Tutorials
Image Based Biometry

Žiga Emeršič, Blaž Meden, Peter Peer

Faculty of Computer and Information Science
University of Ljubljana

Content

Tutorials: Introduction, Biometric Pipeline
Image Based Biometry

Žiga Emeršič, Peter Peer

Faculty of Computer and Information Science
University of Ljubljana

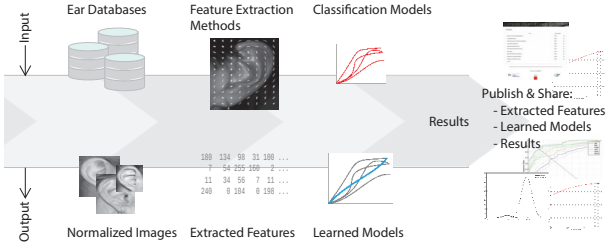
Agenda

- Introduction
- Prerequisites
- Tutorials
- Scientific Assignments
- Previous Achievements

Introduction

- Biometric recognition pipeline:
  - Why?
  - How?

Figure: Biometric recognition pipeline as illustrated by AWE Toolbox work flow diagram [1].



Prerequisites

We will be using:

- Matlab, AWE toolbox,
- OpenCV, OpenBR,
- other.

So you should already be familiar with:

- Matlab,
- C/Python,
- basics of computer vision principles.

Tutorials

Classroom work:

- $\frac{2}{3}$ tutorials:
  - $\frac{1}{2}$ presentations & discussion,
  - $\frac{1}{2}$ independent work,
- $\frac{1}{3}$ seminars.

Scientific Assignments

Work:

- 2 scientific assignments:
  - implementation,
  - written report,
- the final seminar: more in-depth research work.

Let us vote on the deadlines!

Please, be on time with your submissions.

What did previous years' students achieve?

- https://arxiv.org/pdf/1711.09952.pdf [2]
- https://arxiv.org/pdf/1708.06997.pdf [3]
- https://bit.ly/2CAH2eW [4]
- https://www.youtube.com/watch?v=4GmSdNFZ4AM
- https://www.youtube.com/watch?v=0-C0EATdLYI

Z. Emersic, V. Struc, and P. Peer, "Ear Recognition: More Than a Survey," *Neurocomputing*, 2016.

[U+FFFD] Emeršič, D. Štepec, V. Štruc, and P. Peer, "Training convolutional neural networks with limited training data for ear recognition in the wild," in *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, May 2017, pp. 987–994.

[U+FFFD] Emeršič, D. Štepec, V. Štruc, P. Peer, A. George, A. Ahmad, E. Omar, T. E. Boult, R. Safdaii, Y. Zhou, S. Zafeiriou, D. Yaman, F. I. Eyiokur, and H. K. Ekenel, "The unconstrained ear recognition challenge," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*, Oct 2017, pp. 715–724.

——, "The unconstrained ear recognition challenge," in *2017 IEEE International Joint Conference on Biometrics (IJCB)*, Oct 2017, pp. 715–724.

Tutorials: Data, Recognition Toolbox, Performance
Image Based Biometry

Žiga Emeršič, Peter Peer

Faculty of Computer and Information Science
University of Ljubljana

Agenda

- Acquiring Data
- AWE
- Performance Measures

Acquiring Data

Discussion before we begin acquiring image-based datasets:

- Different biometric modalities require different approaches of image acquisition: e.g. faces vs fingerprints.
- Different ways of acquiring images: photo sessions, the Internet, videos, ...
- After we define a biometric modality and a method of acquisition the following questions (should) arise:
  - Do we want images to be as clear as possible? Resolution?
  - Do we want occlusions, bad lightning conditions or always the best as possible?
  - How many images do we need, how many subjects?
  - Grayscale/color?

Storage & Annotation

Discussion after we acquire data:

- Are we allowed to store plain images?
- In what format are we going to store the data?
- Annotations help us with the evaluation process: either to define subsets, or to enhance feature acquisition process and decision process. What annotation attributes are we going to need?

AWE

Annotated Web Ears (AWE) Toolbox & Dataset

- `http://awe.fri.uni-lj.si`
- Written in Matlab, intended to simplify, unify and speed-up the process of ear recognition evaluation. However, you can use any other toolbox, or write the whole pipeline yourself.
- The toolbox already includes Annotated Web Ears Dataset.
- The dataset contains 1.000 images of 100 persons. Images were collected from the web using a semi-automatic procedure.

Verification vs Identification

Refer to the materials from the lectures, for the measures below. Note: this is for recognition – we also need to evaluate detections/segmentations (in the following weeks).

- Identification:
  - Ranks → CMC (Cumulative Match Curve)
- Verification:
  - False/true acceptance (positive) rate → ROC (Receiver Operating Characteristic) curve → AUC, EER; DET (Detection Error Trade-off) curve.
  - Precision, recall → f-measure: $2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$.

You can find the generated results and performance measures in the AWE toolbox in the folder */_output/results/*.

Base definitions

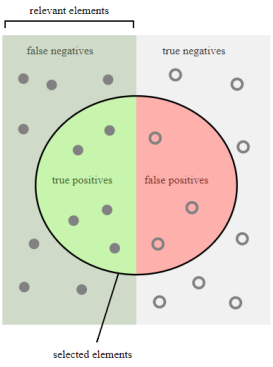


Figure: True/false negatives/positives and their relation to precision and recall (sensitivity). Source: `https://upload.wikimedia.org/wikipedia/commons/2/26/Precisionrecall.svg`.

Curves

- ROC curve: true acceptance (positive) rate vs false acceptance (positive) rate.
- DET curve: false rejection (negative) rate vs. the false acceptance (positive) rate.
- CMC curve: recognition rate vs ranks.

Receiver Operating Characteristic (ROC) Curve

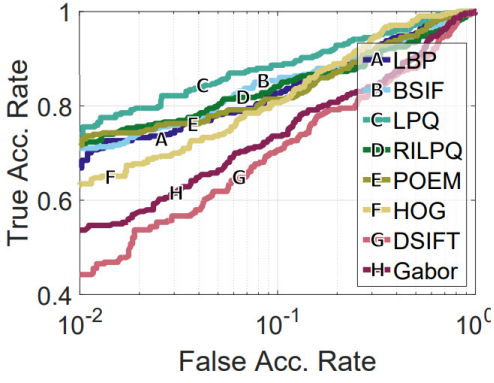


Figure: Receiver Operating Characteristic (ROC) curve plots true acceptance rate (y-axis) vs. false acceptance rate.

Cumulative Match Characteristic (CMC) Curve

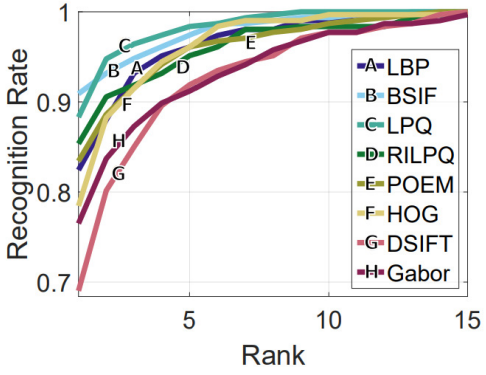


Figure: Cumulative Match Characteristic (CMC) curve plots accuracy in percentage (y-axis) for each rank (x-axis).

Tutorials: Towards Detection Image Based Biometry

Žiga Emeršič, Peter Peer

Faculty of Computer and Information Science
University of Ljubljana

Content

- Local Binary Pattern (LBP).
- Using feature extraction for both, recognition and detection.
- Detection using Viola-Jones.

