Simone Quadrelli

Machine learning project

Academic year 2018 - 2019

Contents

List of Tables

List of Figures

Abstract

1 Introduction

short abstract: what are your going to present in the report
statement of the problem/goal of the analysis and description of the data set(s)
list of three to five findings/keypoints
the analysis with wise commentary
(optional) theoretical background of the used methods
conclusions (should include the findings/keypoints)
the Appendix, containing all the R code

The present project addresses the fruit and vegetable multiclass classification problem. Its objective of this project is to classify a variety of fruits and vegetables whose images are stored in a recently released dataset.

Fruit and vegetable classification is a challenging task, there are two different kind of problems that must be overcome: vegetables and fruits within the same class can differ in shape and colour due to maturity and growth. On the countrary, vegetables and fruits in different classes, such as red apples or grapes, may be very similar both in textures and in colours. Fruit and vegetable classification has a key role in many industries: it can be exploited to make agriculter more autonomous than ever before in history, but it can also be applied to all stages of the supply chain to assess the quality of the products from the producer to the consumer. The ability of harvesting robots can be enhanced by enforcing fruit and vegetables recognition, indeed it may allow them to recognize not only the type of fruit or vegetable but also its maturity or wheter it is rotten. Moreover, autonomous classification provides more consistent quality assessment since it does not depend of the opinion of different workers.

The projects will analyze the classification performance of three machine learning algorithms on different fruits. The k-nearest neighbour (knn) algorithm was used to compute a baseline, a more sophistocated, the support vector machines (SVM) algorithm was later used for classification. The last approach was to construct a convolutional neural network becausethey are knoen to suit well the image classification. The are two different kind of features that were extracted to feed the algorithms: the grayscale and the rgb representation of the images.

The objective of the project are

- analyze the classification performance of the knn and svm using the grayscale on a variety of fruits
- analyze the classification performance of the knn and svm using the grayscale on a variety of fruits
- analyze the classification performance of the knn and svm using the rgb sacle on a variety of fruits

- analyze the classification performance of the knn and svm using the rgb scale on a variety of fruits

Descrivere cosa si trova nelle sezioni successive

2 Dataset and Features

The analyzed dataset, known as *Fruits 360*, was originally published in 2017 by Horea Muresan and Mihai Oltean to address fruit and vegetable classification [1]. The existence of either too small datasets or too low-quality dataset was most compelling problem. On the countrary, this dataset provides more that 70000 images of 114 different fruits and vegetables¿ Each image has a definition of 100 × 100 pixels. To increase the reliability of the results, it contains images of rotated fruits and vegetables. However those images make the training process much harder . As the authors explain, the dataset was produced as follows: fruits and vegetables were planted in the shaft of a low speed motor (3 rpm) and a short movie of 20 seconds was recorded. However, due to the variations in the lighting conditions, the background was not uniform and a dedicated algorithm was exploited to extract the fruit from the background. The images were of such high quality that no preprocessing was required. It is also worth noticing that the dataset was already splitted in train and test set, therefore no further splitting was required.
Fruits and vegetables have several distinct visual characteristics that can be extracted from the images. Colour, shape, size and texture are the most commonly used features[1] in image classification. Among all the possible feauters that can be extracted from an image, colours features play a key role in fruit and vegetable classification. Indeed, they seem to suit the domain and their computation is very fast and efficient [2]. The project focuses on the grayscale and rgb representation of the images.
The following lines introduce the reader to some concept of colour spaces and colour representation. Colour spaces are a specific organization of colors the allows to reproduce the representation of colours. Two different colour spaces were exploited in this project: grayscale and the rgb.

2.1 RGB colour space

The RGB colour model is an additive color model, meaning that red, green and blue are added together to reproduce an wide array of colors. A color in the RGB colour model is described by indicating how much of the red, green, and blue is included. The colour of a pixel is expressed as an RGB triplet (R,G,G). Each component can vary from 0 to 255, since each channel is represented by a byte.

