Conversion of fingerprints using Inconsistent Cycle GANs

Objective

1. To convert user's real fingerprint(RF) of a user into a fake fingerprint(FF)
2. FF should look like a real fingerprint and should not be detectable as a fake image
3. The similarity between RF of the user and his FF should be low. One should not be able to trace RF of a user from his FF.
4. Similarity between different impressions of RF and their corresponding FF should be maintained. These FFs should be similar. Intra user score preserved.
5. One RF of a user to be converted into multiple FFs. Each FF is to correspond to a unique key.
6. The similarity between RF and each FFs, and amongst each FFs has to be low. One should not be able to trace back other fingerprints from any one of the fingerprints.
7. The similarity between RFs of two different fingers and their corresponding FF should be maintained. These FFs should not be similar. Inter user score preserved.

Motivation

In case of security break and fingerprint leakage, user’s original fingerprints should not be compromised

Even if the architecture and the key associated with a user is leaked, one should not be able to trace back the real fingerprint of any user

Abstract

We use Cycle GANs to convert RF to FF of a user. The fingerprint type of RF and FF are different.

Cycle GANs preserve the inter user and intra user scores of fingerprint matching. They also result in FFs look real, with similar distortions and orientations as RFs.

We use the latent space of RFs to change the output image of Cycle GANs. A unique key is associated with each RF. The latent space of RF is modified using the key to prepare unique FF.

The key serves two purpose.

1. To provide a layer of protection in case of data leak. Even if the architecture is invertible, one will need the key associated with the user to trace back to his RF.
2. To associate multiple FFs with the same RF. One single RF can be changed to multiple FF using different keys. The user can be given different keys as per the requirement.

Abstract

Heat map is used to crop the FF. Cropping is required to remove noise outside the fingerprint, and to improve the difference score between user's RF and FF. The center part of fingerprint impression which helps in fingerprint classification is used for subsequent steps.

During conversion, a random key is selected. The latent space of RF and key is input to the ResNet which changes the input latent space to the required FF type.

The ration in which to add the latent space of RF and key is optimized using linear regression.

The final output image has been tested using frequency spectrum for fake image detection.

Neural nets are used for the architecture, invertibility is not possible. RF of the user is thus safe.

Keywords

Fingerprints

Inconsistent Cycle GANs

Latent space

Resnet

Linear Regression

Gabor filter

Heat Map

Harris Corner Detection

VeriFinger

Frequency spectrum

Security

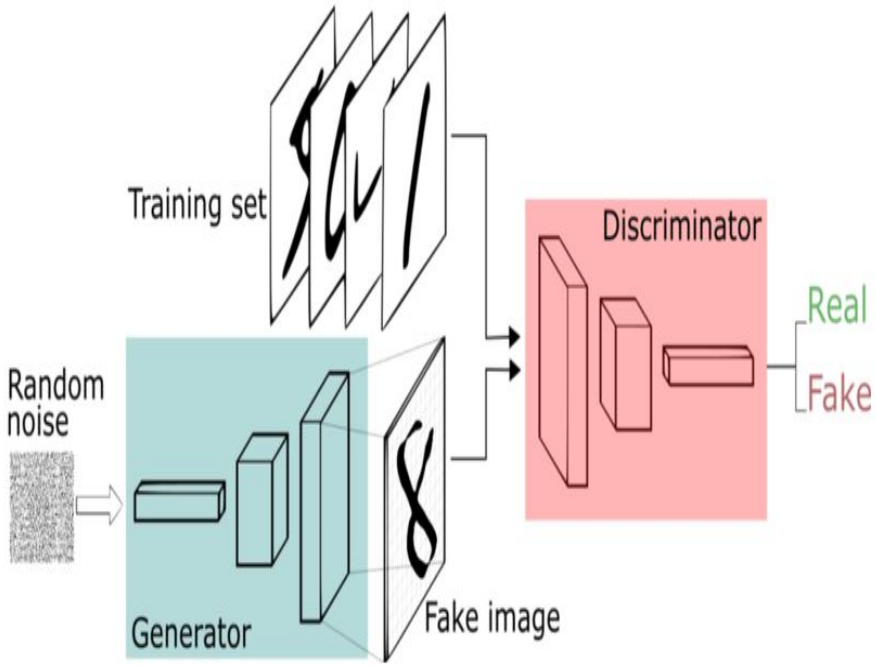
Anguli MSVC

Cycle-GANs

Two neural net architectures work simultaneously.

