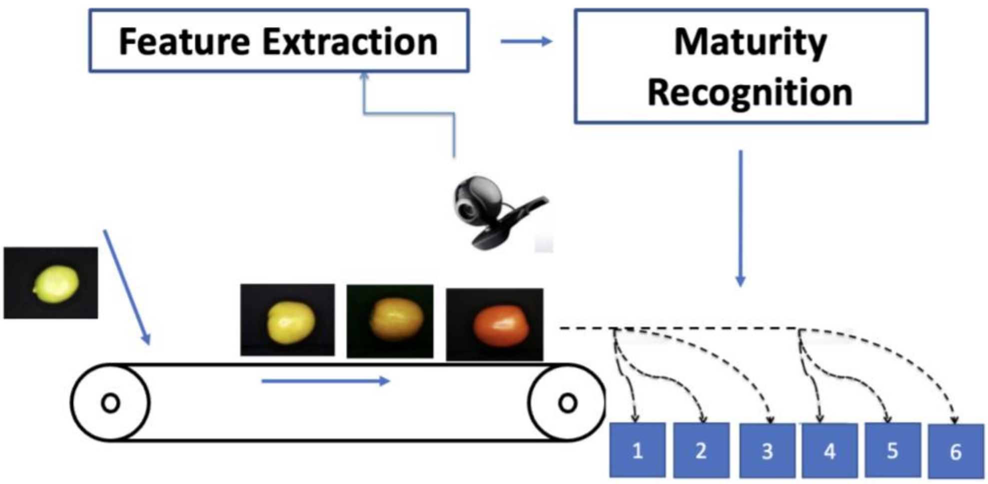
Aprendizagem Computacional / Machine Learning

Assignment TP3: Degree of Maturity Recognition

Authors: Hugo Bronze Canelas de Brito Prata - 2014198526

Luigi Masi – 2021195907



Summary

1. Introduction

The main goal of this work is to develop and implement two types of fuzzy inference system that recognizes the degree maturity of a crop.

1) Mamdani
   - With 9 rules

1. If (G is greenlow) then (CLASS is 6) (1)
2. If (G is green_midlow) then (CLASS is 5) (1)
3. If (G is green_mid) then (CLASS is 4) (1)
4. If (G is green_mid_high) then (CLASS is 3) (1)
5. If (G is green_high) then (CLASS is 2) (1)
6. If (G is green_very_high) then (CLASS is 1) (1)
7. If (R is Red_mid) and (G is greenlow) then (CLASS is 6) (1)
8. If (G is green_high) and (B is blue_mid) then (CLASS is 1) (1)
9. If (R is Red_mid) and (B is blue_low) then (CLASS is 6) (1)

   - With 25 rules

1. If (G is green_very_high) and (B is blue_mid) then (CLASS is 1) (1)
2. If (G is green_very_high) and (B is blue_high) then (CLASS is 1) (1)
3. If (G is green_high) and (B is blue_mid) then (CLASS is 2) (1)
4. If (G is green_high) and (B is blue_high) then (CLASS is 2) (1)
5. If (G is green_mid_high) and (B is blue_high) then (CLASS is 2) (1)
6. If (R is Red_high) and (G is green_mid_high) and (B is blue_mid) then (CLASS is 3) (1)
7. If (R is Red_high) and (G is green_mid_high) and (B is blue_high) then (CLASS is 3) (1)
8. If (G is green_mid) and (B is blue_high) then (CLASS is 4) (1)
9. If (R is Red_mid) and (G is green_mid) and (B is blue_mid) then (CLASS is 4) (1)
10. If (G is green_mid) and (B is blue_low) then (CLASS is 5) (1)
11. If (R is Red_high) and (G is green_midlow) and (B is blue_mid) then (CLASS is 5) (1)
12. If (R is Red_mid) and (G is green_midlow) and (B is blue_high) then (CLASS is 5) (1)
13. If (R is Red_low) and (G is green_midlow) and (B is blue_low) then (CLASS is 6) (1)
14. If (R is Red_mid) and (G is greenlow) and (B is blue_low) then (CLASS is 6) (1)
15. If (R is Red_mid) and (G is greenlow) and (B is blue_high) then (CLASS is 5) (1)
16. If (R is Red_low) and (G is greenlow) and (B is blue_mid) then (CLASS is 6) (1)
17. If (R is Red_high) and (G is greenlow) and (B is blue_high) then (CLASS is 5) (1)
18. If (R is Red_high) and (G is green_midlow) and (B is blue_high) then (CLASS is 5) (1)
19. If (R is Red_high) and (G is green_mid) and (B is blue_mid) then (CLASS is 4) (1)
20. If (R is Red_low) and (G is green_mid) and (B is blue_mid) then (CLASS is 4) (1)
21. If (R is Red_mid) and (G is green_mid_high) and (B is blue_high) then (CLASS is 3) (1)
22. If (R is Red_mid) and (G is green_mid_high) and (B is blue_mid) then (CLASS is 3) (1)
23. If (R is Red_mid) and (G is green_mid_high) and (B is blue_low) then (CLASS is 3) (1)
24. If (R is Red_high) and (G is green_mid_high) and (B is blue_low) then (CLASS is 3) (1)
25. If (R is Red_low) and (G is green_very_high) and (B is blue_mid) then (CLASS is 1) (1)

