

Foundations of Artificial Vision and Biometrics Project

Academic year 2019-2020

GENDER CLASSIFICATION USING   
A SPIDER WEB METHOD

Students: Salvatore Froncillo 0522500858

Pasqualino Gravina 0522500864

# Summary

[Summary](#_Toc40517008)  [1](#_Toc40517008)

[Introduction](#_Toc40517009)  [2](#_Toc40517009)

[Presentation of the project](#_Toc40517010)  [2](#_Toc40517010)

[Objectives](#_Toc40517011)  [3](#_Toc40517011)

[Dataset used](#_Toc40517012)  [3](#_Toc40517012)

[Assigned configurations](#_Toc40517013)  [5](#_Toc40517013)

[Preprocessing](#_Toc40517014)  [6](#_Toc40517014)

[Spiderweb Algorithm](#_Toc40517015)  [6](#_Toc40517015)

[Spiderweb Algorithm 2.0](#_Toc40517016)  [6](#_Toc40517016)

[Stockbooks](#_Toc40517017)  [8](#_Toc40517017)

[SVM Classifier](#_Toc40517018)  [8](#_Toc40517018)

[Neural networks](#_Toc40517019)  [9](#_Toc40517019)

[Neural network with spider web vector](#_Toc40517020)  [9](#_Toc40517020)

[Convolutional neural network](#_Toc40517021)  [10](#_Toc40517021)

[Hybrid neural network](#_Toc40517022)  [11](#_Toc40517022)

[Further Evidence (CelebA dataset)](#_Toc40517023)  [12](#_Toc40517023)

[CelebA dataset](#_Toc40517024)  [12](#_Toc40517024)

["Re-training" model](#_Toc40517025)  [12](#_Toc40517025)

[Cross Tests](#_Toc40517026)  [12](#_Toc40517026)

[Conclusions](#_Toc40517027)  [13](#_Toc40517027)

# Introduction

## Project presentation

Immagine che contiene persona, fotografia, posando, indossando

Descrizione generata automaticamenteGender *classification is* a widely used type of classification, in particular it is used to support different recognition biometrics *.* This type of classification is very useful, for example, in the forensic field, the identification of a subject's gender by biometrics can narrow the search field by about 50%.

One of the most used biometrics is the face; when we analyze the gender from the face we operate with a physical biometry.   
The bone structure of individuals of male or female gender appears to be different and it must, however, be taken into account that often the face can be altered with features typical of the population of the opposite sex to that of the subject (make-up, beard, etc. ) *(Figure 1)* .

Figura

Immagine che contiene persona, guardando, indossando, faccia

Descrizione generata automaticamenteOne technique that could overcome the "texture" problems often encountered in *gender recognition algorithms* could be the use of landmarks. Landmarks are in fact "key" points of the face, generally 68, and after their detection they allow to leave out details of the face that in some cases could be misleading.

To study the distribution of landmarks on faces, reference can be made to the coding of a *pose estimation algorithm .* The algorithm in question uses a cobweb model to divide the face into different sectors, within which the landmarks fall.   
The number of sectors of the web is fixed for each face, regardless of the size of the face; the sector to which each landmark belongs is established by means of simple formulas relating to Cartesian coordinates and circumferences. In *figure 2* we have an example in which we can see the 68 landmarks of the face in green and the spider's web in red.

Figura

The web thus obtained is "unwound" becoming a vector of integers whose values represent the number of landmarks in that sector. This representation is very compact: if we want to consider as an example a spider's web made up of 4 slices per quadrant and 4 circles, we will have an array of 64 elements.

This algorithm has also produced excellent results in estimating the pose of a subject; the purpose of this project is to test its effectiveness by applying it to a *gender classification problem .*

## Targets

In order to obtain the gender classification, the UTKFace image dataset is used . This objective can be achieved by using only the data coming from the algorithm that applies the spider web to the faces in the image. This process is divided into:

* extraction of the face from the image;
* estimation of the position of the face landmarks;
* creation of the array representing the spider web using the provided algorithm;
* train a classifier on the arrays deriving from the algorithm for creating the web;
* estimate the accuracy achieved for *gender classification* .

