Face Anonymisation Techniques

Germans Savcisens[1]

Abstract: As we browse through the internet, we leave many pieces of information that can be used to identify or impersonate us. The project provides an overview of the most common techniques for face image anonymisation. At the same time, it evaluates six implementations of different techniques (ad-hoc and model-based) on tasks, such as naive, reverse and parrot recognition. The results of the experiment show that model-based techniques are more successful, when it comes to the providing good security preservation, while keeping decent utility of the anonymised data record.

Keywords: face recognition; anonymisation; privacy; biometrics

1 Introduction

As the information becomes more available, the privacy becomes of a greater concern. We browse through the internet, interact with different content and share our daily updates, as the result, we leave bunch of privacy-sensitive information online, including the data that is related to our biometric characteristics. This information can be abused to identify or impersonate people. One of the most unique and sensitive types of biometric data is our face.

The goal of the project is to make an overview and implement existing and publicly available face anonymisation techniques. The techniques in question are going to be evaluated on their ability to remove privacy-sensitive information, along with their ability to retain privacy-insensitive information.

2 Related Works

Over the years different methods for face anonymisation have been proposed. The earlier studies focused only on **security preservation**, i.e. *removing as much of the identifying information as possible* [Gr06]. As the result, the anonymised images minimise the probability of a correct recognition without preservation of the privacy-insensitive features. Thus, the output result might not even resemble a face as such (e.g. extreme case is a blackout method).

[1] Technical University of Denmark, DTU Compute, Richard Petersens Plads Building 324 2800, Denmark
s170279@student.dtu.dk

Fig. 1: Ad-hoc techniques: (1) blockout (2) eye mask (3) threshold (4) blur (5) multichannel noise (6) pixelation

However, more recent studies are taking into account the **utility** of the image, i.e. *preserving privacy insensitive features of the original image* [Du14]. Those features are related to facial expressions, skin condition [Gr06], gender, race, age [Du14] etc.

The goal of these studies is to develop a technique that can provide an optimal trade-off between **security preservation** and **utility** of the image [Mu13].

2.1 Ad-hoc techniques

The common methods that do not take into account the **utility** are the so-called *ad-hoc techniques* (examples are presented on Figure 1):

1. **blackout**: face is obscured by the blob, usually a black rectangle [NSM05]
2. **mask**: some region of the face (e.g. eyes, mouth) is obscured by the blob [NSM05]
3. **threshold**: converts pixels to either black or white based on the threshold value [NSM05]
4. **blur**: face is smoothed with a filter with large *strength* (e.g. Gaussian filter) [Gr06]
5. **noise**: randomly change values of random pixels on the image
6. **pixelation**: reducing the resolution of the image region [Gr06]

Blackout and mask techniques are the simplest and yet most reliable when it comes to the security preservation. Both can indeed decrease the probability of correct identification up to 100 % for both naive and reverse recognition [NSM05] (recognition types are explained later). At the same time, random noise that obstructs more that 50 % of the face can also decrease the probability of the correct identification [NSM05].

However, techniques as pixelation, blurring and thresholding are not proved to be reliable techniques [NSM05] [Gr06]. Both performed poorly within the *naive* and *reverse* recognition. Despite the performance of these methods they are still actively used . For example, blurring is used to remove the privacy sensitive information on Google StreetView [Go19].

Fig. 2: Model-based techniques: (1) k-same-pixel k=7 (2) k-same-eigen k=7

2.2 Model Based

One of the earlier model based techniques is a **k-Same** algorithm [NSM05]. It still mostly focused on the **security preservation**, but it did tackle the issue of the **utility** as this technique allows to preserve more information than the ad-hoc ones [NSM05].

The idea is to take k closest faces from a gallery [2] and average them out to produce an anonymized image [NSM05]. As the result, the probability of the correct identification is equal to $1/k$.

The earlier version proposed two techniques **k-same-Pixel** and **k-same-Eigen**. Both methods rely on the Eigenfaces representation of face images [TP91]. Principal Component Analysis is applied on the images from gallery. It gives the eigenvectors, each of them captures some variation in the data. By considering different linear combinations of eigenvectors with highest eigenvalues, it is possible to generate face images. It is also possible to project each face image into this subspace, as well as reconstruct it back without huge loss of information [TP91].

When there is a need to anonymise the image, it is projected onto the Eigenfaces subspace. The k closest faces are found by comparing the newly projected face with the projections of known faces.

The **k-same-pixel** then averages the **original** k closest images and produces the anonymised image. While the **k-same-eigen** averages the **projected representations** of the k closest images and then reconstructs the averaged projection into an original space.

