

## Resilient Image Feature Description through Evolution

**Abstract:** Feature description is an important stage in many different vision algorithms. Image features detected by various detectors can be described using descriptors either with a binary or floating-point structure. This study presents the use of evolutionary algorithms, namely Genetic Algorithms (GA), in order to improve the robustness of the feature descriptors against increasing levels of photographic distortions such as noise or JPEG compression. Original feature descriptors were evolved in order to reduce the descriptor distance for the mentioned test cases. Results, tested using a statistical framework, suggest that the evolved descriptors offer better matching performance for two state-of-the-art descriptors.

**Keywords:** Feature Description, Matching, Genetic Algorithms, Binary Descriptors, Floating Point Descriptors, Optimization.

### 1. Introduction

A *feature* is an image primitive that contains valuable information about the content of the image. Every feature appearing in an image shadows a real-world object. A feature can be in form of a corner [1], an edge [2], a small region (blob) [3] or a segment [4]. Features are represented using *descriptors*, which are calculated using the pixel information around the feature using a variety of methods: A small patch of surrounding pixels can comprise the descriptor, or a more complex description like an oriented gradient histogram [5]. The literature presents many different feature detectors and descriptors. Evaluations of many feature detectors can be found in [6-8]. Based on the review given therein, a good feature detector should be able to detect features that are stable in terms of geometry under different viewing conditions [9, 10], should present significant amount of variation in its neighbourhood so that they will be prominent and provide useful information as well as presenting good localization accuracy [11]. It is also important for the detector to detect such features in a reasonable amount of time, a vital requirement for real-time applications.

It is known that the quality of images can be seriously affected by various imaging conditions including motion, diffusive environment, high compression or low signal to noise rates [12]. The descriptors mentioned above display a significant amount of robustness, or invariance, against such distortions. The inherent methods used in the descriptor extraction algorithms employ a number of approaches to achieve high performance, for instance robustness against noise is achieved using Gaussian smoothing.

The main research question tackled in this paper is whether algorithms for enhancing the results already achieved by these inherent methods can be developed using sophisticated approaches. Evolutionary computation methods such as GA can be a good candidate for improving the robustness of the descriptors.

There are a very limited number of studies focused on the use of evolutionary algorithms [13] for enhancing the performance of features used in computer vision [14]. The work by Chen et al. [15] used Ant Colony Optimization for reducing the image features (not key-points as proposed in this paper) used in classification such as first/second order moments, entropy, *etc.* A different study conducted in [16] aimed to select the optimal key-points extracted by SIFT to be used for face recognition. These optimal set of key-points were the ones producing the best matches. Referred work presented a theoretical approach rather than solid experimental results. A very recent study [17] employed GA for improving the coverage of image features across the image using a robust technique based on spatial statistics as proposed in [7].

A broad classification of the descriptors can be made by looking at their structure: binary and floating-point descriptors. The binary descriptors include BRIEF [18], ORB [19], BRISK [20], FREAK [21] and LATCH [22], whereas the integer and floating-point descriptors are SIFT [23], SURF [24], KAZE [25] and AKAZE [26]. The work presented in this paper aims to improve the original descriptors generated by both binary and floating-point descriptors in order to improve robustness against a variety of changes in imaging conditions such as noise, blur and JPEG compression. A GA was used here to improve the robustness of the image descriptors under severe cases of each condition. A detailed statistical analysis using Mc Nemar's test [27] showed that there are statistically significant differences between the performances of the original and the evolved descriptors.

The rest of the paper is structured as follows: Section 2 gives details of the evolutionary approach for descriptors with binary and floating-point structure. The experimental design is presented in 3, followed by Section 4 where the results are presented with a detailed statistical analysis. Finally, conclusions are drawn in Section 5.

## 2. Evolutionary Approach for Feature Descriptors

A feature descriptor is essentially a vector, either binary or floating-point, describing a local neighbourhood around a feature. This structure of a descriptor is very suitable for representation in form of a chromosome in GA context. The following sections will elaborate more on the chromosome representation, matching methods and the genetic operators used in the evolutionary process used to evolve original descriptors into a form that is more robust against a variety of distortions.

