De-Identifying Facial Images Using Projections on Hyperspheres

P. Chriskos, O. Zoidi, A. Tefas and I. Pitas [1]
[1] Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124 Greece

***Abstract*— A major issue that arises from mass visual media distribution in modern video sharing, social media and cloud services, is the issue of privacy. Malicious users can use these services to track the actions of certain individuals and/or groups thus violating their privacy. As a result the need to hinder automatic facial image identification in images and videos arises. In this paper we propose a method for de-identifying facial images. Contrary to most de-identification methods, this method manipulates facial images so that humans can still recognize the individual or individuals in an image or video frame, but at the same time common automatic identification algorithms fail to do so. This is achieved by projecting the facial images on a hypersphere. From the conducted experiments it can be verified that this method is effective in reducing the classification accuracy under 10%. Furthermore, in the resulting images the subject can be identified by human viewers.**

I. INTRODUCTION

Media sharing has become mainstream in modern times and its volume increases daily. This inconceivable amount of information includes a large amount of visual media that contain information about the actions of the individuals depicted as well as the creators of these media. Large scale sharing, viewing and storing of these media introduces concerns for the privacy of the above mentioned participants. As is usually the case, this visual information is freely available to all Internet users and, as a consequence, dangers arise concerning the privacy of these media creators and the subjects depicted. Face recognition algorithms are able to recognize faces in images and video frames efficiently threatening the privacy of the subjects. Malicious users can utilize video sharing sites and social media to collect information regarding specific individuals and groups fast, freely and without much effort. Another concern for privacy arises from the wide use of video surveillance in public places, which in junction with face identification software allows identification of all persons regardless of suspicion level. Examples of privacy violation can be found in the cases of Google Street View and EverySpace, which among others use visual data to provide services and inevitably invade our everyday privacy, although not intentionally. To tackle this issue new methods must be developed that protect the privacy of the subjects, while maintaining a certain level of image quality. The quality of the final product must allow human viewers to recognize the individuals in a scene.

The proposed method is developed under the following scope. Suppose that a malicious user has trained a classifier to identify certain individuals in a series of images or video frames. With this classifier the malicious user can search for information in visual media about the targeted individuals. If new images shared by these users have been modified by a certain method the trained classifier will fail to recognize the targeted individuals, thus rendering future actions of the targets safe. So the proposed method aims to do just that while at the same time preserving enough visual quality to characterize the end product acceptable for everyday use.

Most face de-identification methods attempt to deceive automatic face recognition methods by also hindering identification by human viewers. These methods aim to destroy the majority, if not all, of the data concerning the depicted individual. Ad-hoc solutions [1] include the use of simple methods such as applying a black mask on parts of the face. Black bars are used in order to cover the eyes, while T-shaped masks cover both the eyes and the nose. Other masks reveal only the mouth and, finally, a black mask can be applied to the entire face, destroying all visual information of the facial image [1]. Additional simple methods include methods that blur the face area using low-pass filters [1], methods that add random noise with a predetermined distribution, methods that use the negative image and methods that swap facial areas, such as eyes, nose, mouth, between images that belong to different individuals [3]. Finally, simple methods also exist that subsample an image leading to pixelation, or that threshold the pixel values [1]. Moreover, more advanced methods exist that implement the k-anonymity model [2], so that all of the de-identified images indiscriminately relate to at least k elements of the initial image set. Other methods explloit characteristics of identification methods such as eigenface-based algorithms, k-anonymity models and PCA or LDA face recognition methods in order to defeat them [4]. Finally, another method exists that reduces the number of eigenvectors used in constructing the final images from basis vectors [5].

A common characteristic of the above methods is hindering recognition by both human viewers and automatic classifiers. In this paper a novel approach is proposed that utilizes projections on hyperspheres in order to defeat classifiers while preserving enough visual information so that human viewers can identify the depicted individuals, contrary to the methods mentioned above.

The rest of the paper is organized as follows. An introduction on hyperspheres is presented in Section II. The proposed de-identification method is described in Section III. The experimental setup and results are presented in Section V. Section VI analyzes a potential attack against this method. Finally the conclusions are drawn in Section VII.