Both methods proved to bias the ability of a face recognition software and hence lower the probability of the correct identification slightly below the 1/k [NSM05] (results can be seen in Figure 2).

Both methods have several limitations. First of all, each image in the gallery should relate to only one person, otherwise, the method might become less secure [NSM05]. Secondly, it needs an already existing and processed gallery set (+ can further reduce reliability if an attacker gets hand on the gallery set). Thirdly, there is no configurable trade-off between

[2] set of biometric references

security preservation and utility of the image [Mu13], and generally it does not provide a natural looking face [Mu13].

2.3 Similarity Based

To overcome the drawbacks of the Model-Based techniques, researchers proposed a **k-Same-M** model [Gr06]. It proposes to use Active Appearance Model (AAM), i.e. parametric model that is defined by mesh fitted on top of a face. The k-same-M algorithm applies PCA not on the image, but on the mesh of the face. So given a face image and it's mesh, k-same-m algorithm outputs an average mesh, which then can be used to generate a face image [Gr06].

It is experimentally proven that k-same-M method can keep the same level of **security preservation** as Model-based methods, at the same time, it is better at preserving the **utility** of an image [Gr06] (only the facial expression), and generally, anonymised faces are looking more natural (e.g. the ghosting effect does not appear) [Gr06]), but still produces artefacts on the anonymised images.

At the same time, it still requires that each image in the gallery is related only to one person. It also does not allow to control for the security preservation vs utility trade-off [Mu13]. More than that, it requires a gallery of images with a predefined (handcrafted) AAM meshes [Gr06].

Further proposed method to improve the k-same-M algorithm allows to have multiple images of the same person in a gallery [Mu13]. Instead of calculating the average in respect to the new image, it is suggested to (1) randomly generate a vector in the PCA subspace, (2) find it's k neighbours, (3) confirm if the projection of the new image is in the subset of the k-neighbours, (4) if confirmed, calculate the average.

2.4 Neural Networks

The latest face anononymisation works involve Deep Learning, specifically Generative Adversial Networks (GANs).

In comparison to the k-same techniques, the GAN methods provide significantly better **utility** [WYL18] [Du14].

One of the projects [RJLR18] developed a GAN based neural network that was able to anonymise faces on the video recordings. The advantage of this model is it anonymizes faces, but preserves the action of the person. As the result, it achieves good results on the action detection benchmark [RJLR18], but it fails to preserve age, gender and color.

Another projects [Du14] [WYL18] implement and train a GAN NN that removes identification features of the face with the possibility to preserve race, gender and age of the

person. It produces high-resolution images with lower probability to introduce any artefacts [WYL18] (comparing to the k-same based techniques).

3 Experiments

In this paper the following techniques are going to be benchmarked: (1) **pixelisation**, (2) **blur**, (3) **multichannel noise**, (4) **k-same-Pixel**, (5) **k-same-Eigen**, (6) **k-same-PartialM**. Only the **security preservation** aspect of each method is considered. The **utility** benchmark is out of scope of this paper; however, some consideration will be given to it.

3.1 Database

Subset of FERET database [Ph98] is used to benchmark the face anonymisation techniques. Due to requirements of the k-same-pixel and k-same-eigen methods only one frontal image per person is used. As the result, **147 frontal face images** are used. All images were converted to a *grayscale*, as well as *normalised* to account for illumination. Further, the images were resized to *180x240*. However, the results can be extended to the original images without any loss of generality.

3.2 Face Recognition Software

One of the most widespread Face Recognition packages is `face_recognition` [Ge18]. It is a publicly available and easily configurable implementation of a system for `Python`.

3.3 Security Preservation

To evaluate the performance of the anonymisation technique within the **security preservation**, the following cases are usually considered:

1. **Naive Recognition**: Matching of original images to the anonymised images [NSM05]. In this setup gallery consist of original images, the anonymised images are matched against the gallery.
2. **Reverse Recognition**: Matching of anonymised images to the original images [NSM05]. In this setup gallery consist of the anonymised images, the original images
3. **Parrot Recognition**: Matching of anonymised images to anonymised images [NSM05][3]

[3] (both probe and reference sets contains images produced by the same anonymisation technique, with same configuration).

Fig. 3: Blur results: (1) k=10 (2) k=25 (3) k=50 (4) k=90

Fig. 4: Pixelisation results: (1) k=94 (2) k=97 (3) k=99 (4) k=100

The evaluation considers all three scenarios.