### 2.1. Binary Descriptors

#### 2.1.1. Matching method

A conventional method for matching binary descriptors is the Hamming distance [28] which can be defined based on the following quantities:

- $f_{00}$ : num. of positions where both descriptors have 0s.
- $f_{01}$ : num. of positions where the first has 0 and the second has a 1.
- $f_{10}$ : num. of positions where the first has 1 and the second has a 0.
- $f_{11}$ : num. of positions where both descriptors have 1s.

Based on these quantities, the Hamming distance  $d_H$  is defined as follows:

$$d_H = \frac{f_{11} + f_{00}}{f_{00} + f_{01} + f_{10} + f_{11}} \quad (2)$$

This distance can also be computed efficiently using options provided by modern processor instruction sets (e.g. *X-OR* operation) [28].

#### 2.1.2. Chromosome representation

A binary feature descriptor is a bit string with constant size, hence can be represented as a binary chromosome:

$$\langle x_1, x_2, \dots, x_n \rangle, x_i \in \{0,1\} \quad (3)$$

where  $n$  is the descriptor size which can be 64, 128 or 256 bits for current binary descriptors. The gene size for the ORB descriptor was chosen as 256.

Starting with an initial population of 20 individuals, the evolutionary operations create new individuals to be added to a population of maximum 100 individuals.

#### 2.1.3. Genetic operators

**Selection:** The GA selects the most suitable (fittest) individuals in order to create new individuals using the cross-over. This selection was implemented using the tournament selection [13].

In this selection type, 10 tournaments are performed with a tournament size of 8. Randomly selected individuals are used to

fill each tournament, then a mating pool is created by selecting the fittest individuals of the tournaments.

**Recombination:** An important stage in evolutionary computing is the phase where new individuals are added to the population. This stage is called recombination (or cross-over). The new individuals carry genetic information inherited from the parents selected during selection.

The recombination was implemented using two-point crossover for the binary descriptors. The length of the chromosome region to be exchanged (*i.e.* cut-length) was selected as 128 bits.

**Mutation:** A third evolutionary operator in the GA is mutation where newly created individuals are subject to random changes in their chromosome structure. For a binary descriptor this change was trivially performed using negation or flip.

### 2.2. Floating-Point Descriptors

#### 2.2.1. Matching method

The matching method used for comparing two floating-point descriptors is selected as the Euclidean distance between two vectors:

$$d_E = \sqrt{\sum_{i=1}^m (x_i - y_i)^2} \quad (4)$$

where  $m$  is the size of the 128 bin floating-point KAZE descriptor for two descriptors  $x$  and  $y$ .

#### 2.2.2. Chromosome representation

The chromosome representation for the floating-point descriptors follows the same notation with the binary descriptor, except for the fact that the genes are from the real number domain ( $\mathbb{R}$ ):

$$\langle x_1, x_2, \dots, x_n \rangle, x_i \in \mathbb{R} \quad (5)$$

#### 2.2.3. Genetic operators

**Selection:** The selection mechanism for deciding the mating pool that will be used in the recombination phase follows the same structure described for the binary descriptors.

**Recombination:** The gene structure in the floating-point description varies from the binary descriptors in nature, and hence requires different genetic operators.

Whole arithmetic recombination was proposed in [29] for chromosomes with floating-point structure as:

$$\begin{aligned} x_{child1} &= \alpha \cdot x_{parent1} + (1 - \alpha) \cdot x_{parent2} \\ x_{child2} &= \alpha \cdot x_{parent2} + (1 - \alpha) \cdot x_{parent1} \end{aligned} \quad (6)$$

which uses  $\alpha$  as an *aggregation weight* for combining the information residing in alleles from both parents. In the equation,  $\alpha \neq 0.5$ , since this would result in the very same genetic information to be transferred to both children.

**Mutation:** Individual diversity was ensured using the mutation operator which generated a random value between zero and unity to be employed as an offset. This offset is added to a randomly selected gene position based on the mutation rate.

### 2.3. Fitness Definition

The fitness of a chromosome regardless of the binary or the floating-point structure is based on the matching performance of the descriptor for a number of distortions described in the experimental framework in Section 3.