Local Binary Pattern

Let's calculate LBP [1] by hand on the board! Generally these are the core ideas:

- In a circular fashion compare pixels with the selected central pixel.
- If value is below/above, store as 0/1. This string is actually a number.
- Plug these number together and you get a long feature vector.
- Improvement 1: cycle-shift each value to the largest/lowest value (local rotational invariance).
- Improvement 2: divide the image into cells, calc LBPs for each cell separately and instead of simply joining the values, calculate histograms.
- Improvement 3: instead of using uniform histogram bins, use separate bins for e.g. values with more than n $0-1$ transitions.

From Recognition to Detection

- And now for the crucial question!
- We used approaches like LPB, HOG, LPQ, DSIFT, etc., for our feature extraction and then these features served us as the basis for the biometric recognition.
- Can we use the same approaches for detection as well? Yes we can! The only question is how. Sliding window is the most trivial approach that you can use right away (however, by far not the best).

Detection

Viola-Jones [2] is arguably the most popular non-NN-based detection algorithm despite its age. It contains a few crucial advances:

- The use of Haar features to respond to feature of faces (let's draw some examples!).
- The use of integral images to drastically speed up the calculation of sub-matrices' sums (let's calculate a toy example together!).
- Adaboost is used to train the classifier and during run-time a cascade of classifiers is used.

Although the training is long and tedious, the algorithm is fast during run-time. Furthermore, trained classifiers are available for many many biometric modalities (and objects).

Viola Jones

Using OpenCV we can use pretrained VJ writing in essence only two lines: cascadeFace = cv2.CascadeClassifier("haarcascade-for-x.xml") detectionList = cascadeFace.detectMultiScale(img, 1.05, 5)

Let's program an example and observe the results!
See the attached python code.


L. Wang and D.-C. He, "Texture classification using texture spectrum," *Pattern Recognition*, vol. 23, no. 8, pp. 905–910, 1990.


P. Viola and M. Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2001.

Tutorials: Segmentation
Image Based Biometry

Žiga Emeršič, Peter Peer

Faculty of Computer and Information Science
University of Ljubljana

Content

- Segmentation Overview
- Thresholding
- K-Means Clustering
- Watershed
- Grabcut
- CNN-based

Segmentation

Many approaches toward segmentation:

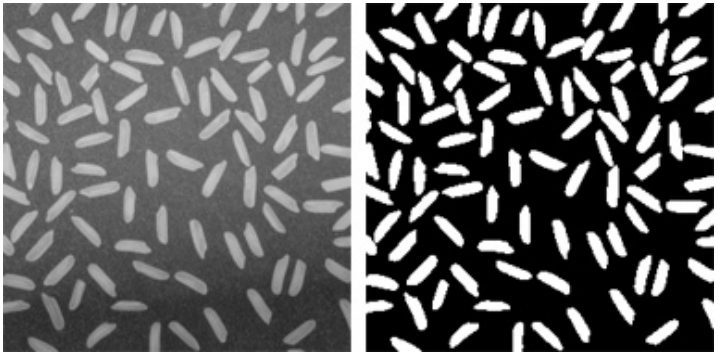
- Thresholding
- Histogram-based (using color/intensity)
- Region-based & Clustering (using color/intensity/texture)
- Edge-based
- Neural Networks
- Many other approaches: Partial Differential Equation, Watershed, Graph-based, Grabcut, ...

Segmentation

- Can we use segmentation for biometric object detection?
- Can we use object detection for biometric segmentation?
- How do we initialize segmentation target?
- Is segmentation more useful than the traditional bounding-box detection for biometric recognition?

Thresholding

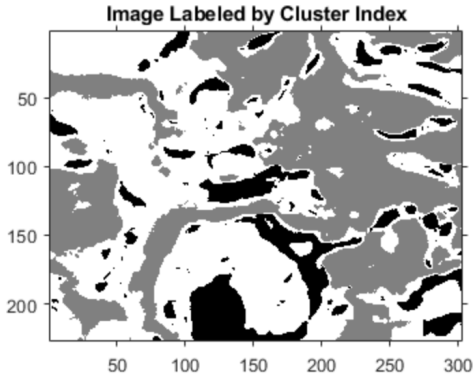
The most trivial segmentation approach – thresholding:



https://www.mathworks.com/help/images/correcting-nonuniform-illumination.html

K-Means Clustering

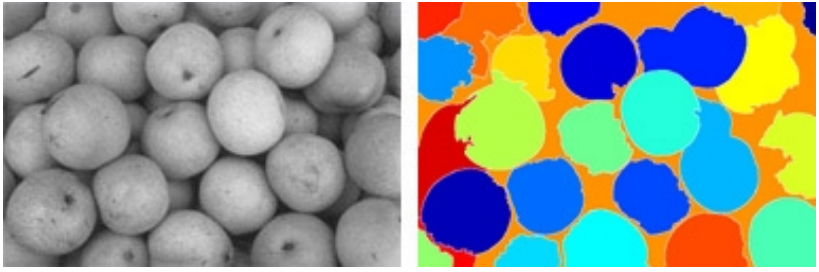
An example of clustering where color distance is used as a criteria:



https://www.mathworks.com/help/images/color-based-segmentation-using-k-means-clustering.html

Watershed

Works best if you can define observed object and "background":



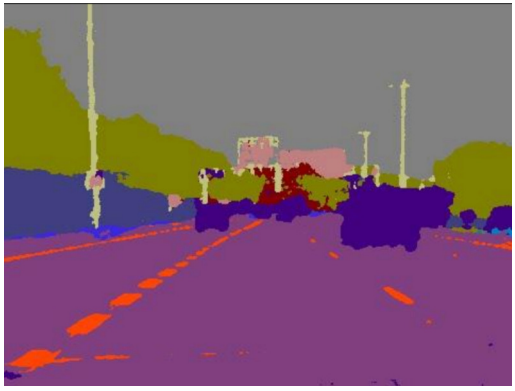
https://www.mathworks.com/help/images/marker-controlled-watershed-segmentation.html

Grabcut

https://docs.opencv.org/trunk/d8/d83/tutorial_py_grabcut.html

CNN-based

Have you seen this video?



https://www.youtube.com/watch?v=CxanE_W46ts

Tutorials: Bibliography
Image Based Biometry

Žiga Emeršič, Peter Peer

Faculty of Computer and Information Science
University of Ljubljana

Content

- Why and where to publish?
- How does publication process look like?
- How to know what literature to follow and read?

History

In the ancient times before the internet:

- Printed publications.
- Physical books and printed journals you had to go through.
- No CTRL+F!!
- How do books and journal look like today?

Publication Process

Who is who?

- Authors write the papers.
- Reviewers review the papers, but who are the reviewers?
- Editor manages the reviewers and the publishing process.
- Publisher employs editors and actually publishes papers.

Publication Process #1

Who earns/looses money?

- Authors write papers for "free" (are paid by their institutions).
- Reviewers review papers for "free" (are paid by their institutions).
- Libraries and researchers need to pay for the access to the papers (paid by their institutions).
- What about the publisher?

Publication Process #2

Who earns/looses money? (open access model)

- Authors write papers for "free" (are paid by their institutions).
- Authors pay the publisher to publish a paper (paid by their institutions).
- Reviewers review papers for "free" (are paid by their institutions).
- Libraries and researchers get free access to the papers.
- What about the publisher?

How to combat €

Two ways of fighting:

- arXiv.org: `https://arxiv.org`
- Sci-Hub (use TOR)

But are these solutions always good? Can we get rid of publishers?

The most common types

The most common types of literature:

- Journal papers.
- Conference papers.
- Books (book chapters).

Papers in computer science generally fall into two groups:

- Original research papers.
- Review papers.