[1] Features are the characteristics of an object that can distinguish it from other objects

2.2 Grayscale colour space

Since the dataset contains rgb images, it was necessary to extraxt a grayscale representation of the pixels to be used as features. In grayscale representation each pixel is a byte, meaning that the possible values it can be in range from 0 to 255.
The conversion algorithm implemented by the library *imager* is the Luma algorithm. Each rgb channel R, G, B of a pixel are passed to a function $\Gamma(x)$ to obtain its transformation, respectively R', G', B'.
The Γ function is defined as:

$$x' = \Gamma(x) = Ax^\gamma \tag{1}$$

where x in the pixel intensity, A is a scalar whose value is often 1 and γ is usually $\frac{1}{2.2}$.
To obtain the final grayscale representation y of a pixel x, the following linear transformation is applied [3]

$$y = 0.21R' + 0.71G' + 0.07B' \tag{2}$$

3 Data analysis

This section provides the reader an insight of the procedure used to analyze the data, it also provides a commantary of the results.
The procedure described in the section is the same for both the grayscale featires and the rgb features, however the differ in the topology of the convolutional neural network. Algorithm **??** provides a brief overview of the analysis.

Algorithm 1 Schema of the analysis

```
Feature extraction
Dimensionality reduction
Extraction of the most significant features
Construction of a baseline using knns
Evaluation of svm
CNN
```

3.1 Feature extraction and dimensionality reduction

The first major step in the analysis is the data extraction and the computation of the data matrix and the vector of the labels concerning the following fruits: *Apple Golden 1*, *Apple Golden 2*, *Apple Red 2*, *Apple Red 1*, *Banana*, *Apricot*, *Blueberry*.
As previously stated, the dataset contains 100x100 pixels images, therefore the data matrix with rgb features has 409440000 elements. Bearing that in mind, the images are rescaled to 50×50 pixels to reduce the computational complexity of the problem. The feature vector of each image is extracted by the functions

of the library *imager*. Once all the data vectors are extracted, a data matrix is created. Each row of the matrix contains the linearized representation of the image. At the same time a vector of labels is computed.
Most learning algorithms are exponentially slow in the number of the feature or at least significantly slow down as the number of feature increases. Furthermore, it is reasonable to belive that many of the pixels are useless since they are just part of the background. As a result, Principal component analysis (PCA) is exploited to reduce the dimensionality of the data and select the most promising features. PCA is one of the most common technique to reduce the dimensionality of the data since it is very efficient and many libraries implement it. PCA computes all the principal components that can be seen as the axis along which data differ the most.
When using the grayscale features the execution time required to compute the PCA is 5 minutes and 30 seconds. However, the execution time required to compute the PCA is about 57 minutes for grb features[2].

