KRYPTONITE

Protection against adversarial attacks with randomisation of recognition algorithm

Svetlana Koreshkova[1]

Grigory Marshalko[2]

[1] JSC «Kryptonite»

[2] Technical committee for standardisation TC 26

September 16, 2020

Biometrics

Biometrics

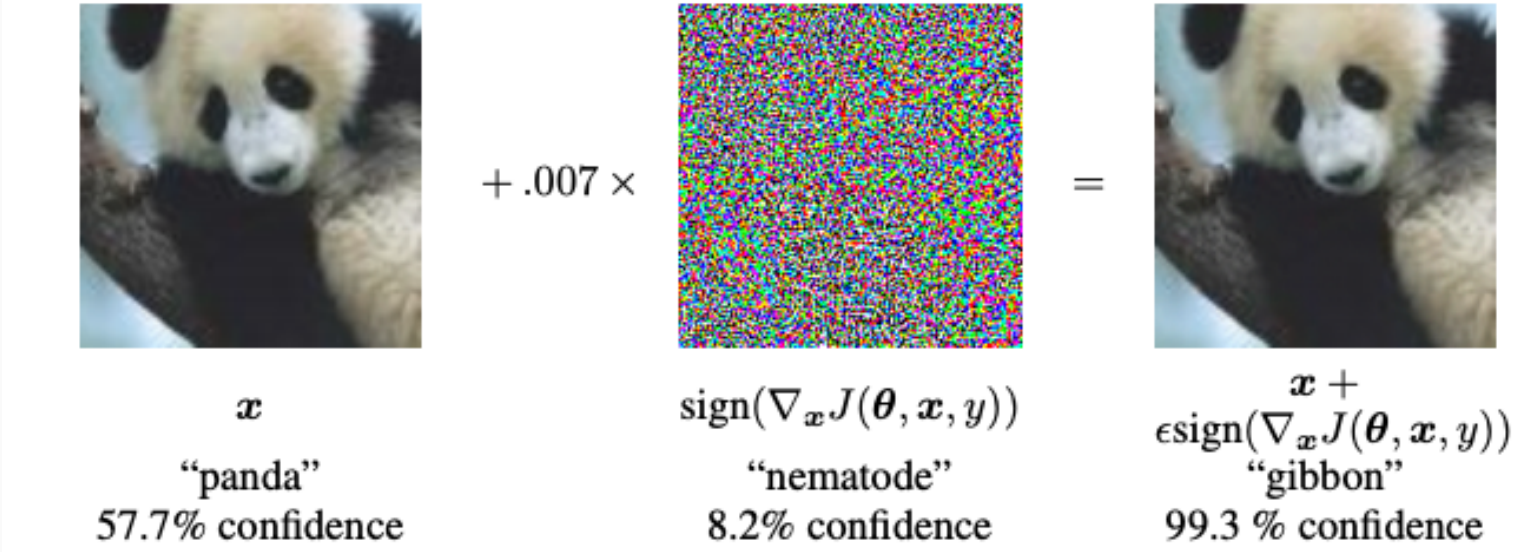
Biometrics are measurable anatomical, physiological, or behavioral characteristics that are inherent in every person. For example:

- fingerprint
- palmprint
- face recognition
- iris recognition
- voice
- gait

Adversarial attack

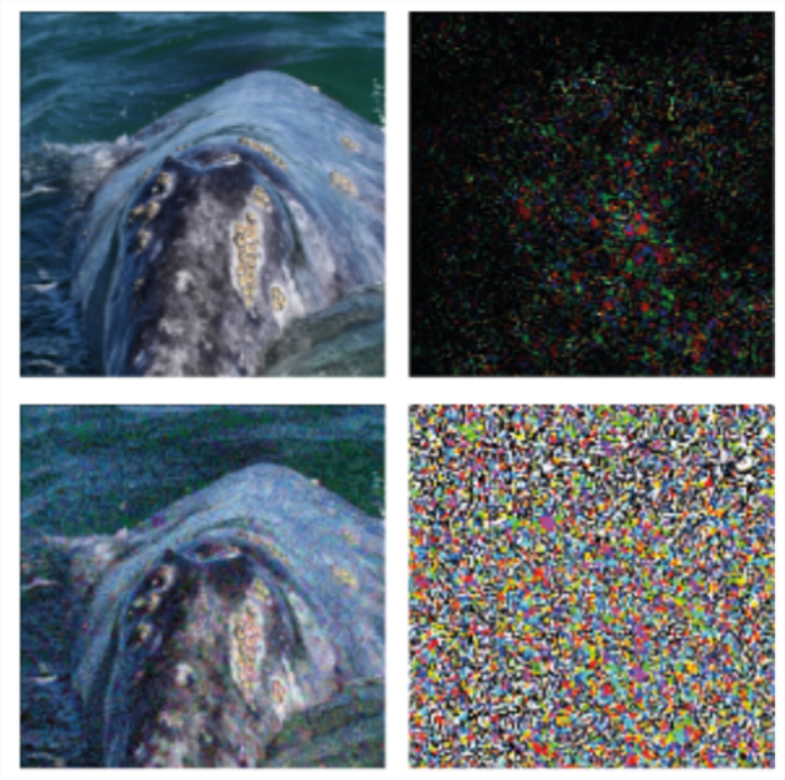
Examples of adversarial attacks

- **Limited-memory Broyden–Fletcher–Goldfarb–Shanno attack (L-BFGS)**: minimization of the loss function towards the target class
- **Fast Gradient Sign Method (FGSM)**: adding some weak noise at each stage of optimization to go in some metric towards the desired class of images



Examples of Adversarial attacks

- **Jacobian-based Saliency Map Attack (JSMA)**: adding perturbations to optimize specific gradient parameters
- **DeepFool**: calculating the minimum noise map, which will spoof the algorithm



Protection against adversarial attacks

Examples of protection

- **Adversarial training**: adding adversarial examples to the training database
- **Gradient masking**: reducing the sensitivity of the model to small disturbances in the input data by smoothing the gradients in the direction

Examples of protection

- **Defensive Distillation**: using a smaller network to evaluate inference function of the recognition system
- **Ensemble Method**: using multiple classifiers and combining their results
- **LBP operator for color images**: combining the values of the LBP operators for one channel

Recognition system based on the LBP algorithm

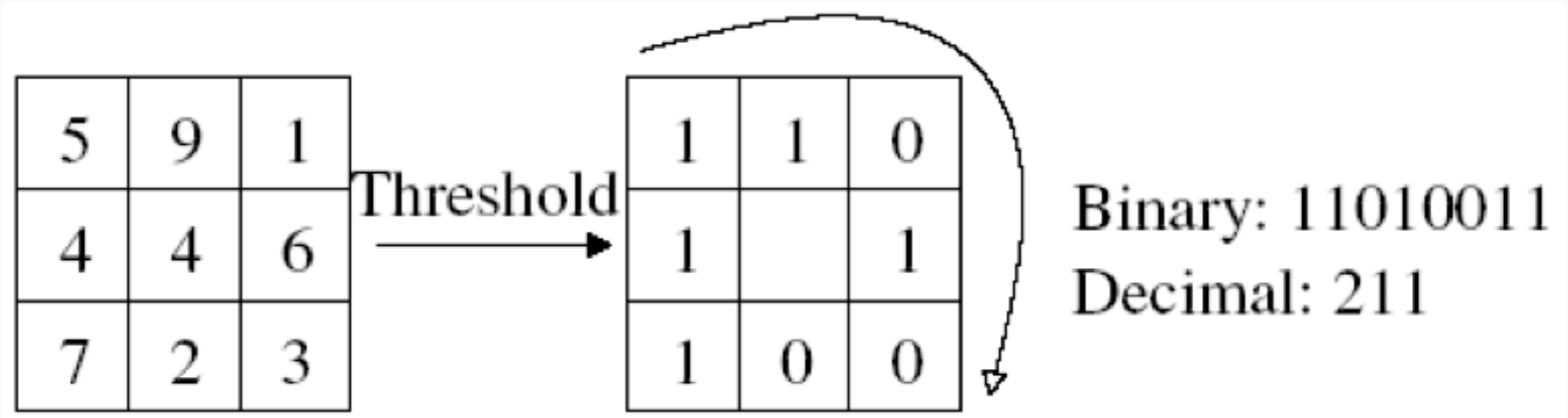
LBP operator

Consider a block $B_Z^{t\times t}, t = 2q+1, q = 1, 2, ...$ of brigtness matrix of image Z. Then operator $LBP : [0, 255]^{t\times t} \to [0, 2^{t^2-1} - 1]$ when applied to the block is defined by the following formula:

$$LBP(B_Z^{t\times t}) = \sum_{q=0}^{t^2-1} 2^q s(p_q - p_{(x^c,y^c)}),$$

where (x^c, y^c) — coordinates of the central pixel $B_Z^{t\times t}$, p_q — brigtness of a pixel with number q (in a certain ordering) from $P_Z^{t\times t}$, $s(x)$ — Heaviside step function.