Literature Rating

How to know which journals are OK? Like these for example:

- American Journal of Engineering Research
- European Journal of Industrial Engineering
- Central European Journal of Energetic Materials

Literature Rating

Journals:

- JCR: `https://plus.si.cobiss.net/opac7/jcr`
- SJR: `https://www.scimagojr.com/journalrank.php?type=p`

Conferences:

- JCR: `http://www.conferenceranks.com/`

How does Slovenian ARRS rank journals vs. conferences?

Where to search for the Literature

- https://scholar.google.com
- https://ieeexplore.ieee.org
- http://citeseerx.ist.psu.edu
- https://mendeley.com
- https://www.scopus.com (Elsevier)
- http://dblp.l3s.de/
- http://isiknowledge.com/ (Web of Science)
- ...

Tutorials: Writing Papers
Image Based Biometry

Žiga Emeršič, Peter Peer

Faculty of Computer and Information Science
University of Ljubljana

Content

- What is the IMRAD structure?
- In what order do we write?
- What goes where?
- How to write each part.
- Overview of some actual examples.

Introduction

- The content of these slides is based on personal experiences from reviewers' comments and the knowledge of more experienced colleagues. They are still only opinions.
- However, these guidelines are helpful in all technical writing, not only scientific papers.

IMRAD

- The structure of IMRAD:
- `https://en.wikipedia.org/wiki/IMRAD`
-
  - **I**ntroduction
  - Related Work
  - **M**ethod
  - Experiments, **R**esults **and** **D**iscussion
  - Conclusion
- What is the order of writing?

IMRAD

- The recommended order of writing (not obligatory):
  - (Related Work,)
  - Method,
  - Experiments, Results and Discussion,
  - Introduction,
  - Conclusion,
  - Abstract.

Tools

- A useful tool to use: `http://www.overleaf.com`
- Why LaTeX?
  - WYSWYG tools (e.g. Word/Writer) are great for writing something shorter, but they are not really made for writing papers: no word breaking, referencing more difficult, equations more difficult, table of contents more difficult, version incompatibility, everything looks worse.
  - It is not all perfect: `https://tex.stackexchange.com/questions/1756/why-should-i-use-latex`.
  - The negative aspect of difficulties compiling Tex can be removed by using online compilers.

General Tips

Writing:

- Avoid using too complicated terms.
- Avoid too complicated sentence structures or very long sentences.
- Avoid adverbs (very, etc.).
- When using an acronym make sure to explain it the first time you are using it.
- Prepare diagrams to support the explanation of your methods.
- Imagine that the reader will be the laziest person on Earth.

General Tips

- Be careful not to repeat yourself.
- The use of active/passive voice is a delicate topic. Generally stick to more active voice; it reads better and is easier to understand for the reader.
- Which is better:
- *"We use a histogram equalization to improve recognition performance."*
- *"During recognition, to improve performance, a histogram equalization is performed."*

Using Present/Past Tense

Using Present/Past Tense:

- Past tense for the related work, what you did and when describing results.
  E.g.: "Xiang showed that, ...", "We found that, ...", "We proposed smth, ...".
- Present tense for for covering conclusions/indications and for future work.
  E.g.: "The results show that, ...".

General Tips

- Make sure you reference every single figure/table/equation.
- Make figures clear, font sizes should match.
- Plots should be in vector format.
- Make images of the correct size and make them visible.
- Check for "Figure X" vs "Fig. X".
- Captions should stand on their own.
- Do not use too many borders in tables, instead make use of white space (if possible).

Writing Abstract

Abstract:

- Three parts: motivation, the approach, findings.
- Stands on its own, this means:
  - no references,
  - no acronyms (some exceptions).
- Polish, polish, polish. Often the only thing people read.

Why does the abstract need to stand on its own?

Writing Introduction

- Roughly consists of the following:
  - State: Current status of the field.
  - Problem: Presentation of the problem (what is currently missing, what is not researched yet, problematic etc.).
  - Solution: Brief description of your solution to the problem, how the solution is structured.
  - Paper structure (optional): Briefly describe what is where in the paper.
- Include an overview diagram of your method already in the introduction.

Writing Related Work

- Adapt to the target audience (e.g. field of the journal).
- Avoid describing into too much details, rather provide a reference and instead cover more work. Make sure that you describe the core idea and not trivial parts of that work.
- Try to cite the initial, original work where applicable.
- Including a tabular overview is especially useful, but not that common.

Writing Methods' Section

Methods:

- Arguably, the most important part of the paper.
- Be clear, include diagrams.
- Within each paragraph try to go from a general to the specific idea, whilst retaining one idea per paragraph.
- Title the section appropriately.
- No details about the experiments.

Writing Experiments, Results & Discussion

Results & Discussion:

- Describe experiments well and present results clearly. Your guide should be, that the reader is able to reproduce your results easily.
- Figures are for displaying a larger overview, whereas tables are for the exact values. Do we need both?
- Discuss not only through the numerical data, but also generalize, what are the trends, possible conclusions, why do you think something is, etc.

Writing Conclusion

Conclusion roughly consists of three parts:

- Summarize what you have done.
- Describe the findings through advantages and also mention limitations. This is good, not only for the community, but also you are ready for the reviewers.
- Wrap it up with what the larger implications are and what are the possible future improvements or where do you think the research will go.

Preparing References

References:

- Make sure you do not miss the latest state-of-the-art.
- Do not include papers that are not important to the work presented in the paper.
- Make sure you do not miss any of the important work that was done. Not covering work from e.g. the last two years can mean two things (both are bad):
  - you did not check the literature well enough, meaning your approach could be obsolete,
  - or actually nothing happened in the last two years, meaning the research has shifted and the whole area is obsolete.
- Make sure that styles match, i.e.: bibtex from Scholar or from IEEEXplore differ, one will contain DOI, other will not, or one will have conference name written differently than the other.

Example #1

Figure 1. Ear images

DSIFT features result in a lower performance, while the binary-pattern-based techniques again perform better and more or less on the same level [**?**]

When we focus only on the LBP we can say that this method may have some difficulties in identifying some characteristics because we have some ears that have some "obstacles" like earrings, hearing aids and in some cases where the colour of the ear is very similar to the colour in the background of the image. This method will also have difficulties if the texture of the image is not in the range that is recognized by LBP is a particular case of the texture spectrum.

Missing citation references and maybe not the best way to arrange images. To solve errors like this, search for "?" before you submit your paper.

Example #2

duce a large variation when perturbed
ıer detection algorithm is a mathematical
ing such differences in which we may even
as eigenvalues to determine how much of
in a given window. As the Harris corner
is largely invariant to rotation and scale,
to try and find all high-frequency corners
nine their relative positions. The idea is
v we take the picture of the same ear, it
corners at similar intervals to other ears
me person in our database. Harris Corner
included in the native MatLab library.

Net is the name of a specific architecture of
:al network designed to classify hundreds of
was trained on 1.2 million high-resolution
ing the top-1 error rate of about 37.5%. One
ibutors to its success was a large database,
usion of a regularisation technique called
domly disables some portion of neurons in
yer in order to prevent overfitting. As the
ork is a class and we are only interested
ld be used to describe our ears, we extract

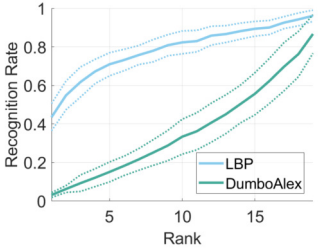


Figure 3. A CMC curve comparison of DumboAlex with LBP

The probe image is tested against the *rank*-nearest neighbours of every image in our dataset. By the CMC (Cumulative

Font sizes do not match.

Example #3

***Abstract*—In this paper we will present experimental evaluation of local phase quantization and dense scale invariant feature transform methods on AWE dataset.**

I. Introduction

Automatic identity recognition from ears has been researched increasingly in the last decade. As fingerprint and iris recognition are well researched some other methods for identification still offer a lot of potential for application.