The fitness of an individual in the GA is defined to minimize the total descriptor distance for different levels of distortions such as noise, blur or JPEG compression. Here, the Hamming distance was employed for the binary descriptors and the Euclidean distance was used for the floating-point descriptors. Let  $d_N$ ,  $d_G$  and  $d_J$  be the distances between the original descriptor and the ones that were extracted from images subject to noise, Gaussian blur and JPEG compression respectively, the fitness function  $f(i)$  for an individual  $i$  in the GA is designed as:

$$f(i) = \frac{1.0}{\sum_{k=1}^l d_N + \sum_{k=1}^l d_G + \sum_{k=1}^l d_J} \quad (7)$$

where  $l$  is the distortion level described in Section 3.1.

This fitness function will result in a lower fitness value for an evolved descriptor as the total distance for each distortion case increases. This enforces the descriptors to evolve so that they will result in lower distance values in case of different levels of a variety of distortions.

## 3. Experimental Design

The following sections will elaborate the experimental framework by presenting the dataset created for different cases of distortion levels and then describe the measures used in the evaluation along with the statistical test employed.

### 3.1. Dataset

The original Oxford database (which can be found at <http://www.robots.ox.ac.uk/~vgg/research/affine/>) was used to create the dataset used in testing the presented evolutionary approach. A new database was developed here by adding different levels of distortions as shown in Figure 1.

Here there are 3 main levels with 5 different sub-levels of distortions including noise, blur and JPEG compression. Details for these levels are presented in Table 1. For the JPEG compression, a lower value from the 0 – 100 scale indicates a higher amount of compression and hence higher distortion in the image as can be seen from Figure 1. The original database included 8 datasets including bark, bikes, boat, graf, leuven, trees, wall and ubc. The newly created database includes a total of  $8 \times 3 \times 5 = 120$  images, a relatively large enough database for

drawing statistically meaningful conclusions [8].

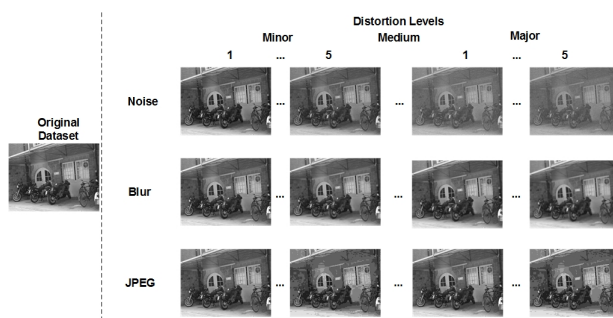


Figure 1 Generated dataset by addition of varying levels of distortions

Table 1. Distortion levels used to create the dataset

	Minor				
Noise	0.01	0.02	0.03	0.04	0.05
Blur	3	5	7	9	11
JPEG Compression	1	85	90	95	100
	Medium				
Noise	0.06	0.07	0.08	0.09	0.1
Blur	7	9	11	13	15
JPEG Compression	1	20	40	60	80
	Major				
Noise	0.11	0.12	0.13	0.14	0.15
Blur	11	13	15	17	19
JPEG Compression	1	10	20	30	40

