# OPERAÇÃO DE UM MANIPULADOR POR MEIO DA DETECÇÃO DE GESTOS BASEADA EM APRENDIZADO DE MÁQUINA

José M. Oliveira-Jr*, Michel C. R. Leles†, Adriano S. Vale-Cardoso†,
Mário C. da Silva-Jr†, Leonardo A. Mozelli‡, Armando A. Neto‡

*Graduação em Engenharia Mecatrônica, UFSJ, Ouro Branco/MG, Brasil

†CELTA, UFSJ, Ouro Branco, MG, Brasil

‡Departamento de Engenharia Eletrônica, UFMG, Belo Horizonte, Brasil

Emails: jmarques.oliveirajunior@gmail.com, mleles@ufsj.edu.br, adrianosvc@ufsj.edu.br, mariocupertino@ufsj.edu.br, mozelli@cpdee.ufmg.br, aaneto@cpdee.ufmg.br

**Abstract**— A Computer Vision based system intended to operate a robotic manipulator is presented in this work. Images representing some pre-configured gestures are captured using a RGB-D sensor, and then applied to a processing stage, based on a Machine Learning technique, known as Support Vector Machines (SVMs). Each gesture indicates a different movement to be executed by the robotic manipulator. After the gesture is identified it is translated into robot commands which are sent through a standard serial interface. The results have showed that the human gestures were successfuly converted into pre-determined robot movements, which means the processing stages worked as expected.

**Keywords**— Machine Learning, Supporting Vector Machines, Manipulator, RGB-D sensors

**Resumo**— Neste trabalho é apresentado um sistema para operação de manipuladores robóticos baseado em técnicas de Visão Computacional. As imagens, representativas de gestos prestabelecidos, são capturadas por meio do sensor do tipo RGB-D e processadas por uma técnica de Aprendizado de Máquina, conhecida como Máquina de Vetores de Suporte (*Support Vector Machines* ou SVM). Uma vez classificados, cada gesto indica um movimento diferente a ser executado pelo manipulador robótico. Os comandos são enviados para esse dispositivo via comunicação serial. Os resultados obtidos demonstram que foi possível a captura, processamento, classificação e envio das informações corretas ao manipulador para execução dos movimentos predeterminados.

**Palavras-chave**— Aprendizado de Máquina, Máquinas de Vetores de Suporte, Manipulador Robótico, sensores RGB-D.

## 1 Introdução

A substituição de operadores humanos por robôs em tarefas arriscadas ou em ambientes hostis representa uma evolução na automatização de processos industriais e domésticos. Embora exista uma longa tradição do uso de robôs na execução de tarefas precisas e repetitivas, em ambientes rígidos e bem estruturados, ainda persistem grandes desafios para o trabalho em ambientes dinâmicos, envolvendo interação e cooperação com humanos (Heyer, 2010).

Os desafios impostos são inúmeros e as abordagens são fortemente interdisciplinares (Latombe, 2012). Entretanto, a complexidade envolvida em muitas atividades ainda requer a participação ativa de especialistas humanos. O objetivo é prover uma experiência de interação remota com o ambiente por meio de robôs da forma natural, em que o grau de autonomia do sistema robótico deve ser ajustado de acordo com as especificações da situação considerada (Niemeyer et al., 2008).

No presente trabalho, é proposto um sistema para operação de um manipulador robótico baseado em técnicas de Visão Computacional. Para tanto, realiza-se o reconhecimento de alguns gestos pré-estabelecidos, capturados por um sensor RGB-D, via técnicas de aprendizado de máquina. Em seguida, os gestos reconhecidos são traduzidos em comandos os quais são enviados ao manipulador robótico, e possibilitam a execução de movimentos padronizados. A Fig. 1 ilustra o diagrama do sistema desenvolvido nesse trabalho.


**Figura 1:** Diagrama ilustrando o arcabouço experimental.

O texto está organizado de acordo com as seguintes seções: Seção 2 trata da revisão bibliográfica; Seção 3 apresenta os conceitos básicos da plataforma desenvolvida; Seção 4 mostra a análise dos resultados; e Seção 5 trata da conclusão e de direções futuras.

## 2 Trabalho relacionados

A Visão Computacional é a área de conhecimento que estuda métodos para extração de informações de uma cena a partir de suas imagens digitais, permitindo entre outras coisas, a tomada de decisões. Dentre as suas principais funcionalidades, é possível destacar: aquisição, processamento, análise e reconhecimento de algum padrão em imagens. Tang et al. (2014) mostram como esse conceito pode ser aplicado no reconhecimento de faces, enquanto Weinland et al. (2011) fazem uma revisão sobre o reconhecimento de ações. Pisharady e Saerbeck (2015) fornecem uma revisão recente de métodos para reconhecimento de gestos baseado em processamento de imagens, enquanto que Ruffieux et al. (2015) discutem um método para segmentação temporal e identificação de gestos. Já Pillajo e Sierra (2013) apresentam uma Interface Homem-Máquina, baseada em sensores RGB-D, similar à proposta neste trabalho, que permite ao usuário operar um manipulador robótico do tipo SCARA, de 3 Graus de Liberdade (GDLs). Em nosso trabalho, utilizamos um manipulador de 5 GDLs. Além disso, é adotado um procedimento de aprendizado para reconhecimento dos gestos, ao passo que em Pillajo e Sierra (2013) mede-se apenas a posição espacial por meio de metadados fornecidos pelo sensor.