Figure 1. This is example of an image that we predicted that classification will fail due to the occlusion and accessories.

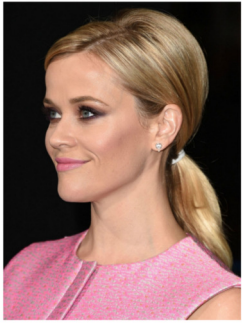


Figure 2. This is example of an image that we predicted that classification will succeed because ear is fully visible with good lightning.

B. Dense Scale Invariant Feature Transform

SIFT descriptors are invariant to scaling, translation and rotation. This descriptors are calculated on a local neighbour-

Abstract too short, images too large.

Example #4

References

1. *Histogram of oriented gradients*, Feb 2018.
2. Ziga Emersic, Vitomir Struc, and Peter Peer, *Ear recognition: More than a survey*Ear recognition: More than a survey, CoRR **abs/1611.06203** (2016).
3. *Local binary patterns*, Feb 2018.

Are these references? :)

Example #5

III. Results

A. HOG

In parameters tuning for HOG I conducted experiments on the several parameters sets.

1.Parameter set: cell size[32 32]px,block size [2 2],orientation 9,block overlapping 50%

2.Parameter set: cell size[32 32]px,block size [2 2],orientation 18,block overlapping 50%

3.Parameter set: cell size[32 32]px,block size [2 2],orientation 9,block overlapping 0%

4.Parameter set: cell size[16 16]px,block size [2 2],orientation 18,block overlapping 50%

5.Parameter set: cell size[16 16]px,block size [2 2],orientation 9,block overlapping 50%

6.Parameter set: cell size[8 8]px,block size [2 2],orientation 18,block overlapping 50%

Not very readable, is it?

Example #6

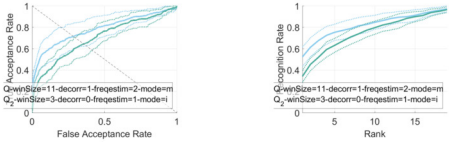


Figure 4. ROC (left) and CMC (right) curve of LPQ method.

proper annotations for the dataset.

For feature extraction it is crucial for good performance of the method to choose the right, optimal, parameters. We measured the performance using the AUC score, ROC and CMC curve. Interpretation of which gave us optimal parameters

Plot labels not readable, fonts also not consistent compared to the text.

Example #7

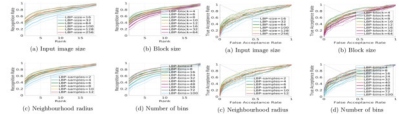


Figure 2: CMC Curves for LBP descriptor

Figure 3: ROC Curves for LBP descriptor

D. Testing

All extractors we test with AWE toolbox and dataset. In AWE dataset are 1000 images of 100 subjects. We test extractors with 5-cross-validation, which is predefined in AWE toolbox. In extractors, we first test every specific parameter and observe how it perform. Then we take some best and default values for every parameter and try different combinations. So we get the best parameter for the specific extractor. On end, we try to improve our results with sharpening image before extractions. For evaluating extractors and their parameters we use Cumulative Match Curve (CMC) for identification problem and Receiver Operating Characteristic (ROC) curve for verification problem.

III. Results

In this section, we represent the results of our evaluation. We try to test different parameters. Boundaries of parameters we decide based on if we increase parameter than performance must improve. If it not improve then we stop testing. We stop too if runtime of evaluating parameter is to slow.

First, we analyze LBP extractor. We can see results on images 3 and 4. For input image size we can see that performance improve for CMC and ROC curve and is best for size 64,100, 128. After that, it drops. Very similar happen with radius (samples) parameter. Here we get the best score at radius 6 to 10. For block size, we see that bigger than block is, worse performance is in our extractor. This probably happens because with a smaller size of the block we can compute more different descriptors and because of that we are more accurate. For a number of bins, we see that when we have more than 8 bins we get always a very similar result. It is very interesting that we get best results in CMC and ROC curve for same parameters. After our test with different parameter combinations, we decide that best parameters for LBP are size=64, block=4, bins=58, samples=8 (Image 6).

Results of our evaluation for DSIFT descriptor are represented on images 4 and 5. Here we can see that descriptor perform best on input image size of 100 and 128, grid size 90 and 100, patch size between 4 and 16, number of bins between 8 and 32. Same as for LBP we get here again best result for CMC and ROC for the same parameters. After combination test best parameters for DSIFT extractor are input size of image=128, patch size=8, grid=100, bins=8 (Image 5).

As we can see from Image 6, we improve our default result of extractors. On end, we try with a sharpening of the image.

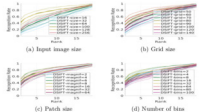


Figure 4: CMC Curves for DSIFT descriptor

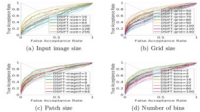


Figure 5: ROC Curves for DSIFT descriptor

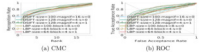


Figure 6: Best results for tested extractors and result with sharpening of image (s=1)

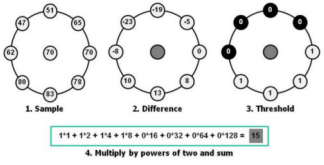
While the author tried to present as many results as possible, these plots are just not readable.

Example #8

Metoda lokalnih binarnih vzorcev (ang. local binary patterns - LBP) je preprosta a učinkovita metoda, ki označi piksle na podlagi upragovanja njihove okolice (slika 2). Vsaka oznaka je predstavljena v obliki binarnega števila. Histogram teh števil lahko uporabimo kot vektor značilk [lbp].

The value of the LBP code of a pixel (x_c, y_c) is given by:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p \qquad s(x) = \begin{cases} 1, if\ x \geq 0; \\ 0, otherwise. \end{cases}$$



Slika 2: Izračun oznake piksla z metodo LBP.

Do not use diagrams/figures from other people, especially while not citing their work. Furthermore, the screenshot of the figure is of bad quality (e.g. gray background).

Example #9


```
********RESULTS********
Rank-1           : 30.84+-5.65 (10)
EER_er           : 35.11+-8.80 (1775)
VER_01FAR_ver        : 4.18+-2.88 (1470)
VER_1FAR_ver         : 14.87+-6.03 (1535)
AUC          : 68.74+-8.42 (3525)
#min(FSc/n)      : 1026
#min(TSc/n)      : 55
#min(Sc/n)       : 1083
#Sc      : 12062
************************
********RESULTS********
Rank-1           : 41.84+-8.31 (10)
EER_er           : 32.95+-8.15 (2046)
VER_01FAR_ver        : 4.57+-4.07 (1291)
VER_1FAR_ver         : 16.49+-5.59 (1499)
AUC          : 73.65+-8.88 (3712)
#min(FSc/n)      : 1026
#min(TSc/n)      : 55
#min(Sc/n)       : 1083
#Sc      : 12062
************************
```

Figure 1. The results from implementing HOG with (32,32) an (8,8) cell size

Someone was in a hurry! :) Solution: start writing the report/paper earlier.

Tutorials: CNN Overview
Image Based Biometry

Žiga Emeršič, Peter Peer

Faculty of Computer and Information Science
University of Ljubljana

Agenda

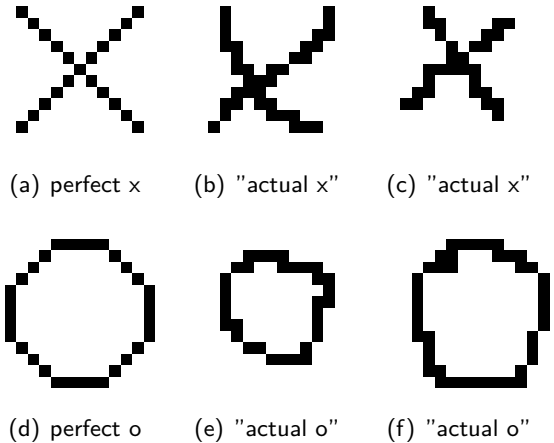
- What are Convolutional Neural Networks (CNN)?
- How do CNN work?
- When to use CNN?
- We recommend the following course: `http://cs231n.github.io/`.