The research leading to these results has been partially supported from COST, Action IC1206 and the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 287674 (3DTVS).

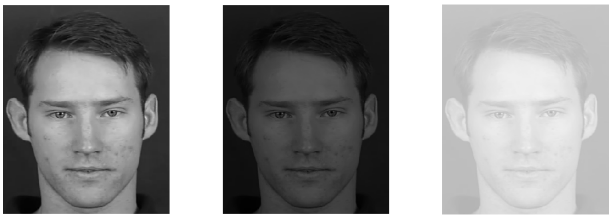

Fig. 1. Left: Original Image,Center: Projection on hypersphere centered at the origin, Right: Projection on hypersphere centered at the image whose pixels contained the maximum allowed value

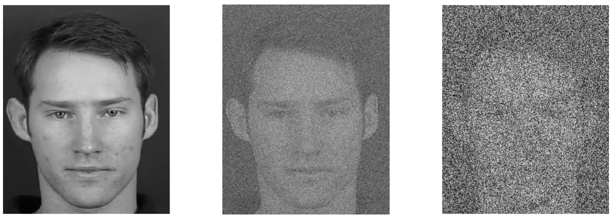

Fig. 2. Left: Original Image,Center: Projection on hypersphere with noise center following the normal distribution with a mean of 0.5 and standard deviation of 0.1, Right: Projection on hypersphere with noise center following the uniform distribution

II. Projection on Hypershperes

A hypersphere [6][7] is a generalization of the ordinary circle in 1 dimension and the ordinary sphere in 2 dimensions to dimensions $n \geq 3$. For any natural number n, a hypersphere $S^{(n-1)}$ with radius R is defined as:

$$x_1^2 + x_2^2 + \ldots + x_n^2 = R^2, \tag{1}$$

where $x_1, x_2, \ldots, x_n$ are n-tuples of points and R is the radius of the hypersphere, which is a positive real number. A hypersphere S^{n-1} can also be defined as the set of points in the n-dimensional space, which are at distance R from a center point. A hypersphere S^{n-1} centered at some origin is defined as:

$$S^{n-1} = \{\mathbf{x} \in \mathbb{R}^n : ||\mathbf{x}|| = R\}. \tag{2}$$

where $\mathbf{x}$ is a point in the n-dimensional space. The projection of a point $\mathbf{x} \in \mathbb{R}^n$ onto S^{n-1} is given by the following equation [9]:

$$P_{S^{n-1}}(\mathbf{x}) = \frac{R}{||\mathbf{x}||}\mathbf{x}, \tag{3}$$

where $P_{S^{n-1}}(\mathbf{x})$ denotes the projection of point $\mathbf{x}$ onto the hypersphere S^{n-1}.

III. De-Identification of Facial Images Based on Projection on Hyperspheres

Each image occupies a position in the n-dimensional space, where the dimensionality n of the image is equal to the number of pixels. Intuitively it is expected that images depicting the same individual with the same pose are bound to lie close together in space forming local clusters, while images depicting different individuals are bound to lie farther apart.

The general idea is to bring images of different individuals closer together in order to prevent classifiers from correctly identifying a subject in an image. The most simplistic approach is to replace the initial image with the average of all the images, or with another image. The purpose of this method however is to preserve enough information from the first image so that human viewers can identify the depicted individual. So instead of replacing the images with the average image we project the images on a hypersphere with radius R centered at the origin.

The structure of the data allows trained classifiers to accurately identify the individual in an image. A way to impair this ability of the classifiers is to undermine this structure. This can be achieved by projecting the images on a hypersphere. This projection distorts the images in such a way that, the new architecture of the data does not allow trained classifiers from discerning between the individuals. Bringing all images near to the center of the hypersphere, image clusters of different individuals are driven closer together. This clashes with the initial idea that the distance between the clusters allows classifiers to correctly classify a subject. Consequently it is expected that this projection method will hinder classifiers from accurately identifying a subject.