Classic LBP operator



Dividing the image into blocks and forming an image histogram

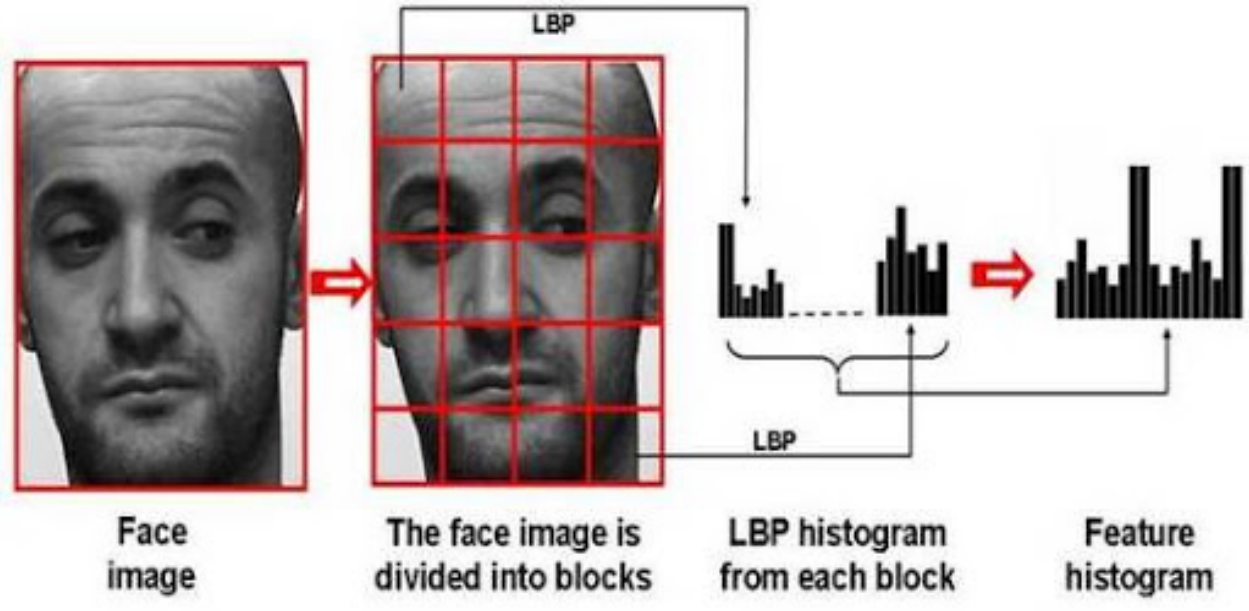


Image classification

The resulting image histograms H_1, H_2 are compared according to the metric L_1.

$$d(H_1, H_2) = \sum_i |H_1(i) - H_2(i)|.$$

The metric is used to determine the closest class of images with the nearest neighbour method:

$$Near(H_Y) = \arg \min_{1 \leq k \leq N} d(H_Y, H_{X_k l}), 1 \leq l \leq M_k,$$

where H_Y — histogram of input image, $H_{X_{kl}}$ — histograms of database images, N — number of image classes, M_k — number of images per class k.

Attack on the LBP algorithm

Selection of uninformative pixels (set P)

- Manual selection
- Select uniform areas using pixel brightness
- Selection by histogram

original

brightness

histogram

Description of the attack

Input: X, Y — brightness matrices of attacking and attacked images.
Output: Z — brightness matrix of the modified image.

- Let H_X is a histogram of X, H_Y is a historgam of Y, $H_Z = H_{Z'} = H_X$
- Let P denote the set of all uninformative pixels
- Do while $d(H_Z, H_Y) < d(H_{Z'}, H_Y)$:
  1. $H_{Z'} = H_Z$
  2. We randomly choose a pixel i from the set P
  3. We select a block with center in pixel i and calculate LBP operator value j for the block
  4. Let $p(j) = H_Z(j) - H_Y(j)$
  5. If $p(j) \geq 0$, then pixel i is removed from P
  6. If $p(j) < 0$, then a small disturbance is added to the pixels in the LBP operator block in Z

Examples of changing the image depending on iterations

attacked image Y

original image X
$d(H_X, H_Y) = 105.03$

1 iteration
$d(H_Z, H_Y) = 53.71$

2 iteration
$d(H_Z, H_Y) = 53.51$

3 iteration
$d(H_Z, H_Y) = 51.86$

Modified LBP algorithm and its properties

Modified LBP algorithm

For the modification, it is proposed that for each case of identification a permutation $\sigma \xrightarrow{\mathcal{U}} \mathcal{S}_n$, where $n = t^2 - 1$, is generated. According to this permutation, the elements from the set $P_Z^{t \times t}$ are selected. So the LBP operator function takes the following form:

$$LBP'(P_Z^{t \times t}) = \sum_{q=0}^{t^2-1} 2^q s(p_{\sigma(q)} - p_{(x^c, y^c)})$$

.

The Fisher-Yates algorithm can be used as a pseudo-random permutation generation algorithm.

