Statistical Dependence of Pixel Intensities for Pattern Recognition

Ievgen Smielik and Klaus-Dieter Kuhnert
Institute for Real-Time Learning Systems
University of Siegen, Germany

***Abstract*—In this paper, we describe an algorithm for speeding up object recognition by reducing the amount of pixels taken into account when processing images. We show that some statistically stable regions can be found on image. Taking just one pixel from each region preserves the most of information of the image. We employ linear dependency between pixel intensity values to organize neighbouring pixels in groups. Bayesian classification was chosen to prove suitability. We present the results that show computation speed increase without significant performance losses.**

I. Introduction

In this work we construct a model that consist of several layers: two-pixel distribution matrices, coefficients of determination and sets of regions(covers). This layers are described in later sections of this paper.

The idea of using correlations of adjacent pixels was used in several works on image encryption to prove goodness of encryption [1], [2], [3], authors note a substantial linear dependence between adjacent pixels on plain-image and show considerable dependence decrease on cipher-images.

Another application of correlation of adjacent pixels is image registration [4]. Authors use a feature vector containing correlation information to determine the best matches between search area and windows.

Correlation of adjacent pixels was used for restoration and denoising in recent paper [5]. Authors show a connection between correlation and noise level, mention empirical, simplified version of θ-regions approach described further in the article.

Other authors in [6] look on adjacent pixels dependence, introduce dissimilarity between pixels and use it for segmentation, this approach looks similar to ours in terms of dividing image into regions, but the basic idea is different: our regions are statistics stipulated, tend to have smaller intensity variations over large set of samples, and segmentation regions tend to connect pixels based only on their intensities.

Adjacent pixel intensity difference was successfully applied to Face recognition task in [7].

Statistical dependence between parts of a scene, or ’context learning’, is getting more attention with recent works as well.

The subject of this article goes beyond and extends correlation-based approaches in terms of combining information from many pixels together rather then treating every pair of pixels separately.

II. Pixel dependency model

In this section we introduce Pixel dependency model and present it’s useful properties.

A. Building a model

We take a dataset of N grayscale images I_i, $i \in [1, N]$. The first step is to compute two-pixel distribution matrices

$$\mathrm{TPD}_{p_1,p_2}(x,y) = \sum_{i=1}^{N} \begin{cases} 1 & \text{if } I_i(p_1) = x \text{ and } I_i(p_2) = y \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where p_i is a point on image; $I_k(p_i)$ is the intensity value of pixel p_i of k-th image; $x, y \in [0, 255]$ for grayscale images. Element $e = TPD_{p_1,p_2}(x_1, y_1)$ is a number of observations of pixel p_1 taking value x_1 on the same image with pixel p_2 taking value y_1 through all images in a dataset and there are different tables for different pairs of pixels p_1, p_2. As a result we have distribution tables for pairs of pixels over all images. An example of one such distribution table is presented on **Fig. 1**. It’s clear to see that the distribution table of one pixel with itself, TPD_{p_1,p_1}, has all points laying on one line, see **Fig. 2**, and presents that pixel intensity probability distribution function, see **Fig. 3**. Closer related pixels tend to have all points near one line in their distribution table and uncorrelated pixels have more spreading.