One(generator G) creates fake image, another(discriminator D) differentiates fake image from real image.

The generator try maximizing the probability of making the discriminator mistakes its inputs as real.

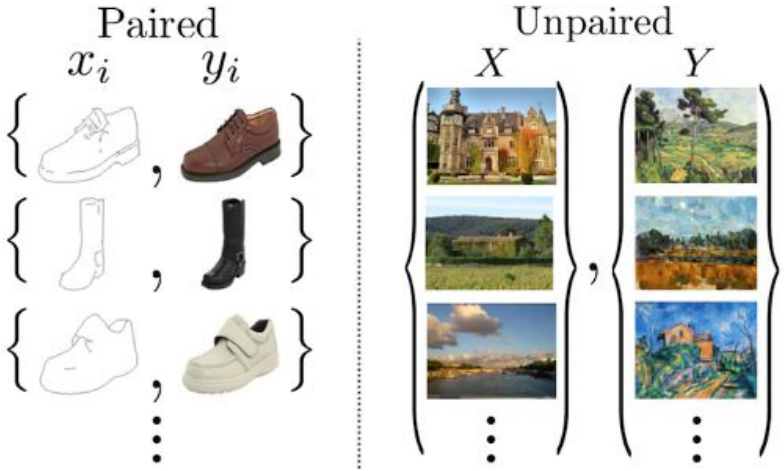


Generative Adversarial Network framework.

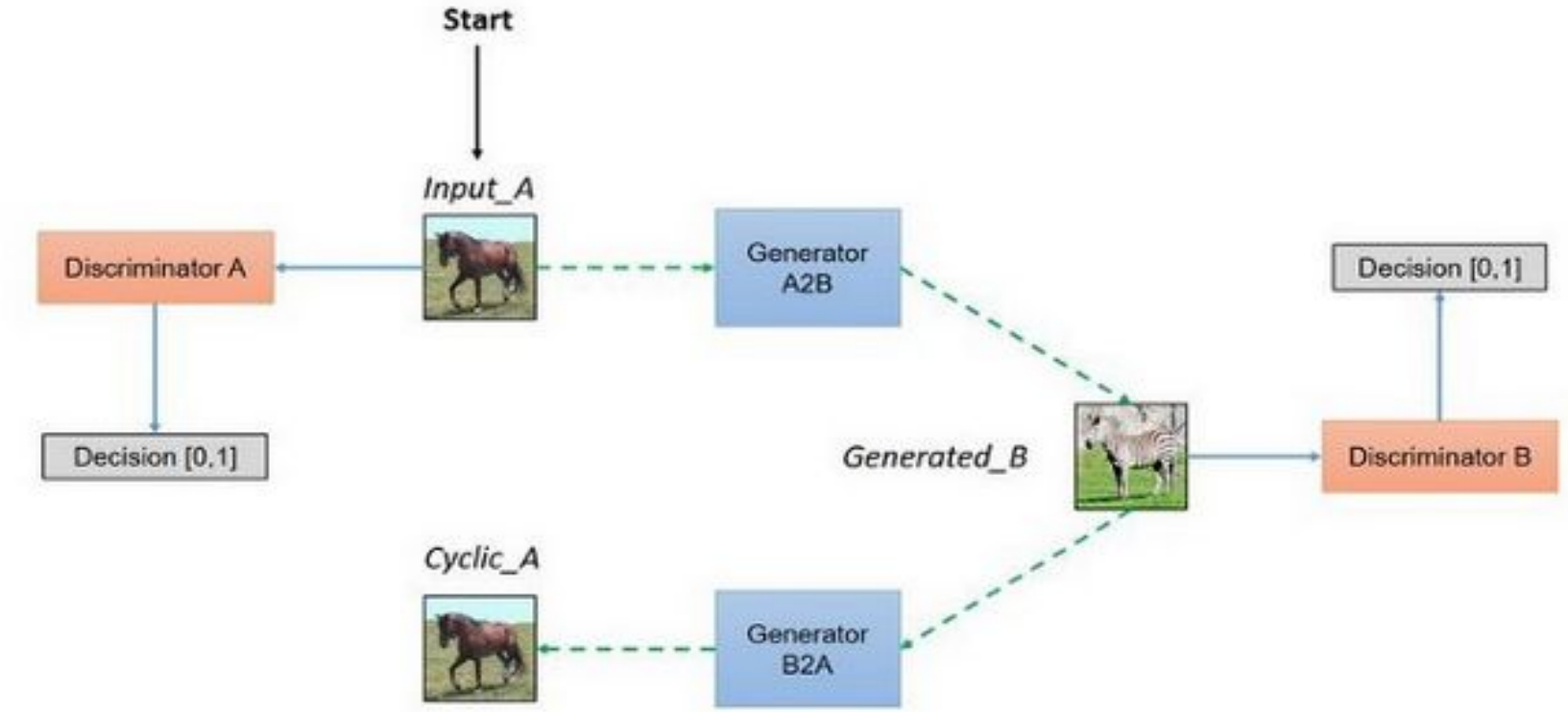
Image-to-image translation is the task of transforming an image from one domain (e.g., images of zebras), to another (e.g., images of horses)

