

INF1771 - INTELIGÊNCIA ARTIFICIAL

TRABALHO 3 – APRENDIZADO DE MÁQUINA

O **objetivo do Trabalho 3** é desenvolver um sistema de reconhecimento de gestos feitos com a mão (Figura 1). Este problema envolve o reconhecimento do formato e também dos movimentos realizados com a mão.

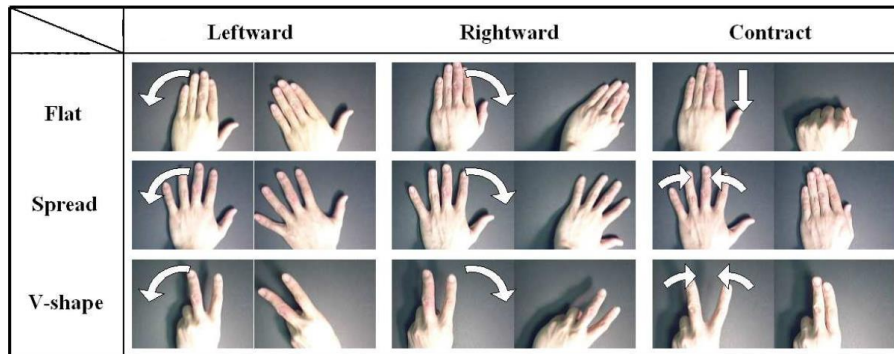


Figura 1: Tipos de gestos que podem ser feitos pelo usuário.

Para essa tarefa, você tem a disposição um conjunto de 90 sequencias de imagens (mais de 64 mil frames) de 9 tipos diferentes de gestos, os quais são compostos por 3 formatos de mão e 3 tipos de movimentos. Esse conjunto de imagens foi criado pelos autores do artigo “*Tensor Canonical Correlation Analysis for Action Classification*” [1].

Os gestos que compõem o conjunto de treinamento são os seguintes:

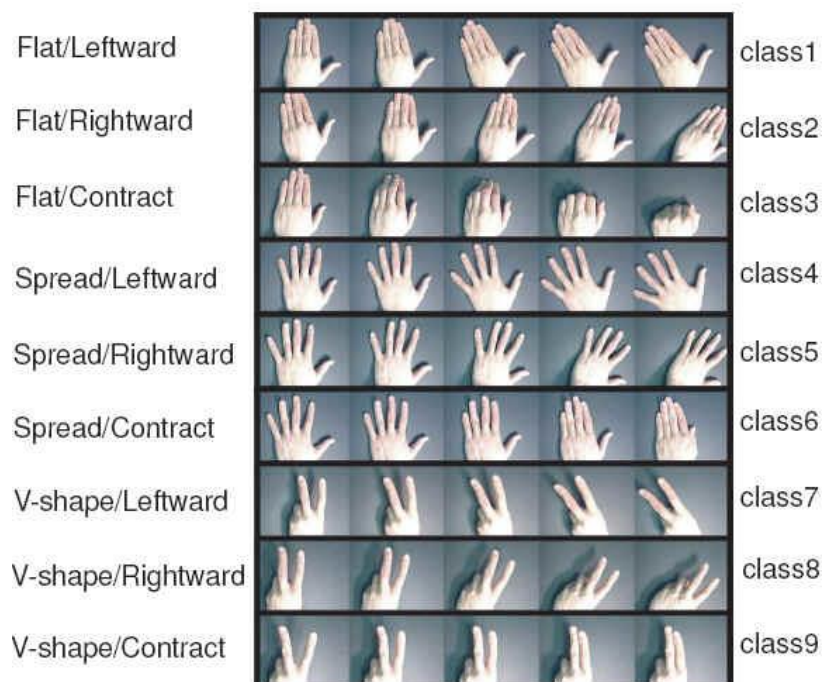


Figura 2: Gestos que compõem o conjunto de treinamento.

Para desenvolver o sistema de reconhecimento de gestos você deve utilizar algum método de **aprendizado de máquina supervisionado**, visto que temos um conjunto rotulado de exemplos para treinamento (sequencias de imagens). Para isso, você deve seguir os seguintes passos:

- 1) Definir quais serão os atributos que serão usados para descrever os exemplos de treinamento.
- 2) Criar um programa para extrair os atributos das sequencias de imagens de forma a gerar um conjunto de treinamento e um conjunto de validação. Normalmente gera-se um único conjunto de dados e depois se divide ele em dois conjuntos (treinamento e validação). Cada sequencia de imagens representando um gesto será um exemplo de treinamento.
- 3) Utilizar 4 algoritmos de aprendizado supervisionado, treinando-os com o conjunto de treinamento e depois realizando classificação do conjunto de testes para verificar qual algoritmo apresenta a melhor taxa de reconhecimento. Devem ser utilizados os seguintes algoritmos:
 - Árvores de Decisão
 - K-Nearest Neighbor (KNN)
 - Support Vector Machine (SVM)
 - Rede Neural (usando backpropagation)

Não é necessário programar os algoritmos, é permitida a utilização de ferramentas de aprendizado de máquina, como por exemplo, o Weka (<http://www.cs.waikato.ac.nz/ml/weka/>).

Se a taxa de reconhecimento estiver muito baixa para todos os algoritmos, deve-se retornar para a etapa 1 e selecionar melhor os atributos para descrever os exemplos de treinamento.

- 4) Com base nos resultados dos testes, você deve implementar o algoritmo que obteve melhores resultados na linguagem de sua preferencia.