Técnicas de Aprendizado de Máquina são comumente utilizadas no processamento das imagens, possibilitando a construção de sistemas capazes de adquirir conhecimento de forma automática (Monard e Baranauskas, 2003). Nölker e Ritter (1999) utilizaram Redes Neurais Artificiais para reconhecimento de gestos. Beluco (2002) aplicou uma rede Multi-Layer Perceptron para classificação de imagens de sensoriamento remoto. Shiba et al. (2005) avaliaram o desempenho de árvores de decisão para rotulação de imagens de satélites. Krizhevsky et al. (2012) apresentaram um significativo trabalho de classificação de imagens utilizando Convolutional Neural Networks. O uso das Máquinas de Vetores de Suporte (*Support Vector Machines*, ou SVMs) é recorrente na detecção, segmentação e classificação de objetos em imagens (Heisele et al., 2001; Malisiewicz et al., 2011).

As SVMs se fundamentam na teoria do aprendizado estatístico, desenvolvida por Vapnik (2013). Em uma tarefa de segregação, o objetivo da SVM é encontrar um hiperplano que separe as classes pela maior margem possível (Mitchell et al., 1997). Em problemas de dados de grande dimensão, SVMs tendem a ser menos susceptíveis a problemas de ajustes de parâmetros, uma vez que requerem menos pesos para configuração (Karystinos e Pados, 2000; Liu e Castagna, 1999). Outra característica atrativa é o uso de funções Kernel nas funções não lineares das SVMs, tornando o algoritmo mais eficiente, uma vez que permite a construção de simples hiperplanos em um espaço de alta dimensão computacionalmente tratável (Burges, 1998).

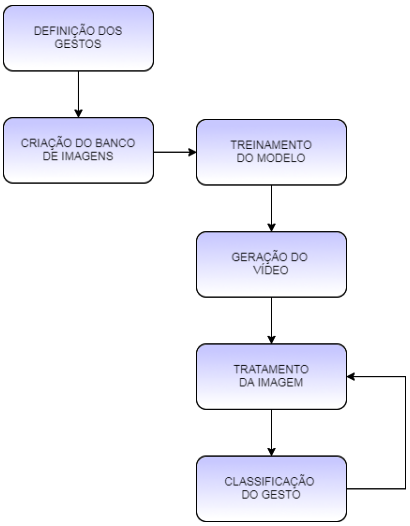


**Figura 2:** Fluxograma do algoritmo para reconhecimento de gestos feitos pelo operador.

No tipo de aprendizado supervisionado utilizado neste trabalho, dado um conjunto de exemplos rotulados na forma $(x_i, y_i)$, em que $x_i$ representa um exemplo e $y_i$ denota o seu rótulo, deve-se produzir um classificador (também denominado modelo, preditor ou hipótese) capaz de predizer precisamente o rótulo de novos dados (Lorena e de Carvalho, 2007). Essa parte do processo é denominada treinamento. Após a criação do modelo, deve-se verificar seu desempenho, classificando amostras não utilizadas no treinamento e comparando tais resultados com os rótulos verdadeiros. Esta etapa denomina-se validação. As SVMs apresentam boa capacidade generalização, i.e., conseguem prever corretamente a classe de dados não apresentados, anteriormente.

## 3 Metodologia

O ciclo do algoritmo se inicia com o treinamento do modelo Máquina de Vetores de Suporte (*Support Vector Machine*, ou SVM), prosseguindo para a exibição do vídeo, obtenção e tratamento da imagem do gesto e, finalmente, sua classificação. Os dois últimos passos são repetidos durante a exibição do vídeo, conforme fluxograma mostrado na Fig. 2.

### *3.1 Definição dos gestos*

O passo inicial consiste na definição dos gestos e da resposta pretendida. Foram definidos três tipos de movimentos a serem reconhecidos pelo sistema: i) abertura e fechamento da mão; ii) contagem do número de dedos na mão; e iii) ângulo formado entre o braço e o antebraço. Dessa forma, três tipos básicos são obtidos: i) um booleano (mão aberta ou fechada); ii) um inteiro (número de dedos na mão); e iii) um ponto flutuante (ângulo

entre braço e antebraço). Os métodos para identificação de mão aberta ou fechada e para contagem do número de dedos são os mesmos (classificador SVM). Já para o cálculo do ângulo entre o braço e o antebraço foi utilizada outra estratégia, discutida posteriormente.

