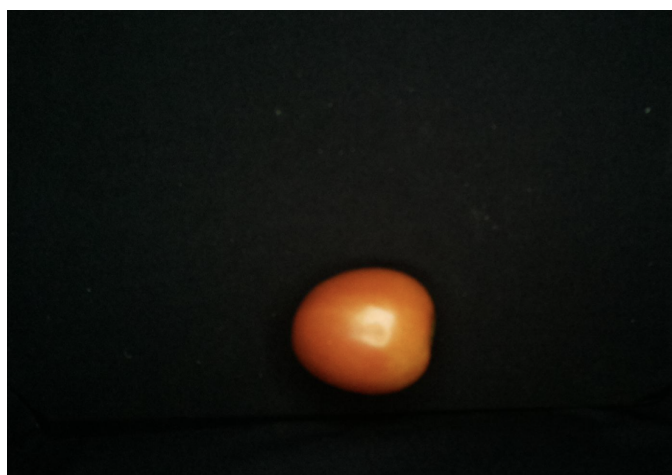


Aprendizagem Computacional

Relatório do Trabalho Prático 3



Degree of Maturity Recognition

2021/2022

Mestrado em Engenharia Informática

PL3
PL3

Pedro Rodrigues
Miguel Rabuge

2018283166
2018293728

pedror@student.dei.uc.pt
rabuge@student.dei.uc.pt

Índice

Dataset	2
Divisão treino/teste	3
Isolamento do Objeto	3
Representação das Features	3
Média por Componente RGB	3
Média em Escala de Cinza	3
Arquitetura do Fuzzy Inference System	4
Mamdani	4
Sugeno	4
Parameterização	4
Resultados Experimentais	4
Conclusões	5

1. Dataset

Neste capítulo descrevemos como foi dividido o dataset e como foi isolado o objeto.

1.1. Divisão treino/teste

A divisão treino/teste foi feita através de um rácio, nomeadamente 70% para treino e os restantes 30% para teste. Para cada uma das 6 classes, são retiradas 70% destas para treino e 30% destas para teste.

1.2. Isolamento do Objeto

O objeto foi isolado recorrendo à aplicação do Matlab, “*colorThresholder*”. Através desta *app* foi possível gerar uma função - “createMask” - que recebe uma imagem $H \times W \times 3$ e retorna uma matriz de booleanos, True onde está o objeto e False onde não está, e uma matriz 3D $H \times W \times 3$, onde o background, que se apresenta na imagem original com vários tons escuros, fica todo com a mesma cor, ou seja, inteiramente preto.

2. Representação das *Features*

Neste segundo capítulo descrevemos o pré-processamento efetuado às imagens para representar o conteúdo do objeto previamente isolado de diferentes formas.

2.1. Média por Componente RGB

Nesta primeira abordagem, cada imagem é reduzida a um vetor de tamanho 3 normalizado em $[0, 1]$. Cada célula deste vetor representa a média de cada uma das componentes RGB presentes na imagem, respetivamente.

O Fuzzy Inference System terá portanto 3 canais de input.

2.2. Média em Escala de Cinza

Nesta segunda abordagem, cada imagem é reduzida ao valor da média em escala de cinza, normalizada em $[0, 1]$, obtida pela seguinte fórmula:

$$Grayscale = 0.2989 \cdot R + 0.5870 \cdot G + 0.1140 \cdot B$$

Onde R, G e B são os valores médios por componente RGB, descrito no subcapítulo 2.1.

O Fuzzy Inference System terá portanto 1 canal de input.

3. Arquitetura do *Fuzzy Inference System*

Neste capítulo descrevemos as arquiteturas utilizadas nos *Fuzzy inference Systems*. Apresentamos também a metodologia através da qual são definidas as regras e alterados as funções de *membership* dos canais de input.

3.1. Mamdani

Para este Fuzzy Inference System utilizamos a seguinte arquitetura:

