cobrindo o pulso, o que facilita o processo de segmentação da mão e detecção da orientação do gesto. Nesta abordagem, foram feitas alterações para permitir a segmentação da mão em fundos variados, e com o pulso coberto.

## III. Soluções Propostas

### A. Redes Neurais Convolucionais

Temos como vantagem do uso de aprendizado profundo a desnecessidade de engenharia de características – a própria rede o faz. Como contrapartida, necessita-se de uma grande quantidade de exemplos de treinamento. Neste trabalho isso foi mitigado pelo uso de *data augmentation*, e transferência de aprendizado ao usar pesos pré-treinados para uma maior e mais desafiadora tarefa de classificação de imagens em 1000 classes: a ImageNet [3]. Foram avaliados dois modelos de redes neurais neste trabalho: *Xception* [1] e *Inception ResNet V2* [14].

A *Inception ResNet V2* [14] é uma rede neural convolucional que faz convoluções fatorizadas e regularizações. Devido a imensa variação na localização da informação em uma imagem, escolher o tamanho do kernel ideal para realizar operações de convolução, que destacam partes salientes da imagem, é uma tarefa difícil. Um tamanho de kernel maior é preferível para a informação que é distribuída mais globalmente, enquanto um kernel de tamanho menor é preferível para informação que é distribuída mais localmente. Para resolver esse problema, as redes da linha *Inception* introduziram os módulos *Inception*, que realizam convoluções em uma entrada com vários tamanhos de filtros além de realizar *max pooling* para obter *features* contidas em subregiões da entrada. Para a

versão residual das redes *Inception* (*Inception ResNet V2*) foi utilizado blocos *Inception* mais baratos que o modelo original

Na *Xception* houve uma modificação na camada *Depthwise Separable Convolution*. Nesta camada existe uma *pointwise convolution* seguida por uma *depthwise convolution* (ordem contrária no modelo *InceptionV3* [15]). *Depthwise convolution* é a convolução espacial canal a canal. A convolução separável em profundidade (*Depthwise convolution*) baseia-se em fatorizar a operação de convolução em duas camadas: uma convolução em profundidade que aplica um filtro para cada canal da entrada; e uma convolução por pontos (pointwise), de tamanho 1x1, responsável por criar novas características por combinações lineares dos canais de entrada e alteram a dimensão. Como resultado, são mais baratas computacionalmente sem perdas de performance significativas em relação às convoluções completas.

### *B. Reconhecimento Baseado em Parâmetros de Forma*

O algoritmo de reconhecimento baseado em parâmetros de forma segue os seguintes passos:

1) Segmentação da figura da mão utilizando *k-means clustering*;
2) Detecção de orientação da mão como horizontal ou vertical;
3) Extração de features relevantes para a classificação (centroide da mão, detecção do polegar, etc.);
4) Classificação a partir das informações obtidas nas etapas 1 a 3.

*1) Segmentação:* A segmentação da mão é realizada utilizando *k-means clustering* para agrupar os pixeis da mão em clusters separados dos pixeis do plano de fundo utilizando a distância euclidiana entre as diferentes cores de cada agrupamento.

Para que o agrupamento funcione, primeiro é necessário transformar a informação de cor das imagens do espaço RGB para o espaço de cor L*a*b*, um espaço de cor baseado em um canal de luminosidade e dois canais cromáticos que permite que a diferença entre duas cores seja dada pela distância euclidiana entre elas. Os dois canais de cor, a* e b*, são baseados na teoria de cores opostas, onde duas cores não podem ser verdes e vermelhas ao mesmo tempo, nem amarelas e azuis ao mesmo tempo.

Após essa transformação, é necessário estabelecer a quantidade de centroides para o *k-means* classificar. Em imagens com condições uniformes (contendo apenas a mão realizando o gesto e um fundo plano, sem detalhes e de cor distinta a de pele), 2 centroides são suficientes para uma boa segmentação. Porém, em casos gerais, mais complexos, são utilizados 5 centroides para que o agrupamento não junte pedaços da mão com informação irrelevante de fundo.

A técnica do *k-means* funcina melhor com imagens maiores em que há maior distinção dos pixeis da mão para os de fundo.

*2) Orientação:* Uma vez segmentada a mão, é necessário determinar sua orientação, vertical ou horizontal. A detecção correta é importante pois a orientação influencia diretamente no método de identificação dos dedos.

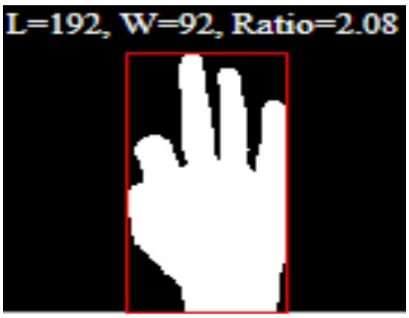


Figura 1. Orientação vertical.