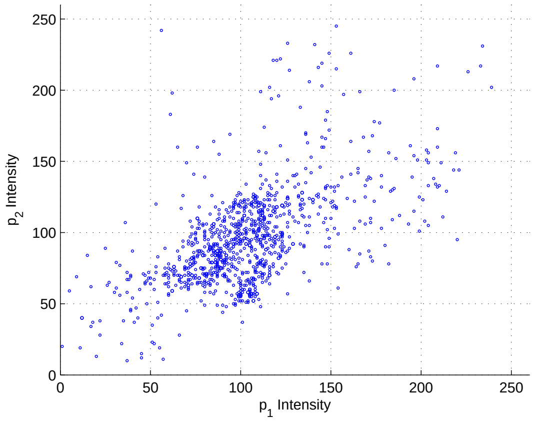


Fig. 1. Two-pixel distribution

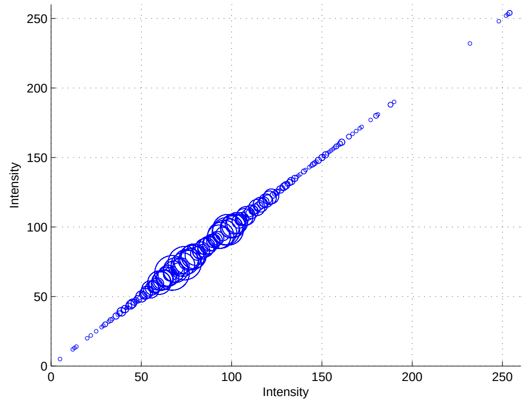


Fig. 2. One-pixel distribution, for pixel (4; 12)

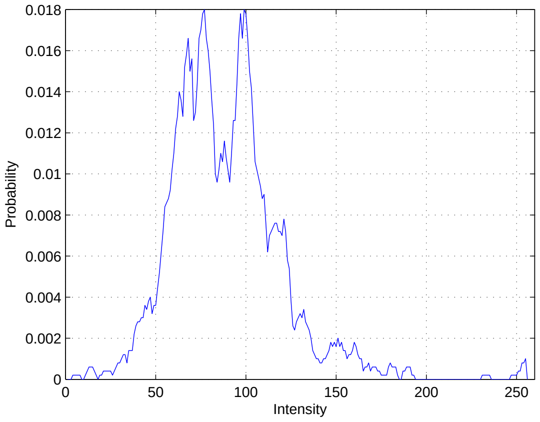


Fig. 3. Probability distribution function, for pixel (4; 12)

In order to find out dependencies between pixel intensities we compute correlation matrices. We follow the approach described at [8]. For each distribution matrix we compute the least squares estimation of liner regression

$$f(x) = \alpha + \beta x \tag{2}$$

That gives us a best fit line for each matrix. We use it as predicted values to compute goodness of fit, which in its turn shows how closely two pixels are related. We compute the total sum of squared deviations in data Y from its mean $\overline{y}$ and corresponding sum of squared residuals (errors)

$$\begin{aligned} SST &= \sum_{i=1}^{N}(y_i - \overline{y})^2 \\ SSE &= \sum_{i=1}^{N}(f_i - y_i)^2 \end{aligned} \tag{3}$$

where y_i is observed and f_i is predicted value. Coefficients of determination are computed as

$$R^2_{p_1,p_2}(x,y) = 1 - \frac{SSE}{SST} \tag{4}$$

A higher value of R^2 indicates a stronger relationship between pixels, see **Fig. 4**. Coefficient of determination of one point with itself, $R^2_{p_0,p_0}$, equals to 1.

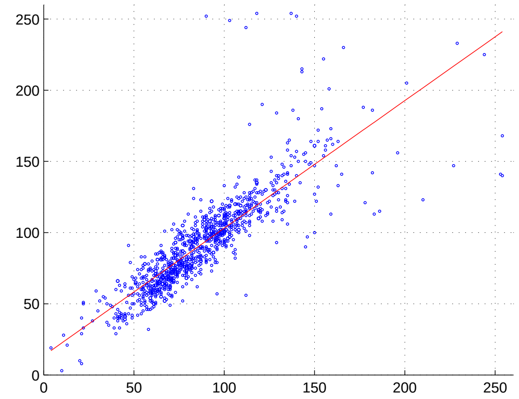


Fig. 4. Linear regression function in red, here $R^2 = 0.86$

For some pixel p_0 let's see into matrix

$$CoD_{p_0}(p_i) = R^2(p_i) \tag{5}$$

where point p_i lays in some neighbourhood of p_0, $|x_0 - x_p| \leqslant k$ *and* $|y_0 - y_p| \leqslant k$. The result of plotting CoD with $k = 6$ is shown at **Fig. 5**.