This project involved 2 candidates, therefore it was requested to:

* obtain the assigned dataset;
* extract the faces of the subjects from the dataset;
* identify the location of the landmarks;
* use multiple web **configurations to create different training sets,** validation   
  sets and test sets ;
* create **two** ensembles to determine the genre, testing it with the various sets described above;
* evaluate the accuracy of the different combinations and evaluate the pros and cons.

## Dataset used

The UTKFace dataset is a large-scale face dataset with an age range from 0 to 116 years. The dataset includes over 20,000 images of faces tagged with age, gender and ethnicity. The images cover large variations in pose, facial expression, lighting, resolution, etc.

The labels referring to the face contained in each image are included in the file name and all follow the same pattern, that is:

[age] \_ [gender] \_ [race] \_ [date & time] .jpg

* [age] is an integer from 0 to 116 and indicates age
* [gender] is 0 (male) or 1 (female)
* [race] is an integer between 0 and 4, indicating white, black, Asian, Indian and others (such as Hispanic, Latino, Middle Eastern).
* [ date & time ] is in the format of yyyymmddHHMMSSFFF , which shows the date and time when an image was inserted into the dataset.

The dataset is provided in two different versions; the first defined In-the-wild or containing the image in full format which also includes the possible presence of other objects and a second version in which the images contain only the aligned and cropped face, this procedure was carried out with Dlib .

To pursue the aim of this project, the second version was used as it did not have huge computing resources available.

During the pre - processing phase, the distribution of the dataset was analyzed on the basis of the features indicated on the label to evaluate which ones could be useful for the set objectives; the following page shows graphs of this analysis.

|  |
| --- |
|  |
| *Graph 1* |
|  |
| *Graph 2* |
|  |
| *Graph 3* |

## Assigned configurations

The four configurations assigned are those with the lowest MAE (Minor Absolute Error ) for the *pose estimation* , and are the following:

1. 4C\_4S\_var2
2. 4C\_4S\_var4
3. 4C\_3S\_inv
4. 5C\_4S\_inv

Parameter C indicates the number of circles that make up the spider web and parameter S the number of slices for each quadrant.   
As for the difference between the first and second configurations, there is a variation in the radius from the center of the web to the circles.

In table 1 it is possible to observe the division of the radius for each configuration:

|  |  |  |
| --- | --- | --- |
| **Configuration** | **Radius Division (from inside to outside)** | **Array size** |
| *4C\_4S\_var2* | 8/15 \* R; 12/15 \* R; 14/15 \* R; R. | 64 |
| *4C\_4S\_var4* | 4/10 \* R; 7/10 \* R; 9/10 \* R; R. | 64 |
| *4C\_3S\_inv* | R / 4; R / 2; 3/4 \* R; R. | 48 |
| *5C\_4S\_inv* | R / 5; 2/5 \* R; 3/5 \* R; 4/5 \* R; R. | 80 |

Table 1

# Preprocessing

## Spiderweb algorithm

At the beginning of the project an algorithm based on the configuration 4 circles 4 slices was provided to extract a "spider web array" from an image.

The algorithm is divided into:

* face detection of the image;
* extraction of the 68 landmarks of the face;
* assignment of each landmark to the corresponding sector of the web.

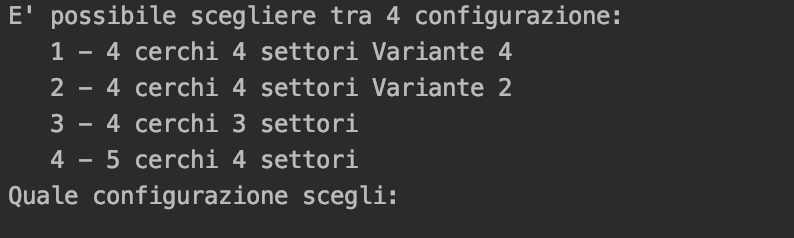
The assigned algorithm turned out to be a good starting point for building a responsive solution of the web algorithm, given the assignment for use with different web configurations, however, several changes were required.

## Spiderweb Algorithm 2.0

Various functions have been added to the aforementioned algorithm to make it more usable and efficient in achieving the assigned objectives.

**Responsive configurations**

The first change was the addition of the function **choice ()** , which allows you to choose from the command line which web configuration to apply to the dataset. ( *Figure 3* ).