2) Sugeno

- With 9 rules

1. If (G is greenlow) then (Class is 6) (1)
2. If (G is green_midlow) then (Class is 5) (1)
3. If (G is green_mid) then (Class is 4) (1)
4. If (G is green_mid_high) then (Class is 3) (1)
5. If (G is green_high) then (Class is 2) (1)
6. If (G is green_very_high) then (Class is 1) (1)
7. If (R is red_mid) and (G is greenlow) then (Class is 6) (1)
8. If (G is green_high) and (B is blue_mid) then (Class is 1) (1)
9. If (R is red_mid) and (B is blue_low) then (Class is 6) (1)

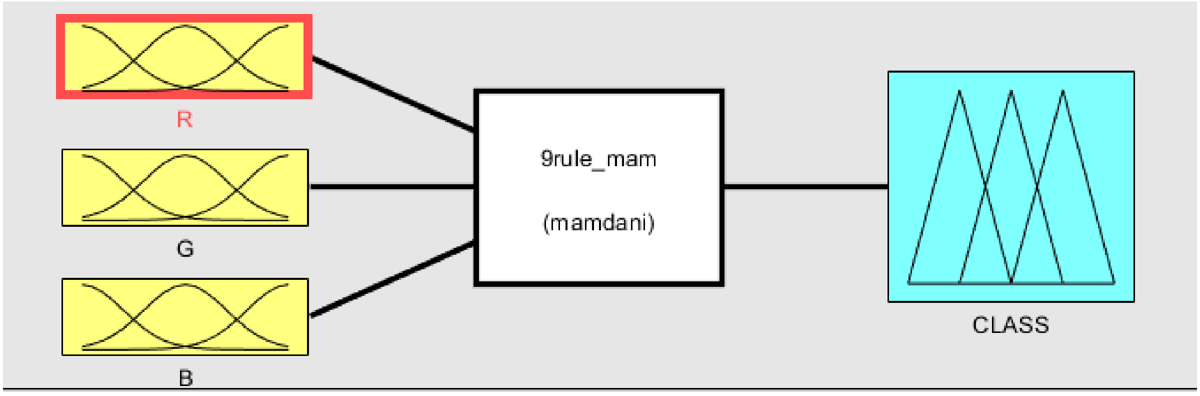
- With 25 rules

1. If (G is green_very_high) and (B is blue_mid) then (CLASS is 1) (1)
2. If (G is green_very_high) and (B is blue_high) then (CLASS is 1) (1)
3. If (G is green_high) and (B is blue_mid) then (CLASS is 2) (1)
4. If (G is green_high) and (B is blue_high) then (CLASS is 2) (1)
5. If (G is green_mid_high) and (B is blue_high) then (CLASS is 2) (1)
6. If (R is Red_high) and (G is green_mid_high) and (B is blue_mid) then (CLASS is 3) (1)
7. If (R is Red_high) and (G is green_mid_high) and (B is blue_high) then (CLASS is 3) (1)
8. If (G is green_mid) and (B is blue_high) then (CLASS is 4) (1)
9. If (R is Red_mid) and (G is green_mid) and (B is blue_mid) then (CLASS is 4) (1)
10. If (G is green_mid) and (B is blue_low) then (CLASS is 5) (1)
11. If (R is Red_high) and (G is green_midlow) and (B is blue_mid) then (CLASS is 5) (1)
12. If (R is Red_mid) and (G is green_midlow) and (B is blue_high) then (CLASS is 5) (1)
13. If (R is Red_low) and (G is green_midlow) and (B is blue_low) then (CLASS is 6) (1)
14. If (R is Red_mid) and (G is greenlow) and (B is blue_low) then (CLASS is 6) (1)
15. If (R is Red_mid) and (G is greenlow) and (B is blue_high) then (CLASS is 5) (1)
16. If (R is Red_low) and (G is greenlow) and (B is blue_mid) then (CLASS is 6) (1)