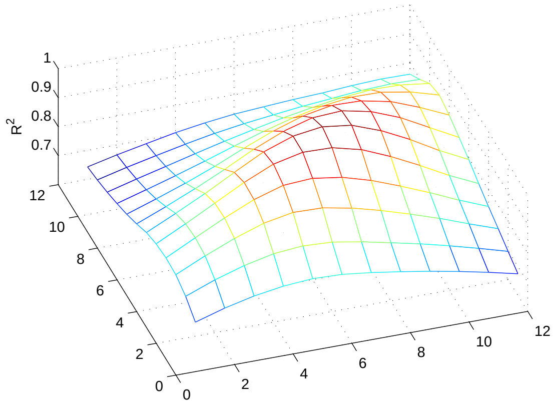


Fig. 5. Coefficients of determination

If we take only those points in which R^2 is greater than some parameter θ

$$CoD^{\theta}_{p_0} = \{p : R^2_{p_0,p} \geqslant \theta\} \tag{6}$$

we have a set of points which intensity values are strongly dependant on the value of p_0. We show that it is possible to use just one point, the center, from region to preserve the information about the whole region. We call CoD^{θ}_{p} a θ-region with center p.

B. θ-regions competition

Let $C = \{CoD^{\theta}_{p} : p \in I\}$ be a cover of I containing all θ-regions. Let's select such a subcover, that overlaps between

θ-regions are less then some small integer l. To do that we use simple yet effective algorithm, see **Algorithm 1**.

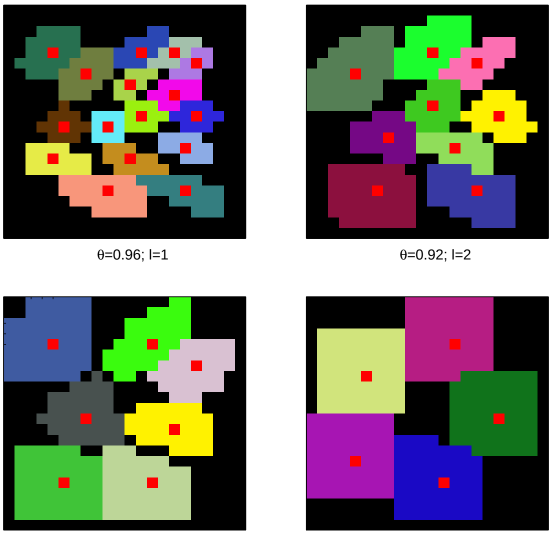
Algorithm 1 Find subcover

```
    $RegionSet \leftarrow \emptyset$
    loop
        $SuitableRegions = \bigcup_{C} r,\ r \bigcap RegionSet \leqslant l$
        if $SuitableRegions = \emptyset$ then
            return $RegionSet$
        end if
        $region \leftarrow random(SuitableRegions)$
        $RegionSet \leftarrow RegionSet \bigcup region$
    end loop
```

In other words, we take regions on random providing they don't intersect with other selected on more than l pixels, we stop when it is not possible to add any more regions. The outcome of **Algorithm 1** is a subcover $S \subseteq C$, let n_S be a number of points covered by S, then the best subcover S_b is

$$S_b = \max_{n_S} S \tag{7}$$

According to approximate theoretical best coverage we perform a certain big amount of iterations with **Algorithm 1**. One example of achieved cover is presented on **Fig. 6**. Pixels marked red on the picture are regions centers. These pixels are to be used in classification, as knowing intensity of region center we can predict intensities of other pixels in a region. By decreasing the amount of pixels taken into account in classification we achieve great performance increase.


θ=0.96; l=1

θ=0.92; l=2

θ=0.88; l=3

θ=0.75; l=1

Fig. 6. Regions with centers. Each plot shows colored θ-regions with centers in red, threshold and intersections limit l

III. Experiments and Results

In our experiments set up we use BioID face database containing about 1500 face images. As a pattern to recognize we select the inner eye corner. Here we provide the results of two experiments: one uses image patches as they are in the database and for the other we use normalized images, which means that all eyes have been made the same size before selecting eye corners. Both object datasets have the same parameters: 1500 eye corner images, 20 by 20 pixels, split into training and testing sets, 1000 and 500 images respectively; background images of the same size and amount cut from random places from full images.

We use simple Bayesian classifier [9] to classify object (C_o) and background (C_b)

$$p(X|C_i) = \prod_{t=1}^{n} p(x_t|C_i) \tag{8}$$

where x_i are pixel intensity values. Object's and background's probabilities were estimated by counting the frequencies of intensity values on eye corner patches and background images respectively. Important point is to normalize intensities on each image taking into account only those pixels we are going to use.