Figure

To make the algorithm totally responsive, it was also advisable to partially modify the existing **add ( ) function .**

**Responsive Resize**

In the original algorithm the images are resized to a fixed number, equal to 512 pixels.   
In the version illustrated below, this operation was made responsive, since with a fixed number it was not possible to identify the landmarks on all the images. Furthermore, it has been noticed that lowering the definition to 256 pixels the algorithm is faster and does not lose much accuracy.   
To take advantage of all the “usable” images, it was decided to gradually increase the resolution of the images whenever it was not possible, at a given resolution starting from 256 pixels, to recognize a face.   
After various tests it emerged that the images that are impossible to process at a definition of 2048 pixels are not usable, for this reason it was decided to set 2048 pixels as the limit.

Ultimately the **responsive\_ resize ( ) function** starts from a resize at 256 pixels and increases up to 2048 pixels, passing through 512 and 1024 pixels.

**Unusable image analysis**

Even applying the **responsive\_ resize ( ) function** there was a certain percentage of images rejected by the algorithm. Further to the analysis, impurities were found in the dataset, such as partially covered faces, images with only eyes and objects not inherent to faces. Another type of impurity found was an incorrect labeling of several images, in fact, some elements are missing labels in the file name.   
These impurities are identified during the label retrieval phase and are not added to the dataset.

Here are some examples of impurities:

|  |  |  |
| --- | --- | --- |
| Immagine che contiene interni, guardando, vicino, primopiano  Descrizione generata automaticamente | Immagine che contiene persona, indossando, vicino, faccia  Descrizione generata automaticamente | Immagine che contiene uccello, cibo, sfocato, rosso  Descrizione generata automaticamente |
| Immagine che contiene persona, uomo, guardando, indossando  Descrizione generata automaticamente | **Immagine che contiene metro  Descrizione generata automaticamente** | Immagine che contiene sedendo, vicino, remoto, tavolo  Descrizione generata automaticamente |

**Data backup**

Finally, the **write\_list\_to\_ file ( ) function has been added** which saves the information output by the algorithm in a file format. csv .

# Classifiers

In this chapter we will analyze in detail the research process carried out to reach the best solution in terms of accuracy.

The types of supervised learning models developed for solving the classification problem are:

* Support Vector Machine (SVM);
* neural network with spider web vector;
* convolutional neural network ;
* hybrid neural network image and cobweb.

In the following paragraphs we will proceed to analyze in detail each type of model and the related results using the assigned configurations.

## SVM classifier

**Support** Vector **Machines** ( **SVM ) are supervised** learning models associated with learning algorithms for classification and regression. Given a set of examples for training, each of which is labeled with the class to which the two possible classes belong, an SVM training algorithm builds a model that assigns the new examples to one of the two classes, thus obtaining a classifier linear binary.   
An SVM model is a representation of the examples as points in space, mapped in such a way that the examples belonging to the two different categories are clearly separated by as large a space as possible. The new examples are then mapped in the same space and the prediction of the category to which they belong is made on the basis of the side in which it falls.

In our project the SVM was the first solution implemented and tested. No n produced excellent results, considering the accuracy score of 0.74.

Below are the results obtained by testing the 4 configurations:

|  |  |
| --- | --- |
| **Configuration** | **Accuracy** |
| 4C\_4S\_var4 | 0.72 |
| 4C\_4S\_var2 | 0.73 |
| 5C\_4S\_inv | 0.74 |
| 4C\_3S\_inv | 0.71 |

Table

## Neural networks

Artificial neural networks are mathematical models composed of artificial neurons inspired by biological neural networks and are used to solve engineering problems of Artificial Intelligence.

Wanting to give a more detailed definition, we could say that **neural networks are mathematical-informatic computation models based on the functioning of biological neural networks, ie models made up of interconnections of information** ; these interconnections derive from artificial neurons and computational processes based on the cognitive science model called "connectionism" .

## Neural network with cobweb vector

The second model developed uses a neural network to which the dataset received in output by the "spider web" algorithm is given as input, therefore, for each image we have an array of integers with a variable size based on the chosen configuration *(Table 1 )* .

