Pegoraro Marco
Master’s degree in Computer Science and Engineering
University of Verona
marco.pegoraro_01@studenti.univr.it

Report

Machine Learning & Pattern Recognition (2019/2020)

Index

Motivation and rationale

In the last decades the field of computer graphics has witnessed an important evolution. The advancement of modelling, digitising and visualising techniques for 3D shapes has led to an increasing amount of 3D models, while the development of new computer hardware and software has made easier the use of 3D data in our daily lives. With all this new resources, emerging research areas have found more interests. One of these is the classification and retrieval of 3D models. Many researchers around the world have decided to explore new mathematical theories and techniques to solve this problem, as can be seen in [2], [3] and [4], and a large number of algorithms have been proposed, each one with its own advantages and disadvantages. In this project I employ the ShapeDNA, an effective shape signature, to classify 10 different classes of 3D meshes undergoing near-isometric transformations or small deformations.

State of the art

ShapeDNA[1] is a numerical fingerprint or signature of surfaces and solids. Its value is computed taking the eigenvalues of the Laplace–Beltrami operator. For the purpose of shape comparison and identification, the eigenvalues need to be normalised dividing by the first non-zero eigenvalue of the spectrum. In this way the signature is independent of the size of the shape. The Laplace–Beltrami spectrum fulfils many fundamental properties such as:

- ISOMETRY: independence of distance-preserving transformations;
- SCALING: independence of the object's size;
- SIMILARITY: the signature depends continuously on shape deformations;
- EFFICIENCY: the effort needed to compute the signature is reasonable;
- PHYSICALITY: intuitive geometric or physical interpretation;
- COMPRESSION: the signature doesn't contain redundant data.

Shape-DNA is useful in several applications, such as copyright protection, database retrieval and quality assessment. Moreover shape identification and comparison can be done using only few eigenvalues, making possible to locate objects rapidly within huge databases.

Objectives

Objective of this project is to classify 10 different classes of 3D meshes undergoing near-isometric transformation or little deformation. I decide to use the ShapeDNA as feature to inspect its property on different shapes and its capability of discrimination.

Methodology

Since ShapeDNA works well with distances in shape comparison, I used the K-Nearest Neighbourhood which computes the distances between the test samples and the K nearest train samples. The prediction is based on the mode of the class between the K nearest train samples. As already shown by Reuter et al.[1], a good distance metric for the ShapeDNA is the euclidean. So, as first test, I computed the squared Euclidean distance between the vectors of features and I set the K to 3. I decided to change metric distance to see if there is a better one. In this report I show the results of the Canberra distance. Moreover, I tried to change the number of K neighbours to 5 keeping the squared Euclidean distance.

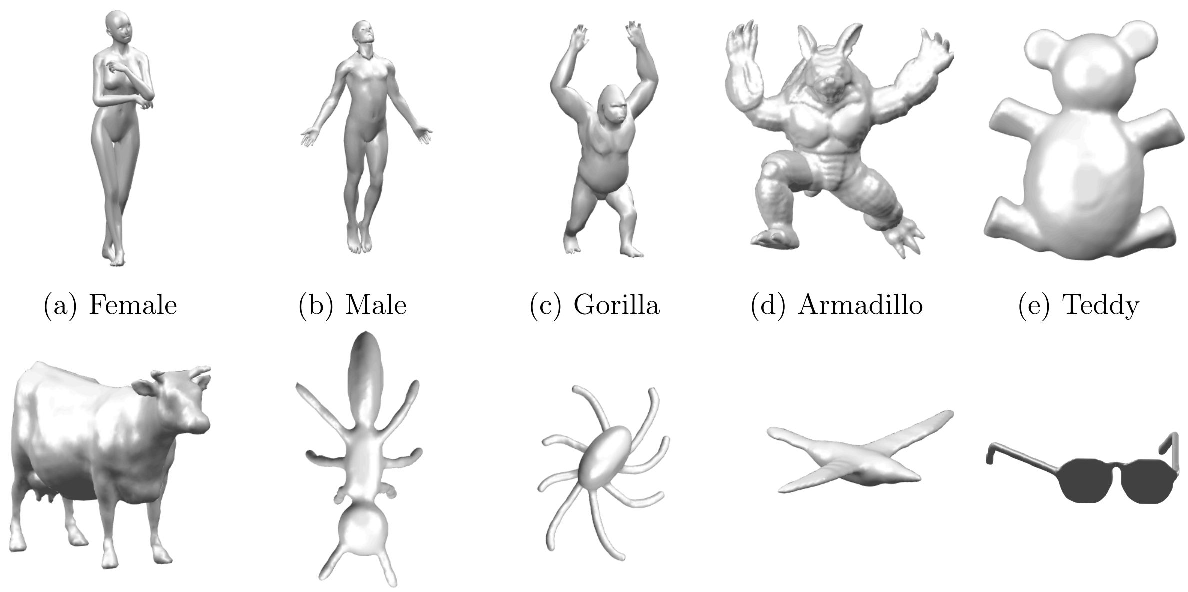
In order to compare more classification models, I did other 3 tests with a different classifier. Since the dataset I selected has a few samples for each class, I chose the Support Vector Machines. Its efficacy in cases, where the number of features is greater than the number of samples, makes it the perfect candidate. Moreover I can also change the type of kernel to solve non-linear problems. In the first test I used a linear kernel with C set to 1. In the next two tests I adopted a RBF kernel: first with C set to 1 and then to 8. This kernel is considered in general as the best choice in literature and setting a higher C penalises more the misclassified data. Since the problem I am facing is multiclass, I trained a SVM for each class as one vs rest. Then I used the higher probability between the SVMs to determine the class. Before giving in input the feature vectors to the SVM, I standardised them removing the mean and then scaled to unit variance.