Convolutional Neural Networks

In the recent years CNN present state-of-the-art in recognition, object detection and other computer-vision tasks.

However, as opposed to the classical approaches where underlying principles are well understood (think about an arbitrary feature extractor or some classification model), here we get a black-box solution that ''just works".

The Problem of Recognition



(a) perfect x (b) "actual x" (c) "actual x"

(d) perfect o (e) "actual o" (f) "actual o"

Figure: Example of two classes (class of X: a–c, class of Y: d–f).

Let's compute!

Let us manually calculate and observe the results of convolutional, pooling and ReLU layers using 7×7 toy example image with 5×5 "X" inside and 3×3 filters.

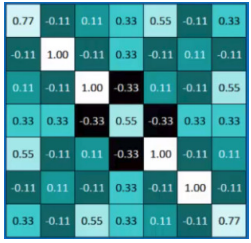


Figure: After convolving a filter.

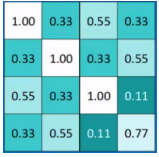


Figure: After pooling with window size 2, and stride/step 2.

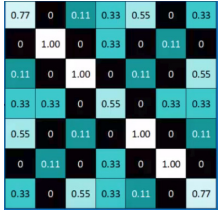


Figure: After rectified linear units.

Figure: `https://www.youtube.com/watch?v=FmpDIaiMIeA`

Let's compute!

- But CNNs are not used only for classification tasks where input is an image and the output is feature vector or class prediction.
- We also want to have detections and pixel-wise segmentations (which are actually classifications, but for every single pixel in the photo).
- We need to upsample somehow. The solution is a sort of transposed convolution (be careful, we are not talking about deconvolutions here):
  - we weigh filter with the value in the image (multiply),
  - position the result into the (larger) target image,
  - sum the overlapping regions.
- The result of this is the image with the same size (usually) as the input image.

Feature Levels

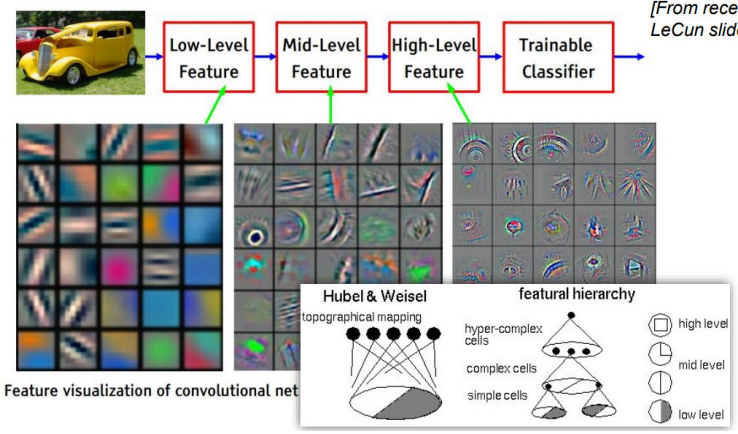


Figure: Filter responses (source: Daniel Skočaj's Deep Learning for Computer Vision slides).

CNN Structure

Some types of layers:

- Convolution Layer (no. of feature matrices, size of the matrices)
- Pooling Layer (window size, step): we shrink the image and introduce non-linearity, by selecting one value within the observed window.
- REctified Linear Units (ReLU) Layer: we floor negative values (with Leaky-ReLU we leave some room) to keep "math" under control and to introduce non-linearity.
- Fully Connected Layer (no. of neurons)

The definition of the elements in the brackets above, and the **number**, and the **order** of layers compose the architecture of the CNN.
After the training when all the weights are set, these trained values form what is known as **model**.

CNN Architecture

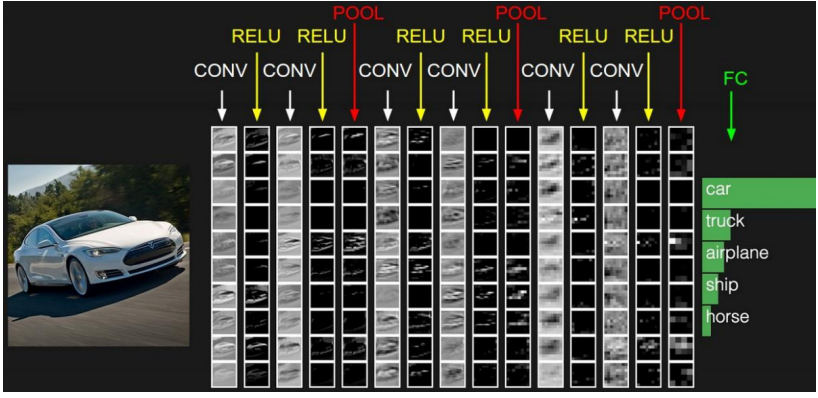


Figure: An example of a CNN architecture (Fei-Fei Li, Andrej Karpathy, Justin Johnson).

Back-propagation is used to set weights (to train the CNN):

- Errors are calculated based on ground-truths and the outputs of the fully connected layer. We measure our unhappiness with outcomes such as this one with a loss function (or sometimes also referred to as the cost function or the objective).
- For each feature pixels weight is increased and decreased – the new prediction is then again evaluated.
- * How do we know whether to increase of decrease the weight, and for how much? Gradient descent, of course!

When we are satisfied with the weights (the errors reach the satisfyingly low values) the learning process is complete.

The problem of overfitting

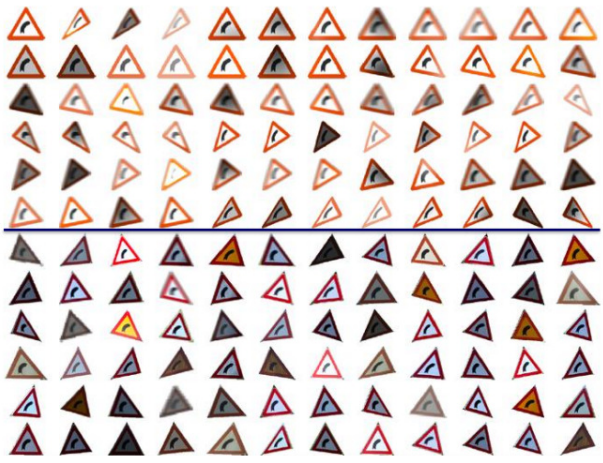


Figure: Artificial data augmentation (source: Daniel Skočaj's Deep Learning for Computer Vision slides).

Tips

- normalize data or do mean subtraction,
- use regularization: (L1, L2),
- use dropout,
- use batch normalization,
- initialize weights randomly and not to zero.

When to use CNN?

The rule of thumb: if you can convolve over you data samples and if it makes sense (you can structure your data as a matrix – an image which means you can do convolution over it), CNN could be appropriate for the task.
Some examples:

- Squirrel detection – OK, because you already have 2D images
- Song recognition – OK, if you manage to represent a song as a matrix
- Student habits of the course participation – No, why?

If your data is still useful after you swap rows (or columns) then CNN are not useful for the task.