17. If (R is Red_high) and (G is greenlow) and (B is blue_high) then (CLASS is 5) (1)
18. If (R is Red_high) and (G is green_midlow) and (B is blue_high) then (CLASS is 5) (1)
19. If (R is Red_high) and (G is green_mid) and (B is blue_mid) then (CLASS is 4) (1)
20. If (R is Red_low) and (G is green_mid) and (B is blue_mid) then (CLASS is 4) (1)
21. If (R is Red_mid) and (G is green_mid_high) and (B is blue_high) then (CLASS is 3) (1)
22. If (R is Red_mid) and (G is green_mid_high) and (B is blue_mid) then (CLASS is 3) (1)
23. If (R is Red_mid) and (G is green_mid_high) and (B is blue_low) then (CLASS is 3) (1)
24. If (R is Red_high) and (G is green_mid_high) and (B is blue_low) then (CLASS is 3) (1)
25. If (R is Red_low) and (G is green_very_high) and (B is blue_mid) then (CLASS is 1) (1)

2. Data set and feature representation

Concerning the influence of the data set and the feature representation on the performance of the recognition system, we noticed that the pre-processing of data is absolutely necessary due to the differences about lighting, maturity, shape and position of the fruit in the pictures. The images needed to be cropped, rescaled, and then evaluated on a by pixel case. In our case, we also need to remove outliers within a class. If any step on the pre-processing is neglected, then the results are compromised. The evaluation is based on colour averages, and so any variation, such as smaller sized images (less pixels) or images with more black area than others, will affect the results as these averages are how we train the classifier.

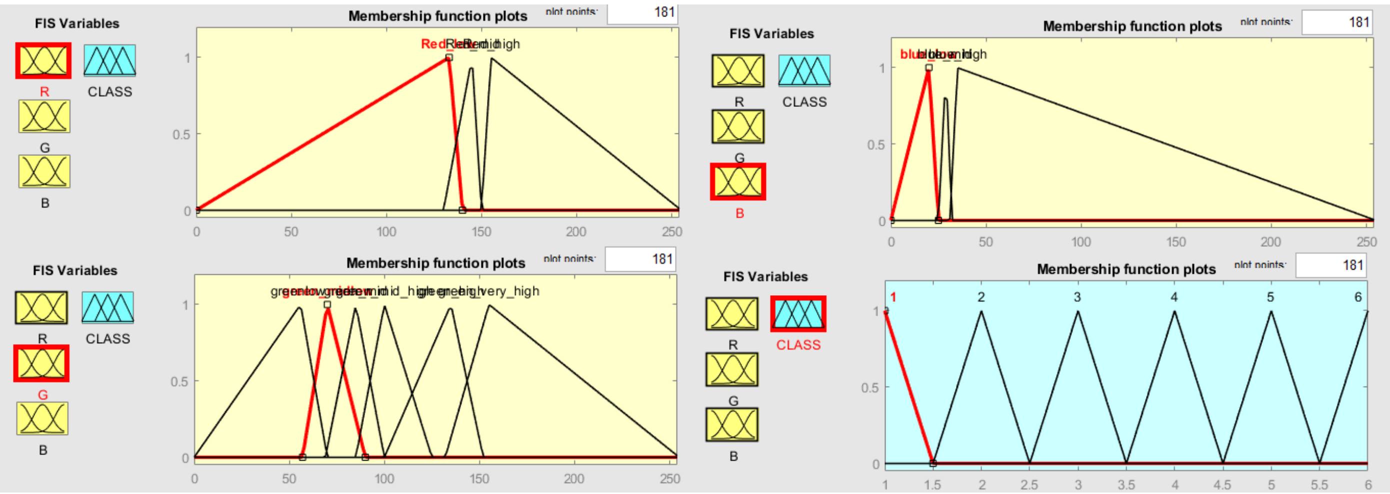
The representation of each class and feature within the architecture also matters. Each class is a simple number [1,6], and each feature (R, G and B colours) is a value [0,255]. R, G and B are separated inputs.