A. Selecting a Center for the Hypersphere

In order to project the images a hypersphere must first be defined. As mentioned in Section II a hypersphere can be defined with a center and a radius. For a center, several alternatives where considered. At first abstract centers where selected such as the origin of the n-dimensional space in which the images reside as well as the image whose pixels contained the maximum allowed value e.g. 255 for 8-bit images. These two centers did provide de-identification which can be easily defeated, since the effects that they introduced where darkening and brightening of the input images respectively. This can be easily defeated by applying the inverse effect on the output images. Despite this fact the origin was used in combination with the mean image as is described below. The output of using the above centers can be viewed in Figure 1. Another abstract center considered was an image of random noise whose values where in the same range as the input images e.g. [0,255] for 8-bit images. Visual results for these centers can be seen in Figure 2. Although the de-identification rates where high, visual quality suffered and as such these centers where not considered any further.

In order to deviate from abstract centers the train dataset image closest to the mean image was selected as a center for the hypersphere. Since this is an actual image, all images that depict the same individual as the median are not de-identified as was found through experiments. A better hypersphere would be one that is closer to the initial images and also includes information from other images in order to deceive face recognition algorithms. Such a center would be the mean image of the dataset.

The mean image is computed using the following equation:

$$\bar{\mathbf{I}} = \frac{1}{N_{im}} \sum_{i=1}^{N_{im}} \mathbf{I}_i \tag{4}$$

where $\bar{\mathbf{I}}$ is the average image, N_{im} is the number of images in the given dataset and $\mathbf{I}_i$ is each individual image in the dataset.

B. Selecting a Radius for the Hypersphere

Since the mean image was selected as the center for the hypersphere a radius is needed in order to fully define the hypersphere. It is possible to manually select the radius by using arbitrary values and then assessing the visual quality as well as the error rate of various face recognition methods. This is however a simplistic approach and for each database a new radius must be selected. As such it would be better if a radius could be calculated depending on the database used. This was achieved using the Support Vector Data Description method or SVDD, which is described in Section IV.

C. Projections Used for De-Identiffication

Two different projections where used in order to de-identify facial images. The first one is the average of the projection on the origin and the mean image. The formula used to calculate the de-identified version $\mathbf{I}_{DID}$ of an image $\mathbf{I}$ is the following:

$$\mathbf{I}_{DID} = \frac{1}{2}\left(\frac{R}{||\mathbf{I}||}\mathbf{I} + \bar{\mathbf{I}}\right). \tag{5}$$

where $\bar{\mathbf{I}}$ is the mean image, R denotes the radius of the hypersphere and $||\mathbf{I}||$ is the measure of image $\mathbf{I}$. This projection method will be referred to as Projection De-Deidentifiaction on Origin or PDID-O for short.

The second projection used was the projection with a hypersphere centered on the mean image. The de-identified image can be calculated using the following formula:

$$\mathbf{I}_{DID} = \left(\frac{R * (\mathbf{I} - \bar{\mathbf{I}})}{||\mathbf{I} - \bar{\mathbf{I}}||} + \bar{\mathbf{I}}\right). \tag{6}$$

and as above $\bar{\mathbf{I}}$ is the mean image, R denotes the radius and $||\mathbf{I}||$ is the measure of image $\mathbf{I}$. This projection method will be referred to as Projection De-Deidentifiaction on Mean Image or PDID-M for short.

Having defined the projections used to de-identify the input images, now let us focus on the value of radius R that should be used in the following section.

IV. Automatic Selection of Radius R

Choosing a small value for radius R allows us to project the initial images close to the center, and subsequently close to each other. This means that images of different individuals will also be close to images from other individuals. Choosing a large value for R, it is possible to project the initial images farther from the center, closer to the initial locations. Therefore the output images will be farther away from each other, and subsequently the clusters of different images will also be farther away. It is suspected that for small values of R the error rates of the classifiers will be high, since the classifiers will be unable to discern between the images from different individuals and as a result will classify them falsely. The value of R will also be responsible for preserving the quality of the initial images. For small values of R the image quality will suffer, while for large values of R the quality of the output images will be closer to that of the initial image. These observations can hint to the choice for the value of parameter R.