The dataset will be composed of 3D meshes. For each class, the shape will be near-isometric. In order to make the task more challenging, the meshes can have small deformations, different connectivity, holes and small differences in some parts. In this way the samples can represent even imperfection due to error in the design of the shapes.

Experiments and results

For my experiments, I have chosen a dataset composed of 10 classes coming from two different datasets: the classes *Female, Male* and *Gorilla* belong to the TOSCA non rigid shape[5]; the classes *Armadillo, Teddy, Fourleg, Ant, Octopus, Bird* and *Glasses* belong to SHREC2007[2]. Figure 1 shows an example from each class in the dataset.

The shapes from TOSCA have holes in the eyes and are the same in each class but with isometric transformation. With this shapes I want to test how well ShapeDNA can identify the same shape in different positions and if it can distinguish 3 different humanoid shapes, especially *Female* and *Male*. The classes from SHREC2007 represent different animals and objects with more variation among shapes of the same class. *Teddy* and *Armadillo* are biped models, but with different structure from the others humanoids in the dataset. *Armadillo* has some shapes with missing parts, while in *Teddy* the models undergo small deformations. *Ant* and *Octopus* are two animals classes both whit 8 limbs, but with different bodies. *Fourleg* is the class with the most variation and it represent animals that stand on four legs. It includes cows, horses, dogs, pigs, bulls, giraffes and



(a) Female (b) Male (c) Gorilla (d) Armadillo (e) Teddy

(f) Fourleg (g) Ant (h) Octopus (i) Bird (j) Glasses

Figure 1: Dataset

camels. I consider this class the most challenging, because it contains many shapes with different characteristics. In this way, I want to test if ShapeDNA can capture the common properties in *Fourleg* and distinguish it from other classes. *Bird* and *Glasses* are the most different classes from the others and they have unique structures. With this dataset I want to capture classes that are different from each other in various way in order to create various kind of challenge in the classification.

Each class in the dataset has 20 samples that have been split in training and testing sets with respectively 16 and 4 samples from each class. Since there is a low number of samples, I adopted a cross validation technique repeating the splitting 5 times.

In the feature extraction phase, I precomputed the ShapeDNA for all the samples and saved it in a file with extension "_DNA.npy". In this way I speeded up the tests without having to recompute the features vectors each time. Even though the ShapeDNA can have dimension higher than 50, in the classification process I used features vectors of 11 elements to compensate the low number of samples.

To evaluate the performance, I computed a confusion matrix for each partition of the dataset and one to summarise all the tests. From each matrix, I extracted the accuracy of the classifier and the precision and the recall for each class. Then I summarised all this values in a bar plot. I also saved a qualitative result displaying the test shapes with the label predicted in red, if it is the wrong prediction, and green, if it is the correct one. At the end of all the tests, I summarised the results in two bar plots: one represents the accuracy, average precision and average recall for each classification method; the other shows the f1 score of each classifier divvied up for each class plus the average of all the classes. All the images can be found in the Results folder of the GitHub project. The confusion matrices for each classification method are also visible at the end of the report from figure 2 to 7.