Problems of CNN

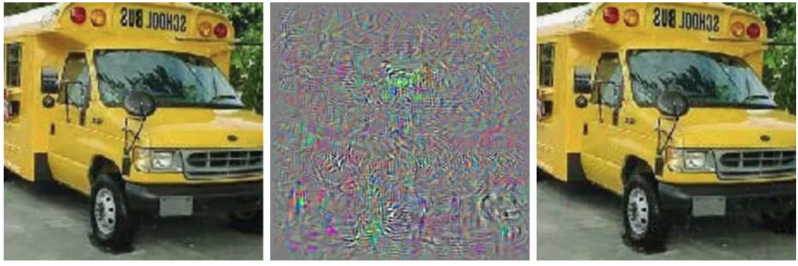


Figure: Ostrich vs. bus problem (source: `https://www.popsci.com/byzantine-science-deceiving-artificial-intelligence`).

We can trick CNNs, so that they fail spectacularly. There is no clear explanation of decisions being made.

Tutorials: CNN Hands-on Image Based Biometry

Žiga Emeršič, Blaž Meden, Peter Peer

Faculty of Computer and Information Science
University of Ljubljana

Agenda

- Deep Learning Frameworks
- Architectures
- Examples

Convolutional Neural Networks

Training convolutional neural networks, even the shallower architectures, requires GPU (preferably Nvidia's GPU with CUDA support).

If you already have a trained CNN you can use CPU for predictions, but for training use only GPU, CPU takes way too much time to be useful.

To repeat: A model is a trained network, whereas architecture is just a blueprint for how the layers are stacked (so no information on weights).

Frameworks

There are many many different frameworks available for deep learning. Often you are forced into one simply because the authors of the architecture/model you are trying to use implemented it in a specific framework.

However, if you can choose, go for the architecture with many models available and is easy to use (easy to develop and change arch, easy to debug, easy to reproduce the results).

-

Frameworks

Many possibilities:

- Caffe (`http://caffe.berkeleyvision.org/`): very popular especially in the initial hype of CNNs, the majority of models was released for Caffe.
- Keras (`https://keras.io/`): it runs on top of TensorFlow/Theano and offers an easy-to-use programming interface.
- GluonCV (`https://gluon-cv.mxnet.io/`): a framework with CNN implementations that guarantee reproducible results. Although arguably not as well developed and popular as Keras you achieve good results quickly.
- MatConvNet (`http://www.vlfeat.org/matconvnet/`): Matlab code with C (Mex) additions. Because it is in Matlab you would usually use this only if you have no other options.
- TensorFlow (`https://www.tensorflow.org/`), Theano (`http://deeplearning.net/software/theano/`), Torch (`http://torch.ch/`), etc.

Architectures

You can assemble your own architecture, but usually it is recommended that you start with an existing architecture. Moreover it is useful to start with a trained model and then finetune it for your domain. There are many options, e.g. (`https://modelzoo.co/`).

Let us see some examples of architectures, each in a different framework:

- A simple toy example for MNIST classification in Keras (the code is attached).
- SegNet in Caffe.
- RefineNet in Matlab.
- A comparison of the traditional Viola-Jones [1] vs. Single Shot Detector (SSD) [2] in TensorFlow.

MNIST Classification in Keras

MNIST is an old dataset of numbers written by hand. Using only a shallow CNN architecture you can predict a corresponding number with over 99% accuracy.

- You can use the code available at `https://raw.githubusercontent.com/keras-team/keras/master/examples/mnist_cnn.py`.
- Only add saving and loading of the train model and loading of an image and then making a prediction on it (see the attached code).

SegNet in Caffe

Architecture is defined in a files similar to JSON. While this is great for portability and generalization, this type of definition is unclear if you want to modify it, add connections etc. (help with a visualization tool), as shown in the figure below:

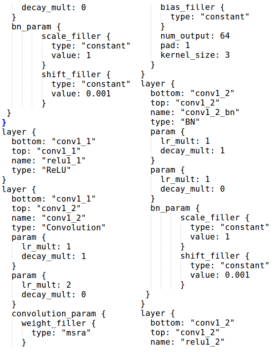


Figure: A section of Caffe's architecture definition.

RefineNet in Matlab

If you thought Caffe's configuration files are unclear, check RefineNet's code written in Matlab:

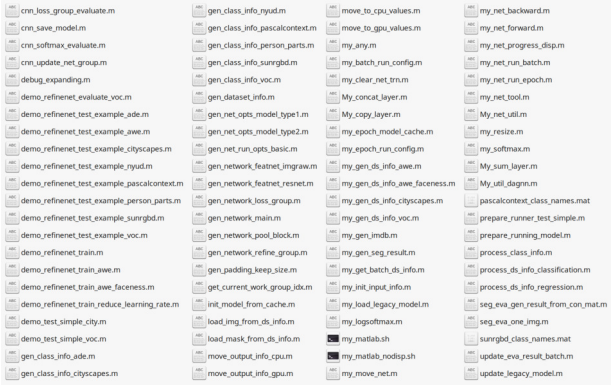


Figure: File structure of RefineNet's Matlab code.

Single Shot Detector

For the comparison of the traditional Viola-Jones [1] vs. Single Shot Detector (SSD) [2] prepared in Pyhton using TensorFlow see and run the attached code.

Figure: Run the code and observe the differences on this image with many many faces.


P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–I.


W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

IMAGE BASED BIOMETRY 2018 / 2019

Face Deidentification: Introducing Privacy in Surveillance Applications

University *of Ljubljana*
Faculty *of Computer and Information Science*

M. Sc. Blaž Meden

TEC | Tecnológico de Costa Rica

What is Deidentification?

- Deidentification is a special kind of so-called **Privacy Enhancing Technology** (PET) that removes identity-related cues from the input imagery.
- Such technology is of paramount importance for **ensuring privacy in various services** such as Google Street View or FourSquare, in multi-media data collections shared between government agencies, or online video-enabled chat rooms and video-conferencing apps.

"I never forget a face,
but in your case
I'll make an exception."

– Privacy Enhancement Technology, such as Face Anonymization / Face Deidentification is of paramount importance nowadays. –

Motivation

Why Face Deidentification research?

- Many developed active surrveillance systems nowadays (25 Million CCTV Cameras worldwide, 4 Million just in UK, 500k in London – average person in London observed 300 times per day!).
- Many developed "smart" face recognition systems, where the basic de-identification methods (e.g. blurring, pixelation) are not capable of preserving user privacy.

Figure 1: Cameras next to the Palace of Westminster's Elizabeth Tower, London.

Motivation

Why Face Deidentification research?

- Currently, there are 170 million surveillance cameras in China and, by 2020, the country hopes to have 570 million!
- The system is being used to help consumers as well as police, who can track people's movements, friends, and even try to predict crime.
- Most of the time, the people's identity does not have to be revealed (especially not to anyone) – **This is where Face Deidentification comes into play**.

Figure 2: A Chinese police surveillance vehicle on Tiananmen Square.

Motivation

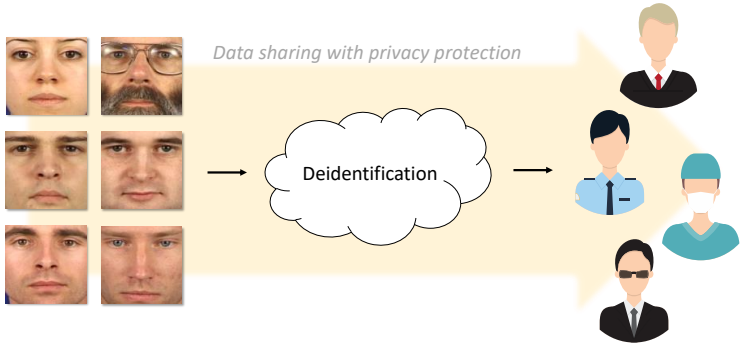


Figure 3: The data needs to be appropriately deidentified before being shared between relevant stakeholders, to prevent misuse of personal information and ensure privacy protection. In our case we are trying to prevent face recognition from video and image data.

