

# Sistema classificador de imagens para personagens do seriado “Os Simpsons”

Giovani Henrique Bertuzzo<sup>1</sup>, Mateus Vieira Torres<sup>1</sup>.

<sup>1</sup>DACOM – Universidade Tecnológica Federal do Paraná (UTFPR)  
Caixa Postal 271 – 87301-899 – Campo Mourão – PR – Brazil

giovani.hbertuzzo@gmail.com, torres@alunos.utfpr.edu.br

**Abstract.** *This report describes the procedure used in the development of an image classification system for the characters of the series "The Simpsons". It uses the concepts and techniques of Artificial Intelligence, in conjunction with some feature extraction libraries, to classify the set of images.*

**Resumo.** *Este relatório, descreve os procedimentos utilizados no desenvolvimento de um sistema classificador de imagens para personagens da série “Os Simpsons”. Este utiliza os conceitos e técnicas ministrados na disciplina da Inteligência Artificial, em conjunto com algumas bibliotecas extratoras de características, para classificação do conjunto de imagens.*

## 1. Introdução

O tema Inteligência Artificial (IA) é muito discutido atualmente, seja no âmbito técnico ou ético. O fato é que esta vem realmente ganhando notoriedade no meio em que vivemos devido a sua grande contribuição em diversas áreas da tecnologia, entre elas, o reconhecimento de padrões, cujo uma das funções é ser capaz de identificar a qual classe um determinado elemento pertence.

Para alguns casos a automatização de tarefas, como por exemplo, uma máquina que é capaz de prever a espécie de um peixe (i.e, Salmão, Bacalhau, Sardinha). Parece muito fácil para o ser humano identificar tais classes à olho nu, porém automatizar esse processo é uma atividade extremamente complexa devido a diferença entre a capacidade do cérebro humano e do computador de fazer relações entre outros processos relacionados.

A tecnologia de reconhecimento de padrões pode abordar diferentes técnicas para solucionar um problema, uma delas é o aprendizado não supervisionado, que fornece ao sistema um conjunto de treino com as classes já classificadas para prover uma base de comparação, ou seja tentar “ensinar” as relações para o sistema, após isso o sistema recebe o conjunto de testes para classificá-los objetivando o maior número de acertos e usando como base o “conhecimento” adquirido com o conjunto de treino.

## 2. Objetivos

O sistema desenvolvido tem como objetivo prever a partir de uma base de dados (imagens), a qual classe de personagens do seriado “Os Simpsons” o elemento analisado pertence. O programa deve ser capaz de diferenciar cinco personagens do seriado, Homer Jay Simpson (Homer), Marjorie Bouvier Simpson (Marge),

Bartholomew Simpson (Bart), Elisabeth Marie Simpson (Lisa) e Margareth Simpson (Maggie). Parte do processo é extração de características dos personagens, que são escolhidas visando maximizar a precisão do sistema.

### 3. Procedimentos

Conforme descrito anteriormente, um sistema com aprendizado não supervisionado, recebe um conjunto de dados, no nosso caso várias imagens contendo os personagens que já foram previamente classificados, deles são escolhidas e extraídas as características que deem mais destaque para a diferença entre os personagens e então é gerado um arquivo “treino.arff” para ser utilizado como conjunto de treino no classificador. A grande dificuldade está em selecionar e principalmente extrair tais características a partir do conjunto de dados. Para isso, utilizamos alguns Extratores implementados na linguagem Java.

*Extratores:* são bibliotecas ou classes cujos métodos tem como função extrair características da imagem. Neste projeto foram utilizados:

- Extrator Borda: retorna a quantidade pixels de contorno na imagem.
- Extrator Cor Predominante: retorna a cor predominante na imagem, desconsiderando o branco.
- Extrator Cores Predominantes: retorna as 5 cores mais recorrentes na imagem.
- Extrator Quantidade de Cores: retorna a quantidade de cores que existem na imagem.
- Extrator Quantidade de Pixels: retorna a quantidade de pixels na imagem.
- Extrator Cores Específicas: retorna a quantidade de pixels pretos, brancos, vermelhos, laranjas, azuis claros e azuis escuros na imagem.

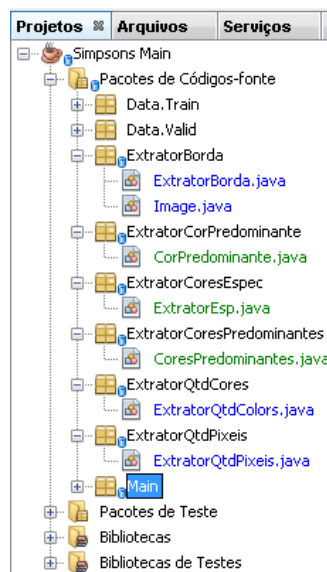


Imagem 1. Extratores

Após a extração das características, o sistema gera um arquivo com extensão “.arff” para ser utilizado no software Weka.

