Fingerprint Liveness Detection Using Local Coherence Patterns
IEEE SIGNAL PROCESSING LETTERS, VOL. 24, NO. 1, JANUARY 2017

Group Number- 25

Name- Dushyant Singh Satyapal

Abhijeet Singh Chahal

Prerit Rajpal

Objective

- To develop an image descriptor for fingerprint liveness detection using the local coherence pattern of a given image. It would enhance the security of biometric recognition frameworks.
- It was found that materials used for making fake fingerprints bring nonuniformity in the captured image due to the replica fabrication process. Thus, we take local coherence patterns to be fed into a linear Support Vector Machine (SVM) classifier to determine whether a given fingerprint is fake or not.

Understanding

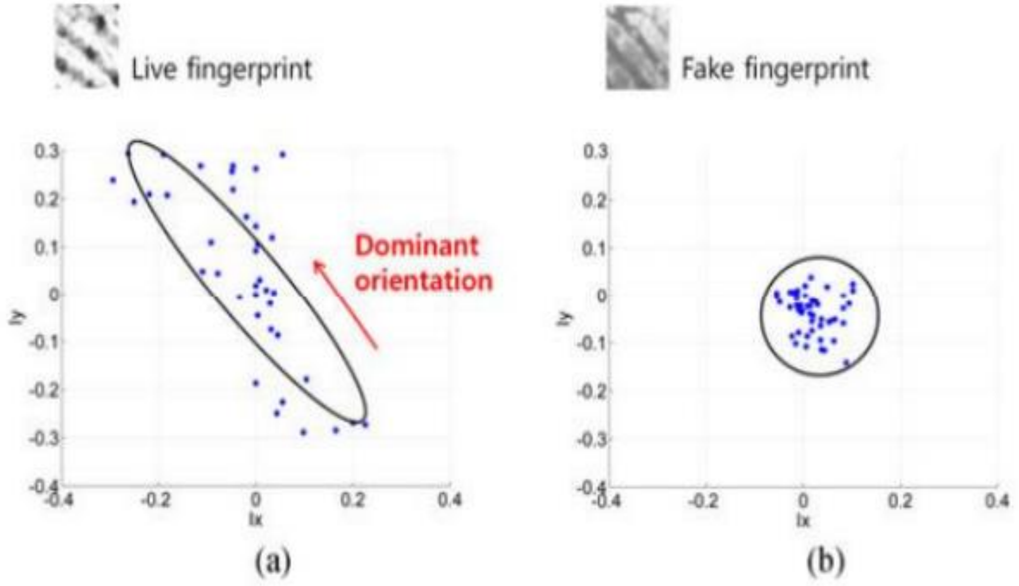


Fig. 1. Gradient distributions obtained from the small local region of live and fake fingerprints (I_x and I_y denote gradients in horizontal and vertical directions, respectively). (a) Result on the live fingerprint. (b) Result on the fake fingerprint. Note that images are from the ATVS dataset [7].

- The fingerprint is a directional structure, and the dispersion caused by ridges and valleys makes the coherence along the dominant orientation.
- The materials employed for making fake fingerprints (e.g., silicone, wood glue, etc.) tend to yield the nonuniformity in the captured image due to the replica fabrication process, we focus on the difference of the dispersion in the image gradient field between live and fake fingerprints. More specifically, we propose to define the local patterns of the coherence along the dominant direction, the local coherence patterns, as our features, which are fed into the linear support vector machine (SVM) classifier to determine whether a given fingerprint is fake or not.

Coherence Computation

- Coherence is a fixed relationship between the phase of waves in a beam of radiation of a single frequency. To compute the directional coherence in the fingerprint image, we first calculate the intensity-gradient distribution as follows-

$C= [c_1, c_2, c_3,\ldots,c_m], c_k=[I(k), I_x(k), I_y(k)]^T$.

- Where $I(k)$, $I_x(k)$, and $I_y(k)$ denote the intensity and image gradients of horizontal and vertical directions at each pixel position (x_k,y_k), M represents the number of pixels belonging to the local region.