3.4 Implementation

3.4.1 Ad-hoc

Anonymisation techniques such as **pixelisation**, **blur** and **multichannel noise** share one implementation. When presented with a face image, the script finds a bounding box of a face using the Haar Cascade (frontal). For all the methods, only pixels within this bounding box are processed and substituted back into the image.

For the **blur**, the script uses a build in `blur` function. For each face image, the script generates 100 blured images with different strength, k, (kernel size), e.g. first image is blurred with `1x1` kernel (identical to the original image), next produced image is blurred with `2x2` kernel and so on (see Figure 3).

For the **pixelisation**, the region of the face is downsampled to be represented in (k) by k pixels. Again, for each face it generates 100 pixelated images with different strength, k; starting from 100x100 and ending up with 1x1 representation (see Figure 4)

For the **multichannel noise**, some k % of the pixels in the face region, where reassigned a random value. At a chosen pixel, a random value was assigned to the red, green, blue channel. For each face, 100 anonymised images with noise ratio k were generated; starting from 1 % and ending up with 100 % coverage (see results in Figure 5).

Fig. 5: Multichannel Noise results: (1) k=40 (2) k=50 (3) k=75 (4) k=95

3.4.2 Model-Based

[4] **k-same-pixel** and **k-same-eigen** were implemented based on the algorithm described in [NSM05]. Implementation does not completely follows the proposed outlines, some implementation details are inspired by the [Ma18] article. The workflow is divided into two parts: *Application of PCA* and *Anonymisation*. The workflow for the Application of PCA:

1. For each image in the gallery a mask is applied (to remove variation in the background).
2. Each face is reshaped into the 1D array.
3. Arrays are standardised around 0 with the standard deviation of 1, and the average face is saved
4. With the use of `scikit_learn` package, the `IncrementalPCA` is applied and the eigenvectors with the 30 largest eigenvalues are kept (the value is found by performing several trials; it does not tend to produce artefacts).
5. Arrays are projected into an Eigenface subspace and saved after.

The workflow for the Anonymisation:

1. The new image is standardised and reshaped into 1D array.
2. The vector is projected into the subspace
3. Euclidean Distance is calculated between the vector and all the projections (from the gallery).
4. *k* closest vectors are chosen. In case of the **k-same-pixel**, the corresponding *k* original 1D vectors are averaged. For **k-same-eigen**, the projected vectors are averaged. The averaged vector is then reconstructed from the Eigenface subspace to the original Image Space. The average vector is added to the reconstructed vector along with scaling by the standard deviation

[4] Examples of produced images can be found in Appendix A

5. Finally, the vector is reshaped to the original shape.

Meanwhile, the **k-same-PartialM** is inspired by the **Enhanced k-same-M** [Mu13]. It does *not* utilises the Active Appearance Model part; thus, operates as **k-same-eigen**,except the averaging step. Thus, after the calculation of the Euclidean Distances:

1. It sets up a threshold: $t_{dist} = max(distance) * 0.45$ (the value is chosen after several trials),
2. It then randomly picks a vector and checks whether the Euclidean Distance is below the threshold,
3. If the requirement of the previous step is satisfied. It picks k closest neighbours around this randomly picked vector and follows the workflow for the **k-same-eigen** (see above).

3.5 Evaluation

To report, evaluate and compare the **security preservation** capabilities of each method the **Detection Error Trade-off** (DET) curve [Ma97] [Na17] is used.

To further explore the performance of each method the plots of **Equal Error Rate** vs *Method Strength* are going to be explored. As well as closer look at the **False Matching Rate** and **False Non-Match Rate**.

During the evaluation of the **Naive Recognition**, the two optimal configuration of each technique will be chosen. To account for **utility** of an image: the Failure-to-Acquire rate is taken into account (i.e. under the assumption that if face-recognition software is unable to acquire a face on the image, the anonymised face has low utility). Model configurations are chosen per criteria: (1st configuration) highest EER with Failure-To-Acquire (FTA) Rate below 50%, (2nd configuration) lowest value of recall + FTA. Assumption is that a technique should minimise the True Matching Rate, *min(recall)*, as well as to provide high utility of the face, *min(FTA)*.

For Naive Recognition statistics is generated for each Strength k (from 1 to 100). However, for the Reverse and Parrot Recognition, the statistics is generated for subset of k (e.g k = 10,20 ,30 ...). Thus, the optimal configurations that are found during evaluation of Naive Recognition might not have statistics in Reverse or Parrot Recognition. In that case, closest configuration will be used instead.