We use unpaired dataset to train cycleGANs

No need of paired images



Block diagram



We feed the model with two types of fingerprints, say ***whorl*** and ***arch***. Aim is to convert whorl into arch. G is converting real whorl into fake arch, and D is differentiating between fake converted arch and real arch fingerprints.

This way we are able to convert real whorl fingerprint into a real looking fake arch fingerprint.

Arch

Left loop

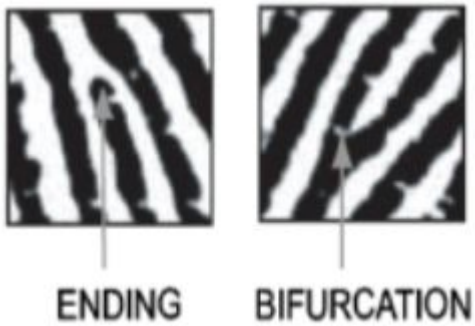
Right loop

Tented Arch

Whorl

Fingerprint Matching

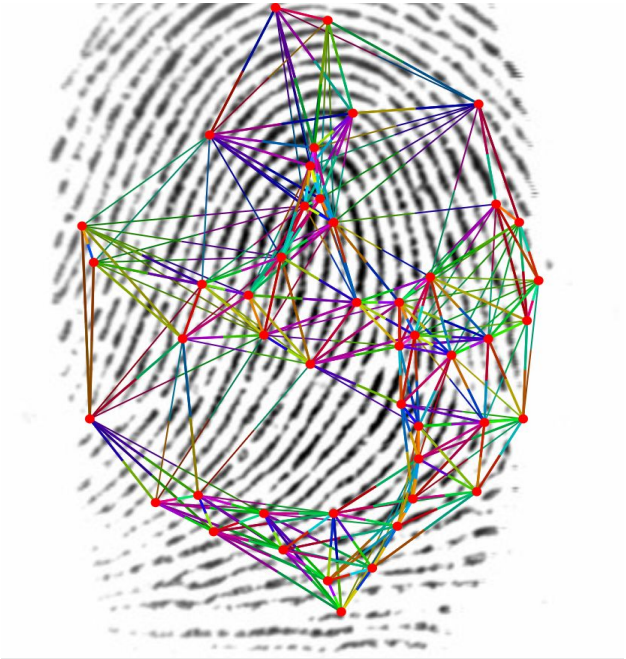
Different fingerprints are matched using level 2 features- Bifurcations and endings, also called Minutiaes



Finding Minutiaes

Minutiae found on the fingerprint image.

Graph determined using angles and distance between minutiaes



Color is determined by edge length and angles. Similar edges have similar colors.

Fingerprints match when the graphs are similar

Root minutiae are blue. Pairing tree is green. Graph of supporting edges is yellow.

Algorithm steps

Registration of a new user follows these steps

1. Detection of the user's fingerprint. Fingerprint is enhanced using Gabor filter.
2. The fingerprint is classified into one of the 5 classes of fingerprints, say type 1.
3. A class of fingerprint different from type 1 is selected randomly, say type 2. Type 2 is the class which the final fake fingerprint will belong to.
4. A cycle gan model has been trained on all type conversions.
5. A random key is selected. This key is a fingerprint which is taken from the key database. Can be of any type.
6. Latent space of real fingerprint and key are added using linear regression, then passed through resnet (decoder) to get the fake fingerprint.
7. Fake fingerprint is enhanced using gabor filter.
8. Heat map is used to crop the fake fingerprint.
9. Harris corner detection is used to find scores between RF and FF. It is made sure that the score between both is lower than threshold. Otherwise a new key image and a new type for FF (Type 2) is chosen for the user.
10. This FF is stored in the database.