To maximize the accuracy of the results, the following parameters were varied in different ways:

* batch size and Epoch of the Fit;

percentage of dropout at each layer ;

* batch normalization of intermediate layers ;
* percentage of data elements in Validation and in Test;
* size and number of intermediate layers ;
* type of optimizer for compiling the classifier;
* type of loss for the compilation of the classifier;
* type of activation function.

It was also attempted to add the two features regarding age and race of the faces to the spider web array, without noticing significant improvements.

The results obtained are shown in table 3:

|  |  |  |
| --- | --- | --- |
| **Configuration** | **Loss** | **Accuracy** |
| 4C\_4S\_var4 | 0.54 | 0.72 |
| 4C\_4S\_var2 | 0.54 | 0.73 |
| 5C\_4S\_inv | 0.50 | 0.75 |
| 4C\_3S\_inv | 0.55 | 0.70 |

Table

We can therefore state that even with the use of neural networks it was not possible to obtain an accuracy percentage greater than 0.75.

## Convolutional neural network

convolutional neural network was created for the classification of genres.   
The **Convolutional Neural Network** ( **CNN** ) represents a highly successful artificial neural network architecture in computer vision applications and also widely used in applications that process media such as audio and video. The most popular application of the   
convolutional neural network, however, remains that of *identifying,* on the part of a computer, what an image represents.   
  
To maximize the accuracy of the result, the following parameters have been changed:

* batch size and Epoch of the Fit;
* percentage of dropout ;
* batch normalization of intermediate layers;
* percentage of data elements in Validation and in Test;
* size and number of intermediate layers ;
* type of activation function.

The accuracy found using this type of neural network is 0.88.

## Hybrid neural network

A next step was the implementation of a hybrid neural network, which combines the convolutional neural network , which takes only the images as input, with the neural network which takes the web array as input. The two networks are concatenated into a single network that will provide a unique result in output.

convolutional neural network used is the same as the convolutional network analyzed in the previous solution, so as to be able to find improvements due only to the addition of the neural network with input to the spider web array.

Also in this case, to maximize the accuracy of the result, the following parameters have been changed:

* patch and number of epochs of the Fit ;
* percentage of dropout ;
* batch normalization of intermediate layers ;
* percentage of data elements in Validation and in Test;
* size and number of intermediate layers ;
* type of activation function.

The results obtained are shown in the table:

|  |  |  |
| --- | --- | --- |
| **Configuration** | **Loss** | **Accuracy** |
| 4C\_4S\_var4 | 0.25 | 0.89 |
| 4C\_4S\_var2 | 0.27 | 0.88 |
| 5C\_4S\_inv | 0.27 | 0.88 |
| 4C\_3S\_inv | 0.26 | 0.88 |

Table

# Further Evidence ( CelebA dataset )

In an attempt to obtain a better percentage in terms of accuracy, it was decided to submit a larger amount of data for training to the model, with better configuration in the neural network test with web only. For this reason, the use of the CelebA dataset was chosen .

## CelebA dataset

CelebFaces Attributes Dataset ( CelebA ) is a large-scale attribute face dataset with over 200,000 celebrity images, each with 40 attribute annotations. The images in this dataset cover large variations in poses and backgrounds.

This dataset differs from UTKFace for the type of attributes used to label the images and the size of the dataset itself; in UTKFace the labels concern biological references (race, age and sex), while in CelebA we focus more on the attributes concerning the appearance conferred on the person by particular objects or signs of recognition (curly hair, glasses, baldness, mustache, make -up, etc ..).

## "Re-training" model

Having first preprocessed the dataset with the spider web algorithm to obtain the array for each image, and then trained the classifier again with the new data, the following result was obtained:

|  |  |  |
| --- | --- | --- |
| **Configuration** | **Loss** | **Accuracy** |
| 5C\_4S\_inv | 0.42 | 0.81 |

Table

## Cross Tests

Finally, the trained classifier was tested on the UTKFace dataset on the CelebA dataset and vice versa. The evidences that emerge from these tests are the following:

|  |  |  |
| --- | --- | --- |
| **Classifier trained on** | **Test dataset** | **Accuracy** |
| UTKFace | UTKFace | 0.74 |
| CelebA | CelebA | 0.81 |
| UTKFace | CelebA | 0.74 |
| CelebA | UTKFace | 0.73 |

Table

Table 6 showed that the UTKFace -trained classifier achieves an accuracy score that remains unchanged across both test datasets.