4 Results

4.1 Naive Recognition

The DET curves for each method can be found in Appendix B. Based on the 1st Criteria, the following optimal configurations were chosen: (7) k-same-pixel with 72 neighbours, (8) k-same-eigen with 72 neighbours, (9) k-same-PartialM with 65 neighbours, (10) Blur with the kernel size 17x17), (11) Pixelisation with the kernel size 17x17 and (12) Multichannel Noise with 9% noise ration. The plot of EER vs Strength of the technique can be found in Appendix C.

Based on the second criteria, the following optimal configurations were chosen: (1) k-same-pixel with 36 neighbours, (2) k-same-eigen with 24 neighbours, (3) k-same-PartialM with 21 neighbours, (4) Blur with kernel size 18x18, (5) Pixelisation (18x18) and (6) Multichannel noise (15%). The Cost vs Strength plot can be found in Appendix C.

The comparison of the optimal models is displayed in Figure 6. It can bee seen that k-same techniques are clustering way above the ad-hoc methods. The EER per technique can be seen in the Figure 7. The top three performing models are k-same chosen by 1st criteria (EER with FTA below 0.5), then goes k-same models chosen by 2nd critera. All the ad-hoc techniques have considerably lower EER, meaning the face-detection software was more successful in those cases.

4.2 Reverse Recognition

The DET Curve for Reverse Recognition can be found in Appendix D. At the same time, the comparison of Equal Error Rates can be seen on Figure 8a. In this case, ad-hoc methods are generally better at tricking the recognition system. At the same time, if we look at the FMR values (see Figure 8b, it can be seen that Pixelation and k-same-pixel and k-same-eigen producing high amount of False Matches. The FNMR varies within 0.9 and 1.0 (not displayed).

4.3 Parrot Recognition

Only False Match Rate and False Non-Match Rate are displayed for this type of recognition. As some techniques provide perfect separation, the rendered DET curve becomes less informative. Figure 9a displays FNMR per technique and Figure 9b displays FMR per technique.

It can be seen that ad-hoc models generally produce less false matches and almost no false non-matches. At the same time, Model Based techniques produce high values in both cases. Meaning that an anonymised face

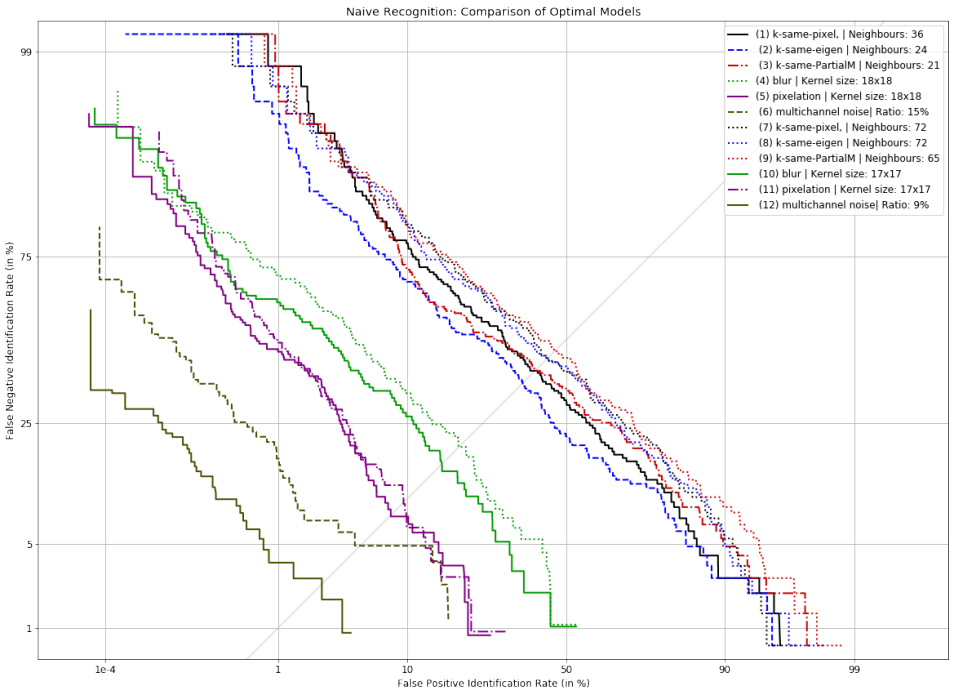


Fig. 6: Naive Recognition: DET curve for the optimal models

5 Discussion

The ad-hoc techniques performed poorly on the Naive Recognition Task, providing twice as low EER score as Model based techniques. At the same time, it did outperform the Model Based techniques in the Reverse Recognition Tasks. It provided high EER, high FMR and high FNMR, which as a result might be a good thing, as the system often will output multiple matches with a small probability that a true identity is within those matches.