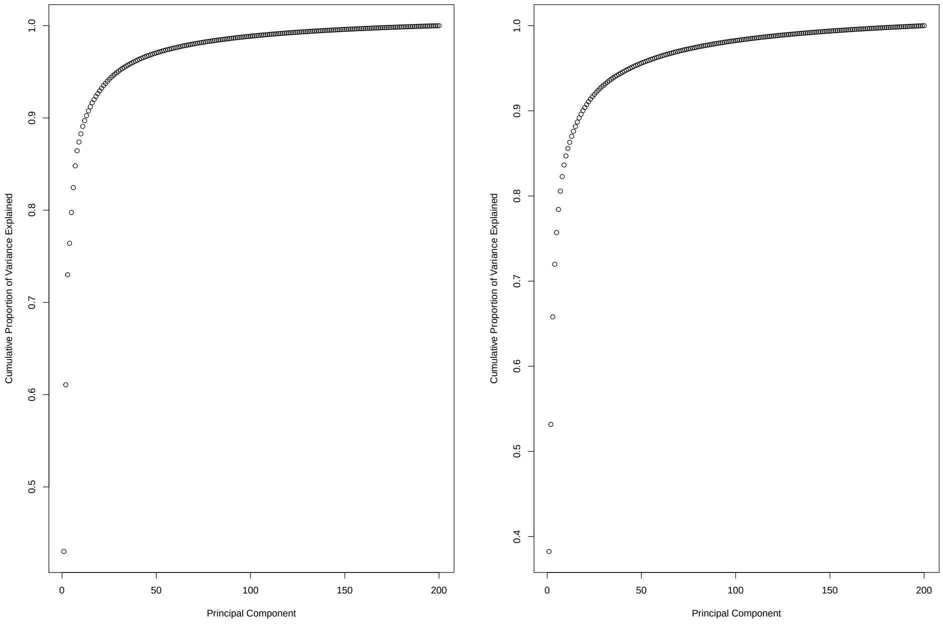

Figure 1: Cumulative variance explaind by the principal components: the figure on the left concerns grayscale feature, the figure on the right concerns rgb features

As figure **??** show, in both cases, using 150 is enough to explain all the variance

[2] All time measurement rely on a Intel Core i5-7200U dual core processor and 8 GB DDR4 RAM

in the data. However, having fixed the number of compunents, rgb components explain less variance than the grayscale components. To compare and analyze the performance of the learning algorithms the following set of principal components $\{3, 10, 50, 100, 150\}$ are used. The rationale behind this choice is to select the number of components that should give the best result having at the same time a fast execution time.

3.2 Baseline and knn

The k-nearest neighbours (knn) algorithm seems to be an appropriate choice to define a baseline because it is known to be one of the most efficient classification algorithm, even if it is very simple. Indeed, the knn classifies a test point with the labels of the closest training point. Moreover, in image classification it seems to perform quite as well as more sophisticated algorithms such as support vector machines. [4]

To evaluate the performace of the knn and the other algorithms, the most natural choice is the accuracy, defined as:

$$\text{accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{3}$$

The number of neighbours (k) chosen to construct the baseline are: $1, 3, 7, 9, 11$. As can be seen from figure **??**, there is no remarkable diffenrence in performace if the number of features exploited is 10 or more. On the other hand, when only three features are used the algorithm achieves lower performances. This difference is even more sharp when rgb features are involved [3], indeed the accuracy lies about 0.10 below the other accuracy lines.

The best number of features to select is 50, since with all the possible value of k knn performs best with 50 features.

If we limit our analysis, considering just the result of knn with at least 10 features 1-nn has the very best results, even if up to 7 neighbours there is no significant difference in performance. Overall, knn reaches a very significant accuracy: the accuracy level when considering rgb features reach up to 0.91, while the accuracy level reached using grayscale features is about 0.82. As expected the rgb features are more reliable and are associated with better results.

Overall, the best model is 1-nn with 50 features.

[3] The figure on the left always refers to the result obtained using the grayscale features. The figure on the right refers to the result obtained using the rgb features.

