# Technical Report for MI-ROZ Seminar Work
# Random Covariances for Texture Classification

Ondřej Podsztavek, podszond@fit.cvut.cz

December 27, 2017

## Abstract

This technical report presents adaptation to iris data and evaluation of random covariances for texture classification algorithm. The algorithm randomly selects regions from a texture, computes covariance descriptors and uses nearest neighbors algorithm to find similar textures. To improve its accuracy on iris data random over-sampling balancing, region cut of normalized iris image are introduced. Then, the algorithm's parameter is optimized. Evaluation shows that with this approach classification accuracy of 40% can be reached when a sufficient training dataset is provided.

## 1 Introduction

This report is about random covariances algorithm applied to iris data. Several adaptation and improvement are proposed in order to get better accuracy than with raw algorithm.

Section 3 describes iris data which should be classified in this work. Section 4 introduces concept of covariance descriptor. Section 5 surveys the random covariances algorithm followed by section 6 about experiments with the algorithm on iris data. Section 7 show the result of the adapted algorithm on two different data splits. Last section 8 proposed further improvement to the algorithm.

## 2 Goal

This seminar work aims to implement a recognition algorithm of irises using the random covariances for texture classification algorithm described in [6]. Given an iris the algorithm should recognize the subject that the iris belongs to.

## 3 Iris Data

Data provided for this work consists of 813 irises of 73 different subjects. Each iris is normalized into $100 \times 600$ RGB image (see sample normalized irises in figure 1). There is different number of samples for a subject ranging from 22 for the subject number 23 to 2 for the subject number 7.

### 3.1 Train and Test Set

There will be two evaluation of the implemented algorithm. First on train to test set ratio 1:1 and second in ratio $1{:}n-1$, where $n$ is the number of iris samples for a subject.

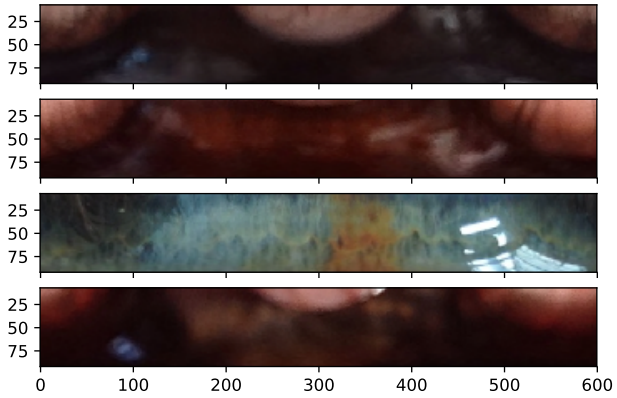


Figure 1: Sample of 3 normalized irises.

## 4 Covariance Descriptor

Let $\mathbf{I}$ be a color image and $\mathbf{F}$ corresponding $d$-dimensional feature image extracted from it by function $\phi$. Function $\phi$ projects from pixel to pixel:

$$\mathbf{F}_{xy} = \phi(\mathbf{I}_{xy})$$

Given a rectangular patch $P$ in $\mathbf{F}$ containing $K$ pixels. Then $\{\mathbf{z}_k\}_{k=1..K}$ is a set of $d$-dimensional feature points inside $P$.

The patch $P$ defined in previous section is represented with $d \times d$ covariance matrix of feature points [5]:

$$\mathbf{COV}_P = \frac{1}{K-1}\sum_{k=1}^{K}(\mathbf{z}_k - \bar{\mathbf{z}})(\mathbf{z}_k - \bar{\mathbf{z}})^T,$$

where $\bar{\mathbf{z}}$ is average of the point set. The covariance matrix combines multiple features, which might be correlated. Diagonal entries represent variance of features, while off-diagonal entries are correlations.

Moreover, covariance matrices are low-dimensional due to symmetry of $\mathbf{COV}$, because the vectorized version is only of size $\frac{(d^2+d)}{2}$:

$$\mathbf{cov}_P = \begin{pmatrix}\mathbf{COV}_{P_{11}} & \mathbf{COV}_{P_{21}} & \mathbf{COV}_{P_{22}} & \dots & \mathbf{COV}_{P_{dd}}\end{pmatrix}^T$$

The vectorized version as pointed out in [4] can be related to entities in an Euclidean space. Therefore, it usage for machine learning is straightforward.