However, the good performance in Reverse Recognition can be influenced by the fact that ad-hoc methods generally had high failure-to-enroll rate (see Appendix E). E.g. due to the fact that, On the other hand, the model-based methods tend to maintain below average FTE for the most time.

In Parrot Recognition, the ad-hoc techniques generally provided high FNMR and quite low FMR. Which again is a good thing, as the images are not matched to the correct person, instead the system will generate candidate list with false matches. However, again it can be due to the fact that FTE rate is high. For example, FTE rate for Blur with a 20x20 kernel is already above 50%. Thus, even thought the **security preservation** is high, the **utility** of the anonymised image is low. More than that, [NSM05] paper shows that if you just compare

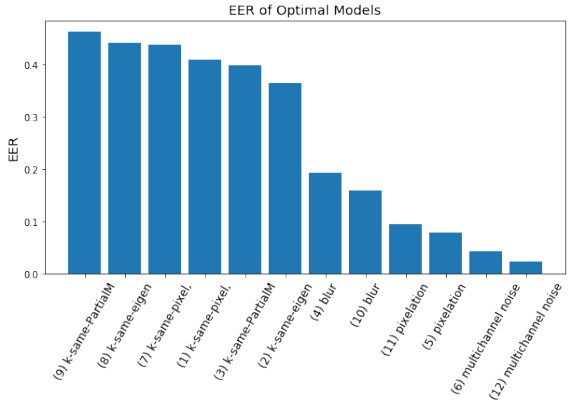
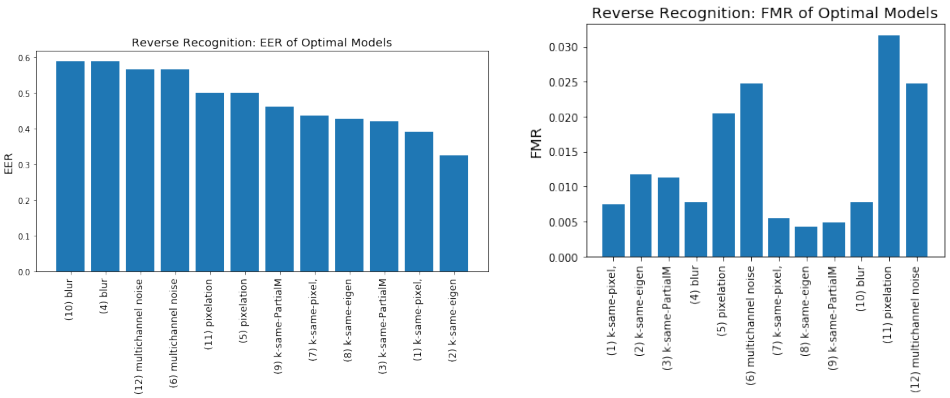


Fig. 7: Naive Recognition: EER per optimal technique (sorted)



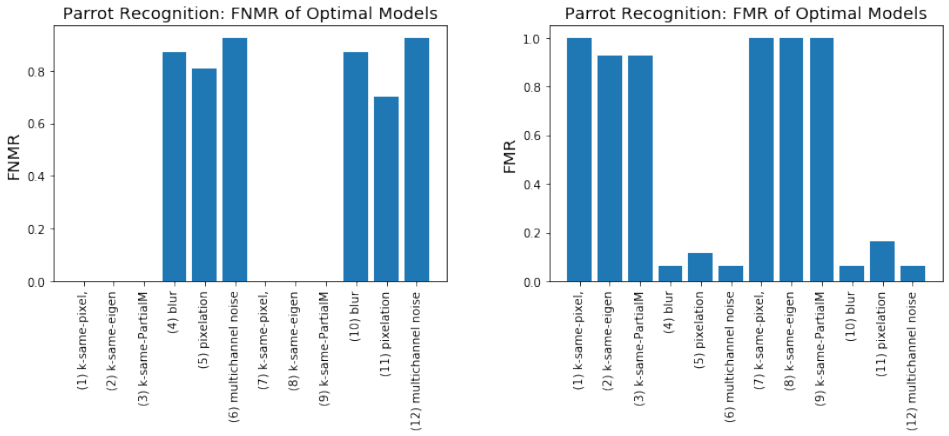
(a) EER per optimal technique (sorted)

(b) False Match Rate per optimal technique

Fig. 8: Reverse Recognition

blurred or pixelated images (without the use of the face recognition software, i.e. without trying to aquire and the face), the FNMR is significantly lower. Meaning it is easy to match images, as the pattern/image produced is more or less unique (even thought it does not resemble a face).