Instead, although the classifier trained with CelebA has reached an accuracy of 0.81 in the test phase with its own dataset, with the cross-test this value decreases. This decrease is attributable to the difference in the composition of the two datasets; in UTKFace there are faces in an age range ranging from 0 to 116 years, with a diversified distribution by race. In CelebA , on the other hand, we have a narrower age range, which does not include faces in childhood and advanced age.

# Conclusions

The search for the best solution began with the use of SVM classifiers; this type has demonstrated, on the 4 configurations used, a maximum accuracy of 0.74.

Considering that the result achieved was not optimal, we continued with the implementation of a classifier based on a neural network which takes as input the dataset given in output by the "spider web" algorithm. Also in this case, testing the algorithm on the 4 configurations, the accuracy limit of 0.75 was not exceeded.

convolutional neural network was created for the classification of genres which led to an accuracy of 0.88.

Subsequently a hybrid neural network was implemented, which joins the convolutional neural network with the neural network that takes the spider web array as input; this model produces an accuracy of 0.89. Although small, the variation in accuracy between the last two models demonstrates a slight contribution from the use of the web.

In an attempt to obtain a better percentage in terms of accuracy in the neural network model with only web vector, it was decided to submit a greater amount of data to this model using the CelebA dataset .

This attempt produced an accuracy of 0.81, given that it is, however, affected by the presence within CelebA of faces depicting only people in a narrower age range than UTKFace .

# Appendix

## Neural Network Graphs with Web Vector - UTKFace

**4C\_4S\_var4 configuration**

|  |  |
| --- | --- |
| *Accuracy* | *Loss* |
| **Immagine che contiene mappa, screenshot  Descrizione generata automaticamente** | **Immagine che contiene testo, mappa  Descrizione generata automaticamente** |

**4C\_4S\_var2 configuration**

|  |  |
| --- | --- |
| *Accuracy* | *Loss* |
| **Immagine che contiene testo, mappa  Descrizione generata automaticamente** | **Immagine che contiene testo, mappa  Descrizione generata automaticamente** |

**Configuration 5C\_4S\_inv**

|  |  |
| --- | --- |
| *Accuracy* | *Loss* |
| **Immagine che contiene mappa  Descrizione generata automaticamente** | **Immagine che contiene mappa  Descrizione generata automaticamente** |

**Configuration 4C\_3S\_inv**

|  |  |
| --- | --- |
| *Accuracy* | *Loss* |
| **Immagine che contiene testo, mappa  Descrizione generata automaticamente** | **Immagine che contiene testo, mappa  Descrizione generata automaticamente** |

## Convolutional Neural Network Charts - images only

|  |  |
| --- | --- |
| *Accuracy* | *Loss* |
| **Immagine che contiene testo, mappa  Descrizione generata automaticamente** | **Immagine che contiene testo  Descrizione generata automaticamente** |

## Hybrid Neural Network Charts

**4C\_4S\_var4 configuration**

|  |  |
| --- | --- |
| *Accuracy* | *Loss* |
| **Immagine che contiene mappa  Descrizione generata automaticamente** | **Immagine che contiene testo, mappa  Descrizione generata automaticamente** |

**4C\_4S\_var2 configuration**

|  |  |
| --- | --- |
| *Accuracy* | *Loss* |
| **Immagine che contiene testo, mappa  Descrizione generata automaticamente** | **Immagine che contiene testo, mappa  Descrizione generata automaticamente** |

**Configuration 5C\_4S\_inv**

|  |  |
| --- | --- |
| *Accuracy* | *Loss* |
| **Immagine che contiene mappa  Descrizione generata automaticamente** | **Immagine che contiene testo, mappa  Descrizione generata automaticamente** |

**Configuration 4C\_3S\_inv**

|  |  |
| --- | --- |
| *Accuracy* | *Loss* |
| **Immagine che contiene mappa  Descrizione generata automaticamente** | **Immagine che contiene testo, mappa  Descrizione generata automaticamente** |

## Neural Network Graphs with Spider Web Vector - CelebA Dataset

**Configuration 5C\_4S\_inv**

|  |  |
| --- | --- |
| *Accuracy* | *Loss* |
|  |  |