Parameters	KNN			SVM		
	square euclidean K=3	canberra K=3	square euclidean K=5	linear C=1	RBF C=1	RBF C=8
Accuracy	88	91,5	86	89,5	89	**93,5**
Precision	88,27	92	90,78	90,78	89,49	**93,71**
Recall	88	91,5	86	89,5	89	**93,5**

Table 1: Results express in percentage

Parameters	KNN			SVM		
	square euclidean K=3	canberra K=3	square euclidean K=5	linear C=1	RBF C=1	RBF C=8
Female	95,24	95	87,8	68,75	84,44	**100**
Male	97,56	93	90,48	79,17	80	**100**
Gorilla	84,21	**100**	85,71	**100**	**100**	**100**
Armadillo	**85,71**	81	76,19	82,93	82,93	80
Teddy	**88,89**	**88,89**	**88,89**	82,93	84,21	84,21
Fourleg	74,29	80	76,47	**91,89**	81	90
Ant	95	95,24	95	**100**	97,56	**100**
Octopus	91,89	91,89	94,74	**97,44**	92,68	**97,44**
Bird	71,79	87,8	64,86	**90,48**	85,71	83,72
Glasses	92,68	**100**	97,44	**100**	**100**	**100**

Table 2: F1 score per class in percentage

As can be seen in table 1, all methods achieve a good result with an average accuracy of 89,5%. The best model is the SVM with a RBF kernel and C set to 8 and it reaches an accuracy of 93,5%. The other parametrizations of the SVM achieve an accuracy of 89%, lower by 4% than the best result. The main difference is the C value which, if set higher, penalizes more the misclassified data. In this case, where there are few samples for each class, fitting more the classification on each sample results in a better performance. The second best accuracy is 91,5% and it's reached by the KNN model with Canberra distance and K set to 3. Increasing the value of K to 5 drops the accuracy to 86%, the lowest in the experiments. This result may be justified by the low number of samples for each class and the resulting low density in the features space. The test with square Euclidean distance achieves an accuracy of 88%, still lower than the SVM models.
Table 2 shows the F1 score for each class. *Glasses* and *Gorilla* are the only classes that are perfectly classified by all the SVM models. The only KNN model that perfectly classifies these two classes is the one with Canberra distance and K=3. The confusion matrices 2 and 4 show that the other two variants often misclassify *Gorilla* with *Glasses* and *Bird*. This behavior suggests a similarity in their spectrum that could be disambiguated adding a feature that describes the external appearance such as silhouettes. Overall,

Female and *Male* are better classified by the KNN models than the SVM models with C set to 1. This may suggest that a classification based on metric distance can capture better the similarity and dissimilarity between very similar shapes. Despite this, their confusion matrices shows that in very few cases the KNN models misclassify this classes with very different shapes such as *Bird* or *Glasses*, while the SVM models mistake only between *Female* and *Male*. This errors are completely overcame by the SVM with RBF and C set to 8. The classes with the lowest F1 scores are *Armadillo* and *Teddy*. The case of *Armadillo* can be justified by the presence of missing parts in some meshes. This blemish can heavily affect the ShapeDNA producing a spectrum with a higher slope than the spectrum of the full shape. These results shows that none of the used classifier are able to find a solid correspondence among the sample of this class. Even though *Teddy* has no cut in his samples, it is often mistaken with *Fourleg* or *Armadillo*. The worst results obtained by the KNN models are in *Fourleg* and *Bird*. The former is a class with a very high interclass variation that can easily lead to mismatching with other classes. Despite this, the SVM models can reach higher values proving to be a better choice in this situation. The latter is a class with a lower variation, but with a less defined structure.

Conclusions

ShapeDNA has demonstrated to be a good feature for non-rigid 3D shapes classification with some lack that can be avoided adding extrinsic information. Despite the small dimensions of the dataset, both KNN and SVM achieved almost perfect results with an accuracy higher than 88%. RBF kernel with higher C leads to the best results classifying perfectly 5 classes.

References

[1] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. "Laplace–Beltrami spectra as 'Shape-DNA'of surfaces and solids". In: *Computer-Aided Design* 38.4 (2006), pp. 342–366.

[2] Daniela Giorgi, Silvia Biasotti, and Laura Paraboschi. "Shape retrieval contest 2007: Watertight models track". In: *SHREC competition* 8.7 (2007).