# 5 Random Covariances

This section introduces the random covariances for texture classification algorithm provided in [6]. Random covariances algorithm starts with feature extraction from each pixel with function $\phi$, which maps a pixel to a $d = 5$ dimensional feature space:

$$\mathbf{F}_{xy} = \begin{pmatrix} I(x,y) & |\frac{\partial I(x,y)}{\partial x}| & |\frac{\partial I(x,y)}{\partial y}| & |\frac{\partial^2 I(x,y)}{\partial x^2}| & |\frac{\partial^2 I(x,y)}{\partial y^2}| \end{pmatrix}^T$$

where $I$ is the pixel intensity and the rest of the features are norms of first and second derivatives calculated with the filters $\begin{pmatrix} -1 & 0 & 1 \end{pmatrix}^T$ and $\begin{pmatrix} -1 & 2 & -1 \end{pmatrix}^T$.

Then $s$ random square patches with random sizes are sampled from $\mathbf{F}$ and for each patch the covariance descriptor is computed. Therefore each image is represented with $s$ covariance descriptors. Given $t$ training images there is a total of $s \cdot t$ covariance descriptors.

A test sample is then classified using $k$ nearest neighbors algorithm. Again $s$ covariance descriptors are extracted from the test image. Based on distances to nearest training covariance descriptors the test sample is assigned to class with the most votes.

# 6 Experiments

This section describes approaches to adapt, modify and improve the random covariances for texture classification algorithm for classification of irises in order to gain reasonable performance.

A half of the data is left out as test set for final evaluation and finally there will be also for classification only from one iris sample as mentioned in subsection 3.1. The rest of the data are used for improving the algorithm accuracy. Thus, this half is further split into train and validation set in ratio 3:1.

## 6.1 Adaptation to Iris Data

The original random covariance for texture classification algorithm was tested on Brodatz texture database [1]. It samples $s = 100$ random covariances from patches of random sizes from $16 \times 16$ to $128 \times 128$ and used $k = 5$ for nearest neighbors algorithm. Furthermore, it did not used the Euclidean properties but computes distances between covariance matrices according to formula discussed in [2]:

$$d(\mathbf{A}, \mathbf{B}) = \sqrt{\sum_{i=1}^{n} \ln^2 \lambda_i(\mathbf{A}, \mathbf{B})}$$

where $\mathbf{A}$ and $\mathbf{B}$ are symmetric positive definite matrices with eigenvalues $\lambda_i(\mathbf{A}, \mathbf{B})$ from $|\lambda\mathbf{A} - \mathbf{B}| = 0$.

To use the algorithm for irises classification the maximal size of patches has been updated to $100 \times 100$ as that is the maximal possible size and the benefit of Euclidean properties of covariance descriptor is used so a simple metric can be applied. Other parameter are adapted with same values. In figure 2 a feature channels of an iris are shown.

Pixel intensities were computed with formula:

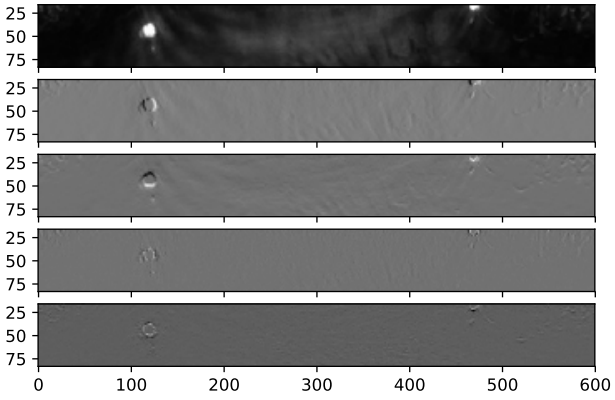


Figure 2: Each feature of an extracted feature image $\mathbf{F}$.

$$I(x,y) = 0.299r + 0.587g + 0.114b$$

where the coefficients are taken from Rec.ITU-R BT.601-7 and the $r$, $g$, $b$ are values of red, green and blue channels of the image respectively.