After a quick analysis, we can see that G has the greatest variation of the three. Because of this, we decided to give more MF's to green: 6 in total, one for each class. We decided to give 3 to R and B: Low, Medium and High. This was because R has very low variation, that is only noticeable on the 6th class (155-132), and because B has low influence in the maturity colour, and although it varies more between its min and max, it does not have a big effect overall due to the low values.

```
CLASS 1
AVG
  148.8506   152.5977    30.1379

MAX
   168    172     68

MIN
   116    131     13

CLASS 2
AVG
  153.5612   137.0102    33.1429

MAX
   182    167     72

MIN
   128     96     12

CLASS 3
AVG
  153.3452    99.7857    30.4405

MAX
   181    144     67

MIN
   121     74      3

CLASS 4
AVG
  154.1971    85.1314    33.5328

MAX
   170    110     59

MIN
   115     64      4

CLASS 5
AVG
  151.0000    70.3904    26.5137

MAX
   165     90     49

MIN
   119     43      5

CLASS 6
AVG
  132.5833    54.6042    18.9792

MAX
   153     74     39

MIN
    92     41      4
```

We used these values to define the range of each MF. As before, if these values are not accurate, then the classifier will not be able to output the correct answers as this is basically how we train this kind of classifier.



All our classifiers used the same MF values to keep the statistical analysis of the results fair. We tried several values for the MFs. We calculated the min, max, and standard deviation of each class. We attempted to tinker with the MF values using these values and came to the conclusion that the best result came from using the standard deviation to calculate an approximation of the left and right ranges of each MF.

We first attempted to create the MFs with gaussian functions, but it proved to be ineffective as the curves were too similar and the classifier couldn't separate each class successfully. We settled for the triangular function, as it proved to be the easier to work with and to create valid results.

3. FIS Architecture

By comparing the two types of architecture, we concluded that Sugeno yielded better results overall. Sugeno FIS possesses more flexibility in the system design and it is well suited for mathematically analysis. Sugeno has a different output MF. Mamdani is more suited for user input rather than mathematical analysis (such as in our case, with the analysis of colour averages being a mathematical analysis) and it is less efficient computation-wise compared to Sugeno. Overall, we found that the main differences between them were the following:

Mamdani FIS

- Output membership function is present
- Crisp result is obtained through defuzzification of rules' consequent
- Non-continuous output surface
- MISO (Multiple Input Single Output) and MIMO (Multiple Input Multiple Output) systems
- Expressive power and Interpretable rule consequents
- Less flexibility in system design

Sugeno FIS

- No output membership function is present
- No defuzzification: crisp result is obtained using weighted average of the rules' consequent
- Continuous output surface
- Only MISO systems
- Loss of interpretability
- More flexibility in system design

On the 9 vs 25 rules architecture: 25 rules is the maximum one usually needs on average for any fuzzy system.