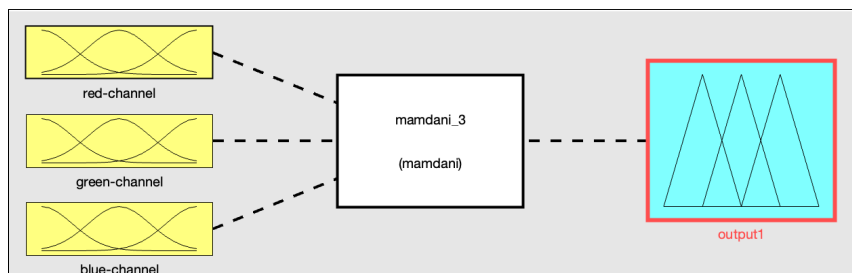


Figura 1 - Mamdani_3

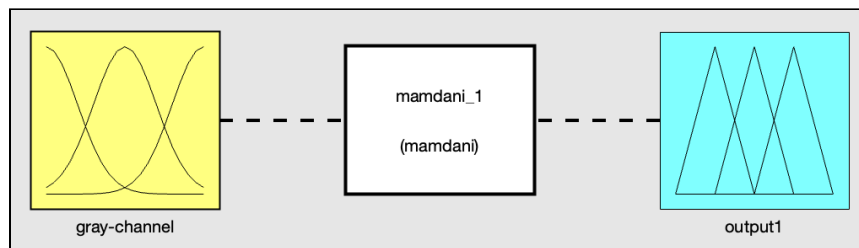


Figura 2 - Mamdani_1

Onde os inputs e outputs são:

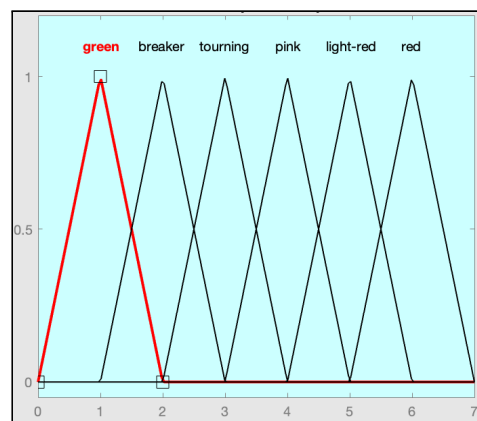
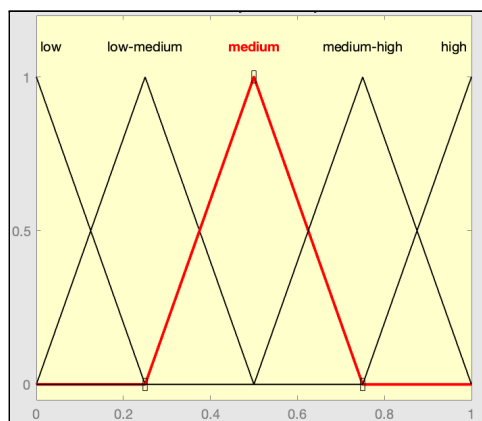


Figura 3(a) 3(b) - Inputs e Outputs

3.2. Sugeno

Para este Fuzzy Inference System utilizamos a seguinte arquitetura:

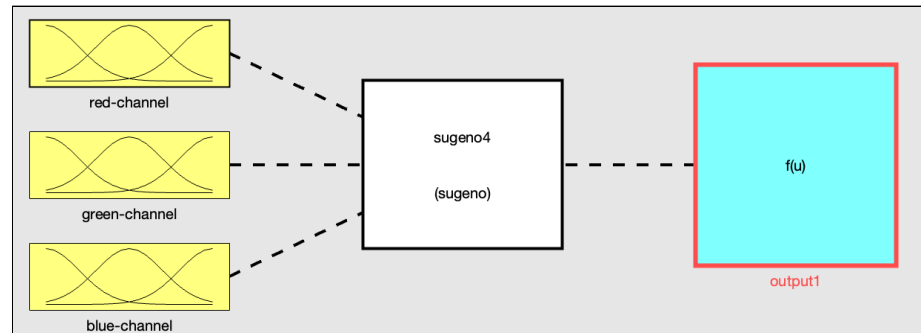


Figura 4 - Sugeno_3

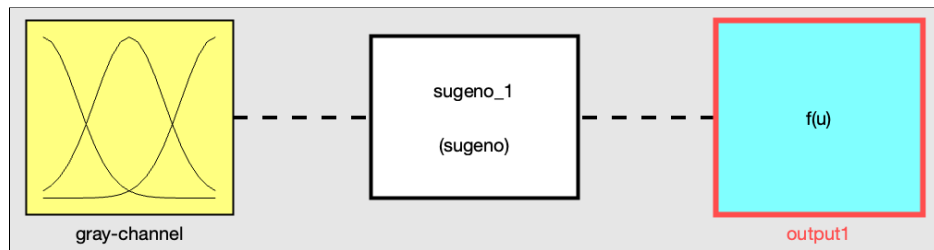


Figura 5 - Sugeno_1

Onde os inputs e outputs são:

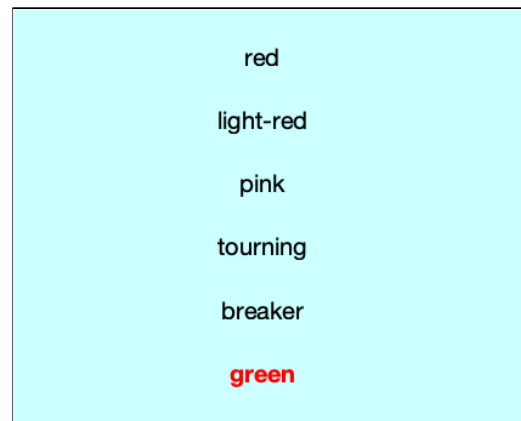
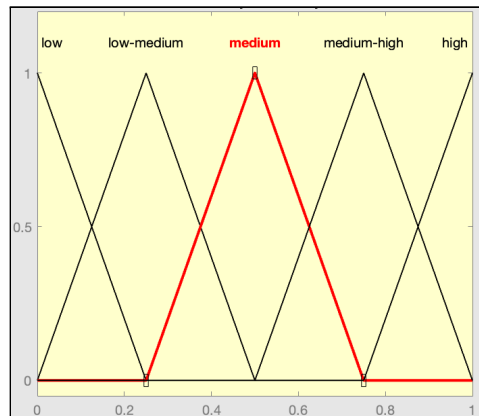


Figura 6(a) e 7(b) - Inputs e outputs

Nota: Existem os *FIS*_1 e *FIS*_3. A estrutura do input, do output e a forma de parametrização das regras e da membership function do input é exatamente a mesma. A única diferença está no número de canais de input, 1 e 3 respetivamente, dado que a representação de features numa escala de cinza utiliza apenas 1 canal, e a média das componentes RGB utiliza 3.

3.3. Parameterização

Em termos de parametrização das funções de *membership* e das regras, é utilizada a função do matlab *tunefis*. Esta função irá, através de um algoritmo genético, adaptar a estrutura das funções de *membership* e definir as regras que entender necessárias. Todos os Fuzzy Inference Systems abaixo são parametrizados desta forma, utilizando 50 como o número máximo de gerações.

4. Resultados Experimentais

Representação	Grayscale		RGB	
Arquitetura	Mamdani_1	Sugeno_1	Mamdani_3	Sugeno_3
Overall	0.53	0.51	0.56	0.52
Classe 1	0.49	0.32	0.54	0.27
Classe 2	0.52	0.64	0.48	0.48
Classe 3	0.42	0.39	0.47	0.53
Classe 4	0.54	0.44	0.66	0.63
Classe 5	0.76	0.83	0.79	0.63
Classe 6	0.09	0.09	0.00	0.36

Tabela 1 - Resultados Experimentais - Accuracy - Max 9 Rules

Representação	Grayscale		RGB	
Arquitetura	Mamdani_1	Sugeno_1	Mamdani_3	Sugeno_3
Overall	0.47	0.55	0.49	0.57
Classe 1	0.73	0.43	0.08	0.54
Classe 2	0.43	0.74	0.55	0.33
Classe 3	0.22	0.33	0.58	0.67
Classe 4	0.59	0.49	0.59	0.58
Classe 5	0.50	0.82	0.63	0.68
Classe 6	0.14	0.09	0.27	0.59

Tabela 1 - Resultados Experimentais - Accuracy - Max 25 Rules

5. Conclusões

- ❖ Que arquitetura apresenta melhores resultados?
 - Ambas as arquiteturas apresentam resultados de *accuracy* semelhantes, com base nas experiências realizadas, de modo que não é possível concluir qual arquitetura apresenta melhores resultados.
- ❖ Como é que os dados e a respetiva representação afetou a performance do sistema?
 - A representação em RGB (3 features) apresentou, de forma geral, melhores resultados em *accuracy* que a de Escala de Cinza (1 feature), o que faz sentido, dado que com a representação RGB, o sistema tem mais informação.
- ❖ O sistema de reconhecimento conseguiu atingir os objetivos?
 - Para algumas classes, existe uma performance bastante significativa, entre cerca de 70% a 80%, dependendo da representação e do sistema, pelo que acreditamos que o sistema de reconhecimento atingiu os objetivos nestas. No entanto, no geral, não, dado que a média de *overall* de teste ronda os 50%.
- ❖ Qual é a capacidade de generalização? Varia consoante a classe a classificar?
 - Em concordância com o que foi enunciado na questão acima, a capacidade de generalização dos sistemas treinados não é grande, dado que se fica pelos 50% de *overall accuracy*. Deste modo, o sistema não é robusto, apesar da *accuracy* variar mediante a classe a ser classificada (a classe 5 consegue ser relativamente bem classificada).