The algorithm adapted this way was trained on training set and evaluated on validation set. Because of random selection of patches the resulting accuracy is not the same for all runs. Therefore, average of 10 runs was taken. The accuracy and standard deviation on validation was $9.36\% \pm 1.63$.

That is very poor performance but when looked at a run it might have reasonable cause. The iris data are unbalanced so there is a unequal number of covariance descriptors for each subject in the $k$ nearest neighbors algorithm. This might possibly lead to missclassification of similar subject due to the unbalance. For instance, iris of subject number 26 with 2 irises in training set is missclassified as subject number 13 which has 5 irises in training set. Overall this mean that there is 300 more covariance descriptors given to the classifier for subject number 13 than for the other one.

For this reason in the next subsection over-sampling of training set is introduced in preprocessing stage.

| **approach** | **accuracy (%)** |
|---|---|
| basic | $9.36 \pm 1.63$ |
| balancing | $17.16 \pm 1.54$ |
| $s = 600$ | $39.82 \pm 2.06$ |
| empirical region | $41.01 \pm 2.17$ |

Table 1: Table of accuracies for different approaches. Each value is average of 10 runs. Standard deviation is also provided.

## 6.2 Balancing

Balancing the training data ensures that there will be same number of covariance descriptors for the random covariance classifier. In general, there are two approaches to balancing — over-sampling and under-sampling.

Under-sampling would left too few samples in training set. For example, random under-sampling is going to left only

one iris for a subject. Thus, over-sampling method is more suitable. Random over-sampling method was chosen. Its use is the same as picking more patches from some irises.

After employing this preprocessing technique the accuracy increased to $17.16\% \pm 1.54$, which is still poor result but the increase is significant.

### 6.3 Tuning the $s$ Parameter

One huge advantage of random covariances for texture classification algorithm is that there is only the $s$ hyperparameter. This section presents its selection in order to improve accuracy.

Original algorithm uses $s = 100$ so it was chosen to evaluate accuracies on validation set for $s \in \{50, 100, 200, \ldots, 900, 1000\}$. The result is presented in figure 3. It shows that after $s = 600$ the performance is not improving significantly. Note that there was only one run for each setting so there might be a deviation as in previous evaluations (see table 1).

The algorithm provided with balanced data and parameter $s = 600$ reaches accuracy $39.82\% \pm 2.06$ on validation set. With such a naive random approach this is already pretty good result in comparison with the previous results.

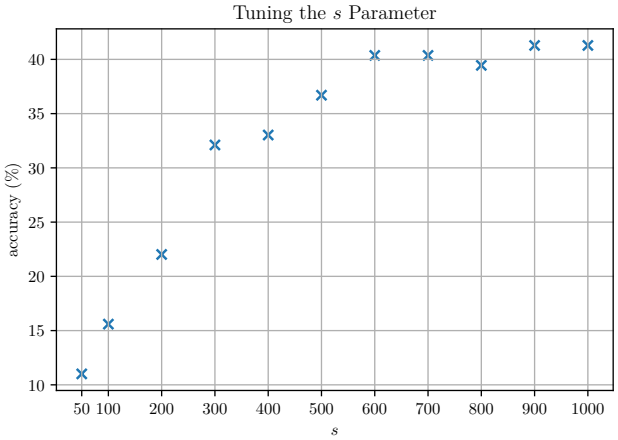


Figure 3: Dependence of validation set accuracy on the $s$ parameter. After $s = 600$ it seems that the performance will not improve. Note that there might be some deviation because there is only one run for each parameter setting.

There is one drawback of increases the $s$ parameter and that is also increases the computational time. In comparison with $s = 100$ it is approximately 6 times slower taking into account both covariance feature extraction and classification of new samples.

### 6.4 Empirical Region Selection

As stated in [3] non-iris occlusions may confuse classifier during iris recognition. Both figures 1 and 4 show examples of iris occlusions by eyelid, eyelash or reflection.

This section introduces preprocessing method to extract only part of and iris image for covariance descriptor computation. Empirically by visualization of iris data it seems reasonable to extract bottom-middle part of each iris because it contains the smallest number of occlusions. Concretely, it means extracting bottom half of an iris and left out 150 pixel on each side. Resulting cut region is shown in figure 4.

