Journal of Universal Computer Science, vol. 26, no. 4 (2020), 454-478
submitted: 4/11/19, accepted: 12/3/20, appeared: 28/4/20 *CC BY-ND 4.0*

A Study on Pattern Recognition with the Histograms of Oriented Gradients in Distorted and Noisy Images

Andrzej Bukała
(Department of Electronics, AGH University of Science and Technology
Kraków, Poland
abukala@agh.edu.pl)

Michał Koziarski
(Department of Electronics, AGH University of Science and Technology
Kraków, Poland
michal.koziarski@agh.edu.pl)

Bogusław Cyganek
(Department of Electronics, AGH University of Science and Technology
Kraków, Poland
cyganek@agh.edu.pl)

Osman Nuri Koç
(Department of Electronics, AGH University of Science and Technology
Kraków, Poland
kocosmannuri@gmail.com)

Alperen Kara
(Department of Electronics, AGH University of Science and Technology
Kraków, Poland
alperenkara.eem@gmail.com)

Abstract: Histograms of oriented gradients (HOG) are still one of the most frequently used low-level features for pattern recognition in images. Despite their great popularity and simple implementation performance of the HOG features almost always has been measured on relatively high quality data which are far from real conditions. To fill this gap we experimentally evaluate their performance in the more realistic conditions, based on images affected by different types of noise, such as Gaussian, quantization, and salt-and-pepper, as well on images distorted by occlusions. Different noise scenarios were tested such anti-distortions during training as well as application of a proper denoising method in the recognition stage. As underpinned with experimental results, the negative impact of distortions and noise on object recognition with HOG features can be significantly reduced by employment of a proper denoising strategy.

Key Words: histogram of oriented gradients, image processing, machine learning, denoising methods

Category: I.4, I.4.4

1 Introduction

Despite great advances in pattern recognition with deep architectures, many image processing systems still rely on low-level lightweight features such as the local binary patterns (LBP), scale-invariant feature transform (SIFT) [Lowe, 2004], speeded up robust features (SURF) [Bay et al., 2008], binary robust independent elementary features (BRIEF), their rotation invariant versions (ORB) [Rublee, 2011] or the histograms of oriented gradients (HOG) [Dalal and Triggs, 2005]. This is usually due to much lower resources required for computation, training and operation of such systems, which are available e.g. in embedded systems or in systems with ensembles of simple classifiers. In this paper we focus on performance of the HOG features in such systems but evaluated in more realistic conditions. That is, performance of all of the above mentioned features have been already investigated and reported, as described in the next section. However, although the HOG features have been also reported, such scrutiny has been performed only with help of the high quality images. In practice, however, we usually face more realistic conditions in which images are frequently affected by different types of noise and distortions. Therefore in this paper we fill this gap. More precisely, in the real applications we often deal with low quality signals, affected by various phenomena caused by the environmental factors and imperfect process of image acquisition, mostly noise and distortions. Under these circumstances it is important to consider to what extent distortions can affect the performance of the HOG based model in the object recognition tasks. The second and more important question is what techniques can be applied to successfully mitigate the negative impact of distortions? In this paper we address these problems and provide some practical advises.

This is an extended version of our previous work [Bukala et al., 2019]. Here we experimentally evaluate the impact of various image distortions on the performance of classification with the HOG features. Our analysis is based on several benchmarks in combination with different classification models. We start with a classifier trained on the undistorted data, while distortions are present during the evaluation. This corresponds to the settings in which either the conditions change during the evaluation stage, or we are forced to train the model on images of a different quality than the ones observed during the evaluation. Afterwards, the case in which the same type of distortions affecting both, training and test data is considered. This setting corresponds to either the case in which overall quality of the data is low during training and testing, or we anticipate the presence of distortions during the testing, but try to induce the artificial distortions during the training as a strategy to mitigate the negative impact of distortions. Finally, in the case of noise, the possibility of applying denoising prior to classification while training the classifier on undistorted data is considered as another strategy of dealing with distortions during the evaluation of the model.