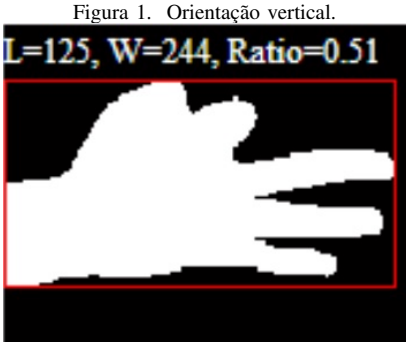


Figura 2. Orientação horizontal.

Dada a mão segmentada de largura L e comprimento C, calcula-se a razão R entre tais medidas:

$$R = \frac{C}{L} \tag{1}$$

assim, tem-se as condições

$$Orientação = \begin{cases} Vertical, & R > 1 \\ Horizontal, & \text{Caso contrário} \end{cases} \tag{2}$$

Portanto, caso o comprimento da mão segmentada seja maior que sua largura, a mesma encontra-se na vertical. E caso a largura seja maior que o comprimento, a orientação é horizontal.

*3) Extração de features:* Nesta etapa, são extraídas informações a respeito do centro de massa da mão (centroide), da presença ou não do polegar e do estado dos dedos, se estão levantados ou abaixados. Tais informações são usadas juntas na classificação e interpretação dos gestos.

*a) Centro de Massa:* O centro de massa divide a imagem da mão em duas partes, uma parte contendo os dedos e outra não contendo. Tal divisão é feita no centro geométrico da imagem.

Para calcular o centroide, foi utilizado o momento da imagem, dado por

$$M_{ij} = \sum \sum x^i y^j I(x, y) \tag{3}$$

onde $M_{ij}$ é o momento da imagem e I(x,y) é a intensidade na coordenada (x,y). Em resumo, o momento da imagem é a

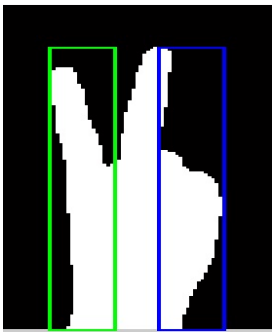

Figura 3. Retângulos formados para detecção de polegar.

média ponderada das intensidades dos *pixels* da imagem. Dado $M_{ij}$, foi possível calcular o centro de massa:

$$X_c, Y_c = \frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \tag{4}$$

onde $X_c$, $Y_c$ são as coordenadas do centroide e $M_{00}$ é a área da imagem binária.

*b) Detecção de polegar:* Nesta etapa, há a identificação da presença ou não do polegar no gesto da imagem. Dados os limiares das laterais, obtidos na segmentação da mão, pega-se 30 *pixels* para dentro da imagem partindo-se de cada uma das duas bordas laterais. Forma-se então dois retângulos de área 30 *pixels* x comprimento da mão.

Calcula-se então a porcentagem de *pixels* brancos dentro de cada retângulo em comparação com o total da imagem. Caso a porcentagem seja menor que 7% em um dos lados, o polegar está presente neste lado. Caso nos dois retângulos conste uma porcentagem maior que 7%, não há polegar no gesto. E caso nos dois lados haja menos de 7% de *pixels* brancos, também não há polegar no gesto.

*c) Identificação dos dedos:* Para a identificação do restante dos dedos são usados dois métodos. Primeiramente, percorre-se todo o contorno da mão segmentada a fim de marcar regiões de picos, pois tais regiões representam os dedos na imagem.

Com os picos detectados, no segundo método, calcula-se a distância euclidiana de todos os picos encontrados com o centro de massa. Dessa forma, tem-se todos os dedos identificados, porém, como alguns podem estar dobrados, todos os picos que tiverem distância euclidiana de valor menor que 75% do maior valor, são declarados insignificantes, ou seja, dobrados.

*4) Classificação:* Por fim, dadas as informações obtidas na etapa de extração de features, gera-se um vetor de cinco bits onde cada bit representa o estado de cada dedo. Ou seja, caso ele se encontre no gesto, o bit correspondente é setado como 1. Caso contrário, é setado como 0. Na figura 4, um exemplo de gesto com os cinco dedos levantados e abaixo o vetor de bits correspondente.

Figura 4. Gesto correspondente ao vetor de bits [1 1 1 1 1].

## IV. Resultados Experimentais

### A. Redes Neurais Convolucionais

Os dois modelos foram avaliados em 3 cenários diferentes, variando *dropout* [11], que previne sobreajuste nos dados de treino (*overfitting*), e *batch size*. A melhor acurácia obtida foi de 91.45%, conforme mostra a Tabela I. Notou-se rápida convergência em um baixo número de épocas, além de *overfitting*, considerando a acurácia de 99% alcançada na validação e no treino. Por isso, foi aumentado o dropout para a segunda *Xception* testada e a acurácia obtida foi melhor. O que já era esperado, considerando que o *dropout* dropa as unidades junto com suas conexões. Assim, o *dropout* não permite que as unidades se co-adaptem muito, prevenindo *overfitting* pois essas co-adaptações das unidades para diminuir a *loss* são complexas e podem não generalizar bem para dados não vistos.