Programa Base

O programa base fornecido demonstra como alguns atributos podem ser extraídos das imagens de treinamento. Os atributos extraídos pelo programa base são:

- 1) Largura do retângulo englobante do maior objeto segmentado encontrado imagem;
- 2) Altura do retângulo englobante do maior objeto segmentado encontrado na imagem;
- 3) Perímetro do maior objeto segmentado encontrado na imagem;

- 4) Área do maior objeto segmentado encontrado na imagem;
- 5) Angulo do retângulo englobante do maior objeto segmentado encontrado na imagem;
- 6) Área do convex hull do maior objeto segmentado encontrado na imagem;
- 7) Aspect Ratio do retângulo englobante do maior objeto segmentado encontrado na imagem;
- 8) Maior eixo da elipse englobante do maior objeto segmentado encontrado na imagem;
- 9) Menor eixo da elipse englobante do maior objeto segmentado encontrado na imagem;
- 10) Angulo da elipse englobante do maior objeto segmentado encontrado na imagem;
- 11) 1º momento invariante de Hu do maior objeto segmentado encontrado na imagem;
- 12) 2º momento invariante de Hu do maior objeto segmentado encontrado na imagem;
- 13) 3º momento invariante de Hu do maior objeto segmentado encontrado na imagem;
- 14) 4º momento invariante de Hu do maior objeto segmentado encontrado na imagem;
- 15) 5º momento invariante de Hu do maior objeto segmentado encontrado na imagem;
- 16) 6º momento invariante de Hu do maior objeto segmentado encontrado na imagem;
- 17) 7º momento invariante de Hu do maior objeto segmentado encontrado na imagem;

O programa base está disponível no seguinte link:

http://edirlei.3dgb.com.br/aulas/ia_2014_1/Trabalho3ProgramaBase_2014_1.zip

O programa base foi desenvolvido utilizando a biblioteca de visão computacional OpenCV (<http://opencv.org/>). Mais detalhes sobre o funcionamento do programa base e como utilizar o OpenCV serão apresentados durante a aula.

Informações Adicionais:

- Durante a etapa de teste dos 4 algoritmos não é necessário implementar todos os classificadores para realizar os experimentos. É permitida a utilização de ferramentas de aprendizado de máquina, como por exemplo, o Weka (<http://www.cs.waikato.ac.nz/ml/weka/>).

- O download das imagens de treinamento pode ser feito acessando o seguinte link:

<ftp://mi.eng.cam.ac.uk/pub/CamGesData/>

Ou copiado durante as próximas aulas (é necessário um pen drive com pelo menos 1 GB livre).

- A divisão dos exemplos de treinamento e teste deve ser feita de forma aleatória. Normalmente 70% dos exemplos devem ser utilizados para treinamento e 30% para testes. Você deve garantir também que os exemplos de cada classe sejam distribuídos nestas mesmas proporções nos dois conjuntos.
- Ao realizar experimentos no Weka, sempre faça *cross-validation* com pelo menos 10 *folds*.
- A implementação de um dos algoritmos pode ser feita em qualquer linguagem (C/C++, C#, Java...), mas aconselha-se utilizar a linguagem C/C++ usando o programa base para ler a sequência de imagens e realizar a classificação.
- Você deve decidir quais atributos serão utilizados pelos classificadores. A classificação pode piorar ou melhorar dependendo do conjunto de atributos utilizado. Os atributos extraídos pelo programa base são apenas exemplos de atributos que podem ser extraídos. Você deve escolher quais atributos serão utilizados e fazer testes para verificar qual conjunto de atributos apresenta melhores resultados.
- Para se conseguir melhores resultados será necessário extrair mais características das imagens de treinamento. O programa base extrai apenas algumas características estruturais, mas existem muitas outras informações úteis que podem ser extraídas computacionalmente das imagens.

Forma de Avaliação:

Será avaliado se todas as etapas do processo foram cumpridas corretamente. A avaliação também será baseada na **apresentação dos resultados** durante a aula.

Essa apresentação deverá conter:

- Descrição da modelagem dos exemplos de treinamento:
 - Atributos selecionados para descrever os exemplos;
 - Justificativa para a escolha dos atributos;
 - Estrutura dos exemplos;
- Descrição dos experimentos realizados:
 - Variações na modelagem dos exemplos;
 - Variações no conjunto treinamento e testes;
 - Variação nos parâmetros dos algoritmos;
- Comparação dos algoritmos analisados:
 - Taxa de reconhecimento;
 - Tempo gasto no processo de treinamento;
 - Tempo gasto no processo de classificação de um exemplo desconhecido;
- Implementação:
 - Descrição de como o algoritmo escolhido foi implementado;
 - Resultados alcançados pelo algoritmo que foi implementado (tempo gasto no processo de treinamento, tempo gasto no processo de classificação de um exemplo desconhecido, taxa de reconhecimento).

Bônus:

- O trabalho que conseguir a maior taxa de reconhecimento receberá 2.0 pontos extras na nota.
- O trabalho que utilizar o algoritmo de reconhecimento de gestos em um jogo de “pedra, papel e tesoura” capturando os gestos da mão dos usuários com uma webcam em tempo-real receberá 3.0 pontos extras na nota.

Data de Entrega:

09/06

Forma de Entrega:

Os trabalhos devem ser **apresentados** na aula do dia 09/06 (segunda) e 11/06 (quarta).

Todos devem enviar o trabalho até o dia 11/06 para o email edirlei.slima@gmail.com.

Não serão aceitos trabalhos enviados depois desta data.

Referências:

[1] T-K. Kim, S-F. Wong and R. Cipolla, Tensor Canonical Correlation Analysis for Action Classification, In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Minneapolis, MN, 2007. http://www.iis.ee.ic.ac.uk/icvl/doc/cvpr07_tcca_final.pdf