The rest of this paper is organized as follows: in Section 2 an overview of related works is provided. In Section 3 the HOG feature descriptors are described. In Section 4 the distortion models used throughout the paper is defined. The conducted experiments as well as the observed results are presented and discussed in Section 5. Finally, Section 6 presents conclusions.

2 Related Works

Noise and distortions are immanent features of all real signal processing frameworks [Grabek and Cyganek, 2019][Cyganek, 2007][Cyganek and Gongola, 2018]. However, relatively little works has been devoted to test performance of common classification patterns in presence of these phenomena. In this section we'll briefly outline the most important ones with special stress on performance of the group of the sparse feature detectors in real situations. Operation of the SIFT and SURF features in presence of deformations and noise is investigated in the work by Khan et al. [Khan et al., 2011]. Similarly, Karami et al. [Karami et al., 2017] evaluated the impact of distortions on SIFT, SURF, BRIEF and ORB features descriptors in the image matching task. Dutta et al. [Dutta et al., 2012] examined an affect of distortions in a commercial face recognition system. Interestingly, various types of signal distortions have been also scrutinized in the context of the convolutional neural networks [Koziarski and Cyganek, 2017, Dodge and Karam, 2016, Karahan et al., 2016, Vasiljevic et al., 2016]. HOG features were successfully used in numerous image recognition tasks, with the most notable examples including human detection [Dalal and Triggs, 2005,] and face recognition [Déniz et al., 2011, Do and Kijak, 2012, Tan et al., 2013], but also problems such as smile detection [Bai et al., 2009], traffic sign recognition [Stallkamp et al., 2012] and handwritten digit recognition [Ebrahimzadeh and Jampour, 2014]. To this day HOG remain important tools in the image recognition, especially in the environment in which the computational resources at our disposal are limited. However, since in such setting low-end image acquisition devices are likely to be used affecting quality of the captured data, it is important to consider the possible impact of introduced distortions on the classification performance. Nonetheless, to the best of our knowledge previous works, except our previous publication [Bukala et al., 2019], didn't examine the impact of low image quality on object recognition in images with help of the HOG features. In this paper we fill this gap.

3 Histograms of Oriented Gradients

Computation of gradients in digital signals constitutes the basic block of majority of signal analysis methods, such as the modern convolutional neural networks, the already mentioned sparse descriptors such as SIFT and SURF, as well as the HOG features which we investigate in our paper. Frequently object recognition based on gradients leads to better results than using only unprocessed intensity or statistical moments computed in discrete signals. Furthermore, it has been observed that gathering the local phases of gradient into histograms provides highly discriminative features. Following this idea many object detection and classification methods were proposed which can be divided into two groups: the global methods, in which histograms of gradients are collected from the entire image [Freeman and Roth, 1995], and the local methods, in which histograms of gradients are computed only in selected regions of the image [Cyganek, 2009]. The method of histograms of oriented gradients, as proposed by Dalal et al. [Dalal and Triggs, 2005], belongs to the latter group. It is based on evaluating normalized histograms of image gradients, computed on a dense grid with uniformly spaced cells. HOG based representation of an image provides several advantages. Namely, the gradient structure conveys information on local shapes in an image; Moreover, gradients are invariant to some of the local geometric and photometric transformations. HOG is computed by dividing the image into small spatial regions, also called cells, and then for each such a cell calculating a 1-D histogram of oriented directions, or edge orientations, over the pixels of that cell. In addition, histogram entries are weighted by the gradient magnitude computed for that entry. Finally, the HOG descriptor is constructed from the concatenated histograms of each cell. However, to increase the invariance to such phenomena as illumination, shadowing, etc. it is useful to further cross-normalize local responses in larger spatial regions. For this purpose, the cells are grouped together into larger and spatially connected regions, called blocks. In our experiments the L1-sqrt normalization is used which can be expressed as follows:

$$f = \sqrt{\frac{\upsilon}{\|\upsilon\|_1 + \epsilon}}$$

where υ is a non-normalized vector, containing all histograms in a given block, $\|\upsilon\|_1$ is its k-norm, This provides better invariance to illumination, shadowing, etc. In the following sections of this paper such a normalized version of the descriptor is called the HOG descriptor.

