Gender Classification using a Spider Web Method

Vincenzo Russo[1]

[1] *Paola Barra, Carmen Bisogni*

Nowadays face classification is a serious problem, in this paper we present an approach based on a feature vector classification extracted using Web-Shaped Model Algorithm [1]. We classify these vectors using a Linear SVM and 2 data-sets CelebA, UTKFace and a combined version of the two. In addition we stratify the dataset making it of even number of males and females.
There are different approaches on Gender Recognition based on images but we decided to use feature vectors due to the convenience and privacy that they can offer.

I. INTRODUCTION

In this study we intend to classify images of individuals from CelebA and UTKFace data-sets, which will be described in detail in the following section. The purpose of this article is to show the differences between using a stratified sample dataset versus a non stratified dataset for gender classification. The feature vector was given by the Web-Shaped Model Algorithm [1].

The aim of the project is to classify genders by using a Web Shaped Model. The motivations for this research involve video surveillance, security. Gender is a very common bio-metric used to support other identification (1:N) bio-metrics. In the forensic field that could be used to reduce the field of research around 50%.

One of the most widely adopted and object of this research is the face. We find that the bone structure is different between male and female. We must note that the face can be altered with typical traits of the population of the opposite gender (e.g. make-up, beard etc...).

II. THE DATA-SETS

A. CelebA

CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations. The images in this dataset cover large pose variations and background clutter. CelebA has large diversities, large quantities, and rich annotations, including:

- 10,177 number of identities,
- 202,599 number of face images, and
- 5 landmark locations, 40 binary attributes annotations per image.

The dataset is divided in ***58%*** females and ***42%*** males. In this dataset the age label is absent, the landmarks were already present but we used the ones retrieved by the Web Shaped Model, we can find the binary attributes values for each image in **list_attr_celeba.csv**.

Index	Definition	Index	Definition	Index	Definition	Index	Definition
1	5o'ClockShadow	11	Blurry	21	Male	31	Sideburns
2	ArchedEyebrows	12	BrownHair	22	MouthSlightlyOpen	32	Smiling
3	Attractive	13	BushyEyebrows	23	Mustache	33	StraightHair
4	BagsUnderEyes	14	Chubby	24	NarrowEyes	34	WavyHair
5	Bald	15	DoubleChin	25	NoBeard	35	WearingEarrings
6	Bangs	16	Eyeglasses	26	OvalFace	36	WearingHat
7	BigLips	17	Goatee	27	PaleSkin	37	WearingLipstick
8	BigNose	18	GrayHair	28	PointyNose	38	WearingNecklace
9	BlackHair	19	HeavyMakeup	29	RecedingHairline	39	WearingNecktie
10	BlondHair	20	HighCheekbones	30	RosyCheeks	40	Young

B. UTKFace

UTKFace dataset is a large-scale face dataset with long age span (range from 0 to 116 years old). The dataset consists of over 20,000 face images with annotations of age, gender, and ethnicity. The images cover large variation in pose, facial expression, illumination, occlusion, resolution, etc. This dataset could be used on a variety of tasks, e.g., face detection, age estimation, age progression/regression, landmark localization, etc. Some sample images are shown as following:

- consists of 20k+ face images in the wild (only single face in one image)
- provides the correspondingly aligned and cropped faces
- images are labelled by age, gender, and ethnicity

The dataset is divided in ***48%*** females and ***52%*** males. The labels of each face image is embedded in the file name, formatted like age_gender_race_date&time.jpg

- **age** is an integer from 0 to 116, indicating the age
- **gender** is either 0 (male) or 1 (female)
- **race** is an integer from 0 to 4, denoting White, Black, Asian, Indian, and Others (like Hispanic, Latino, Middle Eastern).
- **date&time** is in the format of yyyymmddHHMMSSFFF, showing the date and time an image was collected to UTKFace

III. PRE-PROCESSING

For the study we have applied pre-processing The datasets have been balanced by male and female target The motivation was for making the classifier converge faster and having a lower error rate in classification.

In this phase we removed the gender surplus by using a random index which selects elements without replacements and the removes it from the dataset. This gives us more natural results than removing the last n elements. So we divided the pre-processed dataset in an evenly distributed way depending on the individual's gender.

```
import numpy as np

def balance_datasets_random_50_50(path_male,path_female,
    male = loadtxt(path_male, delimiter=',')
    female = loadtxt(path_female, delimiter=',')
    n_row_m = np.shape(male)[0]
    n_row_f = np.shape(female)[0]

    if n_row_f > n_row_m:
        print("more females")
        print(n_row_f - n_row_m)
        idx = random.sample(range(n_row_f), n_row_m)
        female = female[idx, :]
    else:
        print("more males")
        print(n_row_m - n_row_f)
        idx = random.sample(range(n_row_m), n_row_f)
        male = male[idx, :]
    n_row_m = np.shape(male)[0]
    n_row_f = np.shape(female)[0]

    total = np.vstack((female, male))
    print(np.shape(total))
    savetxt(path_dest, total, fmt='%i', delimiter=',')
```