It would be preferable though if radius R was calculated based on the images in each dataset. This can be achieved using the Support Vector Data Description method.

The Support Vector Data Description or SVDD [11] is a method for defining the minimum bounding sphere that encompasses most of or all of the training vectors $\mathbf{x}_i$ where $i = 1, 2, \dots, N$ and N denotes the number of training vectors. This sphere S can be defined by a center u and a radius R, which can be computed by optimizing:

$$\min_{R,\xi,\mathbf{u}} \quad R^2 + c\sum_{i}^{N} \xi_i \tag{7}$$

$$s.t. \quad ||\mathbf{x}_i - \mathbf{u}||_2^2 \leq R^2 + \xi_i \tag{8}$$

$$\xi_i \geq 0, \quad i = 1, 2, \dots, N \tag{9}$$

where ξ_i are the slack variables and c is a parameter that describes the importance of the error in the optimization problem.

Using the Karush-Kuhn-Tucker (KKT) theorem [10] the optimization problem mentioned above can be solved by finding the saddle point of the Lagrangian:

$$\begin{aligned}\mathcal{L}(R, \xi_i, \mathbf{u}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = R^2 + c\sum_{i}^{N}\xi_i - \sum_{i=1}^{N}\beta_i\xi_i \\ - \sum_{i=1}^{N} a_i\big(R^2 + \xi_i - ||\mathbf{x}_i - \mathbf{u}||_2^2\big).\end{aligned} \tag{10}$$

This leads to the following optimality conditions:

$$\frac{\vartheta\mathcal{L}}{\vartheta\mathbf{u}} = 0 \Rightarrow \sum_{i=1}^{N} a_i\mathbf{u} = \sum_{i=1}^{N} a_i\mathbf{x}_i, \tag{11}$$

$$\frac{\vartheta\mathcal{L}}{\vartheta R} = 0 \Rightarrow \sum_{i=1}^{N} a_i = 1 \tag{12}$$

$$\frac{\vartheta\mathcal{L}}{\vartheta\xi_i} = 0 \Rightarrow a_i = c - \beta_i \tag{13}$$

From (11) and (12) the center u is given by:

$$\mathbf{u} = \sum_{i=1}^{N} a_i\mathbf{x}_i \tag{14}$$

Replacing (11), (12) and (13) in $\mathcal{L}(R, \xi_i, \alpha, \beta)$ and using the KKT conditions, optimization problem (7) can be formulated to its dual from:

$$\max_{\boldsymbol{\alpha}} \sum_{j=1}^{N} a_i\mathbf{x}_i^T\mathbf{x}_i - \sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j \mathbf{x}_i^T\mathbf{x}_i, \tag{15}$$

under the condition $0 \leq a_i \leq c$ and $\sum_i a_i = 1$.

After solving 15 radius R can be calculated as:

$$R^2 = \{\min ||\mathbf{x}_i - \mathbf{u}||_2^2, \mathbf{x}_i \text{ is a support vector or } a_i > 0\} \tag{16}$$

With the above approach it is possible to calculate a good estimate of radius R that will provide with the required distortion to de-identify the input facial images.

V. Experimental Results

A. Database Description, Classifiers and Metric Used

Experiments to test the effectiveness of the Projection-DID method where run on the XM2VTS [13] and the Extended Yale B [12] databases. From the XM2VTS database 16 individuals from the first recording where selected and used in the experimental process. The individuals face the camera on a neutral background. The frontal images where isolated and subsequently where cropped to the face area. Finally the images where converted to 8-bit grayscale images. This process resulted in a dataset with 388 train samples and 265 test samples from the 16 videos. Each sample of the above dataset has 128721 dimensions (401×321), with both train and test samples converted into vectors with dimensions 128721×1. The Extended Yale B database contains images from 38 individuals under different lighting conditions. Train and test sets contain 1209 and 1205 samples respectively. These sets where defined by randomly selecting half the images from each individual. Each image has 1200 dimensions (40×30) and was used in vector form with dimensions 1200×1. The train sets mentioned above where used to train classifiers and then the test data where used to measure the efficiency of the proposed method. The three classifiers used in the process where the K-Nearest Neighbour Classifier (KNN) with 1 nearest neighbour, the Nearest Centroid Classifier and the Naive Bayes Classifier. In the case of the KNN classifier varying the number of nearest neighbours to 3 and 5 yielded similar results.