O Weka é um conjunto de algoritmos de aprendizado de máquina que pode ser usado para tarefas de mineração de dados é um software livre e se consolidou como a ferramenta de mineração de dados mais utilizada por estudantes e professores de universidades.

A partir do Weka podemos rodar a base de características extraídas em diferentes classificadores e verificar qual nos resulta um melhor desempenho.

## 4. Resultados Individuais dos Classificadores

Os conjuntos de treino e teste foram executados em diferentes abordagens de classificação, com intuito de obter um melhor desempenho na solução do problema.

Assim fora definido três classificadores, sendo eles, o IBk(Knn), SMO(SVM) e o OneR(One Rule), a seguir será demonstrado o desempenho dos mesmos.

### 4.1 - IBk(Knn)

O algoritmo Knn foi utilizado com 2 métricas, uma utilizando k=3, e outra com k=5. Os resultados obtidos no primeiro caso onde k=3 apresentaram 61,05% acurácia, e para k=5 68,42%.

Esse classificador apresentou um resultado acima de 70% de acerto para todas as classes exceto para a Marge que apresentou uma taxa de acerto de 10%.

```

=== Summary ===

Correctly Classified Instances      65           68.4211 %
Incorrectly Classified Instances    30           31.5789 %
Kappa statistic                    0.5665
Mean absolute error                 0.1708
Root mean squared error             0.3249
Relative absolute error             56.5533 %
Root relative squared error         83.8911 %
Total Number of Instances          95

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0,714    0,200    0,676     0,714    0,694      0,509    0,768     0,635    Bart
      0,769    0,012    0,909     0,769    0,833      0,813    0,906     0,771    Lisa
      0,100    0,024    0,333     0,100    0,154      0,134    0,505     0,145    Marge
      0,750    0,000    1,000     0,750    0,857      0,851    0,948     0,888    Maggie
      0,800    0,214    0,571     0,800    0,667      0,535    0,902     0,743    Homer
Weighted Avg.    0,684    0,134    0,685     0,684    0,670      0,561    0,817     0,662

=== Confusion Matrix ===

  a  b  c  d  e  <-- classified as
25  1  2  0  7  |  a = Bart
 0 10  0  0  3  |  b = Lisa
 7  0  1  0  2  |  c = Marge
 0  0  0  9  3  |  d = Maggie
 5  0  0  0 20  |  e = Homer

```

**Imagem 2.** Matriz confusão do Classificador Knn(k=5)

## 4.2 - SMO(SVM)

O classificador SMO apresentou 65,26% de acerto nas classificações, porém também apresentou baixo desempenho para a classificação do personagem Marge.

```
=== Summary ===

Correctly Classified Instances      62          65.2632 %
Incorrectly Classified Instances    33          34.7368 %
Kappa statistic                    0.5118
Mean absolute error                 0.2669
Root mean squared error             0.3562
Relative absolute error             88.4064 %
Root relative squared error         91.9706 %
Total Number of Instances          95

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0,829    0,317    0,604    0,829    0,699      0,494    0,740    0,569    Bart
      0,769    0,037    0,769    0,769    0,769      0,733    0,887    0,669    Lisa
      0,100    0,012    0,500    0,100    0,167      0,189    0,627    0,175    Marge
      0,417    0,000    1,000    0,417    0,588      0,620    0,872    0,594    Maggie
      0,680    0,143    0,630    0,680    0,654      0,524    0,826    0,542    Homer
Weighted Avg.    0,653    0,161    0,672    0,653    0,627      0,518    0,788    0,537

=== Confusion Matrix ===

  a  b  c  d  e  <-- classified as
29  3  1  0  2 | a = Bart
 0 10  0  0  3 | b = Lisa
 9  0  1  0  0 | c = Marge
 2  0  0  5  5 | d = Maggie
 8  0  0  0 17 | e = Homer
```