When the user signs in henceforth, his RF and key will be taken as input and converted to FF. This new generated FF will be matched with the database of stored FFs. If a match is found, user is given access.

Keys and Latent space

The vector input to the GAN architecture is the latent space of the input. We use this vector to generate different images from the input. The latent space of user's RF is modified using key's latent space.

The modification is done using linear regression to get the optimized result which does not match the RF or the key.

The objective function for optimization is =

0.3*(feature difference between modified gan image and source image) + 0.3*(feature difference between modified gan image and key image) + 0.4*(classification probability score predicted by trained classification model)

The objective function is a pay off between altering the feature points but also preserving the overall shape of a fingerprint belonging to the particular class.

Different keys can be used to generate different outputs. Multiple keys can be associated with a single user.

Heat map activation

https://keras.io/examples/vision/grad_cam/

https://www.pyimagesearch.com/2020/03/09/grad-cam-visualize-class-activation-maps-with-keras-tensorflow-and-deep-learning/

Using Grad-CAM, we can visually validate where our network is looking, verifying that it is indeed looking at the correct patterns in the image and activating around those patterns.

Grad-CAM works by (1) finding the final convolutional layer in the network and then (2) examining the gradient information flowing into that layer.

The output of Grad-CAM is a heatmap visualization for a given class label (either the top, predicted label or an arbitrary label we select for debugging). We can use this heatmap to visually verify where in the image the CNN is looking

This concept has been used to crop our final images to accommodate most contributing pixels and neglecting others.

Validation of results

Intra user- Results show that intra user scores of a user are preserved

Inter user- Results show that inter user scores of users are preserved

Frequency spectrum- The fake fingerprints are not distinguishable from real fingerprint using frequency graph. Both RF and FF give same frequency spectrum.

DET Graph- The results are shown graphically using DET graph. We can decide on the threshold using this graph.

Datasets used

Anguli MSVC: 1200 total fingerprints of each type, used

Illustration of all conversions

We have five types of fingerprints

1. Arch
2. Left loop
3. Right loop
4. Tented Arch
5. Whorl

The conversions will be from each type of fingerprint to every other type vv

Types of fingerprints

Arch

Left loop

Right loop

Tented Arch

Whorl

Illustrations of their minutiaes

Arch | Left loop | Right loop | Tented Arch | Whorl

Example of a conversion

Real fp: Whorl

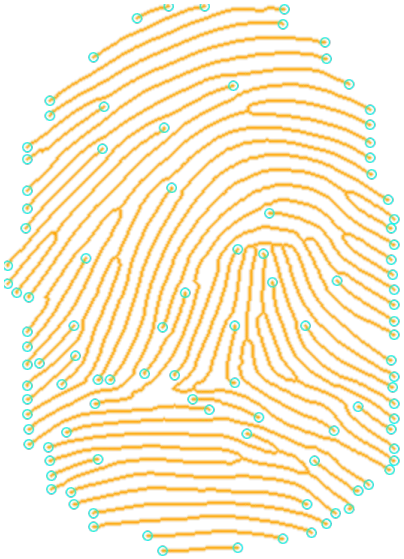
Fake fp: Tented Arch

Minutiae Mask Real fp
Arch

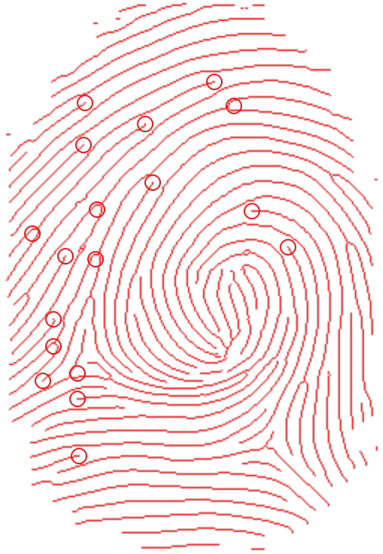
Minutiae Mask Fake fp
Tented Arch

Illustration of their Minutiaes

All minutiaes of real fp
Whorl



All minutiaes of fake fp
Tented Arch



Real fp

Fake fp

Common minutiaes of both fingerprints

Intra user variation

Two goals:

1. Scores between the real and fake fingerprint should be low
2. Different impressions of same fingerprint should produce same fake fingerprint. This is verified by matching scores between real impressions and their fake conversions.