In order to calculate the difference between the initial and de-identified images and to measure the degradation of quality introduced by the Projection-DID method, the mean Mean Square Error (mMSE) metric was used. To calculate the mMSE the images must be in vector form $np \times 1$, where np is the number of pixels in each image. As such the formula that is used to calculate the mMSE is:

$$mMSE = \frac{1}{N_{im}} \sum_{i=1}^{N_{im}} \left[\frac{1}{np} \sum_{j=1}^{np} \left(\mathbf{I}_i(j) - \hat{\mathbf{I}}_i(j) \right)^2 \right] \tag{17}$$

where N_{im} is the total number of images, np is the number of image pixels, $\mathbf{I}_i$ is the i^{th} original image and finally $\hat{\mathbf{I}}_i$ is the i^{th} output image of the applied method. All calculations for the mMSE are done with the images having values in the range $[0, 1]$, after they where divided by 255.

These two datasets contain only a small number of individuals compared to the datasets that an attacker would use to identify a target. It is intuitively expected that if the Projection-DID methods succeed in protecting privacy in these small datasets it will achieve even higher levels of privacy protection in large datasets.

B. Results for the PDID-O Method

This method uses formula 5 to de-identify the input images. The radius used for the PDID-O was calculated using the SVDD method. For the XM2VTS dataset the calculated radius was $R = 67.4034$ and for the Yale B dataset the value for radius R was calculated to be $R = 17.4241$.

In order to test the above radii in respect to error rates and visual quality, other values where also used in the experimental process. For the XM2VTS dataset Table I summarizes the results for different radii and classifiers. As it can be seen more values where selected near the calculated radius in order to assess the effectiveness of the calculated radius. Visual results can be seen in Figure 3 and Figure 4.

For the XM2VTS dataset the results are presented in Table I from which we can conclude that parameter R plays a large role in the error rates that are displayed by the error rates, as well as the mMSE. As suspected increasing radius R reduces the error rates displayed by the classifiers. For a radius of 10 very high error rates are observed reaching 97.36% for the NBC classifier and with an mMSE of 0.06046. Increasing the radius leads to a decline of the mMSE while error rates remain almost the same for a radius $R = 30$ and slightly falling by about 3% for radii $R = 50$ and 70. For a radius with a value of $R = 100$ error rates fall sharply to 49.06% for the KNN classifier and for $R = 120$ the same error rate is 26.04%. The mMSE is also reduced from 0.06046 for $R = 10$, to 0.02829 for $R = 70$ and reaches 0.01216 for a radius $R = 120$.Focusing on the values near the calculated value of $R = 67.4034$ and more specifically from 50 to 80 it can be observed that although the mMSE varies, the error rates remain stable for all three classifiers. The error rate is 90.57% for the KNN and NC classifiers, while slightly higher for the NBC classifier at 93.58%, both being high enough to offer privacy protection. From the results in Table I we can conclude that the calculated radius R by the SVDD method is a really good choice for de-identifying facial images and retaining an acceptable level of quality for this dataset and the PDID-O method. From these results, we propose the value of 70 for radius R for the XM2VTS dataset since $R = 70$ provides high error rates and acceptable image quality. Finally it can be verified from the results that increasing radius R causes a decline in error rates for all classifiers also for the mMSE, as we approach the initial image by increasing the radius R of the hypersphere.

