

Extração de características dos personagens da família Simpsons

Noemi Pereira Scherer¹

¹Universidade Tecnológica Federal do Paraná (UTFPR)

Campo Mourão – PR – Brasil

{noemi}, noemischerer13@gmail.com

Abstract. *This work consists of training and validating a set of images of Simpsons family characters. For this purpose, extraction characteristics, three different types of classifiers, and two classifier fusion are defined for the amount of correctness and final error.*

Resumo. *Esse trabalho consiste em treinar e validar um conjunto de imagens de personagens da família Simpsons. Para isso, são definidas características de extrações, três diferentes tipos de classificadores, e duas fusão de classificadores para a quantidade de acerto e erro final.*

1. Introdução

Reconhecimento de padrões é uma área da ciência cujo objetivo é a classificação de objetos dentro de um número de categorias ou classes. Esses objetos de estudo variam de acordo com cada aplicação, podem ser imagens, sinais em forma de ondas (como voz, luz, rádio) ou qualquer tipo de medida que necessite ser classificada. Tendo aplicação em vários campos, tais como ciência da computação [1].

2. Objetivo

O objetivo desse trabalho é reconhecer características dos cinco personagens da família Simpsons: Bart, Homer, Lisa, Meggie e Merge, utilizando um programa que aplique características para classificação, três tipos de classificadores, e dois métodos de fusão de classificadores. Para o reconhecimento de características existem duas bases de dados, uma de treinamento, contendo todas as imagens dos personagens para serem treinadas e extraídas as características, e outra de validação, para extrair características e determinar se o treinamento está correto.

3. Materiais e métodos

- Linguagem Matlab;
- Características para reconhecimento de padrão da imagem;
- Três diferentes classificadores: K-NN, Random Forest e Bayes Naives;
- Weka para uso dos classificadores;
- Dois tipos de fusão de classificadores: Regra da Soma e do Máximo.

4. O código

4.1.Reconhecimento.m

É utilizado arquivos de treinamento, definido as classes de cada personagem para a saída do arquivo, Bart representa o 0, Homer 1, Lisa 2, Meggie 3 e Marge 4. O programa encontra-se na pasta onde estão armazenadas todas as imagens de treinamento para a extração de características. A variável arquivo recebe todos os dados do diretório, e dentro de um laço de repetição percorre cada arquivo respectivo do personagem, no qual é treinado características para extrair padrões.

A extração é realizada através de pixel, com o objetivo de encontrar pixels correspondentes às cores especificadas. Existe uma tolerância = 5, ou seja, a cor pode variar 5 para mais ou para menos. Para cada acerto, são normalizados as variáveis de pixels, dividindo ela pela quantidade total de pixels.

Para obter a cor correta de cada característica, é definida e escolhida uma imagem que melhor representa cada personagem, plotada e extraída suas configurações, utilizando o `impixelinfo` (Figura 1). Quando plotada, ela mostra todas as cores em RGB, assim, é criada uma imagem para cada características recebendo a sua respectiva cor. Porém, é preciso apenas multiplicar o valor RGB encontrado por 255.

Após normalizada todas as características, elas são salvas em arquivos `.txt`, contendo a especificação de cada personagem treinado.