4 Image Distortion Models

In this section a short characteristics of the distortions used in the experiments are presented and discussed.

(a) $\sigma = 0$ (b) $\sigma = 0.1$ (c) $\sigma = 0.2$

Figure 1: Gaussian noise applied on example image.

4.1 Gaussian noise

The Gaussian noise is an additive distortion of pixel values with probability function expressed by the normal distribution, given as follows:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where μ represents mean value and σ the standard deviation. In digital images the front-end photon acquisition sensors, as well as further electronic modules in the camera, are the main the source of this type of noise [Cyganek and Siebert, 2009]. An example of an image affected by the Gaussian noise is shown in Figure 1.

(a) $\sigma = 0$ (b) $\sigma = 2$ (c) $\sigma = 4$

Figure 2: Gaussian blur applied on example image.

4.2 Gaussian blur

A Gaussian blur is another type of image distortion which can also be modeled with the help of the normal distribution function, affecting values of each pixel of an image. A visual effect of this type of distortion reflects the fact of passing only the low-frequency components of the image signal. This leads to smoothing of sharp edges and corners, also resembling the out-of-focus camera effect, as shown in (figure 2). The Gaussian function in two dimensions is:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

where x is the distance from the origin in horizontal axis, y is the distance from origin in vertical axis, and σ is standard deviation of the Gaussian distribution. This function produces a surface with concentric circles with a Gaussian distribution from the center point. Values from this distribution are used to build a convolution matrix which is then applied on the source image. It is interesting to note that in some algorithms of gradient computation in images, Gaussian blurring is sometimes considered as a preprocessing step [Cyganek and Siebert, 2009].

4.3 Quantization noise

(a) $q = 0$ (b) $q = 0.2$ (c) $q = 0.4$

Figure 3: Quantization noise applied on example image.

The quantization noise arises as a result of the quantization of the continuous intensity signal to discrete levels of the image sensor. It is dependent on the number of bits in the analog-to-digital converter, which limits the number of possible values to a set of discrete values. This type of noise can be modeled by adding a random value η from range:

$$-\frac{1}{2}q \leq \eta \leq +\frac{1}{2}q$$

, where q denotes a quantization level. On the other hand, value of η follows an uniform probability distribution p, expressed as follows:

$$p(x) = \begin{cases} \frac{1}{x_{max} - x_{min}} & \text{for } x_{min} \leq x \leq x_{max} \\ 0 & \text{otherwise} \end{cases}$$

where x_{max} and x_{min} stand for the maximum and minimum values of the argument x. The random variable η takes the values $\pm\frac{1}{2}q$ with a uniform distribution, where q is a quantization parameter that is chosen in the experiments. An example of the images affected by the quantization noise is presented in figure 3.

4.4 Salt-and-pepper noise

(a) $p = 0$ (b) $p = 0.05$ (c) $p = 0.15$

Figure 4: Salt-and-Pepper noise applied on example image.

The salt-and-pepper noise can be caused by errors in the analog-to-digital converters or memory as well as by the deteriorating phenomena in the image sensor of the camera. An image distorted in this way has erroneously bright pixels in dark areas or vice versa. Figure 4 presents an example of this phenomena. This type of noise can be modeled by combination of multiplicative and additive components, as follows:

$$\hat{s}(x) = (1 - \mu)s(x) + \mu\beta$$

where $\hat{s}(x)$ and $s(x)$ stand for distorted and pure signal respectively, μ is a random variable with probability $p = Pr(\mu = 1)$ and β is a random variable satisfying the equation $Pr(\beta = s_{max}) = Pr(\beta = s_{min}) = 0.5$.

4.5 Occlusions

Another common type of distortions encountered in practice are image occlusions. They arise naturally as an effect of interaction of the light rays with solid objects on a pathway between the object of interest and the camera sensor. Occlusions are also influenced by the scene geometry, as well as by front-end of the camera. In our experimental setup, a randomly positioned black square is placed in an image to simulate occlusions. Figure 5 shows an example of this phenomena.