Coherence Computation- Singular Value Decomposition

- To find the dominant orientation and its coherence based on the intensity-gradient distribution, we adopt the Singular Value Decomposition (SVD) because it decomposes the given decomposition into independent axes with the corresponding energy.
- Therefore, the dominant orientation and its energy in the intensity-gradient distribution can be efficiently estimated by computing the SVD of C as follows:

$$\mathbf{C} = \mathbf{U}\mathbf{W}\mathbf{V}^T = \mathbf{U}\begin{bmatrix} w_1 & 0 & 0 \\ 0 & w_2 & 0 \\ 0 & 0 & w_3 \\ \vdots & \vdots & \vdots \end{bmatrix}[\mathbf{v}_1 \;\; \mathbf{v}_2 \;\; \mathbf{v}_3]^T$$

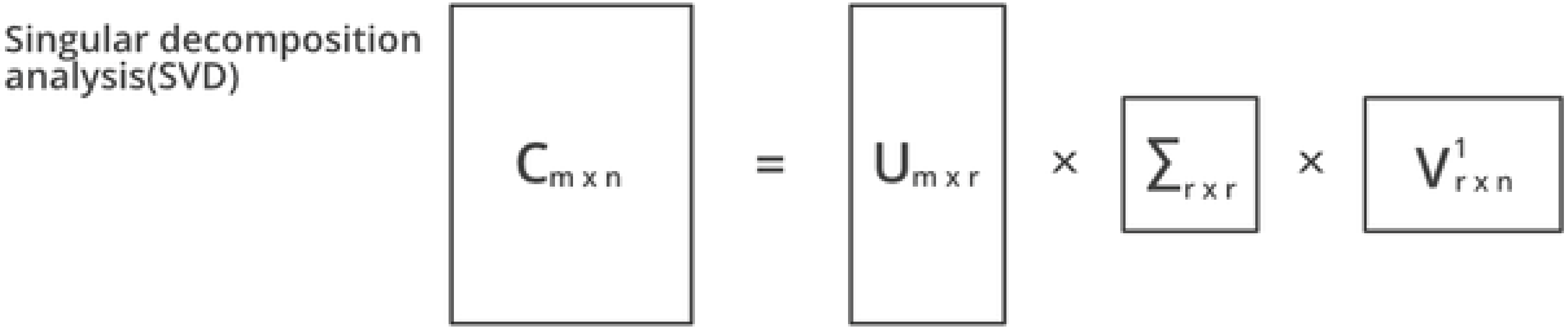
Coherence Computation-

- Therefore, the independent columns we have obtained in W from Singular Value Decomposition are the energies in the directions of $v_1, v_2, v_3.$
- We define coherence :

$$c = w_1 - w_2 - w_3$$

- **Note-** Here, the larger the value c is, the higher the directional coherence of ridges and valleys is.

Pictorial representation for Singular decomposition analysis (SVD)



Local Binary Pattern

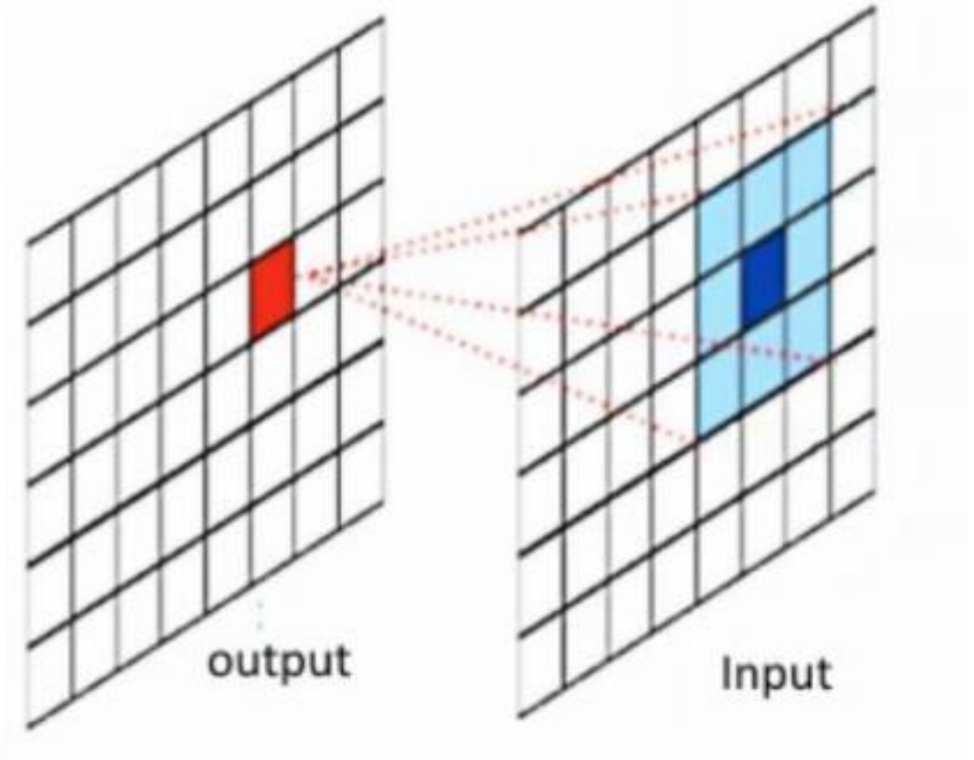
- The main idea behind LBP is to describe the neighborhood of the image element using binary codes. This method is usually used to study their local properties and identify the characteristics of individual parts of the image.
- The main characteristics-

  1-Low calculation cost

  2- Resistance to fluctuations in image grayscale values.