### 3.2. Performance Measures

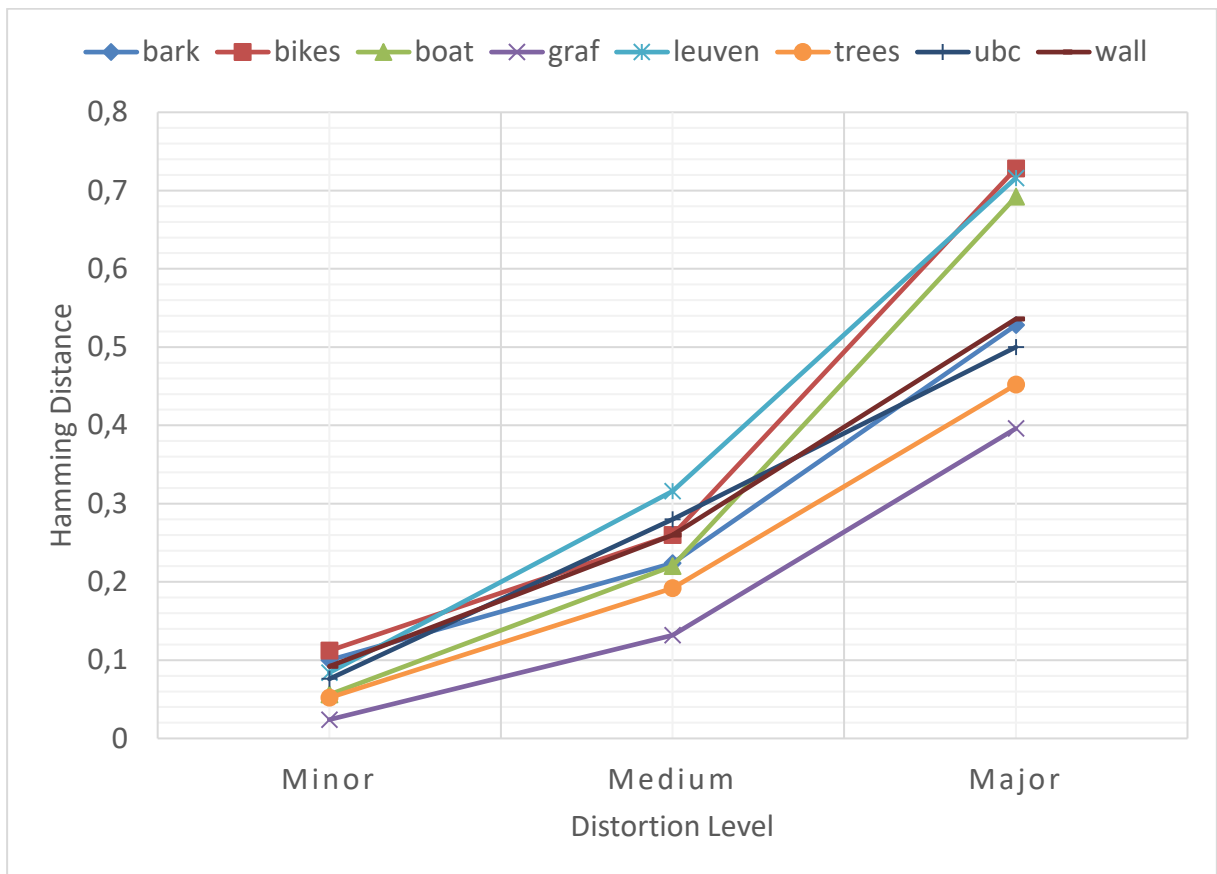
The measures for evaluating the results are the same measures actually used for finding feature matches. The Hamming distance was used to measure the performance of the binary descriptor and the Euclidean distance was employed for the floating-point descriptor.