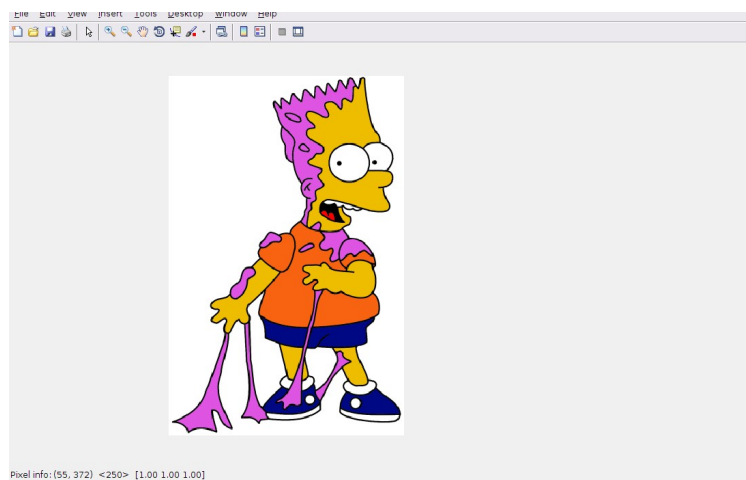


Figura 1: Extração das cores RGB do personagem Bart usando `impixelinfo`

4.2.ReconhecimentoValida.m

O arquivo encontra-se dentro do diretório Valida, no qual possui todas as imagens de teste para o reconhecimento dos Simpsons. O Programa é o mesmo que o `Reconhecimento.m`, porém, difere na quantidade de imagens para cada um dos personagens e os nomes dos arquivos a qual são salvas as características encontradas.

5. Procedimentos e resultados

5.1.Procedimentos

O trabalho consiste em quatro etapas: o desenvolvimento do código para extração de características, aplicação de três classificadores nas extrações, a fusão de classificadores e por fim, a análise total dos resultados obtidos.

1. **Extração de Características:** É utilizado o Matlab para implementação dessa etapa. São criados dois programas, o Reconhecimento.m, no qual recebe os dados de treino, que se encontra na pasta Treinamento. ReconhecimentoValida.m, que tem como entrada os dados de validação, salvo na pasta Valida. A descrição de ambos os códigos encontra-se no tópico 4. Para ambos os programas, são definidas e encontradas dez características no total, sendo duas para cada personagem da família Simpsons.
2. **Classificadores:** Knn, *Bayes Naive* e *Random Forest*, são utilizados. Cada um é aplicado utilizando o programa Weka.
3. **Fusão de classificadores:** Tem como objetivo juntar os três classificadores utilizado e gerar uma porcentagem final de acerto e erro na definição de cada personagem. Nesse trabalho, são utilizado duas fusões, a Regra da Soma, no qual a classe que apresentou a maior soma, é a classe correta; e a Regra do Máximo, considera a classe com maior valor a correta. Para os resultados, utiliza-se o *LibreOffice*, no qual a soma e o máximo são aplicado através de fórmula em cada dado de cada classificador.

São escolhidas como características de extração as cores que mais prevalece no personagem, utilizando o `impixelinfo` para cor correta. Para o Bart, utiliza-se a cor da camiseta e do shorts; para o Homer, a cor da barba e dos sapatos; na Lisa, a cor do vestido e do colar; a cor da roupa e da chupeta para a Meggie; e por fim, a cor do vestido e do cabelo para a Marge.

Para saber os resultados de acertos e erros, utiliza-se o weka, para escolher o classificador desejado (K-NN, Bayes Naives e Random Forest). Antes de utilizar o Weka, é preciso criar um arquivo no formato que ele reconheça os dados (*.arff*). Para isso, cria-se um novo arquivo, um para os dados de treinamento e outro para validação, acrescenta os dados gerados pelo código do arquivo *.txt* e os seguintes parâmetros, ver Figura 2:

- @relation: Nome do programa
- @attribute x numeric: Número de características. No total são dez, ou seja, dez attributes, no qual x, vai de 00 até 09.
- @attribute class {0, 1, 2, 3, 4}: Quantidade de classes trabalhadas, são cinco, uma para cada personagem.
- @data: a partir dele encontra-se os dados obtidos do arquivo de teste.

```

@relation valida.simpsons
@attribute 00 numeric
@attribute 01 numeric
@attribute 02 numeric
@attribute 03 numeric
@attribute 04 numeric
@attribute 05 numeric
@attribute 06 numeric
@attribute 07 numeric
@attribute 08 numeric
@attribute 09 numeric
@attribute class {0,1,2,3,4}

@data
0.538502 0.002550 0.001938 0.008169 0.010200 0.000000 0.000000 0.000000 0.002008 0.000000 0
0.311395 0.002344 0.001738 0.004815 0.000000 0.000000 0.000000 0.000000 0.001323 0.000000 0
0.559644 0.004662 0.001433 0.005530 0.000000 0.000000 0.000000 0.000000 0.002187 0.000000 0
0.540463 0.001436 0.000939 0.004226 0.013376 0.000000 0.000000 0.000000 0.001261 0.000000 0
0.440090 0.000341 0.002588 0.010567 0.385379 0.000000 0.000000 0.000000 0.002123 0.000000 0
0.618341 0.173630 0.000876 0.006338 0.004840 0.000000 0.000000 0.000000 0.001402 0.000000 0
0.014298 0.091490 0.003209 0.002806 0.528226 0.000000 0.000000 0.000000 0.001663 0.000000 0
0.049307 0.002783 0.121867 0.007923 0.585216 0.000000 0.000000 0.000000 0.001225 0.000000 0
0.632287 0.001208 0.002023 0.003452 0.004026 0.000000 0.000000 0.000000 0.001379 0.000000 0
0.535347 0.008832 0.001409 0.003278 0.018187 0.000000 0.000000 0.000000 0.000709 0.000000 0
0.005371 0.006053 0.002659 0.005697 0.000000 0.000000 0.000000 0.000000 0.001485 0.000000 0
0.006686 0.002645 0.002308 0.003988 0.000000 0.000000 0.000000 0.000000 0.000605 0.000000 0
0.500427 0.001380 0.006541 0.005835 0.000000 0.000000 0.000000 0.000000 0.001913 0.000000 0
0.006488 0.040069 0.015686 0.007213 0.000000 0.000000 0.000000 0.000000 0.002192 0.000000 0
0.466999 0.070915 0.000953 0.002279 0.000000 0.000000 0.000000 0.000000 0.001923 0.000000 0
0.544146 0.002194 0.001814 0.002607 0.000000 0.000000 0.000000 0.000000 0.001943 0.000000 0
0.464113 0.002385 0.002930 0.003093 0.000000 0.000000 0.000000 0.000000 0.001689 0.000000 0
0.586689 0.019641 0.001294 0.002339 0.000000 0.000000 0.000000 0.000000 0.003060 0.000000 0
0.573878 0.004688 0.000764 0.008024 0.000000 0.000000 0.000000 0.000000 0.001554 0.000000 0
0.055892 0.000859 0.002387 0.009227 0.660339 0.000000 0.000000 0.000000 0.001882 0.000000 0

```

Figura 2: Arquivo validação no formato .arff

No treinamento.arff e valida.arff, existem 11 colunas. As dez primeiras são as características e a quantidade obtidas delas (Camisa e shorts do Bart, barba e sapato do Homer, vestido e colar da Lisa, roupa e chupeta da Meggie, e cabelo e vestido da Marge, respectivamente). A última coluna é o personagem, sendo 0 = Bart, 1 = Homer, 2 = Lisa, 3 = Meggie e 4 = Marge.

A quantidade de linhas representa a quantidade total de dados trabalhados, no arquivo de treinamento as 78 primeiras pertencem ao Bart, as próximas 61, 33, 30 e 24, ao Homer, Lisa, Meggie e Marge, respectivamente. Já no arquivo valida, ele contém menos dados, a ordem continuou a mesma, porém o valor mudou, 35, 25, 13, 12 e 10.

No Weka, o primeiro passo de todos é selecionar o arquivo de treinamento.arff na aba “Preprocess” → “Open a set of instances from a file”. Feito isso, na aba “Classify” é configurado os classificadores.

1. KNN: Precisa configurar para o tipo IBK. *Choose* → *Classifiers* → *Lazy* → *IBK*. Escolher o arquivo valida.arff para gerar os dados. Em *Test options*, selecionar *Supplied test set* → *Set* → *valida.arff*. Gerar uma saída de predictions, encontrada em More Options. Apertar Start e avaliar os resultados. Os predicts gerados são salvos em um arquivo.txt.predict para serem utilizados depois na fusão de classificadores.
2. Bayes Naive: Mesmos passos do KNN, porém alterado para tipo J48, em *Choose* → *Classifier* → *Bayes* → *Bayes Naive*.
3. Novamente, igual ao KNN, mas para o tipo randomForest. *Choose* → *tree* → *RandomForest*.