Listing 1: Code example

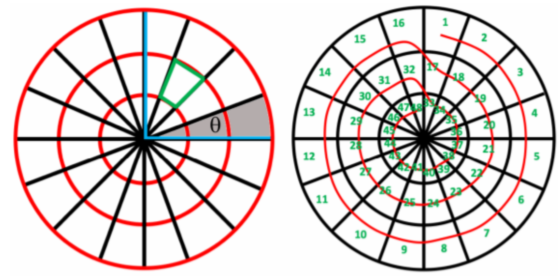

Fig. 3. Left: the web-shaped model applied on the face and centered on the landmark P_33; the radius of the web may change according to the farthest landmark from the center; red lines identify the *circles*, blue lines a *quarter*, the gray region is a *slice* of width θ, and green contour identifies a *sector*. Right: the order in which the sectors are analysed to build the pose feature vector, starting from the outer circle of the model.

Fig. 4. An application of the cascade: the first model identifies the face landmarks; the elements of the second one are circles, slices, and their sectors where the procedure places the landmarks to build a feature vector.

IV. LANDMARKS EXTRACTION

After merging the two data-sets we used the Web-Shaped Model for Head Pose Estimation.[1]
To build a pose feature vector, each landmark needs to be associated with a specific sector of the model. This process is carried out by first detecting the circle, the quarter and then the slice to which a specific landmark belongs to.
For this study we used 4 circles 4 sectors. Giving as output a 64 dimensions vector.

V. MODEL USED

A. Support Vector Machines

Despite the high training cost we decided to use Linear Support Vector Machines because of the simplicity and the good classification generalization for binary problems. Here follows the definition of a Linear SVM:

We are given a training dataset of n points of the form $(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)$, where the y_i are either 1 or -1, each indicating the class to which the point $\vec{x}_i$ belongs. Each $\vec{x}_i$ is a p-dimensional real vector. We want to find the "maximum-margin hyper-plane" that divides the group of points $\vec{x}_i$ for which $y_i = 1$ from the group of points for which $y_i = -1$, which is defined so that the distance between the hyper-plane and the nearest point $\vec{x}_i$ from either group is maximized.

Any hyper-plane can be written as the set of points $\vec{x}$ satisfying

$\vec{w} \cdot \vec{x} - b = 0,$
where $\vec{w}$ is the (not necessarily normalized) normal vector to the hyper-plane. This is much like Hesse normal form, except that $\vec{w}$ is not necessarily a unit vector. The parameter $\frac{b}{\|\vec{w}\|}$ determines the offset of the hyper-plane from the origin along the normal vector $\vec{w}$.

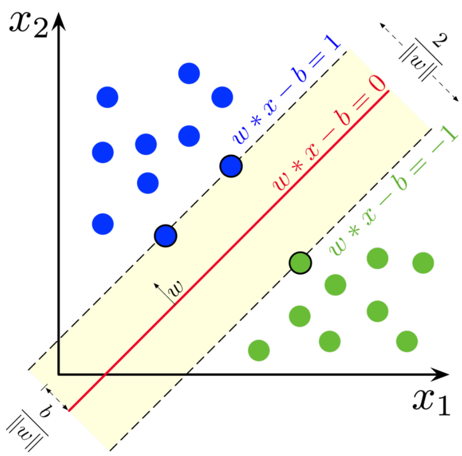

FIG. 1. SVM Example

B. Linear Kernel

Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a new data point will be in. In the case of support-vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a (p-1)-dimensional hyper-plane. This is called a linear classifier. There are many hyper-planes that might classify the data. One reasonable choice as the best hyper-plane is the one that represents the largest separation, or margin, between the two classes. So we choose the hyper-plane so that the distance from it to the nearest data point on each side is maximized. If such a hyper-plane exists, it is known as the maximum-margin hyper-plane and the linear classifier it defines is known as a maximum-margin classifier; or equivalently, the perceptron of optimal stability.

VI. FEEDING PROCESS

This process has been repeated for each classification. A matrix of n rows and 65 columns was created from the numpy arrays given by the Web-Shaped Model Algorithm [1].

- The first column which contains the label 0 for female, 1 for male.
- The remaining columns contain the feature vector of 64 elements extracted from the face by the Algorithm [1].

VII. DATA-SETS STRATIFICATION

In statistical surveys, when sub-populations within an overall population vary, it could be advantageous to sample each sub-population (stratum) independently. Stratification is the process of dividing members of the population into homogeneous subgroups before sampling. The strata should define a partition of the population. That is, it should be collectively exhaustive and mutually exclusive: every element in the population must be assigned to one and only one stratum. Then simple random sampling or systematic sampling is applied within each stratum. The objective is to improve the precision of the sample by reducing sampling error. It can produce a weighted mean that has less variability than the arithmetic mean of a simple random sample of the population.