LBP method steps

- 1. convert the image into grayscale.
- **2-** For each pixel(gp) in the image, select the P neighborhoods surrounding the central pixel. The coordinates of gp are given by-
  $$(gc\_x - R\sin(2\pi p/P), gc\_y + R\cos(2\pi p/P))$$
- **3-** Take the center pixel (gc) and set it as a threshold for its P neighbors.
- **4-** Set to 1 if the value of the adjacent pixel is greater than or equal to the value of the center pixel, 0 otherwise.



- **5-** Now compute the LBP value: Sequentially counterclockwise, write a binary number of digits adjacent to the center pixel. This binary number (or its decimal equivalent) is called **LBP-central** pixel code and is used as a characteristic selected local texture.

$$LBP(gp_x, gp_y) \sum_{p=0}^{P-1} S(gp - gc) \times 2^p$$

- P- number of sampling points on a circle of radius R
- R- determines the spatial resolution of the method or operator
- gc- the intensity value of the central pixel
- gp- the intensity of the neighboring pixel with index p.

$$s(x) = \begin{cases} 1 & if \quad x \geq 0 \\ 0 & if \quad x < 0. \end{cases}$$

Pseudo Code

- To implement this method, we require 3 for loops

```
For i in height range:
     For j in width range:
          Select a chunck of the image to compute its LBP
value
          For each block neighbors:
              check if interpolation is needed
              Compute_LBP(bock)
              Add the result
          Update the matrix with the value of LBP
```

LCP

- LCP is the advanced version of LBP.
- LCP is defined as-

$$f_{\mathrm{LCP}}(x,y) = \sum_{1 \le i \le n} 2^{i-1} \mathrm{LCP}^i(x,y)$$

$$\mathrm{LCP}^i(x,y) = \begin{cases} 1, & \text{if } c(x,y) > c(x_i,y_i) \\ 0, & \text{otherwise} \end{cases}$$