Tabela I
Resultados obtidos usando CNN.

| | **Dropout** | **Acurácia (Top-1)** |
|---|---|---|
| **Xception-1.0** | 60% | 89.82% |
| **Xception-2.0** | 80% | 91.45% |
| **Inception ResNetV2** | 80% | 91.45% |

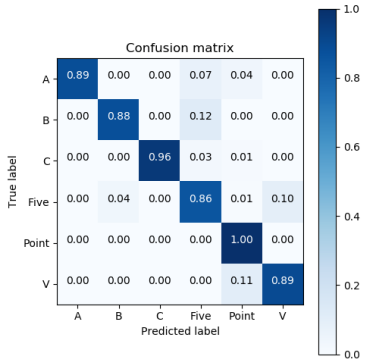


Figura 5. Matriz de confusão para o modelo Xception que obteve a maior acurácia.

### B. Reconhecimento Baseado em Parâmetros de Forma

O método foi avaliado utilizando as imagens do conjunto de teste para comparação com o desempenho do método de

Tabela II
RESULTADOS OBTIDOS USANDO SHAPE PARAMETERS.

| Gesto | Acurácia |
|---|---|
| A | 16.7% |
| B | 0.0% |
| C | 0.0% |
| Five | 10.4% |
| Point | 37.0% |
| V | 38.9% |
| **Total** | **16.9%** |

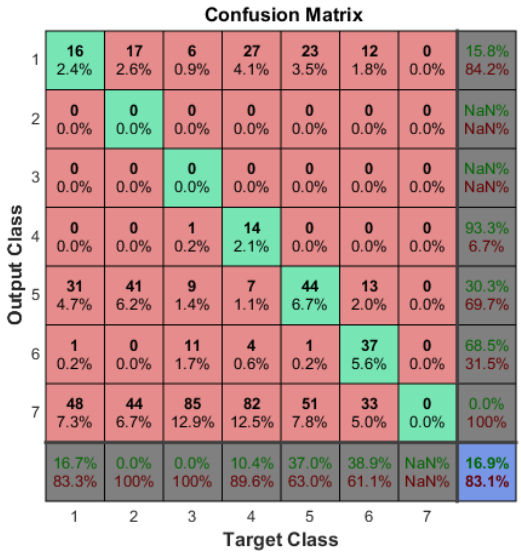


Figura 6. Matriz de confusão para o método utilizando *shape parameters*. As classes enumeradas de 1 a 6 são [A, B, C, FIVE, POINT, V], respectivamente, enquanto a classe 7 representa falhas de segmentação/classificação.

utilizando redes neurais convolucionais. A Tabela II resume os resultados obtidos utilizando o método descrito em [9].

Como mostrado na tabela, observa-se que as taxas de acerto para todos os gestos foram insatisfatórias, tendo gestos que não foram reconhecidos em nenhuma tentativa.

## V. CONCLUSÃO

Os resultados obtidos para o modelo utilizando Redes Neurais Convolucionais foram satisfatórios, visto que a melhor acurácia relatada por [12] foi de 78.22%. Enquanto que para o modelo que utiliza parâmetros de forma, a acurácia alcançada foi muito baixa. O fato do modelo ter sido desenvolvido para reconhecimento de gestos em fundo uniforme e com a mão reta e de frente para a câmera influenciou diretamente na segmentação da mão. Pois no dataset testado, poucas imagens apresentavam uniformidade no fundo juntamente com a correta posição da mão em relação a câmera. Dessa forma, a segmentação foi prejudicada, o que ocasionou em resultados baixos de acerto no reconhecimento dos gestos. Uma outra forma de segmentação, adequada a fundos heterogêneos, é o indicado para aumentar as taxas de acerto deste modelo.

Dessa forma, é correto concluir que o método utilizando Redes Neurais Convolucionais é mais adequado para ser utilizado em reconhecimento de gestos quando o ambiente das imagens não é um ambiente controlado.

## REFERÊNCIAS


[1] François Chollet. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint*, pages 1610–02357, 2017.

[2] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[4] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[5] Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning*, pages 1764–1772, 2014.

[6] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, et al. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*, 2015.

[7] Sébastien Marcel. Sebastien marcel static hand posture database. https://www.idiap.ch/resource/gestures/, 1999.

[8] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010.

[9] Meenakshi Panwar. Hand gesture recognition based on shape parameters. 2012.

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[12] Gjorgji Strezoski, Dario Stojanovski, Ivica Dimitrovski, and Gjorgji Madjarov. Hand gesture recognition using deep convolutional neural networks. In *International Conference on ICT Innovations*, pages 49–58. Springer, 2016.

[13] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[14] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[15] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[16] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[17] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.