### *3.2 Criação do banco de imagens*

A primeira etapa consiste na criação de um conjunto de imagens rotuladas, isto é, associadas às classes previamente definidas, para treinamento e validação do modelo. Essa etapa pode ser dividida em duas partes: a obtenção das imagens, e a preparação dos dados para os passos seguintes. A obtenção do banco de imagens segue o algoritmo mostrado a seguir:

**Programa *Principal_gerar_imagens:*** Criação do banco de imagens

```
qte_imagens = n;   % quantidade de imagens desejadas
iniciar_kinect();
cont = 0;          % contador auxiliar de imagens
while(true)
    meta_dados = mostrar_video();
    if (pessoa_localizada)
        limites = gerar_roi(meta_dados);
        imagem = tratar_imagem(limites,meta_dados);
        salvar_imagem(imagem);
        cont = cont + 1;   % incrementando o contador
        if (cont >= qte_imagens)
            break          % interrompe o laço while
        end if
    end if
end while
```

A comunicação entre o sensor RGB-D e o MATLAB® oferece algumas bibliotecas para a programação[1]. Uma delas permite o fornecimento das posições das articulações da pessoa, conforme visto na Fig. 3. Cada par de dados em `jointIndices` (relativos aos pixels na imagem) e cada trio de dados em `jointCoordinates` (relativos à coordenadas globais) referem-se a posição de uma parte do corpo (marcados com uma cruz na imagem), como cabeça, ombros, cotovelos e mãos, dentre outros.

A função `gerar_roi(meta_dados)`, descrita a seguir, é responsável pela segmentação da Região de Interesse (*Region of Interest*, ou ROI), ou seja, cria uma sub-imagem contendo somente a região da mão. As informações das partes do corpo fornecidas pelo sensor RGB-D são usadas como referência para a função. É realizada uma compensação do tamanho da sub-imagem em função da distância da mão ao sensor, visando preservar a proporção e o tamanho da ROI, mesmo com diferentes posicionamentos da pessoa na cena.

Na segunda parte dessa etapa (preparação dos dados) para o treinamento e validação do modelo

[1]Disponível em: https://www.mathworks.com/help/imaq/examples/using-the-kinect-r-for-windows-r-from-image-acquisition-toolbox-tm.html. Acesso em Março de 2018.

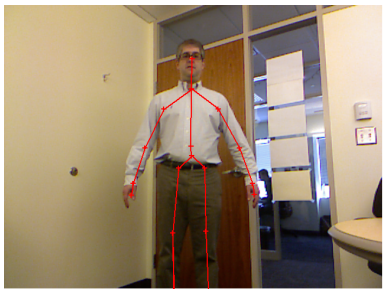


**Figura 3:** Dados fornecidos pelo sensor RGB-D: índices e coordenadas das articulações. Fonte: *Using the Kinect® for Windows® V1 from Image Acquisition Toolbox*$^{TM}$.

**Função gerar_roi (meta_dados):** Criação da sub-imagem contendo somente a mão

```
pixel_mao = meta_dados.JointImageIndices(12,:);   % posição da mão na imagem
coordenada_mao = meta_dados.JointWorldCoordinates(12,:); % distâncias da mão ao sensor

profundidade = calcular_profundidade(coordenada_mao);   % o raio calculado será
raio = calcular_raio(profundidade);                     % relacionado ao tamanho
                                                        % da sub-imagem criada
ponto1 = pixel_mao_x - raio;                            % delimitação da sub-imagem
ponto2 = pixel_mao_x + raio;
ponto3 = pixel_mao_y - raio;
ponto4 = pixel_mao_y + raio;
roi = [ponto1 ponto2; ponto3 ponto4];                   % retorno da função
```

criado a partir da SVM, o conjunto de imagens obtido deve ser transformado em matrizes (em que cada linha representa os *pixels* de uma imagem) e os rótulos transformados em vetores, em que cada elemento representa a classe a qual a imagem pertence. Antes da transformação em uma matriz, é realizado o redimensionamento de todas as imagens, padronizando o tamanho dos elementos gerados. Isso anula os efeitos de distorções causadas pela distância da pessoa ao sensor que não foram compensados pela função `gerar_roi()`. Assim, são criados a partir do banco de imagens dois grupos distintos: a matriz de treinamento e o vetor de rótulos de treinamento, usados na criação do modelo; e a matriz de teste e o vetor de rótulos de teste, utilizados para verificação do desempenho do modelo.

### *3.3 Treinamento do modelo*

A Fig. 4 apresenta os conceitos referentes à criação de um classificador de forma simplificada. Pode-se observar que os dados são manipulados da seguinte maneira: cada elemento (cada imagem) é equivalente a uma linha $x_i$; atrelado a essa linha há uma classe referência $y_i$, tida como verdadeira. A partir disso, o algoritmo traça um hiperplano separando tais classes, objetivando maximizar a margem entre as classes, conforme a seguir:

**Função svm_multiclasse (BancoTreinamento, RotulosTreinamento):**

```
numClasses = contar_classes(RotulosTreinamento);  % conta a quantidade de classes
                                                  % diferentes nos dados
for k = 1 : numClasses
    OneVersusAll  = (RotulosTreinamento==u(k));  % treinando k modelos svm
    modelos(k) = treinar_svm(BancoTreinamento,OneVersusAll);
end
```