During the work, we found out a formula that gave us a suggestion about an optimal total number of rules

$$Number\ of\ Rules = \frac{Number\ of\ inputs\ \cdot Number\ of\ variables}{Number\ of\ MF\ per\ input}$$

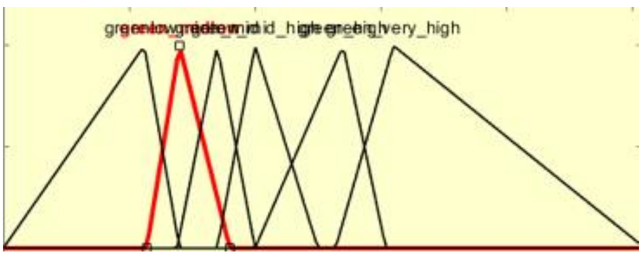
In our case, this gave us a number closer to 9 rather than 25, a total of 6. Because of this, we felt that 25 rules was excessive and that it would overcomplicate the classification. **“no free lunch theorem”.** In theory more rules should help us make more precise rules instead of general rules. But we were unable to.

4. Results

Our results show us that it is easier to adjust the classification of extremes, such as *Very Green/Green* and *Very Mature/Mature*. Our classifiers had a general difficulty in classifying classes closer to the centre of maturity (Pink, Tourning). This could be due to the lesser variation of colour in these middle stages, and similarity to extreme classes, and so an higher difficulty in classifying the fruit was seen. We also believe the dataset is divided into too many classes, as for example, there are pictures within class 4 that to the human eye would effectively be class 5 or even 6, “Muestra11_3” and others. We believe that 4 classes would be more appropriate, should we have a say in the creation of the dataset.

Since our approach involves averages of several fruits, the classifiers themselves are robust enough to receive different inputs and to maintain the average percentage of correct answers, so long as the same pre-processing is applied beforehand.

We believe these types of classifiers are better suited for singling out a lower number of classes or classes with more distinct features, rather than a general classification of many classes with similar averages. This affected the results by giving a maximum average of 60% correct answers. Although, the best results for a given class was 95% for *Very Mature* and 84% for *Very Green*, again proving that the classification is easier for extreme classes such as *Very Green* and *Very Mature*, as the classes after/prior to that rank were then classified with less precision (50s).



9 rules fared better, by being more concentrated on the variation of green, while 25 rules overcomplicated the problem and yielded worst results even with the additional tweaking of the classification given by having more rules available.

We believe better results would be possible, by either focusing on less classes, or by tweaking the colour value averages better. There are also other approaches to Fuzzy Logic which could have been more successful. One of them would be tunfis, which is able to generate rules which may be more effective than the rules created manually though the app.

We created several rules that included one, two or the three colour values, but the best results always came from rules which focused on green and to some extent the lowest and largest value of red.

```
 9rule_mam
CLASS 1 correct predictions: 54 %
CLASS 2 correct predictions: 57 %
CLASS 3 correct predictions: 33 %
CLASS 4 correct predictions: 51 %
CLASS 5 correct predictions: 73 %
CLASS 6 correct predictions: 35 %
Average correct predictions: 51 %

 9rule_sug
CLASS 1 correct predictions: 62 %
CLASS 2 correct predictions: 48 %
CLASS 3 correct predictions: 39 %
CLASS 4 correct predictions: 53 %
CLASS 5 correct predictions: 61 %
CLASS 6 correct predictions: 90 %
Average correct predictions: 59 %
```

```
 25rule_mam
CLASS 1 correct predictions: 51 %
CLASS 2 correct predictions: 59 %
CLASS 3 correct predictions: 31 %
CLASS 4 correct predictions: 54 %
CLASS 5 correct predictions: 45 %
CLASS 6 correct predictions: 45 %
Average correct predictions: 48 %

 25rule_sug
CLASS 1 correct predictions: 54 %
CLASS 2 correct predictions: 55 %
CLASS 3 correct predictions: 47 %
CLASS 4 correct predictions: 27 %
CLASS 5 correct predictions: 37 %
CLASS 6 correct predictions: 85 %
Average correct predictions: 51 %
```