### 3.3. Evaluation

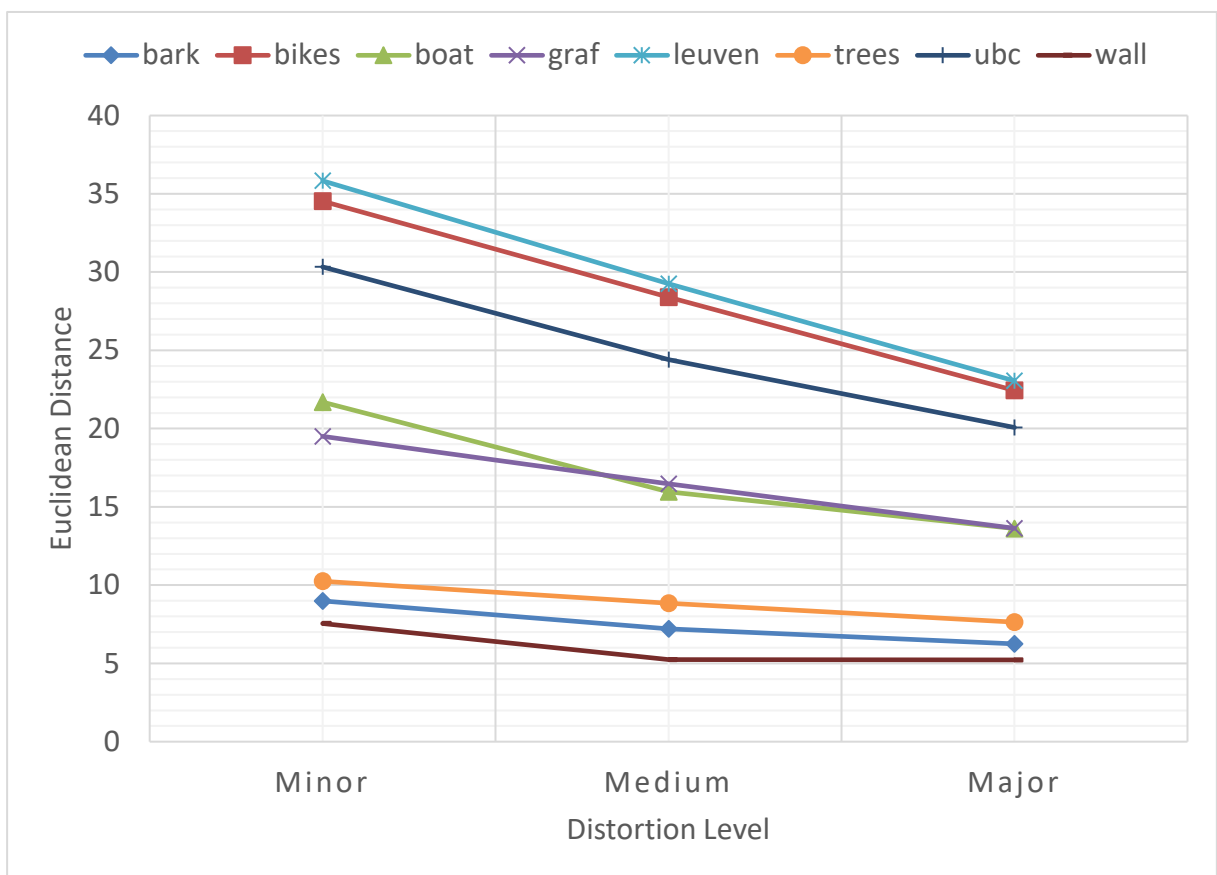
The evaluation method adopted for comparing the original descriptors with the evolved descriptors is the Mc Nemar's test which is a non-parametric test, a variant of  $\chi^2$  test. The test creates  $2 \times 2$  contingency tables in order to compute z scores. This test has a published past of usage by medical research community [30] and has recently been used for performance comparison in computer vision for the first time by Clark [31] and later for machine learning by Bostanci [27]. The test is significantly robust against Type-I error, *i.e.* detecting a difference when there is none.

### 3.4. Results

This section presents results of the statistical analysis performed in order to compare the performances of the original descriptors with the evolved descriptors. Figure 2 demonstrates the effect of the evolutionary process for the binary and the floating-point descriptors.



(a)



(b)

**Figure 2.** Effect of evolution for 3 levels of distortions for different datasets for the (a) binary descriptor (ORB) (b) floating-point descriptor (KAZE).

In the plots given above, eight datasets created by adding varying levels of distortions were tested for the descriptors. The change from the original descriptor structure after the evolutionary process was demonstrated here. The binary descriptor had more changes in its gene structure as the level of distortion increased.

Note that this distance is normalized by the total number of bins used by the descriptor, hence the normalized distance values.

It is interesting to see that the floating-point descriptor required less change in its gene structure through the evolutionary process. Lower levels of distortions created a higher amount of need for

change in the descriptor.

Tables 2 and 3 show the actual descriptor distances between matched images for each couple of images in the datasets *e.g.* matching the first image from the bark dataset with the rest five images in the same dataset. The benefit of the evolutionary process presented in this work is clear from the results, *i.e.* the evolved descriptors result in a smaller descriptor distance for both binary and floating-point descriptors.

**Table 2.** Average Hamming distance for the ORB descriptor for the datasets for five noise, blur and JPEG compression cases under each distortion level.