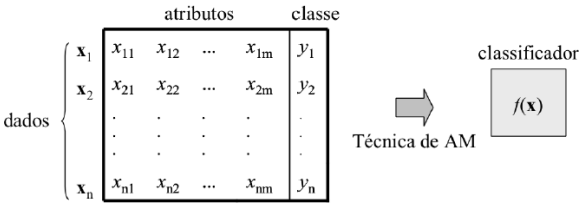


**Figura 4:** Criação de um classificador utilizando Aprendizado de Máquina Supervisionado.

Na classificação é utilizada a estratégia *One-Versus-All* (Bottou et al., 1994). São construídos $k$ modelos SVM, sendo um valor inteiro representando o número de classes. O $k$-ésimo modelo SVM é treinado com todos os exemplos da enésima classe com os rótulos positivos e todos os outros exemplos com rótulos negativos (Hsu e Lin, 2002). Finalmente, verifica-se se cada novo elemento pertence a cada uma das classes.

Logo após a criação do classificador, utiliza-se o banco de testes para verificar a precisão do modelo. O objetivo é calcular o desempenho do modelo em imagens não vistas no treinamento. A matriz de confusão é a forma de representação da qualidade obtida de uma classificação digital de imagem, sendo expressa por meio da correlação de informações dos dados de referência (compreendido como verdadeiro) com os dados classificados (Prina e Trentin, 2015). A precisão do modelo é feita com base na matriz de confusão, calculando-se a razão entre o somatório das amostras classificadas corretamente (somatório da diagonal principal) e incorretamente. Outras métricas de qualidade podem ser obtidas a partir de tal matriz, como sensibilidade, Medida F1, acurácia, especificidade, dentre outras (Matos et al., 2009).

### *3.4 Imagens: geração, tratamento e classificação*

Após o treinamento, o programa segue para a aquisição das imagens em tempo real. Para melhor organização e portabilidade do código foi criada uma função chamada `iniciar_kinect`. Nessa função são realizadas todas as configurações necessárias para obtenção dos dados e dos vídeos fornecidos pelo sensor RGB-D. Após esta inicialização, utiliza-se um laço de repetição que se repete a cada imagem obtida.

Um procedimento comum no processamento de imagens é a binarização. Aqui, procede-se da seguinte forma: i) obtêm-se a ROI (proximidades da mão) com a função `gerar_roi()`; ii) normaliza-se a imagem em escala de cinza, visando minimizar possíveis interferências do plano de fundo; e iii) utiliza-se como limiar para binarização o valor médio da imagem em escala de cinza normalizada. Dessa forma, pode-se segmentar e binarizar a imagem da mão.

A partir do vídeo obtém-se a imagem a ser classificada. Essa classificação é executada com os objetos já criados, conforme descrito na subseção 3.3. A imagem tratada (após passar pelo processo de segmentação, normalização e binarização) é transformada em um vetor, representando o elemento a ser classificado. O resultado indica a qual classe a imagem obtida pertence.

### *3.5 Análise utilizando PCA*

A Análise das Componentes Principais (*Principal Component Analysis*, ou PCA) é uma técnica cujo objetivo é extrair características importantes e representá-las em um novo conjunto de variáveis ortogonais, chamadas componentes principais (Wold et al., 1987; Abdi e Williams, 2010). Dessa forma, é possível representar os dados obtidos utilizando um número menor de dimensões. Neste trabalho, vetores de dimensão 4.900 são utilizados (oriundos de imagens de 70x70 *pixels*). A visualização e o tratamento de cada elemento torna-se simplificado aplicando a técnica do PCA. Tal estratégia também foi aplicada na verificação da eficiência do modelo.

### *3.6 Ângulo do braço*

Já para o algoritmo que calcula o ângulo do braço, são utilizados os dados das articulações fornecidos pelo sensor, também chamados de metadados, da mesma forma que exemplificado na Fig. 3.

Dada as posições no espaço das articulações (ombro, cotovelo e pulso), são calculados dois vetores, correspondendo ao braço, $u$, e antebraço, $v$. Esses vetores correspondem, respectivamente, às distâncias entre ombro e cotovelo e entre cotovelo e pulso. O ângulo entre os vetores é dado por:

$$\nu = \arctan \frac{|u \times v|}{u \cdot v}. \quad (1)$$

Foi criado um *buffer* com o intuito de suavizar mudanças bruscas na saída, cujo tamanho é configurado no programa principal. O funcionamento se assemelha a uma fila: a cada imagem obtida é adicionado ao vetor o novo ângulo calculado e removido o valor mais antigo. A saída final é a média de todos os elementos de tal vetor. Dessa forma, valores espúrios tem menor relevância na saída final da função.

## 4 Resultados

Para a classificação do estado da mão (aberta ou fechada) foram utilizadas 197 amostras, sendo 59 reservadas para validação. Para a classificação dos dedos mostrados, o banco de cada classe foi composto por aproximadamente 597 amostras, sendo reservadas 176 para validação. Exemplos são mostrados na Fig. 5. Os modelos treinados produziram as matrizes de confusão, apresentadas nas Tabelas 1 a 4.