(a) $f = 0$ (b) $f = 0.2$ (c) $f = 0.4$

Figure 5: Simulation of occlusions by positioning a black rectangle at random position in an image. The parameter f controls size of the occluding rectangle.

5 Experiments

5.1 Experimental Set-up

Datasets. In order to provide a reliable and varied data for our experiments we used four different image databases. Differences between datasets include both content and size of image data. The German Traffic Sign Recognition Benchmark, later referred as GTSRB [Stallkamp et al., 2012] consists of around 50,000 road sign images divided into 40 classes. For the purpose of our experiments, all images were resized to 32×32 pixels. The STL-10 database [Coates et al., 2011] contain 13,000 96×96 images divided into 10 classes describing such objects, as airplanes, birds and cars. The MNIST database [LeCun et al., 1998] is made of 70,000 32×32 grayscale images presenting handwritten digits. Finally we used grayscale version of FERET [Phillips et al., 2000] facial recognition database. Over 14,000 256×256 images are labelled into 2337 classes.

Preprocessing. To provide a benchmark for classification with HOG features we trained all classifiers also on vectorized images. For that purpose FERET data was downscaled 4 times. Furthermore, all datasets were normalized to 0-1 range prior to feature computing.

Image distortion models. In Gaussian noise models we tested different distortion levels, using $\sigma \in \{0.025, 0.05, ..., 0.25\}$. For Salt-and-Pepper noise we varied the probability of flipping a pixel $p \in \{0.02, 0.04, ..., 0.2\}$. Quantization noise levels were chosen within range $q \in \{0.05, 0.1, ..., 0.5\}$. Standard deviation for Gaussian blur ranged $\sigma \in \{0.5, 1, ..., 5\}$. For experiments with occlusion we changed the fraction of image being concealed $p \in \{0.1, 0.2, ..., 0.8\}$. In tests with random distortion levels values were chosen within the same range, independently for each image.

Histograms of Oriented Gradients were calculated using fixed number of pixels per cell and cells per block. Best parameters were found with 5-fold cross-validated grid-search independently for each dataset. Most time and memory consuming cases were removed from the results giving final parameters used in all tests, which are presented in Table 1. For the STL-10 database we tested cells of 8×8, 10×10, 12×12 and 16×16 pixels, respectively. For the GTSRB database tested values were 3×3, 4×4, 6×6 and 8×8, 3×3, 4×4 and 6×6 for the MNIST, as well as 24×24, 28×28, 32×32, 40×40, 48×48 for the FERET, respectively. For each setting, we also tested 1×1, 2×2 and 3×3 cells per block values.

Classification. All experiments utilized four classifiers for comparison: the *K-nearest neighbors* (KNN), the *Random Forest Classifier* (RFC), the *Linear Discriminant Analysis* (LDA), as well as the *Support Vector Machine* (SVM) with a linear kernel. Classifier hyperparameters were fitted for each individual case with the use of 5-fold cross-validated grid-search within range: $C \in \{0.001, 0.1, 1, 10, 100\}$ for SVM, $n\_estimators \in \{5, 10, 25, 50, 100\}$ for RFC, $n\_components \in \{1, 2, 4, 6, 8\}$ for LDA and $n\_neighbors \in \{1, 5, 10, 25, 50\}$ for KNN.

5.2 Classification of distorted images

First experimental case focused on analyzing the impact of distortions with known intensity on the classification accuracy. For each scenario both HOG and vectorized image representations were examined. Furthermore, we considered two settings: distortions applied either only to the test data, or to both training and test data. This was done to simulate situation in which distortion levels vary between data used to train classifiers and test data. This can happen due to uncertainty of possible distortions on evaluated data. Maximum distortion levels were chosen to cause near to random classification scores.

The results for this part of the experimental study were grouped by the distortion type and are presented in Figures 6 through 12. As predicted, in all cases, presence of distortions severely impacted the classification accuracy, even for distortions of seemingly small values. As can be seen on examples in Section 4, that levels of distortion do not affect the possibility of image recognition by a human examiner.