		Minor		Medium		Major	
		Original	Evolved	Original	Evolved	Original	Evolved
bark	Noise	4.8472	4.8288	11.2312	11.2264	17.8912	17.896
	Blur	6.5856	6.5144	11.3136	11.1424	16.7328	16.3232
	JPEG	8.4576	8.4728	10.5416	10.6024	13.0648	13.1624
bikes	Noise	6.9248	6.9056	15.224	15.2184	21.7992	21.7928
	Blur	7.4440	7.3768	12.448	12.2664	17.9360	17.4272
	JPEG	7.6624	7.6864	10.5400	10.5824	13.5680	13.6928
boat	Noise	4.4832	4.4864	9.884	9.9024	14.2824	14.2912
	Blur	8.5632	8.5232	14.3176	14.1600	20.5000	19.9824
	JPEG	5.7584	5.7744	7.6056	7.6544	9.5464	9.69920
graf	Noise	3.9064	3.9000	8.7128	8.6928	13.0112	13.0248
	Blur	6.9824	6.9696	12.4624	12.3464	18.5896	18.2944
	JPEG	5.3736	5.3784	7.0080	7.0584	8.8904	8.9840
leuven	Noise	4.6760	4.6448	10.5088	10.5192	15.2504	15.2208
	Blur	7.1376	7.0744	11.9624	11.7168	17.088	16.5608
	JPEG	5.5448	5.5824	7.5584	7.6312	9.6408	9.8096
trees	Noise	3.9712	3.9576	9.5112	9.4984	15.2728	15.2768
	Blur	7.2184	7.1840	12.6944	12.5264	18.7808	18.4072
	JPEG	6.6760	6.6848	8.3568	8.4208	10.2680	10.4112
wall	Noise	4.8616	4.8528	11.0976	11.0944	16.924	16.9232
	Blur	8.3664	8.3032	14.1552	13.936	20.7256	20.3696
	JPEG	6.6264	6.6496	8.8416	8.9088	10.9672	11.0688
ubc	Noise	5.6456	5.6176	13.4128	13.4072	21.368	21.3568
	Blur	8.4328	8.3632	14.3656	14.1712	21.0336	20.6096
	JPEG	10.4168	10.4368	13.0288	13.1176	15.8816	15.9920

**Table 3.** Average Euclidean distance for the KAZE descriptor for the datasets for five noise, blur and JPEG compression cases under each distortion level.

		Minor		Medium		Major	
		Original	Evolved	Original	Evolved	Original	Evolved
bark	Noise	0.3094	0.3043	0.3243	0.3204	0.3521	0.3496
	Blur	0.2489	0.2440	0.2378	0.2348	0.2562	0.2539
	JPEG	0.3427	0.3375	0.3395	0.3355	0.3461	0.3434
bikes	Noise	0.4666	0.4374	0.4742	0.4507	0.4754	0.4586
	Blur	0.4119	0.3849	0.3800	0.3605	0.3694	0.3555
	JPEG	0.4832	0.4544	0.4828	0.4592	0.4797	0.4615
boat	Noise	0.4159	0.4010	0.4169	0.4068	0.4339	0.4262
	Blur	0.3212	0.3120	0.2845	0.2779	0.2980	0.2931
	JPEG	0.4265	0.4127	0.4249	0.4152	0.4262	0.4187
graf	Noise	0.3817	0.3685	0.3862	0.3754	0.3896	0.3808
	Blur	0.3171	0.3058	0.2956	0.2872	0.3069	0.3017
	JPEG	0.3882	0.3757	0.3888	0.3778	0.3906	0.3819
leuven	Noise	0.4706	0.4362	0.4744	0.4494	0.4762	0.4566
	Blur	0.4031	0.3745	0.3668	0.3447	0.3557	0.3389
	JPEG	0.4794	0.4466	0.4796	0.4547	0.4751	0.4554
trees	Noise	0.3410	0.3348	0.3536	0.3483	0.3806	0.3765
	Blur	0.2591	0.2543	0.2476	0.2445	0.2867	0.2845
	JPEG	0.3677	0.3616	0.3635	0.3583	0.3631	0.3587
wall	Noise	0.4981	0.4699	0.5003	0.4798	0.5015	0.4853
	Blur	0.4079	0.3875	0.3808	0.3645	0.4074	0.3956
	JPEG	0.5069	0.4799	0.5038	0.4832	0.4998	0.4830
ubc	Noise	0.3103	0.3063	0.3356	0.3331	0.3697	0.3678
	Blur	0.2300	0.2277	0.2441	0.2429	0.3125	0.3112
	JPEG	0.3452	0.3414	0.3444	0.3418	0.3504	0.3485