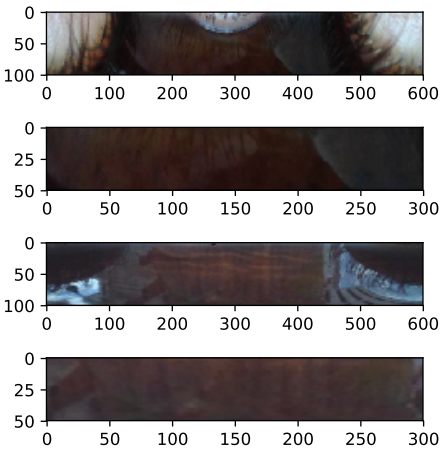


Figure 4: Two examples of iris occlusion with corresponding cut regions. This figure show that by selection bottom-middle part of irises reflection, eyelid and eyelash occlusions might be remove from an iris.

Again accuracy average of 10 runs was measured giving $41.01\% \pm 2.17$. This is an improvement but it is so small that probably the occlusions are not big problem for the random covariances algorithm.

## 7 Evaluation

As described in subsection 3.1 the final algorithm was evaluated on the two data splits. The resulting algorithm consist of random over-sampling and region cut off together with setting $s = 600$.

For test set of half the size the original data, which is unseen by the algorithm during the tuning procedure described in section 6, the accuracy average from 10 runs is $42.14\% \pm 0.93$. That is higher that any result in table 1. It shows that the number of training samples is crucial for this algorithm as train and validation sets were combined and used as training data in this evaluation stage. On the other hand the accuracy gap is not so big.

Second evaluation consist of 73 irises as training data — one iris for a subject. The proposed algorithm has $20.04\% \pm 0.41$ accuracy when give such training data.

Overall, the result is not satisfactory but proves that the random covariances for texture classification algorithm is usable. Next section gives some ideas how to further improve its performance on iris data.

## 8 Discussion

This section discusses the proposed approach and gives some idea how to improve its accuracy.

The random covariances for texture classification has shown that it can classify irises with around 40% accuracy. Two preprocessing techniques were introduces in order to improve it result.

To further increase its classification accuracy different metric might be used. This work uses standard Euclidean distance, while the original algorithm uses distance between two covariance matrices.

There might be more preprocessing step to gain some performance. For example, different features might be extracted from an iris with function $\phi$ as color information. Moreover, the irises might be split according to color before classification.

Lastly a more sophisticated approach than the one shown in subsection 6.4 for detection of region where to compute the covariance descriptor should be used. It should result in better feature extraction without noise and thus improved performance. For example, algorithm for non-iris occlusion detection described in [3] would be used.

Concerning the computational time, which for $s = 600$ is much higher that for smaller $s$ it is not a big problem because there is a algorithm for fast computation of covariance matrices provided in [6] not implemented in this work.

One disadvantage of this algorithm is that it accuracy depends on randomly extracted covariances. This instability is a problem but this work shows that the standard deviation is not so high.

# 9 Conclusion

This work present adaptation of random covariances for texture classification to iris data recognition. Two preprocessing procedures are introduces and the $s$ parameter is optimized. Evaluation shows that in such setting the algorithm can reach around 40% accuracy. Furthermore, some additional step to improve accuracy are discussed.

# References


[1] Phil. Brodatz. *Textures*. Dover Publications, Mineola, N.Y., 1999.

[2] Wolfgang Förstner and Boudewijn Moonen. *A Metric for Covariance Matrices*, pages 299–309. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[3] Michal Haindl and Mikuláš Krupička. Unsupervised detection of non-iris occlusions. *Pattern Recognition Letters*, 57(Supplement C):60 – 65, 2015. Mobile Iris CHallenge Evaluation part I (MICHE I).

[4] Marco San, Marco Crocco, Marco Cristani, Samuele Martelli, and Vittorio Murino. Heterogeneous auto-similarities of characteristics (hasc): Exploiting relational information for classification. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 809–816, Washington, DC, USA, 2013. IEEE Computer Society.

[5] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 30:1713–1727, 03 2008.

[6] Oncel Tuzel, Fatih Porikli, and Peter Meer. *Region Covariance: A Fast Descriptor for Detection and Classification*, pages 589–600. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.