Example showed by converting two impressions of same fingerprint to same type and matching their scores - Whorl to Right type conversion

Steps for showing image matching

Person 1
Impression 1
Real
Whorl

Conversion

Person 1
Impression 1
Fake
Right loop

Person 1
Impression 2
Real
Whorl

Conversion

Person 1
Impression 2
Fake
Right loop

Person 1
Impression 1
Real
Whorl

Conversion

Person 1
Impression 1
Fake
Right loop

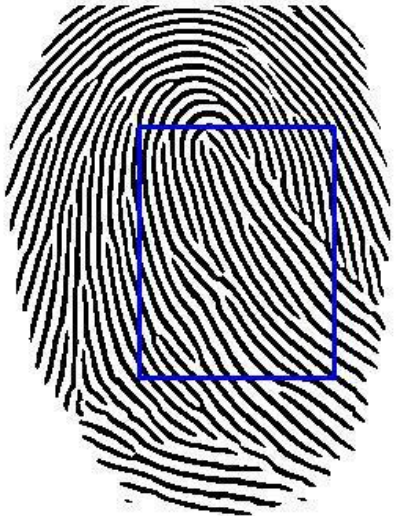
Person 1
Impression 2
Real
Whorl

Conversion

Person 1
Impression 2
Fake
Right loop

Person 1
Impression 1
Real
Whorl

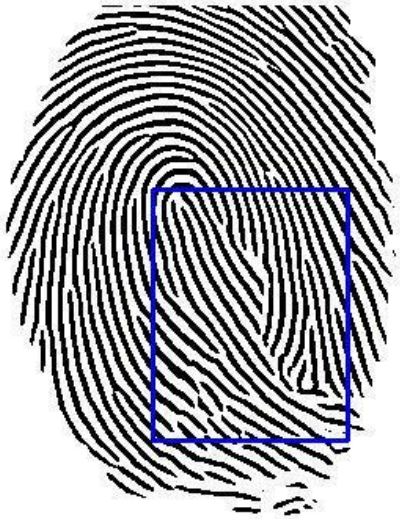
Conversion



Person 1
Impression 1
Fake
Right loop

Person 1
Impression 2
Real
Whorl

Conversion



Person 1
Impression 2
Fake
Right loop

Person 1
Impression 1
Real
Whorl

157

Conversion

Person 1
Impression 1
Fake
Right loop

349

176

Person 1
Impression 2
Real
Whorl

119

Conversion

Person 1
Impression 2
Fake
Right loop

Cropped scores

Person 1
Impression 1
Real
Whorl

Conversion

34

Person 1
Impression 1
Fake
Right loop

197

70

Conversion

2

Person 1
Impression 2
Real
Whorl

Person 1
Impression 2
Fake
Right loop

IMPORTANCE OF USING KEY IMAGE

Person 1
Impression 1
Real

Conversion
with simple
cycle gans

20

Person 1
Impression 1
Fake

48- score
between both
fake image

Conversion
with modified
cycle gans
(key image)

25

Person 1
Impression 2
Fake
With key

INTRA USER SCORES without KEY

Person 1
Impression 1
Real

Conversion

19

Person 1
Impression 1
Fake
Without key

225- score between 2 impressions

92- score between 2 fake impressions

Person 1
Impression 2
Real

Conversion

21

Person 1
Impression 2
Fake
Without key

INTRA USER SCORES with KEYS

Person 1
Impression 1
Real

225- score
between 2
impressions

Person 1
Impression 2
Real

24

Conversion →

Key image

Conversion →

18

Person 1
Impression 1
Fake
With key

89- score
between 2
impressions
with keys

Person 1
Impression 2
Fake
With key

Table

Scores between 6 impressions of same person

Person 1	1	2	3	4	5	6
1	8	193	212	215	246	271
2	91	34	197	224	241	256
3	82	110	2	217	235	230
4	99	128	97	4	264	221
5	97	113	85	75	34	261
6	102	158	111	101	155	29

	Real and fake
	Real and real
	Fake and fake

Inter user variation

Two goals:

1. Scores between the real and fake fingerprint should be low
2. Impressions of different fingerprint should produce different fake fingerprint. This is verified by matching scores between real impressions and their fake conversions.