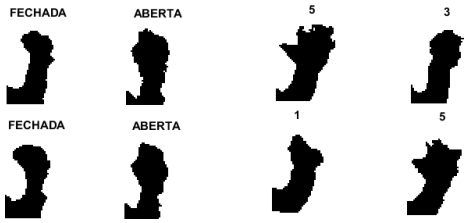


**Figura 5:** Exemplos contidos no banco de imagem para classificação dos gestos.

**Tabela 1:** Matriz de confusão, durante treinamento, de estados mão aberta ou fechada.

| | | Previsto | | |
|---|---|---|---|---|
| | | Aberta | Fechada | Total |
| Real | Aberta | 62 | 0 | 62 |
| | Fechada | 0 | 76 | 76 |
| | Total | 62 | 76 | 138 |

Em nenhum dos casos houve classificações erradas, ou seja, para o banco de imagens de validação, correspondente a aproximadamente 30% do total, o modelo fez todas as previsões corretamente. Com isso, alcançou-se a precisão de 100%. Vale ressaltar que, apesar de uma quantidade de dados relativamente baixa (aproximadamente entre 100 e 200 imagens por gesto), a técnica SVM foi capaz de classificar corretamente as imagens, tanto nos dados de treinamento quanto nos dados de validação, indicando uma razoável separabilidade entre as classes.

Para uma melhor análise do desempenho do código de criação dos modelos, adicionou-se ruído aleatório ao banco de imagens, em diferentes intensidades. Esse experimento foi dividido em duas partes: na primeira tanto os bancos de treinamento quanto o de validação foram contaminados com ruído. Em seguida, apenas o banco de teste apresentava as distorções, visando avaliar o desempenho do classificador quando submetido a imagens ruidosas ausentes no período de treinamento.

Foram aplicados incrementos de 2% no ruído inserido. Para cada quantidade de ruído foram treinados 10 modelos diferentes cuja precisão foi calculada, cuja média é mostrada na Figs. 6 e 7.

Como esperado, pela análise das Figuras 6 e 7, verifica-se que o aumento do ruído aplicado às amostras implica na redução do desempenho do classificador. Outra observação válida é que, quando o modelo é treinado já com as imagens distorcidas, seu resultado é superior ao caso em que as distorções estão ausentes do treinamento. Nota-se que a classificação do números de dedos é mais suscetível a presença de ruído.

Nesses experimentos, a precisão para o banco de treinamento se manteve em 100%, mesmo com as imagens totalmente aleatórias, havendo uma grande diferença em relação ao resultado no banco de dados de validação. Esse fenômeno é conhecido como *overfitting* (Karystinos e Pados, 2000; Liu

**Tabela 2:** Matriz de confusão, durante validação, de estados mão aberta ou fechada.

| | | Previsto | | |
|---|---|---|---|---|
| | | Aberta | Fechada | Total |
| Real | Aberta | 27 | 0 | 27 |
| | Fechada | 0 | 32 | 32 |
| | Total | 27 | 32 | 59 |

**Tabela 3:** Matriz de confusão, durante treinamento, de contagem de dedos.

| | | Previsto | | | |
|---|---|---|---|---|---|
| | | 1 | 3 | 5 | Total |
| Real | 1 | 132 | 0 | 0 | 132 |
| | 3 | 0 | 140 | 0 | 140 |
| | 5 | 0 | 0 | 140 | 140 |
| | Total | 132 | 140 | 140 | 421 |

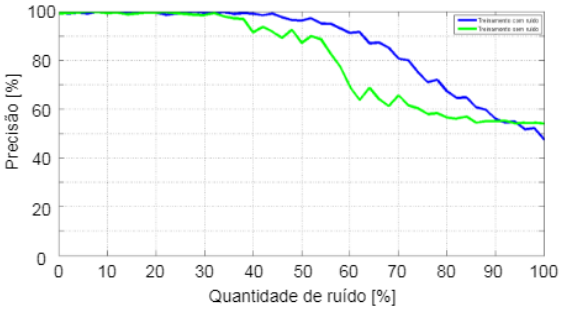


**Figura 6:** Precisão média obtida para validação na classificação da mão (aberta ou fechada) em imagens com ruído: treinamento com ruído (em azul) e sem ruído (verde).

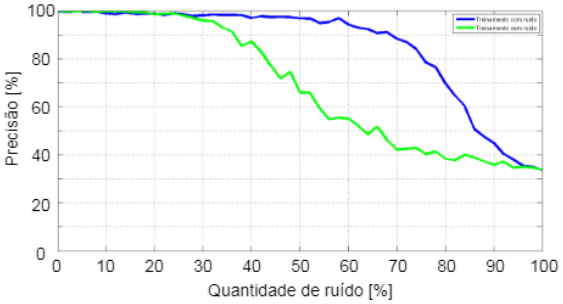


**Figura 7:** Precisão média obtida para validação na classificação de gestos (número de dedos) em imagens com ruído: treinamento com ruído (em azul) e sem ruído (verde).

e Castagna, 1999). A dimensão dos dados, muito maior que o número de amostras, torna o modelo mais suscetível a este tipo de problema.