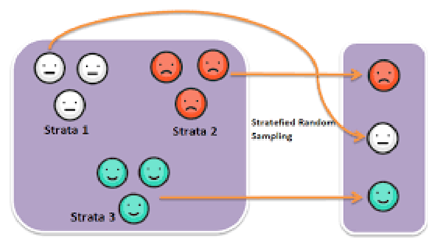

FIG. 2. Stratified Sampling Example

We found the following surplus of individuals by a given gender:

	Surplus	Gender
CelebA	30672	F
UTKFace	901	M
Combined	30878	F

The surplus has been removed from the dataset using random extraction without replacement.

VIII. EXPERIMENTS

The experiments took place on a Windows laptop with Intel Core i7-8550U, 8 GB DDR4 RAM, 512 GB SSD. Two main experiment was:

1. The study of the data-sets with stratified sampling

The data-sets used for the experiments are:

1. CelebA
2. UTKFace
3. CelebA + UTKFace

From the experiments conducted we can assume that a good pre-processing and balancing can improve the training time of the model.

We achieved a Mean Absolute Error of $\approx$ 0,30 for each experiment, which is fairly good for a binary classification problem.
The results were as expected because we used a Linear SVM and the they are consistent with the model used. Since we used a simple model such as the aforementioned we can be sure that it didn't over-fit the data and can generalize well on new data.

IX. PERFORMANCE METRICS USED

In statistics, mean absolute error (MAE) is a measure of errors between paired observations expressing the same phenomenon. Examples of Y versus X include comparisons of predicted versus observed, subsequent time versus initial time, and one technique of measurement versus an alternative technique of measurement. MAE is calculated as:

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}..$$

It is thus an arithmetic average of the absolute errors $|e_i| = |y_i - x_i|$, where y_i is the prediction and x_i the true value. Note that alternative formulations may include relative frequencies as weight factors. The mean absolute error uses the same scale as the data being measured. This is known as a scale-dependent accuracy measure and therefore cannot be used to make comparisons between series using different scales.[2] The mean absolute error is a common measure of forecast error in time series analysis,[3] sometimes used in confusion with the more standard definition of mean absolute deviation. The same confusion exists more generally.

MAE is not sensitive towards outliers and given several examples with the same input feature values, the optimal prediction will be their median target value.

X. RESULTS

	MAE	Time (hours)
CelebA	0,29	6,5
UTKFace	0,31	0,45
Combined	0,30	8

TABLE I. Results

		Predicted	
		Male	Female
Actual	Male	0,71	0,29
	Female	0,31	0,69

TABLE II. CelebA Confusion Matrix

		Predicted	
		Male	Female
Actual	Male	0,68	0,32
	Female	0,32	0,68

TABLE III. UTKFace Confusion Matrix

		Predicted	
		Male	Female
Actual	Male	0,70	0,30
	Female	0,31	0,69

TABLE IV. CelebA+UTKFace Confusion Matrix

	Precision	Recall	F1-Score	Support
Female	0,70	0,71	0,70	25789
Male	0,70	0,69	0,70	25789
Accuracy			0,70	51578

TABLE V. Accuracy for CelebA

	Precision	Recall	F1-Score	Support
Female	0,68	0,69	0,68	3421
Male	0,68	0,68	0,68	3421
Accuracy			0,68	6842

TABLE VI. Accuracy for UTKFace

	Precision	Recall	F1-Score	Support
Female	0,69	0,70	0,70	29315
Male	0,70	0,69	0,69	29314
Accuracy			0,69	58629

TABLE VII. Accuracy for CelebA+UTKFace

XI. CONCLUSIONS

This research showed that the Web Shape model is useful for classifying genders and so reduce by the 50% the uncertainty of identifying a subject. This could be useful for a first effective filtering in a bio-metric identification system.

The model could be improved by tweaking hyper parameters or use a more suitable model for classify these vectors.

As further developments we could:

- Normalize the vector data in the range [0,1].
- Choose a better classifier for binary classification task using sparse low dimensional vectors.
- Tweak the linear SVM hyper-parameters.
- Using a non-linear kernel for the SVM.

ACKNOWLEDGMENTS

We thank Ph.D. Paola Barra and Carmen Bisogni for giving me the opportunity of working on an interesting problem of gender classification.
We thank the BIPLab and Professor Michele Nappi for teaching us the fundamentals of biometrics.
We thank again Paola Barra, Silvio Barra, Carmen Bisogni, Maria De Marsico and Michele Nappi for the chance of using this advanced algorithm for landmarks extraction.

[1] P. Barra, S. Barra, C. Bisogni, M. D. Marsico, and M. Nappi, Web-Shaped Model for Head Pose Estimation: An Approach for Best Exemplar Selection (2020).