Based on FNMR, the model-based techniques seem to be correctly matched in *all* cases, which might imply that the k-same techniques would not be able to stand this type of attack. However, we can see that False Match Rate is close to 1, meaning that the candidate list will contain most of candidates in the reference database. As such the model-based techniques can preserve the privacy in that case.



(a) FNMR per optimal technique (sorted)

(b) FMR per optimal technique

Fig. 9: Parrot Recognition

When it comes to comparison of the k-same techniques. The **k-same-PartialM** is performing slightly better, it also needs less number of neighbours to achieve same metrics as other k-same implementations. This might be due to the randomness of the anonymisation. For small amount of neighbours it does not guarantee to be close to the original image, but as you increase the k it gets more stable. Meanwhile, for large k, the images produced by all of the k-same techniques gets more and more similar to each other.

6 Conclusion

All in all, Model-Based technique seem to provide good performance in Naive Recognition and Parrot Recognition. It is outperformed by the ad-hoc methods in Reverse Recognition, in terms of **privacy preservation**.

However, if we also take into account the importance of image **utility**, the Model-Based techniques are indeed more superior in anonymising face images.

References

[Du14] Du, Liang; Yi, Meng; Blasch, Erik; Ling, Haibin: GARP-face: Balancing privacy protection and utility preservation in face de-identification. In: IEEE International Joint Conference on Biometrics. IEEE, pp. 1–8, 2014.

[Ge18] Geitgey, Adam: , Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning, Nov 2018.

[Go19] Google: , What is Street View?, 2019.

[Gr06] Gross, Ralph; Sweeney, Latanya; De la Torre, Fernando; Baker, Simon: Model-based face de-identification. In: 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06). IEEE, pp. 161–161, 2006.

[Ma97] Martin, Alvin; Doddington, George; Kamm, Terri; Ordowski, Mark; Przybocki, Mark: The DET curve in assessment of detection task performance. Technical report, National Inst of Standards and Technology Gaithersburg MD, 1997.

[Ma18] Mallick, Satya: , Eigenface using OpenCV (C /Python), Jan 2018.

[Mu13] Muraki, Tomoya; Oishi, Shintaro; Ichino, Masatsugu; Echizen, Isao; Yoshiura, Hiroshi: Anonymizing face images by using similarity-based metric. In: 2013 International Conference on Availability, Reliability and Security. IEEE, pp. 517–524, 2013.

[Na17] Nautsch, Andreas; Meuwly, Didier; Ramos, Daniel; Lindh, Jonas; Busch, Christoph: Making likelihood ratios digestible for cross-application performance assessment. IEEE Signal Processing Letters, 24(10):1552–1556, 2017.

[NSM05] Newton, Elaine M; Sweeney, Latanya; Malin, Bradley: Preserving privacy by de-identifying face images. IEEE transactions on Knowledge and Data Engineering, 17(2):232–243, 2005.

[Ph98] Phillips, P Jonathon; Wechsler, Harry; Huang, Jeffery; Rauss, Patrick J: The FERET database and evaluation procedure for face-recognition algorithms. Image and vision computing, 16(5):295–306, 1998.

[RJLR18] Ren, Zhongzheng; Jae Lee, Yong; Ryoo, Michael S: Learning to anonymize faces for privacy preserving action detection. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 620–636, 2018.

[TP91] Turk, Matthew; Pentland, Alex: Eigenfaces for recognition. Journal of cognitive neuroscience, 3(1):71–86, 1991.

[WYL18] Wu, Yifan; Yang, Fan; Ling, Haibin: Privacy-Protective-GAN for Face De-identification. arXiv preprint arXiv:1806.08906, 2018.