A fim de avaliar o comportamento dos bancos de dados criados, utilizou-se o PCA para redução das dimensões. Na Fig. 8 são apresentados os gráficos de dispersão do banco de imagens para classificação de gestos (aberta ou fechada) após o PCA: à esquerda, sem ruído; à direita, com ruído. Já na Fig. 9 estão os gráficos de dispersão para classificação de gestos referentes ao número de dedos. Pela análise das Fig. 8 e 9 observa-se uma grande interferência do ruído na disposição dos dados, tanto para duas ou três classes. Intuitivamente, percebe-se uma maior separabilidade nos casos sem ruído. Isso é demonstrado nas figuras a

**Tabela 4:** Matriz de confusão, durante validação, de contagem de dedos.

| | | Previsto | | | |
|---|---|---|---|---|---|
| | | 1 | 3 | 5 | Total |
| Real | 1 | 56 | 0 | 0 | 56 |
| | 3 | 0 | 60 | 0 | 60 |
| | 5 | 0 | 0 | 60 | 60 |
| | Total | 56 | 60 | 60 | 176 |

seguir, que mostram o desempenho do classificador em função do número de dimensões após passar pelo PCA.

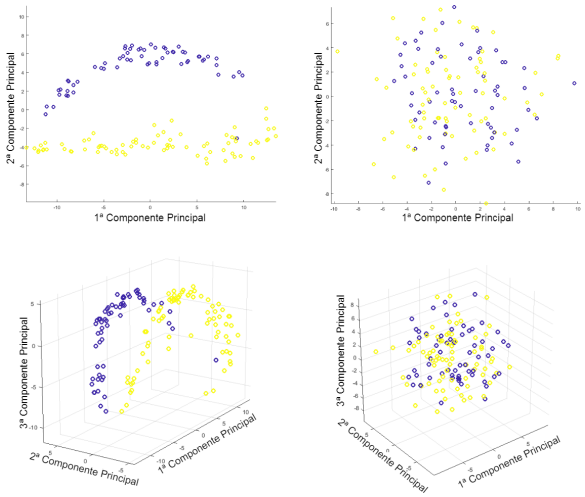


**Figura 8:** Gráficos de dispersão para classificação de gestos (aberta ou fechada) após o PCA: à esquerda, sem ruído; à direita, com ruído. Cada elemento representa uma imagem.

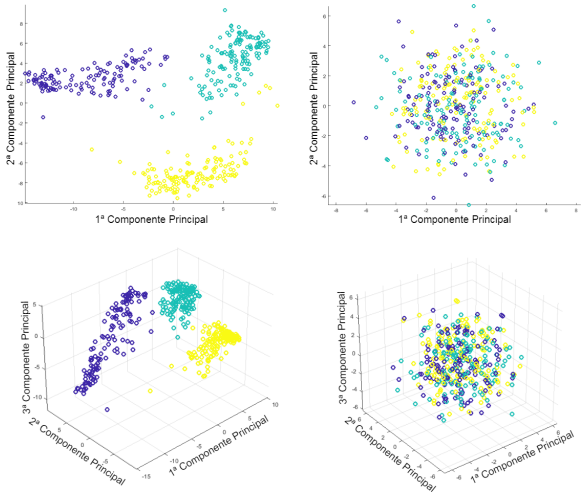


**Figura 9:** Gráficos de dispersão para classificação de gestos (número de dedos) após o PCA: à esquerda, sem ruído; à direita, com ruído. Cada elemento representa uma imagem.

A Fig. 10 apresenta a precisão do classificador em função do número de dimensões para o banco sem ruído em duas situações: na parte superior o classificador de gestos, na inferior, o contador de dedos. A precisão sobe rapidamente para 100% mesmo com poucas dimensões (em torno de três). Isso acontece em ambos os bancos de imagens (treinamento e teste). Já no banco de imagens com ruído (Fig. 11) o comportamento é diferente: a precisão no treinamento sobe gradualmente de acordo com o número de dimensões enquanto no teste se mantém em aproximadamente (1/número de classes). Isso caracteriza o chamado *overfitting*. A precisão atinge o máximo para o banco de treinamentos com cerca de 250 dimensões para o contador de dedos e aproximadamente 40 para o classificador de gestos.