We do classification using all pixels and using only regions' from S_b centers. The results of experiments for classification using all pixels are presented in **Table I** and **Table II**.

TABLE I
Confusion matrix for Classification using all pixels on real images

	Predicted Object	Predicted Background
Object	466	34
Background	107	393

TABLE II
Confusion matrix for Classification using all pixels on normalized images

	Predicted Object	Predicted Background
Object	448	52
Background	95	405

Classification results using regions' centers are presented in form of a plot, where true positive and true negative detections are plotted against the threshold θ, see **Fig. 7** and **Fig. 8** for results on real and normalized images respectively. Straight lines on the plots show results with all pixels.

Time taken by classification using all pixels is about 160ms. Computation time is linear relatively to used pixels, or regions, as shown at **Fig. 9**.

A relation between the number of regions and threshold θ differs for real and normalized images, see **Fig. 10**, it mean that regions on normalized images are bigger because dependency is stronger, correlation is stronger, there is less distortion in data.

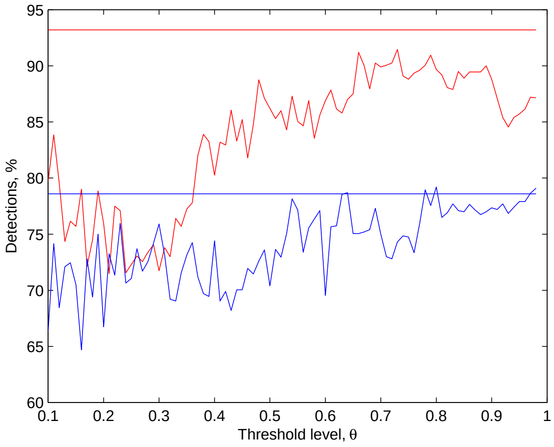


Fig. 7. Detections on real images plot. Correct object detection are shown in red, correct background - in blue.

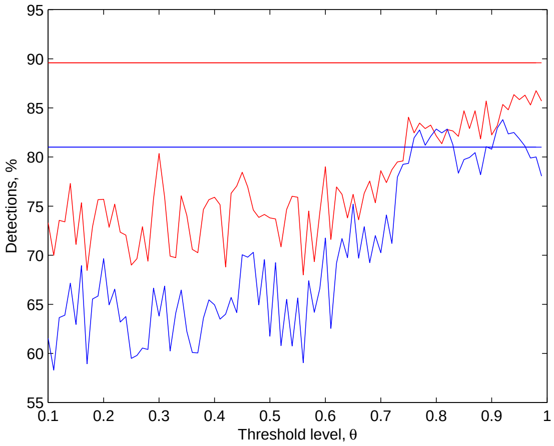


Fig. 8. Detections on normalized images plot. Correct object detection are shown in red, correct background - in blue.

Fig. 11 plot presents a relation between computing time and detection rate. It shows high detection rate even for low time.

IV. Conclusion

In presented work we have showed that proposed method can be applied to boost processing speed without significant performance losses. Practically the best result on real images is achieved with threshold $\theta = 0.8$: it gives only 4% less true positives with approximately the same amount of true negatives compared with all-pixels set up and is 14 times faster. Our approach is stable on low threshold values and gives satisfactory results. This concept is fast and robust though we literally destroy the image by drastically reducing amount of pixels taken into account. We have showed the way to find strong relationships in the image. This method can be used with more complex systems, is not restricted to neither classification nor Bayesian inference, and later research is needed to discover other choices of using Coefficients of determination.