A k-same anonymised images

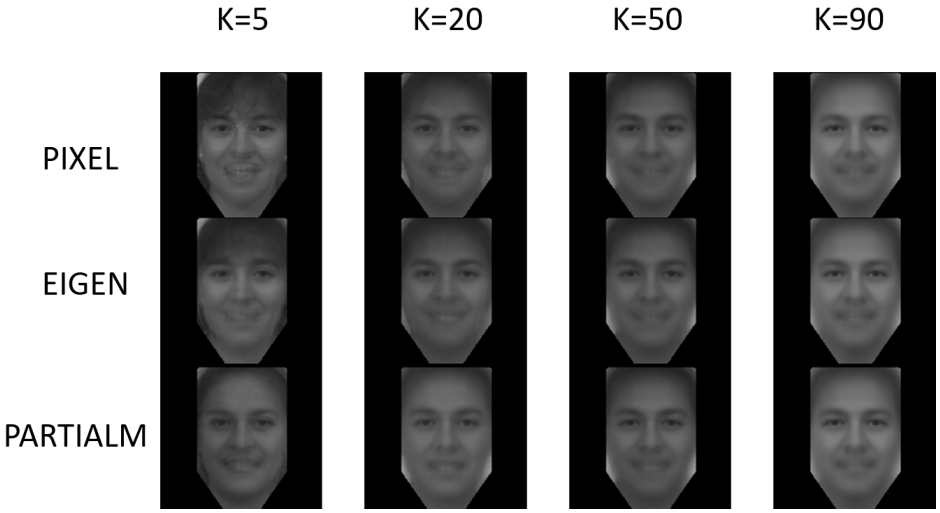


Fig. 10: Results of k-same algorithm

B Naive Recognition: DET Curves per technique

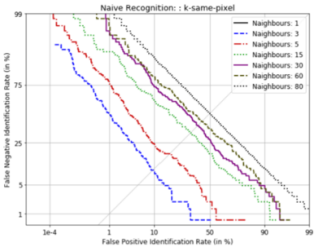


Fig. 11: Naive Recognition: k-same-pixel

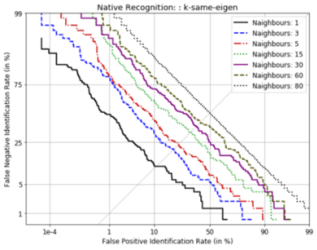


Fig. 12: Naive Recognition: k-same-eigen

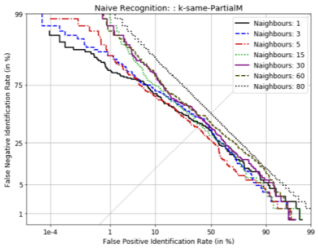


Fig. 13: Naive Recognition: k-same-PartialM

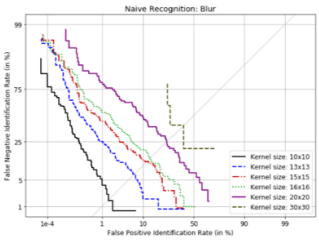


Fig. 14: Naive Recognition: blur

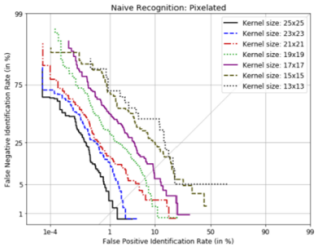


Fig. 15: Naive Recognition: pixelated

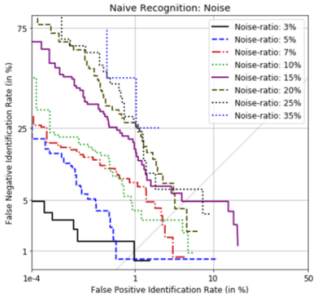


Fig. 16: Naive Recognition: multichannel noise

Fig. 11: DET Curves

C Optimal Models

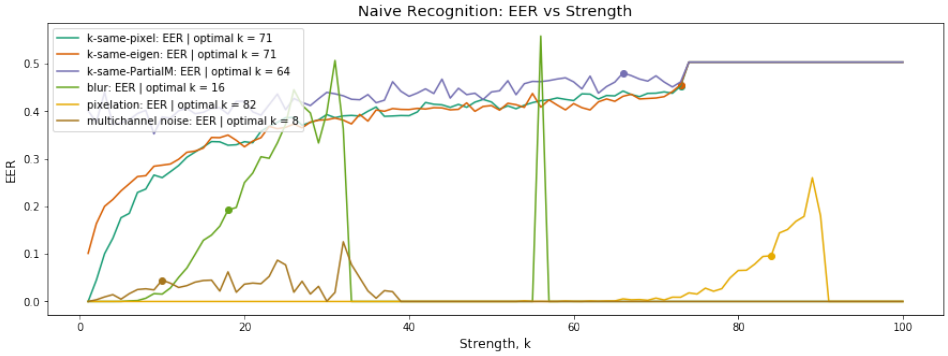


Fig. 12: 1st Criteria: EER per technique (Naive Recognition)