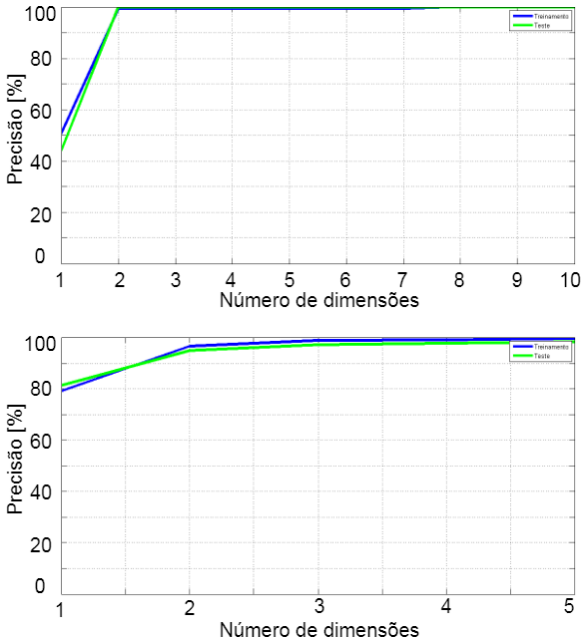


**Figura 10:** Precisão em função do número de dimensões do PCA (banco sem ruído). Treinamento em azul e teste em verde. Parte superior: classificador de gestos; parte inferior: contador de dedos.

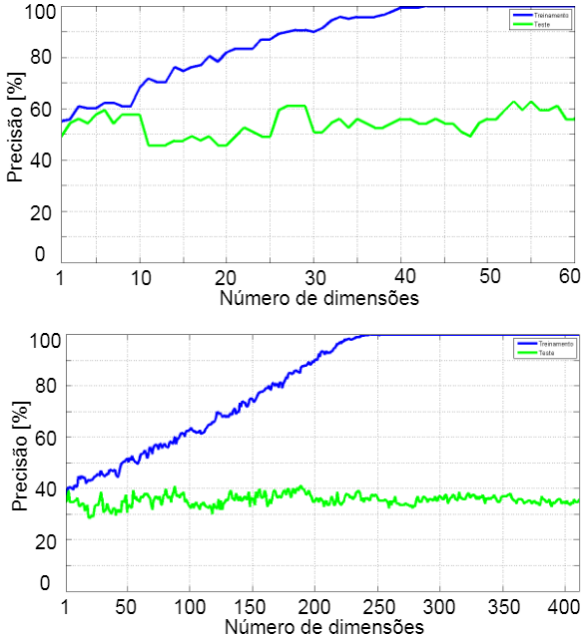


**Figura 11:** Precisão em função do número de dimensões do PCA (banco com ruído). Treinamento em azul e teste em verde. Parte superior: classificador de gestos; parte inferior: contador de dedos.

As próximas etapas referem-se à classificação em tempo real. Com ajuda da localização do ponto da mão fornecido pelo Microsoft Kinect®, extrai-se a região de interesse (ROI), obtendo uma ima-

**Figura 12:** Amostra: versão em escala de cinza e após a binarização.


**Figura 13:** Experimento com o sistema real: reconhecimento de gestos em tempo real.

gem em escala de cinza (Fig. 12). Na função de tratamento da imagem estabelece-se um padrão de tamanho e realiza-se a binarização. O limiar utilizado é a média dos valores de escala de cinza da imagem extraída, já que a mão ocupa grande área na figura. Os resultados são apresentados na Fig 13. Na parte superior, a classificação entre mão aberta e fechada; na central, a contagem dos dedos; e, na inferior, o cálculo do ângulo do braço. manipulador robótico.

Com base nos resultados apresentados foi possível enviar comandos para o manipulador, utilizando-se a biblioteca desenvolvida por Miranda et al. (2015). Esse experimento é omitido aqui em função da limitação de espaço. Destaca-se, entretanto, que tal biblioteca foi revista, ampliada e aperfeiçoada, de modo a acomodar as necessidades inerentes à realização do projeto apresentado.

## 5 Conclusão e trabalho futuro

Nesse trabalho foi proposto um sistema para operação de manipuladores robóticos baseado em reconhecimentos de gestos a partir do uso de uma Máquina de Vetores de Suporte (*Support Vector Machine*, ou SVM) em imagens fornecidas por um sensor RGB-D. Um dos problemas encontrados no desenvolvimento do projeto foi o *overfitting* - relacionado ao fato do número de dimensões (ou *features*) ser muito maior que o número de amostras. Para contornar essa questão, foi utilizada a técnica de PCA, que se mostrou bastante adequada. Os resultados obtidos demonstraram que, a partir da solução proposta, foi possível a captura, processamento, classificação e envio das informações corretas ao manipulador para execução dos movimentos predeterminados. Isso mostra a viabilidade de uso desse sistema em tempo real, cujas aplicações são diversas. Como trabalho futuro pretende-se abordar a área de tecnologia assistiva, como discutido em Miranda et al. (2015), substituindo-se a comunicação serial por uma comunicação via web. Além disso, pretende-se estender o número de comandos a serem enviados, visando formar um alfabeto mais adequado para aplicações assistivas.

## Agradecimentos

Trabalho apoiado por Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) e Fundação de Amparo à Pesquisa de São Paulo (FAPESP).

## Referências

Abdi, H. e Williams, L. J. (2010). Principal component analysis, *Wiley interdisciplinary reviews: computational statistics* **2**(4): 433–459.