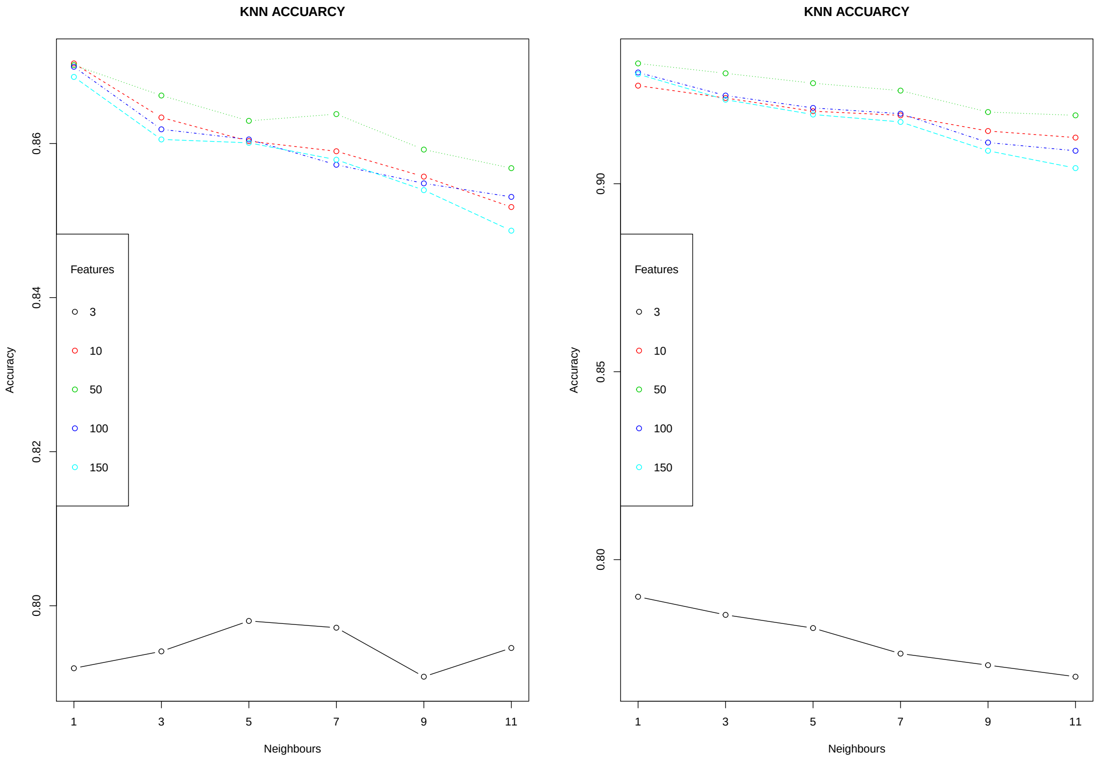

Figure 2: Test accuracy knn

The execution time of the algorithm is acceptable and does not exceed 30s. It is worth noting that the execution time, given the number of features, have just neglegible variation when the number of neighbours changes.

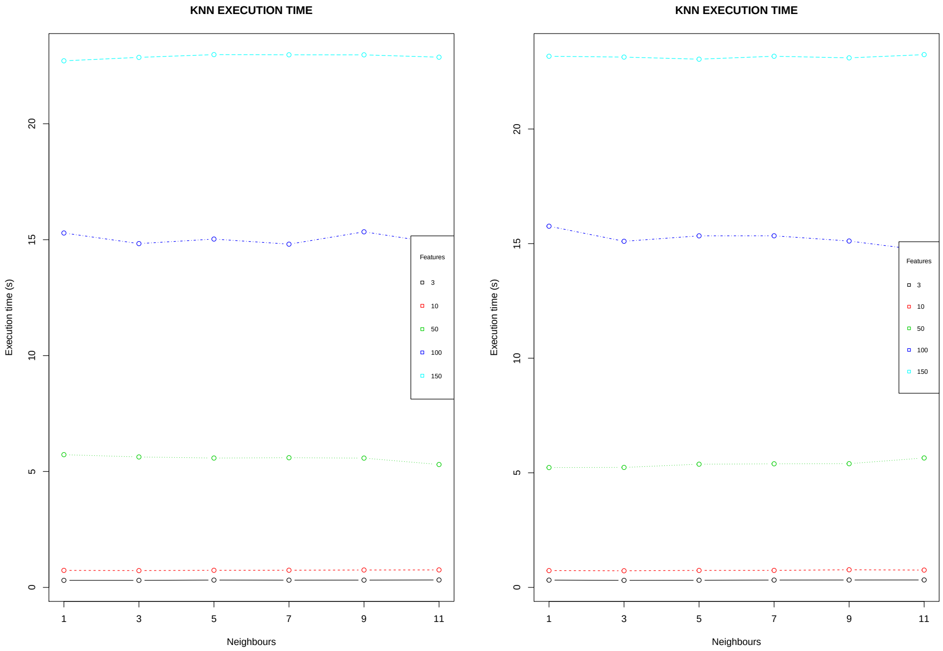

Figure 3: Execution time knn

Overall, the baseline is very promising but hard to improve. The knn algorithm seems to be very suitable to classify images.

