Sweets Classification Algorithm Using Computer Vision and Machine Learning

David Medina Ospina[1]

ABSTRACT ***Abstract*— This technical report describes an specific implementation for a computer vision algorithm that processes a set of sweets images and classifies them using machine learning. Also, it shows the main challenges with this implementation and how those were partially solved**

I. INTRODUCTION

All technical and theoretical aspects about this computer vision algorithm implementation were synthesized in this document, along with all the specifications about the machine learning classification algorithm and the reasons why it was chosen and how it was implemented.

For the purpose of this solution, python was choose as the programming language, opencv for the image processing and scikitlearn for the machine learning algorithm

II. PROBLEM DESCRIPTION

Computer vision is one of the most important innovations of this time, it helps computers understand images and what are in them. Classify objects based on images is a common problem both for the industry and the academy, it requires processing the images, converting them into a data structure and train a machine learning classification algorithm to categorize what is in them specifically.

In this specific implementation, 10 types of sweets were used to train the algorithm. The program must count the number of products and prompt the category they belong to. The categories are:

Fig. 1: List of sweets to classify

- `nr_jet_azul` : Total number of Jet Azul
- `nr_flow_negra` : Total number of Jumbo Flow Negra
- `nr_flow_blanca` : Total number of Jumbo Flow Blanca
- `nr_jumbo_naranja` : Total number of Jumbo Mix Naranja
- `nr_jumbo_roja` : Total number of Jumbo Roja
- `nr_chocorramo` : Total number of Chocorramo
- `nr_frunas_verde` : Total number of Fruna Verde
- `nr_frunas_naranja` : Total number of Fruna Naranja
- `nr_frunas_roja` : Total number of Fruna Roja
- `nr_frunas_amarilla` : Total number of Fruna Amarilla

III. SOLUTION

The problem was divided into two big problems: first, processing the image and extract the features and second, train the classification algorithm with the collected data. Before getting to code the solution, some decision had to be made, the most important was which features to use.

Because the problem consist of image processing and categorization, color is the most obvious feature to use, in this case was used the average RGB(Red, Green, Blue), but since some object share the same RBG(such as the Fruna Naranja and the Jumbo Mix Naranja), the area was also introduced as a feature.

For the first part, OpenCV was used to resize the camera and adjust some settings like exposure and brightness, then each frame of a video(possibly a camera live caption) is processed, using an OpenCV function [1] the background is removed, after that various filters are applied in order to increase the erosion and bright the colors, once that is completed, the most notorious contours are found and a minBoundRectangle is drawn surrounding the biggest contour using another OpenCV function [2], the area of the rectangle represents a very close approximation of the real area of the product. Using that exact same frame the average RGB is also calculated. Finally, these four features are written in a .csv file to train the model.

For the second part, scikitlearn, specifically the Random Forest Algorithm was used as a classification method. Basically, the csv containing the data is loaded using pandas [3] and the model is trained.

IV. EXPERIMENTAL RESULTS

The experimental results represented with a confusion matrix show an accuracy rate of approximate 98 %, from a hundred data collected the algorithm only failed to predict 2

REFERENCES

[1] OpenCV: Background Substraction
`https://docs.opencv.org/3.1.0/db/d5c/tutorial_py_bg_subtracti`
[2] OpenCV: Contour Features
`http://opencv-python-tutroals.readthedocs.io/en/latest/py_tu`
[3] Pandas : Data Analysis Library
`https://pandas.pydata.org/`

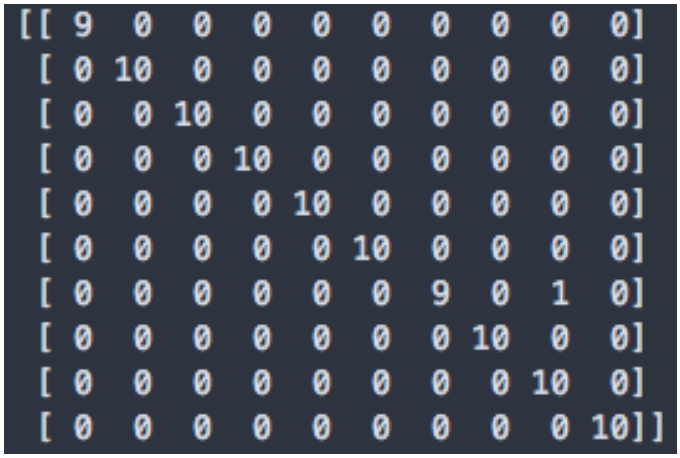

Fig. 2: Confusion Matrix