Beluco, A. (2002). Classificação de imagens de sensoriamento remoto baseada em textura por redes neurais.

Bottou, L., Cortes, C., Denker, J. S., Drucker, H., Guyon, I., Jackel, L. D., LeCun, Y., Muller, U. A., Sackinger, E., Simard, P. et al. (1994). Comparison of classifier methods: a case study in handwritten digit recognition, *Proc. of the 12th IAPR Int. Con. on Pattern Recognition*, Vol. 2, IEEE, pp. 77–82.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition, *Data mining and knowledge discovery* **2**(2): 121–167.

Heisele, B., Ho, P. e Poggio, T. (2001). Face recognition with support vector machines: Global versus component-based approach, *8th IEEE Int. Conf. on Computer Vision*, Vol. 2, IEEE, pp. 688–694.

Heyer, C. (2010). Human-robot interaction and future industrial robotics applications, *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 4749–4754.

Hsu, C.-W. e Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines, *IEEE Trans. on Neural Networks* **13**(2): 415–425.

Karystinos, G. N. e Pados, D. A. (2000). On overfitting, generalization, and randomly expanded training sets, *IEEE Trans. on Neural Networks* **11**(5): 1050–1057.

Krizhevsky, A., Sutskever, I. e Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, pp. 1097–1105.

Latombe, J.-C. (2012). *Robot motion planning*, Vol. 124, Springer Science & Business Media.

Liu, Z. e Castagna, J. (1999). Avoiding overfitting caused by noise using a uniform training mode, *Int. Joint Conf. on Neural Networks*, Vol. 3, IEEE, pp. 1788–1793.

Lorena, A. C. e de Carvalho, A. C. (2007). Uma introdução às support vector machines, *Revista de Informática Teórica e Aplicada* **14**(2): 43–67.

Malisiewicz, T., Gupta, A. e Efros, A. A. (2011). Ensemble of exemplar-svms for object detection and beyond, *IEEE Int. Conf. on Computer Vision*, IEEE, pp. 89–96.

Matos, P., Lombardi, L., Ciferri, R., Pardo, T., Ciferri, C. e Vieira, M. (2009). Relatório técnico "métricas de avaliaçao", *Universidade Federal de Sao Carlos* .

Miranda, V. R., Medeiros, R., Mozelli, L. A., Souza, A. C. S., Alves-Neto, A. e Cardoso, A. S. V. (2015). Controle de um manipulador robótico via eletrooculografia: uma plataforma para tecnologia assistiva, *XII Simpósio Brasileiro de Automação Inteligente* pp. 1613–1618.

Mitchell, T. M. et al. (1997). Machine learning. 1997, *Burr Ridge, IL: McGraw Hill* **45**(37): 870–877.

Monard, M. C. e Baranauskas, J. A. (2003). Conceitos sobre aprendizado de máquina, *Sistemas Inteligentes Fundamentos e Aplicações*, 1 edn, Manole Ltda, Barueri-SP, pp. 89–114.

Niemeyer, G., Preusche, C. e Hirzinger, G. (2008). Telerobotics, *Springer handbook of robotics*, Springer, pp. 741–757.

Nölker, C. e Ritter, H. (1999). Grefit: Visual recognition of hand postures, *International Gesture Workshop*, Springer, pp. 61–72.

Pillajo, C. e Sierra, J. E. (2013). Human machine interface hmi using kinect sensor to control a scara robot, *IEEE Colombian Conf. on Communications and Computing (COLCOM)*, IEEE, pp. 1–5.

Pisharady, P. K. e Saerbeck, M. (2015). Recent methods and databases in vision-based hand gesture recognition: A review, *Computer Vision and Image Understanding* **141**: 152–165.

Prina, B. Z. e Trentin, R. (2015). Gmc: Geração de matriz de confusão a partir de uma classificação digital de imagem do arcgis®, *Simp. Brasileiro de Sensoriamento Remoto* **17**: 131–139.

Ruffieux, S., Lalanne, D., Mugellini, E. e Khaled, O. A. (2015). Gesture recognition corpora and tools: A scripted ground truthing method, *Computer Vision and Image Understanding* **131**: 72–87.

Shiba, M. H., Santos, R. L., Quintanilha, J. A. e KIM, H. (2005). Classificação de imagens de sensoriamento remoto pela aprendizagem por árvore de decisão: uma avaliação de desempenho, *Simp. Brasileiro de Sensoriamento Remoto* **12**: 4–319.

Tang, Y., Sun, Z. e Tan, T. (2014). Slice representation of range data for head pose estimation, *Computer Vision and Image Understanding* **128**: 18–35.

Vapnik, V. (2013). *The nature of statistical learning theory*, Springer science & business media.

Weinland, D., Ronfard, R. e Boyer, E. (2011). A survey of vision-based methods for action representation, segmentation and recognition, *Computer vision and image understanding* **115**(2): 224–241.

Wold, S., Esbensen, K. e Geladi, P. (1987). Principal component analysis, *Chemometrics and intelligent laboratory systems* **2**(1-3): 37–52.