*Imagem 3. Matriz confusão do Classificador SMO*

## 4.3 – OneR(One Rule)

O classificador OneRule apresentou uma acurácia de 69,47%, um resultado também satisfatório, porém apresentou a mesma falha dos outros classificadores ao apresentar uma taxa de acerto de apenas 10% para a classe Marge. O ponto forte do algoritmo ficou na classe Bart, onde classificou corretamente 31 instâncias das 35 existentes, ou seja 88,60%.

```
=== Summary ===

Correctly Classified Instances      66          69.4737 %
Incorrectly Classified Instances    29          30.5263 %
Kappa statistic                    0.5713
Mean absolute error                 0.1221
Root mean squared error             0.3494
Relative absolute error             40.4382 %
Root relative squared error         90.2145 %
Total Number of Instances          95

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
      0,886    0,400    0,564    0,886    0,689      0,475    0,743    0,541    Bart
      0,846    0,049    0,733    0,846    0,786      0,752    0,899    0,642    Lisa
      0,100    0,012    0,500    0,100    0,167      0,189    0,544    0,145    Marge
      0,750    0,000    1,000    0,750    0,857      0,851    0,875    0,782    Maggie
      0,560    0,000    1,000    0,560    0,718      0,696    0,780    0,676    Homer
Weighted Avg.    0,695    0,155    0,750    0,695    0,676      0,588    0,770    0,579

=== Confusion Matrix ===

  a  b  c  d  e  <-- classified as
31  3  1  0  0 | a = Bart
 2 11  0  0  0 | b = Lisa
 9  0  1  0  0 | c = Marge
 3  0  0  9  0 | d = Maggie
10  1  0  0 14 | e = Homer
```

*Imagem 4. Matriz confusão do Classificador OneR*

## 5. Fusão de Classificadores - Vote (3 classificadores)

Por último, como requisitado no documento do trabalho, fora utilizado o método de combinação de classificadores Vote do Weka. Os classificadores usados na combinação foram os mesmos demonstrados anteriormente, a regra de combinação escolhida foi o voto majoritário e o resultado geral obtido pelo método Vote foi de 70,52% de acurácia ou seja uma leve melhora no desempenho. A combinação apontou um excelente resultado para as classes Bart e Lisa que apresentaram uma taxa de acerto (TP Rate) maior que 80% como podemos observar na imagem a seguir.

```
=== Summary ===

Correctly Classified Instances      67          70.5263 %
Incorrectly Classified Instances    28          29.4737 %
Kappa statistic                    0.5868
Mean absolute error                 0.1866
Root mean squared error             0.3026
Relative absolute error             61.7993 %
Root relative squared error         78.1148 %
Total Number of Instances          95

=== Detailed Accuracy By Class ===
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0,886	0,383	0,574	0,886	0,697	0,489	0,795	0,690	Bart
	0,846	0,049	0,733	0,846	0,786	0,752	0,928	0,837	Lisa
	0,100	0,012	0,500	0,100	0,167	0,189	0,627	0,180	Marge
	0,750	0,000	1,000	0,750	0,857	0,851	0,955	0,899	Maggie
	0,600	0,000	1,000	0,600	0,750	0,725	0,920	0,848	Homer
Weighted Avg.	0,705	0,149	0,754	0,705	0,687	0,601	0,849	0,724	

```
=== Confusion Matrix ===

 a b c d e <-- classified as
31 3 1 0 0 | a = Bart
 2 11 0 0 0 | b = Lisa
 9 0 1 0 0 | c = Marge
 3 0 0 9 0 | d = Maggie
 9 1 0 0 15 | e = Homer
```

*Imagem 5. Matriz confusão do método Vote*

## 5. Conclusão

Em geral observamos que o classificador OneRule obteve o melhor desempenho entre os classificadores individuais, foi constatado também que existe algum problema provavelmente nas características da classe Marge que confundem ou dificultam a classificação. Alguns testes também foram realizados isolando as características para determinar qual característica é mais discriminatória.

Como esperado ao remover as características quantidade de pixels na imagem, quantidade de pixels borda e quantidade de cores na imagem não constatamos nenhuma mudança significativa no desempenho do classificador. Através dos testes conseguimos verificar que a característica mais determinante é a cor predominante, apenas com essa característica isolada fomos capaz de obter desempenho maior do que o com o conjunto total.

Portanto vale ressaltar a importância da escolha das características pois uma boa característica pode gerar um bom resultado na classificação e uma escolha ruim de característica pode diminuir drasticamente o desempenho do seu sistema ou seja qualidade é melhor do que quantidade.

## 6. Referências

SHAMS, Rushdi. **Weka Tutorial 13- Stacking Multiple Classifiers (Classification)**. Canada: Vídeo, 2015. Color. Disponível em: <<https://www.youtube.com/watch?v=Nje8mblA7bs>>. Acesso em: 03 dez. 2016.

**Class OneR**. Dica de Leitura. Disponível em: <<http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/OneR.html>>. Acesso em: 03 de dez. 2016.

**Class IBk**. Dica de Leitura. Disponível em: <<http://weka.sourceforge.net/doc.dev/weka/classifiers/lazy/IBk.html>>. Acesso em: 03 de dez. 2016.

**Class SMO**. Dica de Leitura. Disponível em: <<http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html>>. Acesso em: 03 de dez. 2016.

**Class Vote**. Dica de Leitura. Disponível em: <<http://weka.sourceforge.net/doc.dev/weka/classifiers/meta/Vote.html>>. Acesso em: 03 de dez. 2016.