Examples of basic and modified images for $t = 3$

original

20.1

18.7

Identification results table for standard and proposed methods

LBP algorithm for database	LBP algorithm for input image	Algorithm threshold	FAR, %	FRR, %
Basic	Basic	9	0	34.3
Basic	Basic	10	0	14.3
Basic	Basic	11	75	5.7
Basic	Modified	9	0	100
Basic	Modified	10	0	100
Basic	Modified	11	0	100
Modified	Modified	9	0	34.3
Modified	Modified	10	0	14.3
Modified	Modified	11	75	5.7
Modified	Basic	9	0	100
Modified	Basic	10	0	100
Modified	Basic	11	0	100

Properties of the modified algorithm

Consider a probability distribution f, describing a corresponding histogram of a block, $P(x_i = f_i), i = \overline{0, 2^{t^2-1} - 1}$. Let $r = 2^{t^2-1}$.

Definition

«Weak» permutation is a permutation, leaving elements with equal probabilities in their classes, i.e.
$(f_0, ..., f_{r-1}) = (\underbrace{f'_0, ..., f'_0}_{a_0}, ..., \underbrace{f'_k, ..., f'_k}_{a_k}), k < r.$

Definition

«Strong» permutation for distribution is a permutation with
$D_{f,f^\pi} = max_i|f_i - f_i^\pi| > \epsilon.$

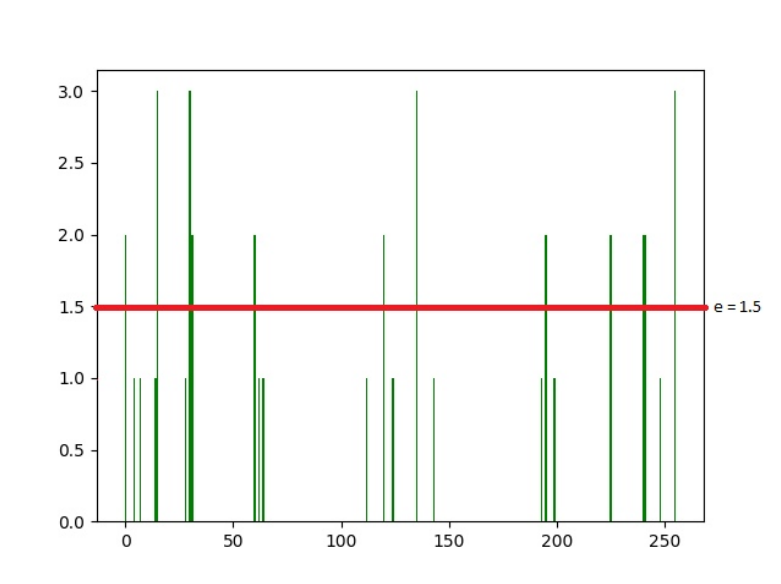
Estimating the number of «strong» permutations

Lemma

The number of «strong» permutations is equal to

$$r! - \sum_{k=0}^{min\{|F_0|,|F_1|\}} \binom{|F_0|}{k} \binom{|F_1|}{|F_1|-k} \binom{|F_1|+|F_\epsilon|-k}{|F_\epsilon|} |F_0|!|F_1|!|F_e|!,$$

where $F_0 = \{f_i : f_i = 0, i = \overline{0, r-1}\}$,
$F_1 = \{f_i : 0 < f_i \leq \epsilon, i = \overline{0, r-1}\}$ and $F_\epsilon = \{f_i : f_i > \epsilon, i = \overline{0, r-1}\}$.



The table of cardinality of the sets of permutation elements depending on the parameter ϵ and the corresponding number of permutations.

ϵ	$\lvert F_0\rvert$	$\lvert F_1\rvert$	$\lvert F_\epsilon\rvert$	Number of «weak» permutations	Number of «strong» permutations
2	179	46	29	$\approx 2^{1576}$	$\approx 2^{1684}$
5	179	61	14	$\approx 2^{1621}$	$\approx 2^{1684}$
7	179	65	9	$\approx 2^{1629}$	$\approx 2^{1684}$
9	179	67	7	$\approx 2^{1636}$	$\approx 2^{1684}$
11	179	69	5	$\approx 2^{1642}$	$\approx 2^{1684}$
13	179	71	4	$\approx 2^{1654}$	$\approx 2^{1684}$
17	179	72	2	$\approx 2^{1653}$	$\approx 2^{1684}$
20	179	73	1	$\approx 2^{1661}$	$\approx 2^{1684}$

Conclusion

In this work

- The modified LBP algorithm is developed
- Comparison of FAR and FRR for the original and the modified systems is performed
- The characteristics of the proposed algorithm are estimated
- The dependence of the number of «strong» permutations on the parameters of the system is estimated

Thanks for your attention!