Example showed by converting two impressions of different fingerprint to same type and matching their scores - Whorl to Right type conversion

Steps for showing image matching

Person 1
Impression 1
Real
Whorl

Conversion →

Person 1
Impression 1
Fake
Right loop

Conversation →

Person 2
Impression 1
Real
Whorl

Person 2
Impression 1
Fake
Right loop

Person 1
Impression 1
Real
Whorl

Conversion

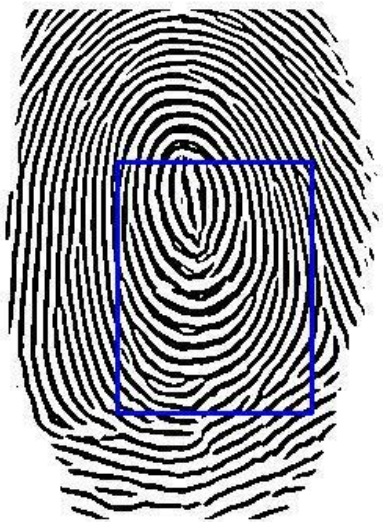
Person 1
Impression 1
Fake
Right loop

Person 2
Impression 1
Real
Whorl

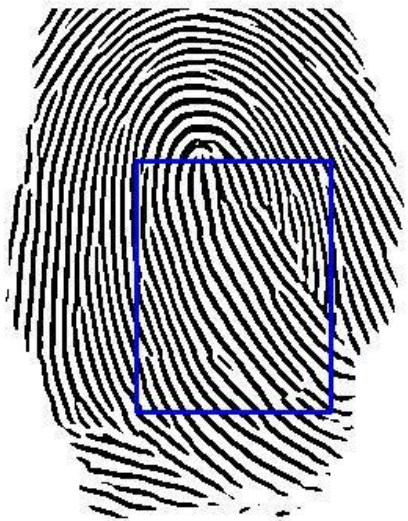
Conversion

Person 2
Impression 1
Fake
Right loop

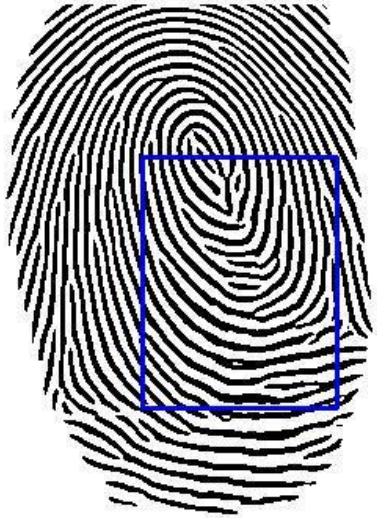
Person 1
Impression 1
Real
Whorl



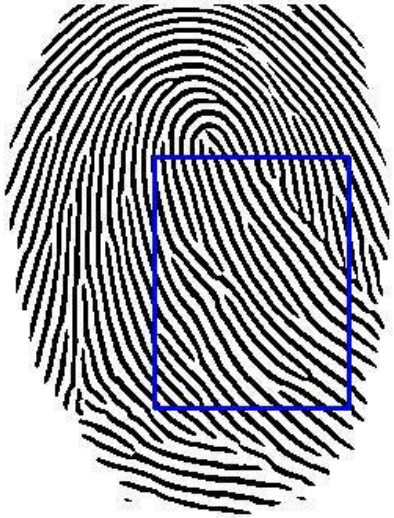
Conversion



Person 1
Impression 1
Fake
Right loop



Conversion



Person 2
Impression 1
Real
Whorl

Person 2
Impression 1
Fake
Right loop

Person 1
Impression 1
Real
Whorl

Conversion

174

Person 1
Impression 1
Fake
Right loop

7

43

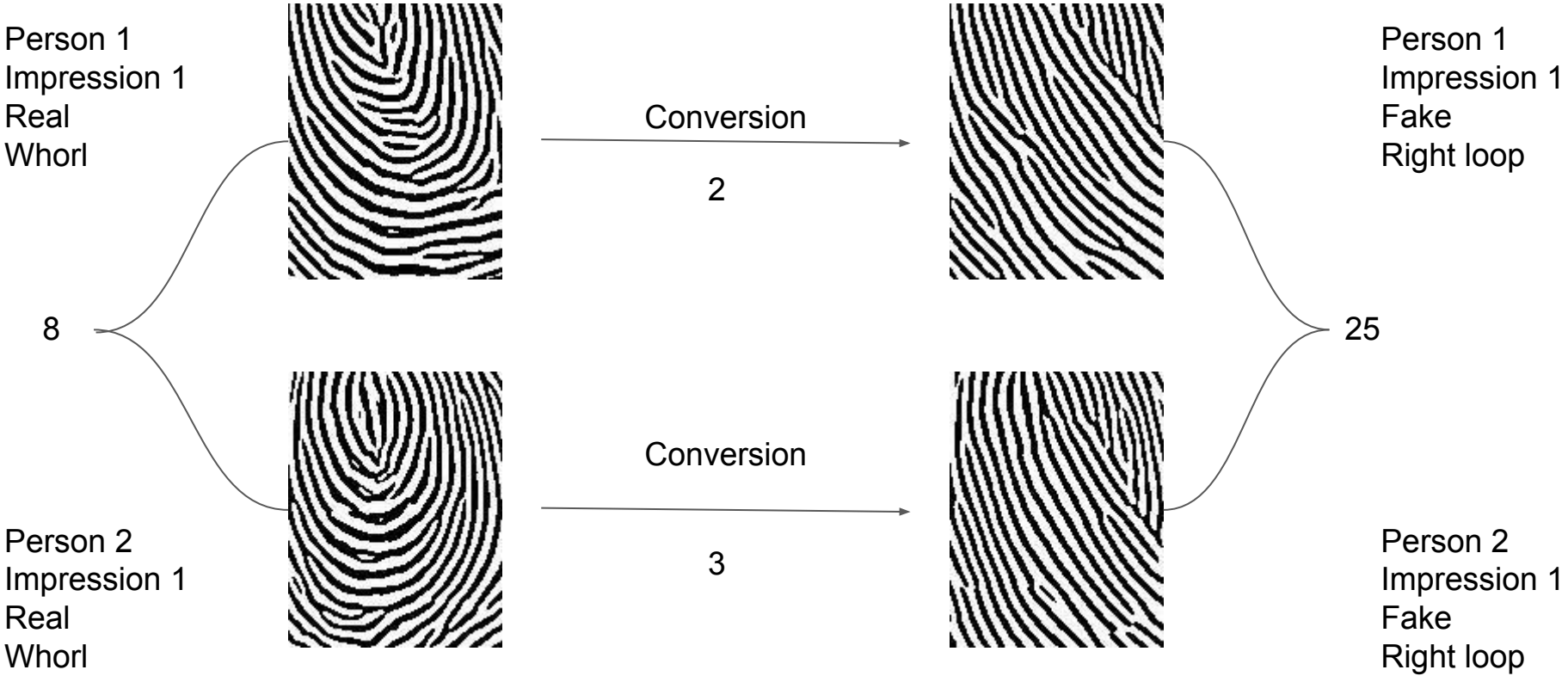
Person 2
Impression 1
Real
Whorl

Conversion

114

Person 2
Impression 1
Fake
Right loop

Cropped scores



INTER USER SCORES without KEY

Person 1
Impression 1
Real
Whorl

19

Conversion →

Person 1
Impression 1
Fake
Without key

62- score between 2 impressions of 2 people

52- score of 2 fake impressions of 2 people without key

Conversion →

21

Person 2
Impression 1
Real
Whorl

Person 2
Impression 1
Fake
Without key

INTER USER SCORES with KEYS

Person 1
Impression 1
Real
Whorl

62- score between 2 impressions of 2 people

Person 2
Impression 1
Real
Whorl

26

Conversion →

Key image

Conversion →

25

Person 1
Impression 1
Fake
With key

48- score between 2 fake fingerprints of 2 people with key

Person 2
Impression 1
Fake
With key

Table

Scores between 6 impressions of two different person

Person 1 and person 2	1	2	3	4	5	6
1	14	3	6	2	2	3
2	2	6	2	1	2	5
3	24	4	2	6	4	8
4	6	11	8	11	5	4
5	5	8	9	10	5	2
6	12	9	25	4	17	3

	Real and fake
	Real and real
	Fake and fake