For the Yale B dataset the radius R that was calculated using the SVDD method has the value $R = 17.4241$. For this R and radii in the same area, the error rates are shown in Table II. As can be seen for a small radius $R = 10$, error rates for all classifiers are high. Increasing the radius leads to low error rates for the KNN classifier, while the NBC and NC classifiers display high error rates. This observation mean that the radius that is computed using the SVDD method

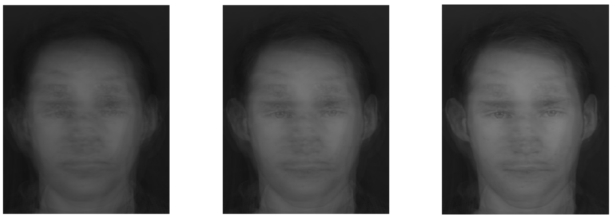

Fig. 3. Results for PDID-O with Left: $R = 10$, Middle: $R = 30$, Right: $R = 50$

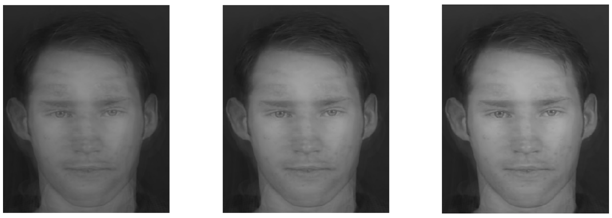

Fig. 4. Results for PDID-O with Left: $R = 70$, Middle: $R = 100$, Right: $R = 120$

TABLE I

ERROR RATES FOR PDID-O (XM2VTS)

	Classifiers			mMSE
Radius	KNN	NC	NBC	
10	93.21 %	93.21 %	97.36 %	0.06046
30	93.21 %	93.21 %	93.58 %	0.04818
50	90.57 %	90.57 %	93.58 %	0.03746
60	90.57 %	90.57 %	93.58 %	0.03268
67.4034	90.57 %	90.57 %	93.58 %	0.02939
70	90.57 %	90.57 %	93.58 %	0.02829
80	90.57 %	90.57 %	93.58 %	0.02428
100	49.06 %	48.30 %	61.89 %	0.01745
120	26.04 %	26.04 %	54.72 %	0.01216

is a good estimate of the radius that should be used in order to de-identify the images sufficiently. For the selected radii the mMSE displays at first a decline from $R = 10$ to $R = 17.4241$ and then increases. In this case the estimate by the SVDD method is not ideal and a smaller radius should be used to attain high de-identification rates. As such we propose a value of $R = 10$ for the Yale B dataset.

In both datasets apart from simply using the original images the LDA method was applied. The results gave varying error rates that where either slightly higher than the ones with the original images and some where lower. In the case of the XM2VTS dataset the images where resized to 40×30. In this case the radius R calculated with the SVDD method was $R = 0.9819$. For this radius the NBC and NC classifiers gave the same error rates at with the original images and the ones with LDA giving 96.23% and 93.21% respectively. The KNN classifier showed error rates at 93.21% for the initial images and 91.32% for the LDA. For the Yale B dataset and a radius of $R = 10$ the NC classifier displays the same error rates at 79.17%. In the case of the NBC classifier the error rate increases if LDA is used from 72.61% to 87.14%. Finally for the KNN classifier there is a drop from 89.96% to 79.50% which is still an acceptable de-identification rate.

TABLE II

ERROR RATES FOR PDID-O (YALE B)

	Classifiers			mMSE
Radius	KNN	NC	NBC	
5	94.94 %	94.19 %	92.94 %	0.04760
10	89.96 %	79.92 %	72.61 %	0.02878
15	60.83 %	88.13 %	82.57 %	0.02038
17.4241	48.30 %	90.37 %	86.14 %	0.02005
20	38.67 %	91.95 %	89.38 %	0.02239

C. Results for the PDID-M Method

This method projects the input image on a hypersphere centered on the mean image using formula 6. The radius calculated using the SVDD method did not provide adequate de-identification with the PDID-M method and the radii used here found empirically. For the XM2VTS dataset the radius proposed is $R = 10$ and for the Yale B dataset $R = 2$. This is a drawback of this method, since the radii cannot be calculated automatically. Error rates for the XM2VTS dataset can be are displayed in Table III and visual results can be seen in Figure 5 and Figure 6. From the results in Table III it can be seen that the PDID-M method gives high error rates with lower mMSE compared to the PDID-O method. From a $R = 4$ with error rates at 96.23% for all classifiers a slight drop is displayed up to a radius of $R = 10$ for which value the error rates are 90.19% for the three classifiers used. Beyond this value the error rates drop sharply and for a radius of $R = 14$ the KNN classifier displays an error rate of 53.21%.