[3] Z. Lian et al. "SHREC '11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes". In: *Eurographics Workshop on 3D Object Retrieval*. Ed. by H. Laga et al. The Eurographics Association, 2011. ISBN: 978-3-905674-31-6. DOI: 10.2312/3DOR/3DOR11/079-088.

[4] Z. Lian et al. "Non-rigid 3D Shape Retrieval". In: *Eurographics Workshop on 3D Object Retrieval*. Ed. by I. Pratikakis et al. The Eurographics Association, 2015. DOI: 10.2312/3dor.20151064.

[5] R. Kimmel A. M. Bronstein M. M. Bronstein. *TOSCA dataset*. Web page. URL: http://tosca.cs.technion.ac.il/book/resources_data.html.

Figure 2: Confusion matrix of the KNN classifier with square euclidean distance and K=3.

Figure 3: Confusion matrix of the KNN classifier with canberra distance and K=3.

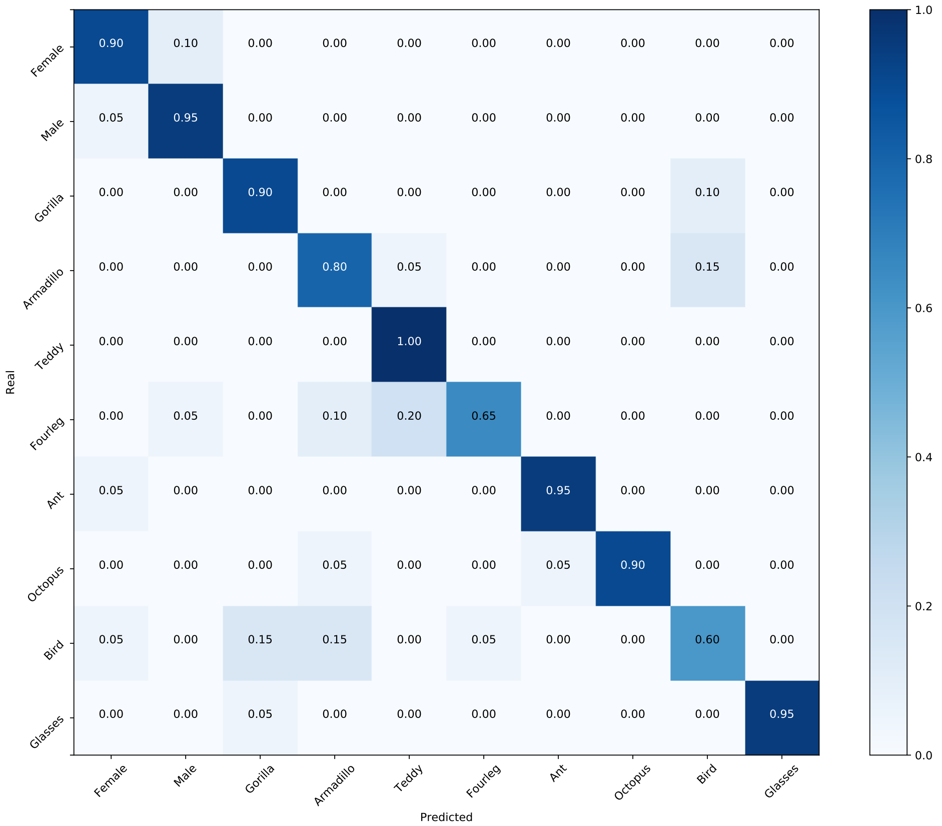


Figure 4: Confusion matrix of the KNN classifier square euclidean distance and K=5.

