Machine Learning and Pattern Recognition

Fingerprint Spoofing Detection

Giovanni Gadi - Inaam El Helwe
Politecnico di Torino
s308685@studenti.polito.it, s3069796@studenti.polito.it

September 1, 2023

The objective of this report is to develop a classifier capable of distinguishing between genuine and manipulated fingerprint images through analyzing the result of applying different machine learning algorithms. The fingerprint images are mapped into a lower-dimensional space(around few hundred dimensions), but to ensure computational feasibility dimensions were reduced to 10 only. Noting that the embedding components have no physical interpretation.The original Dataset consists of 7 classes, however for this project the Dataset has been binarized, groupping all spoofed fingerprint samples into class 0 and authentic fingerprint samples into class 1.

1 Feature Analysis

Our training dataset consists of 2325 samples from which 1525 belong to class 0 and 800 belong to class 1, then one class is twice as much present as the other. In the following we plot some histograms to visualize the distribution of the different features before applying any pre-processing techniques

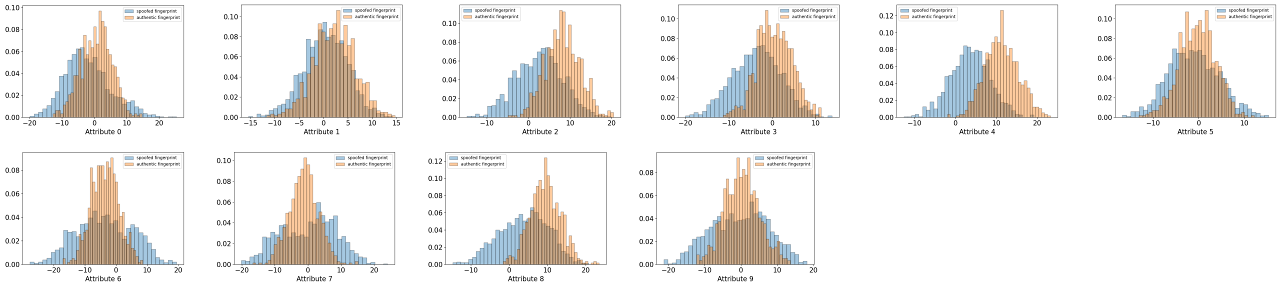
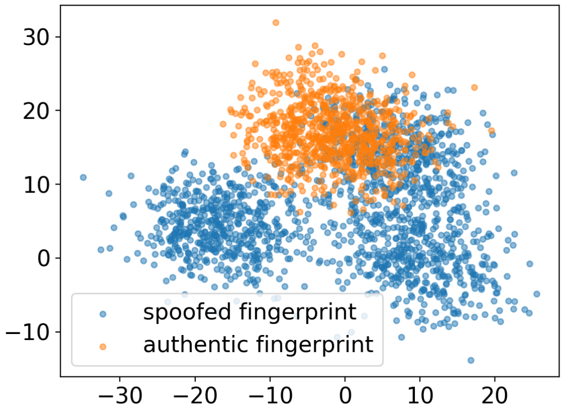


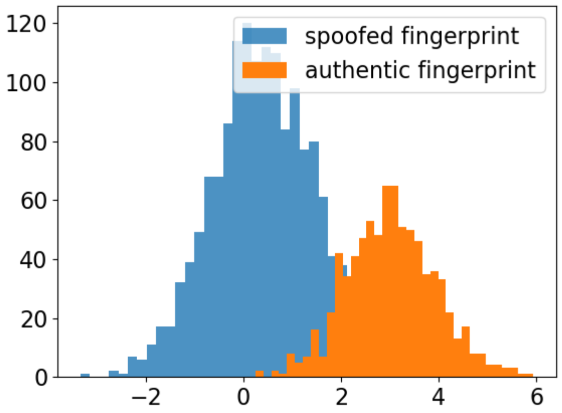
Figure 1: *Raw features distribution of the fingerprints Dataset*

We can see that for target class features can be very well estimated by a Gaussian distribution while Gaussian densities might not be sufficient for non-target class.It's interesting to note that for some features, the distributions allow us to distinguish the two classes for example attribute 4 which can be considered the most discriminant feature.

We can apply PCA(with 2 dimensions) and LDA transformations in order to have a more general insights of our dataset. We get the following figures:

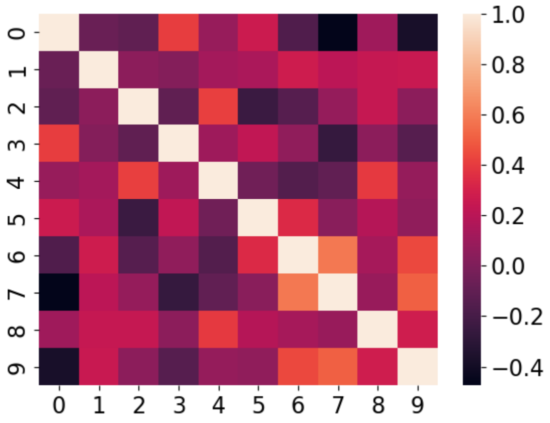


(a) *PCA plot for training set*

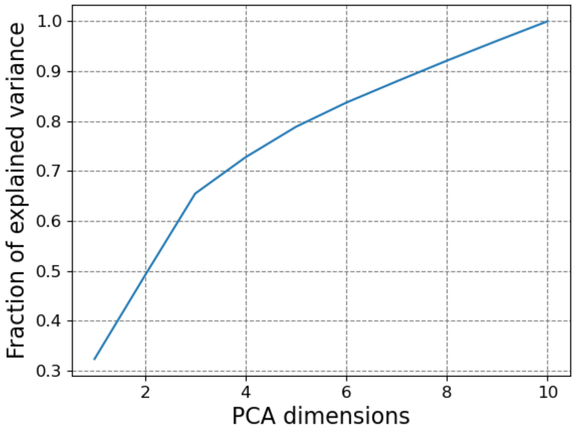


(b) *LDA plot for training dataset features*

From the PCA scatter plot we can tell that the non-target label(spoofed fingerprint) is characterized by multiple clusters while the target class by only one. LDA shows that features are linearly separable with a small area where it's hard to discriminate the two classes and can cause some errors, so we don't expect to obtain poor performances from linear models (Logistic Regression for example) but given the distribution of the features that we observed in the scatter plot we expect non-Gaussian and/or non-linear models(e.g MVG) to be significantly better for the task.

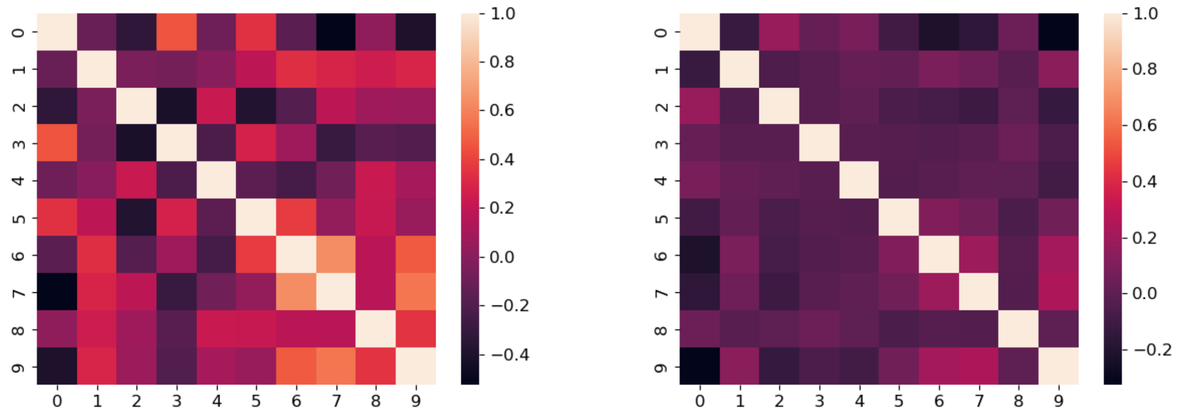


(a) *Heatmap for all dataset*



(b) *Explained variance*

Features Correlation: We will study now features correlations through Pearson correlation of our features. We start by analyzing the heatmap for all dataset Fig(a), it can be clearly seen that some features are highly correlated for example attributes 9-7, 6-7 also 3-0. While others are not(dark colors in the heatmap). So applying reduction techniques like PCA can be beneficial. This assumption is well validated in Fig(b) where we can see that reducing our dimensions to 9 rather than 10 will maintain 96% of the dataset variance and around 92% with 8 directions.



(a) *Heatmap for spoofed fingerprint*

(b) *Heatmap for authentic fingerprint*

Observing the Heatmap of each class alone we can see that for authentic fingerprint the features are weekly correlated, whereas non-target (spoofed fingerprint) class shows significantly larger correlations, so applying PCA might remove useful information that characterize the target class.

2 Validation

2.1 Training Protocol

In the following section we will evaluate different models and select the most suitable one for our application. In order to perform our analysis we will adopt a K-Fold protocol with K=5, this approach usually makes the process of model selection more reliable as we have larger amount of data for both training and validation purposes. It's important to note that no transformation is applied on the whole training data before splitting it, in order not to bias the validation. Regarding model evaluation, we measure performance in terms of normalized minimum detection cost, which measures the cost we would pay if we made different decisions using the recognizer scores. Our target application is defined by the triplet $(\pi_T = 0.5, C_{fn} = 1, C_{fp} = 10)$ π_T being the prior(unbiased in this case) and C_{fn} and C_{fp} being the respectively the costs of the false positive and false negative cases. However, we will as well consider biased cases toward one of the classes(the non-target class) $(\pi_T = 0.1, C_{fn} = 1, C_{fp} = 10)$

2.2 Gaussian Classifier