The error rates for the Yale B dataset are displayed in Table IV. For a radius $R = 1$ the KNN classifier displays an error rate at 96.21% while the NBC a much lower error rate at 88.13%. For $R = 2$ both the previous classifiers drop to 95.02% and 83.32% respectively. The NC also displays a drop in error rate from 92.61% for a radius of $R = 1$ to 89.21% for $R = 2$. The mMSE is at 0.04384 for $R = 1$ and for $R = 2$ the mMSE value drops to 0.03307. The values for the mMSE in the case of the Yale B dataset are close for both the PDID-O and PDID-M method, unlike the case of the XM2VTS dataset as mentioned above. For higher values for radius R all error rates drop below 90%. For $R = 3$ the KNN and NC classifiers display a difference of 1% at 88.71% and 89.71% respectively, while the NBC remains almost stable in comparison with a radius $R = 2$ at 83.14% and the mMSe dropping to 0.02396. For values beyond $R = 3$ error rates drop sharply with a minimum of 76.51% for $R = 4$ and to a minimum of 66.14% for $R = 66.14\%$ both displayed by the KNN classifier.

As in the PDID-O method the LDA method was applied to the initial images. The results gave varying error rates that where either slightly higher than the ones with the original images and some where lower. As mentioned above the XM2VTS dataset images where resized to 40×30. In this case the radius used was $R = 0.8$. For this radius the NBC and NC classifiers displayed equal error rates for the original images and the ones with LDA giving 96.23%

TABLE III

Error rates for PDID-M (XM2VTS)

	Classifiers			mMSE
Radius	KNN	NC	NBC	
4	96.23 %	96.23 %	96.23 %	0.01954
6	90.19 %	94.72 %	96.23 %	0.01804
8	90.19 %	90.19 %	90.19 %	0.01660
10	90.19 %	90.19 %	90.19 %	0.01522
12	66.04 %	71.70 %	90.19 %	0.01390
14	53.21 %	53.58 %	73.58 %	0.01265

TABLE IV

Error rates for PDID-M (Yale B)

	Classifiers			mMSE
Radius	KNN	NC	NBC	
1	96.76 %	92.61 %	88.13 %	0.04384
2	95.02 %	89.21 %	83.32 %	0.03307
3	88.71 %	89.71 %	83.15 %	0.02396
4	76.51 %	89.96 %	81.74 %	0.01652
5	66.14 %	90.54 %	81.41 %	0.01075

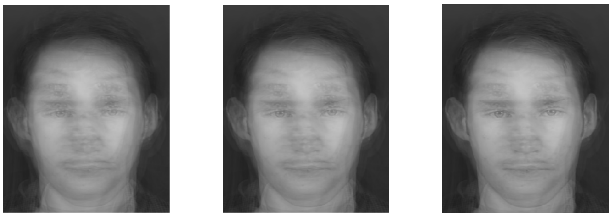

Fig. 5. Results for PDID-M with Left: $R = 4$, Middle: $R = 6$, Right: $R = 8$

and 90.19% respectively. The KNN classifier displayed error rates at 90.19% for the initial images and 96.60% for the LDA. In the case of the Yale B dataset a radius of $R = 2$ was used. The NC classifier displays the same error rates at 85.89%. Error rates of the NBC classifier the error rate increases with LDA from 82.49% to 89.79%. Finally for the KNN classifier error rates from 94.52% to 89.21%.

VI. Possible Attack Against The Projection-DID Methods

As can be seen in the above figures a malicious user trying to defeat the Projection-DID methods could use an averaging filter in order to reduce the ghosting effects introduced by this method and then use a sharpening method in order to increase the correct classification of the classifiers, bringing the output image closer to the initial image. Various low pass filter sizes were used and sharpening filters and the error rates of the classifiers did not diverge from the high error rates reported above. As a result such an attack does not defeat the proposed methods.