3.3 SVM

Support vector machines are often used in image classification. Thier most basic implamentation is based on a linear classifier but the dataset seems to be hardly linearly separable. Therefore the gaussia kernel was applied to allow generic curves to be used. The kernel that was used, is the gaussian kernel. The gaussian kernel can be seen as a linear combiantion of polynomial kernels, therefore it can be seen as the most genral kernel implemented in the library. The kernel allow for more accurate predictions but slows a little bit down the algorithm execution time.
The plot on the left show the accuracy of the svm on the grayscale features. The model reaches a lower accuracy with respect to the accuracy of the knn algorithm. It is less precise by 0.02/0.03. On the countrary svm using rgb features performs as well as the knn algorithm.
In both cases the best accuracy, as before, is reached when the number of features is 50.

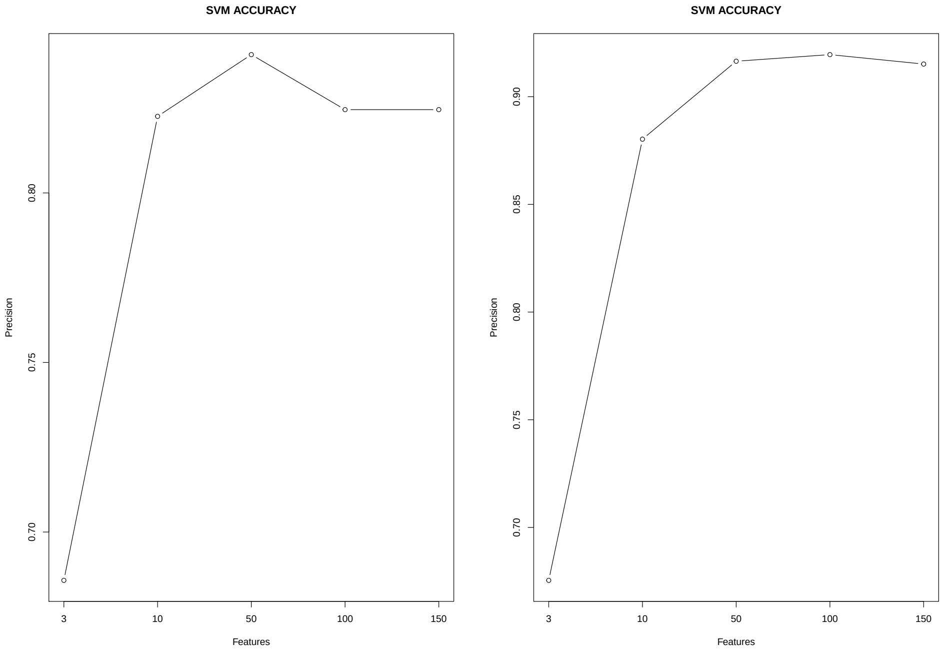

Figure 4: Test accuracy svm

The execution time of the svm is way higher than the execution time of the knn, however it remains a reasonable execution time.