A robust statistical test was also performed in order to investigate whether the performance differences between the original and the evolved feature descriptors are indeed statistically significant. The results of the Mc Nemar's test are depicted in Tables 5 and 6. For both the binary descriptor and the floating-point descriptor, the evolution resulted in smaller distances (shown in the plots with *c*). Statistical significance of these comparisons are

represented with *z* scores. Note that these results should be analyzed using Table 4 (given in [27, 31]) which presents the confidence levels as an indicator of the statistical significance based on *z* scores. In terms of statistical significance, the binary case resulted in significant results for 3 out of 8 datasets, while the floating-point descriptor yielded statistically significant results for the complete dataset.

**Table 4.** Confidence levels corresponding to *z* scores for one-tailed and two-tailed predictions

<i>z</i> score	One-tailed Prediction	Two-tailed Prediction
1.645	95%	90%
1.960	97.50%	95%
2.326	99%	98%
2.576	99.50%	99%

**Table 5.** Mc Nemar's test results for the binary descriptor. *c* is the number of times when the original and the evolved descriptor resulted in a smaller Euclidean distance. *z* is the measure of the statistical significance of *c*. Cases in which the performance difference was statistically significant for one and two-tailed predictions with confidence levels above 90% are denoted with \*.

	bark		bikes		boat		graf		leuven		trees		ubc		wall	
	<i>c</i>	<i>z</i>	<i>c</i>	<i>z</i>	<i>c</i>	<i>z</i>	<i>c</i>	<i>z</i>	<i>c</i>	<i>z</i>	<i>c</i>	<i>z</i>	<i>c</i>	<i>z</i>	<i>c</i>	<i>z</i>
Original	17	1.3568	17	1.4907	21	0.2981	16	1.7889	18	1.1926	14	2.3851	18	1.1926	16	1.7889
Evolved	28		28		24		29*		27		31*		27		29*	

**Table 6.** Mc Nemar's test results for the floating-point descriptor. *c* is the number of times when the original and the evolved descriptor resulted in a smaller Euclidean distance. *z* is the measure of the statistical significance of *c*. Cases in which the performance difference was statistically significant for one and two-tailed predictions with confidence levels above 90% are denoted with \*.

	bark		bikes		boat		graf		leuven		trees		ubc		wall	
	<i>c</i>	<i>z</i>	<i>c</i>	<i>z</i>	<i>c</i>	<i>z</i>	<i>c</i>	<i>z</i>	<i>c</i>	<i>z</i>	<i>c</i>	<i>z</i>	<i>c</i>	<i>z</i>	<i>c</i>	<i>z</i>
Original	0	6.5591	0	6.5591	0	6.5591	0	6.5591	0	6.5591	0	6.5591	0	6.5591	1	6.2610
Evolved	45*		45*		45*		45*		45*		45*		45*		44*	

## 4. Conclusion

This paper presented an evolutionary approach for increasing the robustness of binary and floating-point descriptors. The test cases were comprised of photographic distortions such as noise, blur and JPEG compression of varying severity. A comparison was conducted for analysing their performance against the original descriptor using a new dataset derived from a popular dataset.

Results show that the evolved descriptors are different from the original descriptors since they were evolved in order to accommodate differences in local image regions due to the various types of distortions. It was shown that the performance of the evolved descriptor is significantly, in statistical terms, better than the original. Smaller descriptor distances between the evolved descriptor and the ones extracted from datasets suggest better matching performance under different conditions.