Related work – Face Deidentification

Existing techniques for Face Deidentification can be grouped into:

- Ad-hoc (naive) methods, often use with video sequences (but also on still images).
  - Usually used naive methods: pixelation, blurring, black-boxing.
  - Many of them are prone to various attacks or have other weaknesses (parrot attack or not offering formal privacy guarantees).
- Formal techniques that provide upper bounds on the reidentification risk, but are limited to still images only.
  - Based on formal anonymity schemes, such as: k-same, L-diversity, t-closeness, etc.
- Complete survey on deidentification in multimedia content available by Ribarić *et al.* (2016).

Related work – Ad-hoc Methods

Ad-hoc methods can be applied to video or still image data.
On still images:

- Image scrambling (Dufaux and Ebrahimi 2010), puzzling (Bhattarai *et al.* 2014), filtering (Letournel *et al.* 2015).
- Image foveation in DCT (discrete cosine transform) domain by Alonso *et al.* (2017).
- Q-far deidentification, based on AAMs (Active Appearance Models) by Šamaržija *et al.* (2014).
- Part-based deidentification by aggregating facial components from different donors by Mosaddegh *et al.* (2015).
- Facial expression preserving deidentification based on variational adaptive filtering which preserves most important facial regions by Letournel *et al.* (2015).

Related work – Ad-hoc Methods (Hardware-based Implementations)

Ad-hoc methods can be applied to video or still image data.
On video data:

- The PrivacyCam, hardware implementation, scrambling JPEG coefficients, by Chattopadhyay and Boult (2007).
- PriSurv system, replacing persons with color blocks, by Chinomi *et al.* (2008).
- Automatic face masking (detection, tracking, masking) by Chen *et al.* (2009).
- TrustCam, which encrypts regions of interest, by Winkler and Rinner (2010).
- Deidentification camera for real-time privacy protection, hardware implementation of Gaussian blurring and pixel thresholding by Mrityunjay and Narayanan (2011).

Related work – Formal Methods on Still Images

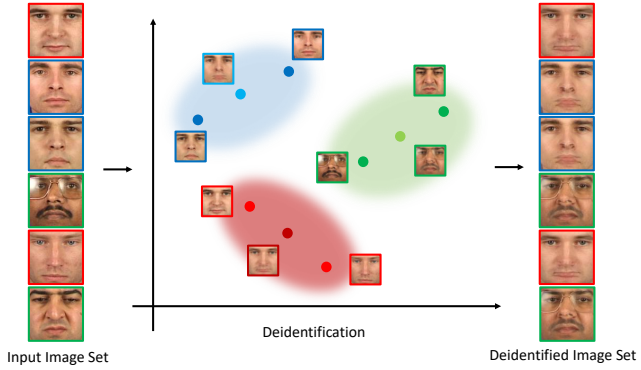


Figure 4: The idea behind k-anonymity mechanisms. Anonymity is ensured by replacing k images from the input set with the same surrogate image. To preserve some of the information, surrogate images are computed as cluster centroids (in this case $k = 2$).

Related work – Formal Methods on Still Images

Table 1: Overview of some of the existing formal deidentification techniques based on the k-Anonymity scheme.

Year	Algorithm	Domain	# Assumptions
2005	k-Same-Pixel [Newton *et al.* (2005)]	Pixel	Client specific image set.
2005	k-Same-Eigen [Newton *et al.* (2005)]	PCA space	Client specific image set.
2005	k-Same-Select [Gross *et al.* (2006)]	AAM, PCA space	Need to specify selection criteria prior to deidentification.
2008	k-Same-Model (or k-Same-M for short) [Gross *et al.* (2008)] (also known as the (ϵ, k)-map algorithm)	AAM	Model parameters obtained during fitting are not unique due to ambiguities.
2013	Driessen–Dürmuth's algorithm [Driessen and Dürmuth (2013)]	PCA space, Gabor wavelets	Not achieving very strong k-Anonymity; human recognition is still possible.
2014	k-Same-furthest-FET [Meng *et al.* (2014)]	AAM, PCA space	Neutral emotion not available explicitly; FET not satisfying k-Anonymity; efficacy experimentally proven.
2014	GARP-Face [Du *et al.* (2014)] (Gender, Age, Race Preservation)	AAM	Utility-specific AAMs for ethnicity, gender, expression.
2015	k-Diff -furthest [Sun *et al.* (2015)]	AAM	Distinguishable client specific image set.
2017	k-SameClass-Eigen [Meng and Shenoy (2017)]	PCA, LDA space	Depends on LDA classifier accuracy; may fail on unknown faces.

Related work – Generative Models

- GNNs – Generative Neural Nets

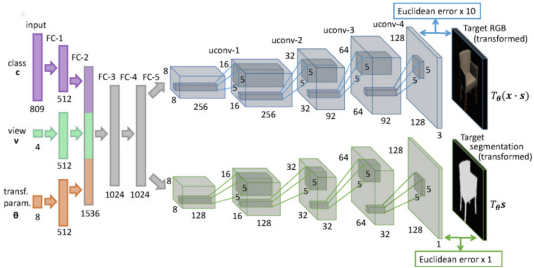


Figure 5: GNN architecture proposed by Dosovitsky *et al.* (2015) for generating furniture images. Our work uses similar architecture (with modified input parameters and without segmentation pipeline).

- GANs – Generative Adversarial Nets initially introduced by Goodfellow *et al.* (2014).

Related work – Encoder-Decoder Models

- Encoder-Decoder Networks – SegNet, U-Net, CapsNet, Autoencoders, Variational Autoencoders and similar approaches.

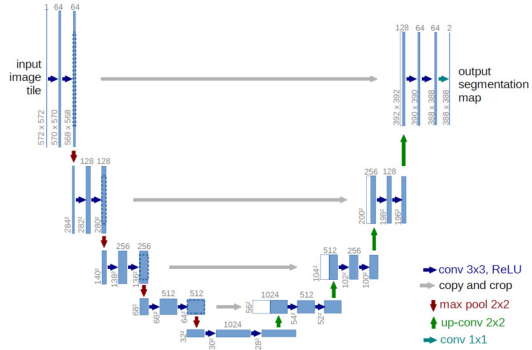


Figure 6: Popular U-Net architecture proposed by Ronneberger *et al.* (2015) for precise image segmentation of biomedical images.

Concept – Deidentification Pipeline

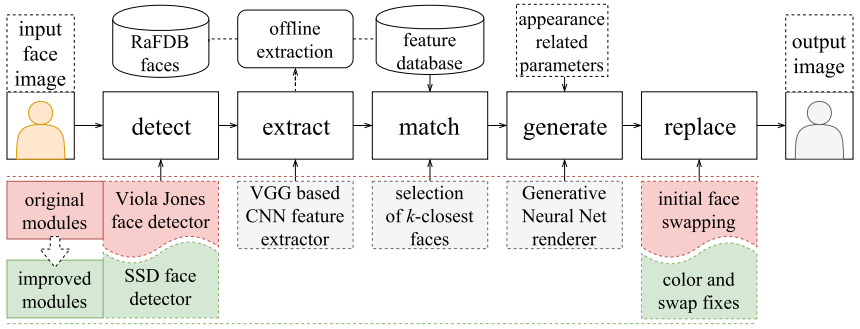


Figure 7: Overview of our deidentification pipeline concept with added module improvements. Viola-Jones face detector was replaced with CNN-based Single Shot Detector (SSD) trained for face detection. Face swapping module was updated with more advanced masking system with skin color corrections. Pipeline can process still images or video frames.