Por fim, é realizada a função de classificadores utilizando a regra da soma e do máximo. O arquivo **FusaoClassificadores_Kittler** possui 5 planilhas, uma com os dados do Knn.txt.predict, outra com BayesNaive.txt.Predict, com o RandomForest.txt.pred, e outras duas para a aplicação da Soma e Máximo. É necessário passar todos os dados dos

arquivos.txt.predict obtido do weka de cada classificador, e aplicar a fórmula da soma e máximo juntando todas as planilhas com valores dos três classificadores.

5.2.Resultados

Taxa de acertos dos classificadores estão da Figura 4, 5 e 6.

Correctly Classified Instances	41	41.4141 %
Incorrectly Classified Instances	58	58.5859 %
Kappa statistic	0.2224	
Mean absolute error	0.2381	
Root mean squared error	0.477	
Relative absolute error	78.3773 %	
Root relative squared error	122.6086 %	
Total Number of Instances	99	

Figura 4: Taxa de obtida K-NN

Correctly Classified Instances	34	34.3434 %
Incorrectly Classified Instances	65	65.6566 %
Kappa statistic	0.0569	
Mean absolute error	0.2748	
Root mean squared error	0.4007	
Relative absolute error	90.469 %	
Root relative squared error	102.9993 %	
Total Number of Instances	99	

Figura 5: Taxa de obtida RandomForest

Correctly Classified Instances	24	24.2424 %
Incorrectly Classified Instances	75	75.7576 %
Kappa statistic	0.0882	
Mean absolute error	0.3091	
Root mean squared error	0.5003	
Relative absolute error	101.7352 %	
Root relative squared error	128.602 %	
Total Number of Instances	99	

Figura 6: Taxa de obtida Bayes Naive

As matrizes de confusão são:

- K-NN
 - a b c d e <-- classified as
 - 19 5 2 7 2 | a = 0
 - 7 11 4 2 2 | b = 1
 - 6 2 3 3 0 | c = 2
 - 4 2 0 6 1 | d = 3
 - 2 3 0 4 2 | e = 4
- Random Forest
 - a b c d e <-- classified as
 - 26 6 0 2 1 | a = 0

13 6 2 4 1 | b = 1

12 1 0 1 0 | c = 2

9 3 0 0 1 | d = 3

6 3 0 0 2 | e = 4

- Bayes Naive

a b c d e <-- classified as

4 0 25 2 4 | a = 0

5 4 14 0 3 | b = 1

2 0 11 1 0 | c = 2

1 0 9 2 1 | d = 3

0 1 5 2 3 | e = 4

A taxa final usando as fusão de classificadores:

ACERTO	43,00000	14,33%
ERRO	56,00000	18,67%

Figura 7: Regra da soma

ACERTO	36,00000	12,00%
ERRO	63,00000	21,00%

Figura 8: Regra do Maximo

6. Conclusão

O objetivo do trabalho é definir características e determinar o acerto do classificador ao reconhecer os cinco personagens da família Simpsons, para isso são utilizadas ferramentas, como classificadores e fusão de classificadores. Por ser as cores como extração de características, a porcentagem de acerto se torna baixo, pois não é algo completamente específico de cada personagem. Outra característica visível, o Bart é o que mais obten acerto, e muitos classificadores ainda o considerava a melhor solução, mesmo sendo outro personagem. Isso pode acontecer por ele possuir um maior número de dados analisados, e muito de suas cores estão presentes nos outros Simpsons também. De forma geral, extrair características não é algo simples, e os resultados podem variar, para mais ou para menos, de acordo com o classificador escolhido. As carcterísticas devem sempre ser algo único, para a taxa de acerto ser sempre alta.

7. Referências

- [1] Definição de reconhecimento de padrões:
https://pt.wikipedia.org/wiki/Reconhecimento_de_padr%C3%B5es.