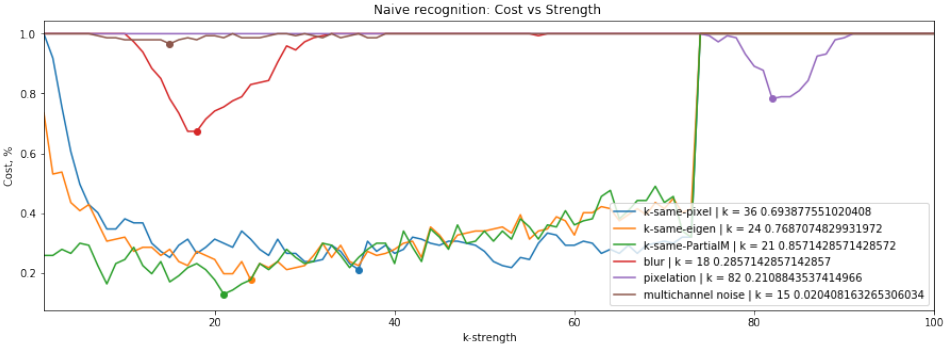


Fig. 13: Cost for techniques (Naive Recognition)

D Reverse Recognition: DET Curves for optimal Models

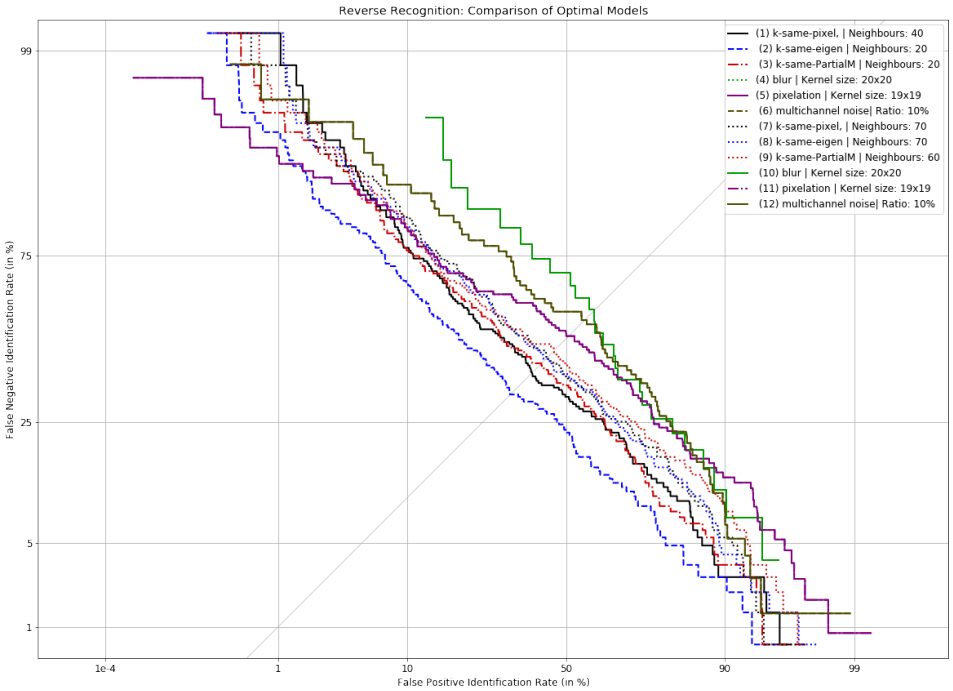


Fig. 14: Reverse Recognition: DET Curve for the optimal Models

E FTE

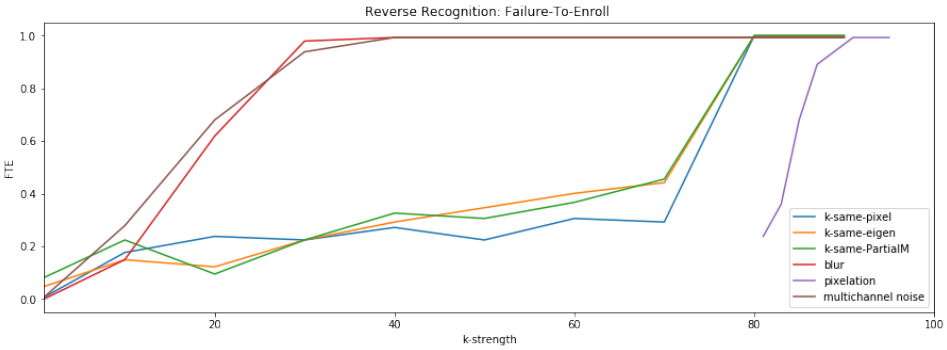


Fig. 15: Reverse Recognition: Failure-to-Enroll