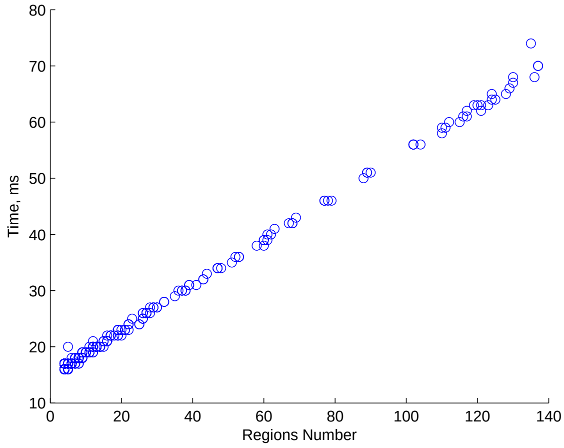


Fig. 9. Used pixels vs Time

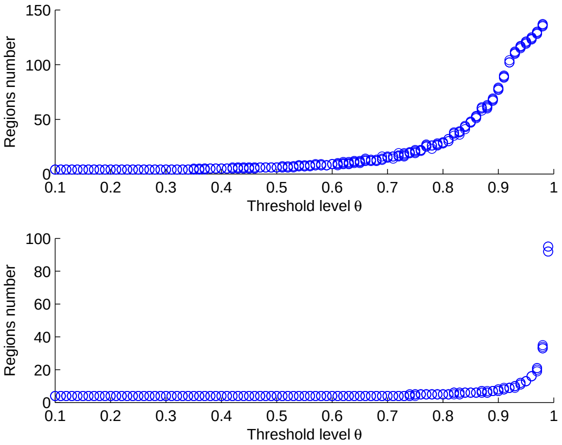


Fig. 10. Regions number vs Theta for real images on top and for normalized images in the bottom

Acknowledgment

The authors would like to thank German Research Foundation (DFG) for funding.

References

[1] Muhammad Khurram Khan, Jiashu Zhang, An Intelligent Fingerprint-Biometric Image Scrambling Scheme, Proceedings of the 3rd International Conference on Intelligent Computing: Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence, August 21-24, 2007, Qingdao, China

[2] Guanrong Chen, Yaobin Mao, Charles K. Chui. A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos, Solitons & Fractals, Volume 21, Issue 3, July 2004, pp. 749761

[3] Rakesh S., Ajitkumar A. Kaller, Shadakshari B. C. and Annappa B. Image Encryption using Block Based Uniform Scrambling and Chaotic Logistic Mapping. International Journal on Cryptography and Information Security (IJCIS), Vol.2, No.1, March 2012

[4] Ranganath, H.S.; Shiva, S.G.; , "Correlation of Adjacent Pixels for Multiple Image Registration," Computers, IEEE Transactions on , vol.C-34, no.7, pp. 674-677, July 1985

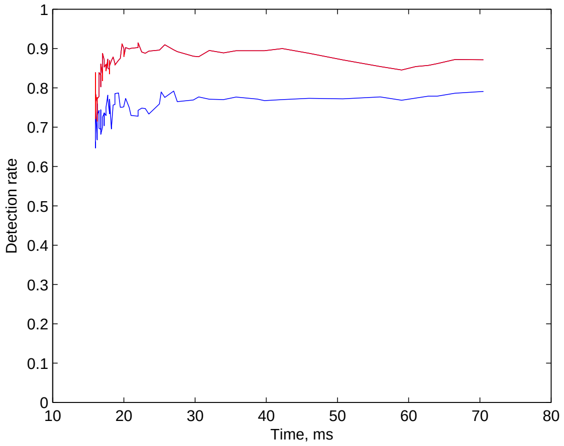


Fig. 11. Time vs Detection rate for real images

[5] Levin, Anat; Nadler, Boaz; Durand, Fredo; Freeman, William T. Patch complexity, finite pixel correlations and optimal denoising. Supplemental material for conference paper at ECCV 2012

[6] Gueguen, L.; Soille, P.; Frequent and Dependent Connectivities. Mathematical Morphology and Its Applications to Image and Signal Processing, Springer Berlin / Heidelberg, 2011, pp. 120-131

[7] Feifei Lee; Kotani, K.; Qiu Chen; Ohmi, T.; , ”Face recognition based on the combination of histogram features and rough location information of facial parts,” Signal Processing, 2008. ICSP 2008. 9th International Conference on , vol., no., pp.2065-2069, 26-29 Oct. 2008

[8] Chatterjee, S.; Hadi, A.; and Price, B. Regression Analysis by Example, 3rd ed. New York: Wiley, pp. 21-50, 2000.

[9] Kantardzic, M. Data Mining - Concepts, Models, Methods, and Algorithms, IEEE Press, Wiley-Interscience, pp. 146-149, 2003, ISBN 0-471-22852-4