Concept – Insight into Face Swapping Module

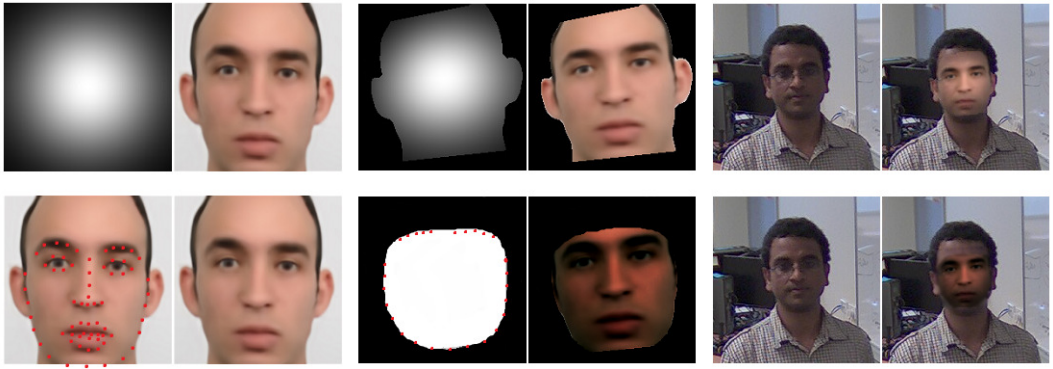


Figure 8: Top row shows the original face swapping approach with hard masks for face replacement (version 1). The bottom line shows the improved face replacement procedure, based on the detected face landmarks (version 2).

Concept – Clustering Procedure

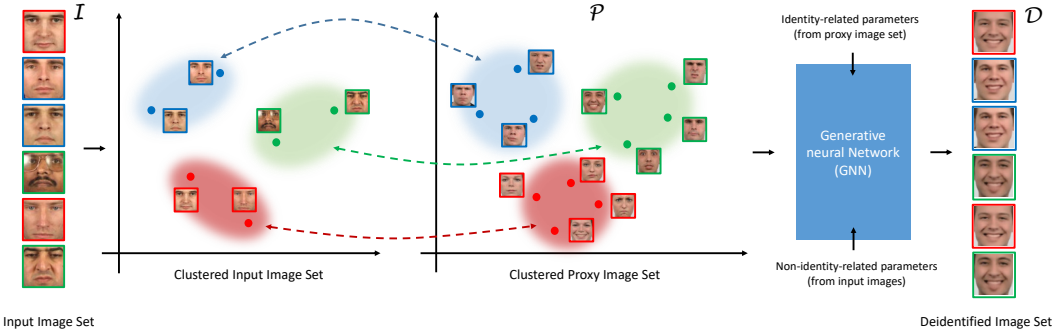


Figure 9: Clusters of k images from I are mapped to the same image in D. Surrogate faces are generated from proxy set P by GNN.

Concept – Generative Architecture

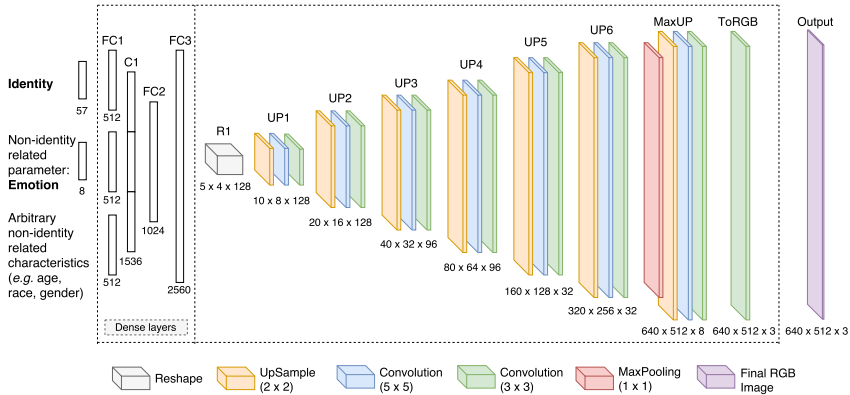


Figure 10: Our architecture is built using fully connected (dense) layers (denoted as FC1, FC2, FC3) and concatenation layer (denoted as C1), followed by six deconvolutional layers (performing upsampling and convolution, from layer UP1 to UP6 as denoted in the figure).

Deidentification Examples

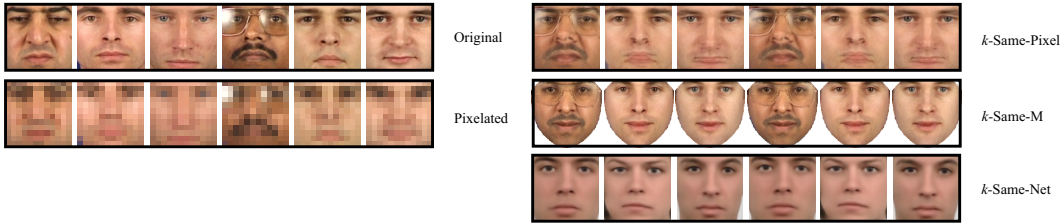


Figure 11: Qualitative deidentification results: the original images, pixelated images, the k-Same-Pixel algorithm ($k = 2$), the k-Same-M algorithm ($k = 2$) and our k-Same-Net approach ($k = 2$).

What about Facial Expressions?

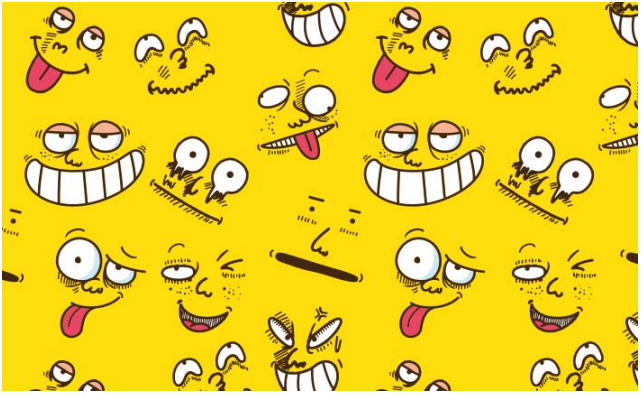


Figure 12: Facial Expressions could be useful when analyzing the imagery from the forenzics point of view in criminal investigations – other soft biometric traits could be preserved as well in order to improve the utility of the deidentified data.

Examples of generating various Facial Expressions

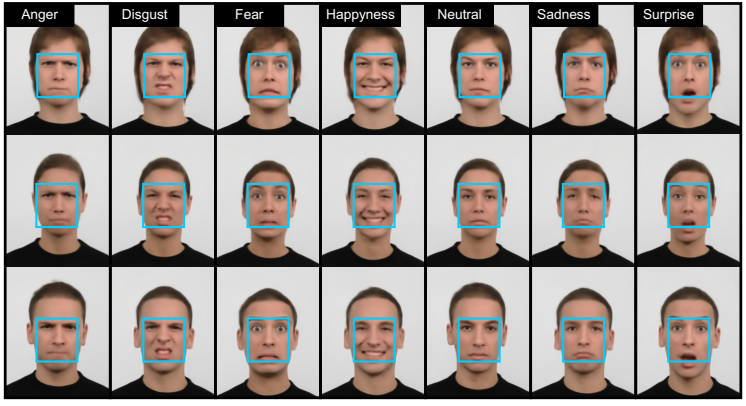


Figure 13: GNN can produce various facial expressions for every identity. Each identity is a mixture of k identities ($k = 2$) from the training (proxy) set. Note that all images appear natural and with no visible arifacts (ghosting or non-natural patterns).

To conclude...

"You can read his mind in his face.
– Yes, it's usually a complete blank."

– It is essential to know what to expose
and what to conceal before releasing
personal data (such as facial imagery)
to external stakeholders. –