VII. Conclusions

In this paper we proposes two methods that de-identify facial images using projections on hyperspheres. In order to calculate a good radius R for the PDID-O method to define the hypersphere the SVDD method was used. The radii given by the SVDD gave radii values that provided high error rates and at the same time acceptable image quality. Error rates where high, attaining 93.58% for the XM2VTS dataset using the Naive Bayes Classifier and the radius $R = 67.4034$. For the Yale B dataset the highest error rate was 92.12% with the Nearest Centroid Classifier and a radius $R = 17.4241$. In the case of the PDID-M method, the radii given by the SVDD did not provide adequate de-identification so the values where selected empirically. The highest error rates with the proposed radii where 90.19% for $R = 10$ for the XM2VTS dataset and 95.02% for $R = 2$ for the Yale B dataset. Comparing the two proposed methods it can be seen that the PDID-M method performs better compared to the PDID-O method. For simlar values of mMSE (about 0.012) the minimum error rate is 26.04% for the PDID-O method and 53.21% for the PDID-M method which is more than double the error rate for PDID-O. To summarize, from the above results it can be concluded that the two proposed Projection-DID methods serve the purpose of providing privacy protection by attaining high error rates from classifiers and providing an end image that can be characterized as acceptable for everyday use.

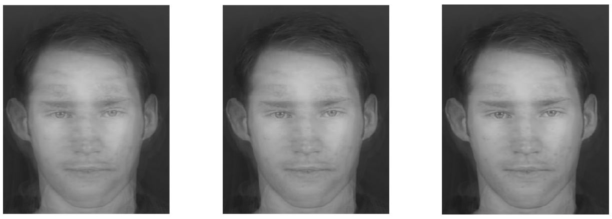

Fig. 6. Results for PDID-M with Left: $R = 10$, Middle: $R = 12$, Right: $R = 14$

References

[1] E.Newton, L.Sweeney and B.Mali, ”Preserving Privacy by De-identifying Facial Images”, *in IEEE Transactions on Knowledge and Data Engineering*, 2005, pp. 232-243.

[2] R. Gross, L. Sweeney, J. Cohn, F. de la Torre and S. Baker, ”Face De-Identification”, *in Protecting Privacy in Video Surveillance*, 2009, pp. 129-146.

[3] S. Mosaddegh, L. Simon and F. Jurie, ”Photorealistic Face de-Identi cation by Aggregating Donors’ Face Components”. *in Asian Conference on Computer Vision*, 2014, Singapore, pp.1-16

[4] B. Driessen and M. Drmuth, ”Achieving Anonymity Against Major Face Recognition Algorithms”, *in Cryptology ePrint Archive*, 2013.

[5] P.J. Phillips, ”Privacy Operating Characteristic for Privacy Protection in Surveillance Applications”, *in Audio- and Video-Based Biometric Person Authentication*, 2013, pp. 869-878.

[6] D.M.Y. Sommerville, *An Introduction to the Geometry of n Dimensions*, Methuen, Dover, New York; 1958, pp. 135-137.

[7] E.W. Weisstein, ”Hypersphere”, *MathWorld, A Wolfram Web Resource*, 2014, `http://mathworld.wolfram.com/Hypersphere.html`.

[8] E.W. Weisstein, ”Ball”, *MathWorld, A Wolfram Web Resource*, 2014, `http://mathworld.wolfram.com/Ball.html`.

[9] S. Theodoridis, K. Slavakis and I. Yamada, ”Adaptive Learning in a World of Projections”, *in IEEE Signal Processing Magazine*, 2011, pp. 97-123.

[10] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, England; 2004, pp. 244.

[11] D.M.J. Tax and R.P.W. Duin, ”Support Vector Data Description”, *in Machine Learning*, vol. 54, 2004, pp 45-66.

[12] A. Georghiades, P. Belhumeur and D. Kriegman’s, ”From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose”, *in PAMI*, 2001.

[13] K. Messer, J. Matas, J. Kittler, J. Luettin and G. Maitre, ”XM2VTSbd: The Extended M2VTS Database”, *in Proceedings 2nd Conference on Audio and Video-base Biometric Personal Verification (AVBPA99)* Springer Verlag, New York, 1999.