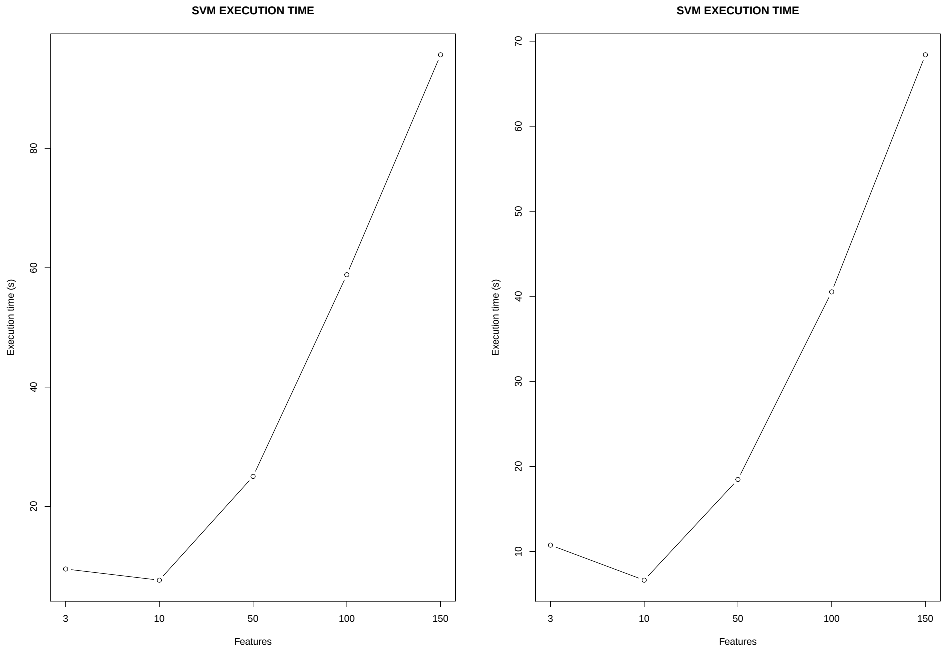

Figure 5: Execution time svm

Overall, the svm does not improve the result defined in the baseline that remains still very significant. Only a negliglible result is reached when using the rgb features.

3.4 CNN

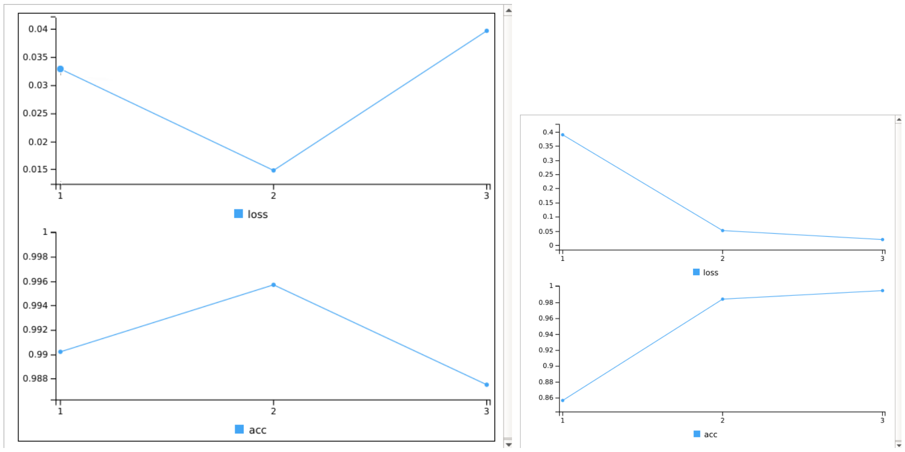

Figure 6: Execution time svm

4 Conclusions

RISULTATI MIGLIORI CONFRONTO CON LA LETTERATURA

5 Appendix: Code

References

[1] *Horea Muresan, Mihai Oltean, Fruit recognition from images using deep learning, Acta Univ. Sapientiae, Informatica Vol. 10, Issue 1, pp. 26-42, 2018.*

[2] *Khurram Hameed, Douglas Chai, Alexander Rassau, A comprehensive review of fruit and vegetable classification techniques. Image and Vision Computing 80 (2018) 22-44.*

[3] *Samuel Macedo, Givanio Melo, Judith Kelner, A comparative study of grayscale conversion techniques applied to SIFT descriptors, SBC Journal on Interactive Systems, volume 6, number 2, 2015.*

[4] *Sandeep Kumar, Zeeshan Khan, Anurag jain, A Review of Content Based Image Classification using Machine Learning Approach, International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970) Volume-2 Number-3 Issue-5 September-2012*