- where n is the number of sampling pixels in the neighbor region of 3× 3 pixels

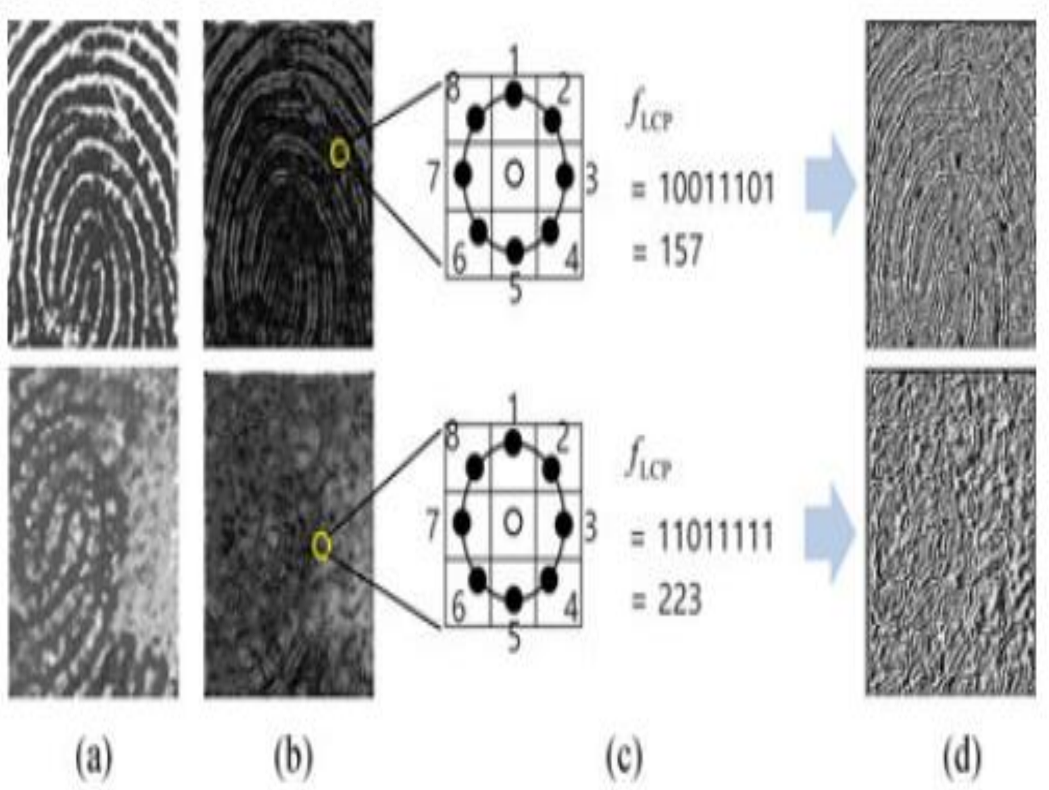


Fig. 2. (a) Input images (top: live, bottom: fake). (b) Coherence map scaled from [0, 255]. (c) Computation procedure for f_{LCP}. (d) LCP images.

LCP Histogram

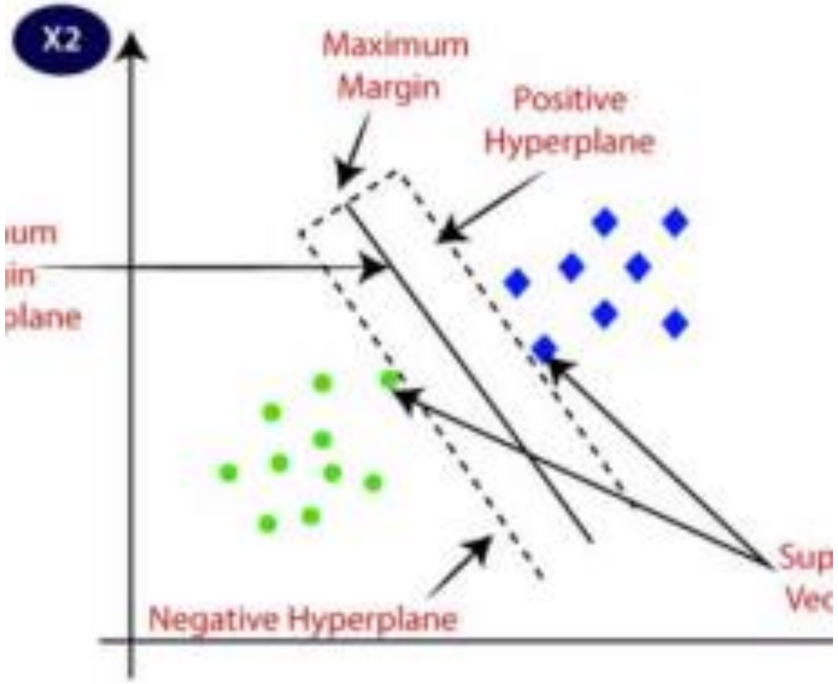
- We numerically define the LCP histogram $h=(h_1,h_2,h_3,h_4,.......,h_n)$.
- Finally, we compute the average of all the LCP histograms constructed in the fingerprint region as our baseline.
- F_{LCP} is the feature vector of the fingerprint image fed to the classifier.

$$h_k = \frac{N_k}{\sqrt{\sum_{q=1}^{59} N_q^2 + \delta}}, \quad N_k = \sum_{\tilde{f}_{LCP}(x,y) \in k} 1$$

$$\mathbf{F}_{\mathrm{LCP}} = \frac{1}{M} \sum_{k=1}^{M} \mathbf{h}_k$$

Support Vector Machine (Classifier)

- The objective of the support vector machine algorithm is to find a hyperplane in N-dimensional space (N- the number of features) that distinctly classifies the data points. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can quickly put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These severe cases are called support vectors, and hence algorithm is termed a Support Vector Machine.



Conclusion

- We have used a simple and novel local descriptor for fingerprint liveness detection.

Our key observations are –

1. The application of LBP can significantly reduce time and computing costs for feature extraction

2. The fabrication process mostly divides the directional structures in the fingerprint image (i.e., ridge and valley). Therefore, we propose exploiting the local coherence patterns along the dominant orientation in the intensity-gradient distribution of the given fingerprint image.

References

- https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
- https://www.geeksforgeeks.org/singular-value-decomposition-svd/
- https://www.geeksforgeeks.org/singular-value-decomposition-svd/
- https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
- https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
- https://en.wikipedia.org/wiki/Local_binary_patterns
- https://medium.com/swlh/local-binary-pattern-algorithm-the-math-behind-it-%EF%B8%8F-edf7b0e1c8b3
- https://www.geeksforgeeks.org/singular-value-decomposition-svd/
- https://wiseai.dev/blogs