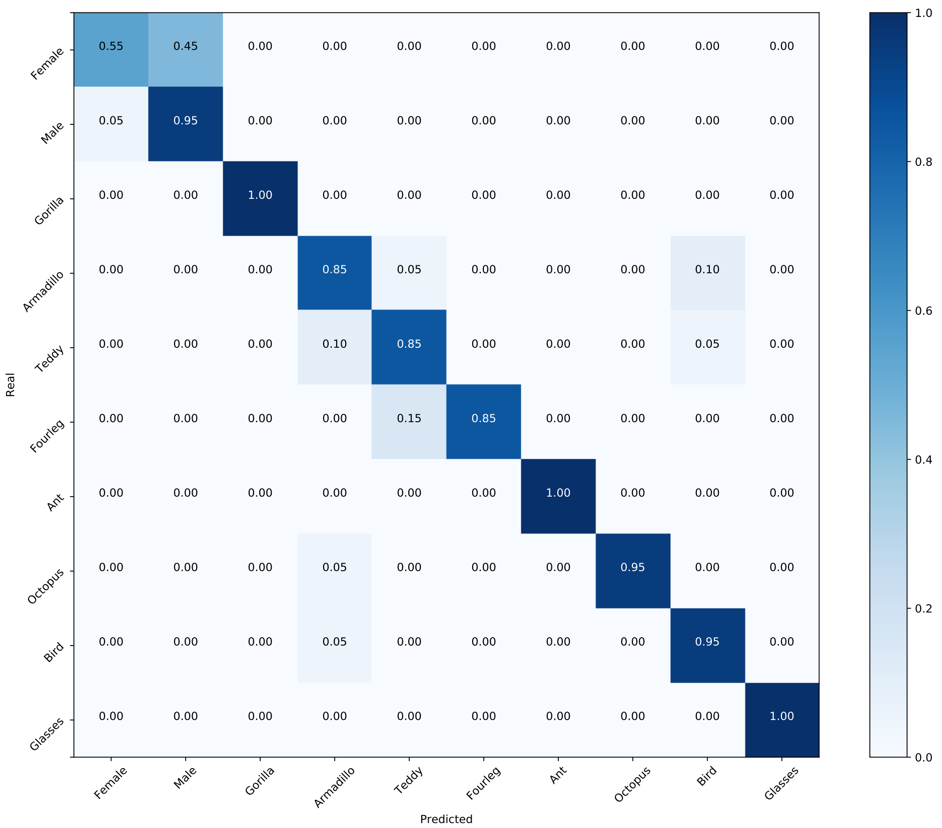


Figure 5: Confusion matrix of the SVM classifier with linear kernel and C=1.

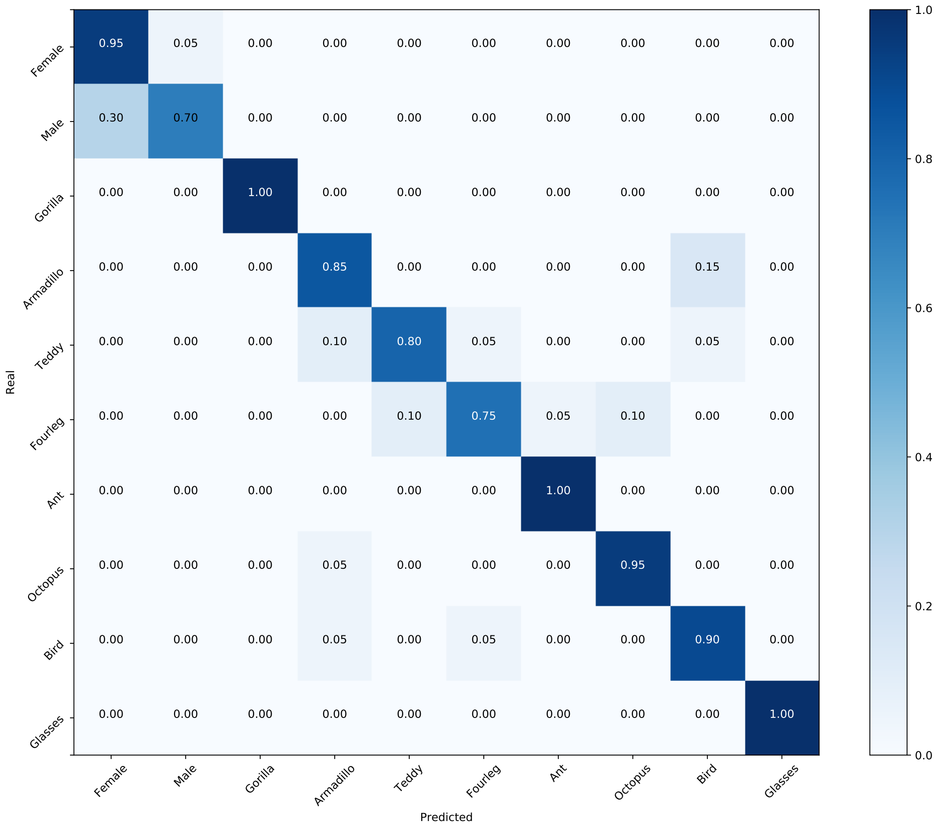


Figure 6: Confusion matrix of the SVM classifier with RBF kernel and C=1.

Figure 7: Confusion matrix of the SVM classifier with RBF kernel and C=8.