This study has shown that current state-of-the-art feature description algorithms have still room for improvement for creating completely robust feature descriptors. The evolutionary algorithms require a significant amount of time to generate satisfactory results. Real-time applications require fast detection and description of image features. Future work will investigate how the lessons learned from these results can be incorporated into the descriptor extraction process.

## References

- M. Zuliani, C. Kenney, and B. S. Manjunath, "A mathematical comparison of point detectors," in *IEEE Computer Vision and Pattern Recognition*, 2004.
- M. Shneier, "Extracting linear features from images using pyramids," *Systems, Man, and Cybernetics, IEEE Transactions*, vol. 12, pp. 569-572, 1982.
- G. R. Cross and A. K. Jain, "Markov random field texture models," *Pattern Analysis and Machine Intelligence*, no. 1, pp. 25-39, 1983.
- J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 731-737, 1997.
- N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", in *Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, vol. 1, pp. 886-893, 2005.
- T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors:A survey", *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177-280, 2008.
- E. Bostanci, N. Kanwal, and A. Clark, "Spatial statistics of image features for performance comparison," *Image Processing, IEEE Transactions*, vol. 23, pp. 153-162, 2014.
- N. Kanwal, E. Bostanci, and A. Clark, "Evaluation method, dataset size or dataset content: How to evaluate algorithms for image matching?," *Journal of Mathematical Imaging and Vision*, 2016.
- C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *International Journal of Computer Vision*, pp. 151-172, 2000.
- O. M. Mozos, A. Gil, M. Ballesta, and O. Reinoso, "Interest point detectors for visual slam," *Current Topics in Artificial Intelligence*, vol. 4788, pp. 170-179, 2007.
- S. Smith and J. Brady, "Susan: A new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45-78, 1997.
- E. Cohen and Y. Yitzhaky, "Blind image quality assessment considering blur, noise, and jpeg compression distortions," in *Optical Engineering+Applications, International Society for Optics and Photonics*, 2007.
- A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing* SpringerVerlag, 2003.
- S. Cagnoni, E. Lutton, and G. Olague, *Genetic and evolutionary computation for image processing and analysis*. Hindawi Publishing Corporation, 2007.
- L. Chen, B. Chen, and Y. Chen, "Image feature selection based on ant colony optimization," in *AI 2011: Advances in Artificial Intelligence*, vol. 7106 of LNCS, pp. 580-589, Springer, 2011.
- D. Ozkan, "Feature selection for face recognition using a genetic algorithm", Department of Computer Engineering, Bilkent University, 2006.
- E. Bostanci, "Enhanced image feature coverage: Key-point selection using genetic algorithms," *The Imaging Science Journal*, doi: 10.1080/13682199.2016.1254939, 2017.
- M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," *Computer Vision, ECCV 2010*, pp. 778-792, 2010.
- E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Computer Vision (ICCV)*, pp. 2564-2571, IEEE, 2011.
- S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *ICCV*, pp. 2548-2555, IEEE, 2011.
- A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer Vision and Pattern Recognition (CVPR)*, pp. 510-517, IEEE, 2012.
- G. Levi and T. Hassner, "Latch: Learned arrangements of three patch codes," *arXiv preprint arXiv:1501.03719*, 2015.
- D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91-110, 2004.
- H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer vision, ECCV 2006*, pp. 404-417, Springer, 2006.
- P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "Kaze features," *ECCV 2012*, pp. 214-227, Springer, 2012.
- P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerate features in nonlinear scale spaces," *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 34, no. 7, pp. 1281-1298, 2011.
- B. Bostanci and E. Bostanci, "An evaluation of classification algorithms using Mc Nemar's test," in *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications*, vol., pp. 15-26, 2013.
- E. Bostanci, "Is hamming distance only way for matching binary image feature descriptors?," *IET Electronics Letters*, vol. 50, pp. 806-808, May, 2014.
- Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Artificial intelligence, Springer, 1996.
- M. Fagerland, S. Lydersen, and P. Laake, "The Mc Nemar test for binary matched-pairs data: mid-p and asymptotic are better than exact conditional," *BMC Medical Research Methodology*, vol. 13, no. 1, 2013.