Table 1: HOG parameters chosen for each dataset

Dataset	pixels per cell	cells per block
GTSRB	(4, 4)	(2, 2)
STL-10	(16, 16)	(3, 3)
MNIST	(4, 4)	(1, 1)
FERET	(32, 32)	(1, 1)

In 15 out of 16 cases considering undistorted images classification of HOG features resulted in better performance than vectorized images, with the sole exception of FERET database classification with RFC classifier. The presence of distortions, especially of higher intensity, using vectorized images turned out superior in some of the cases. Classification of STL-10 database with LDA classifier actually resulted in improved accuracies in the presence of noise, especially when applied to both training and evaluation parts of the dataset. Most likely added noise acted as a regularization factor in that case. Adding the same type of distortions to training data also benefited the classification with HOG feature

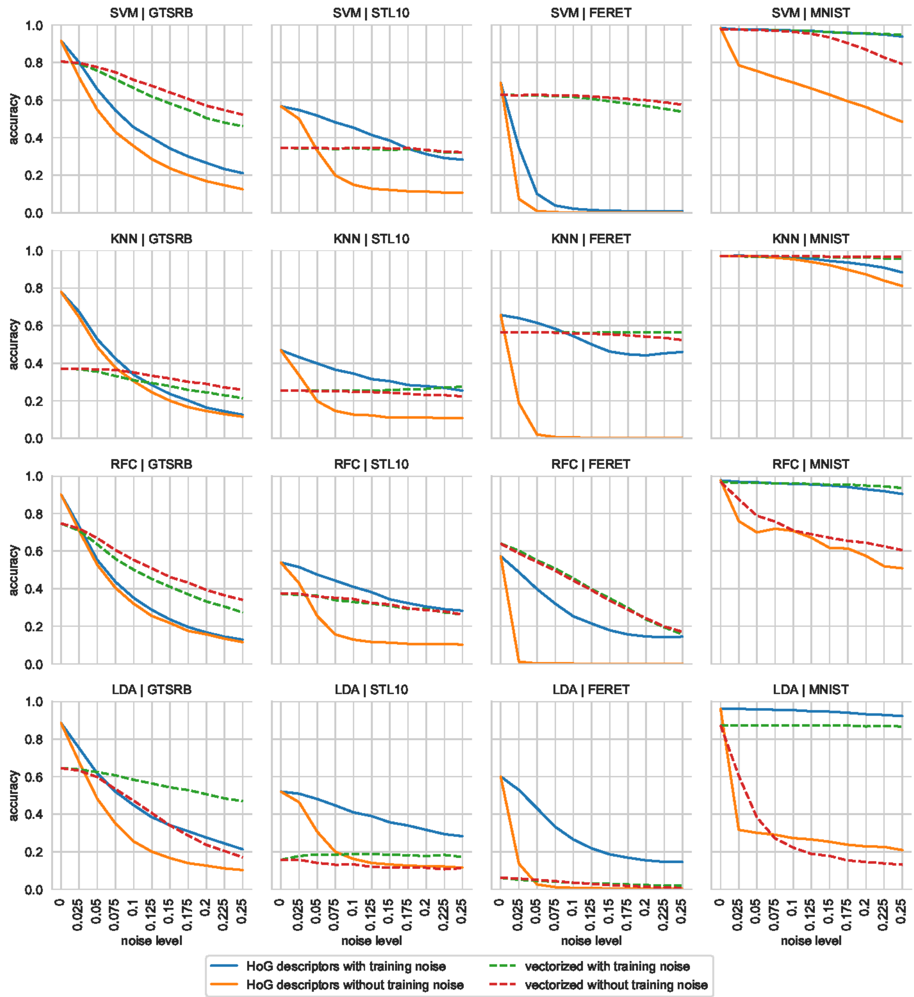

Figure 6: Classification results after applying Gaussian noise.

descriptors in almost all of the cases.

5.3 Applying denoising prior to classification

In the second part of the experiment our goal was to examine the effects of denoising algorithms on classification of previously distorted images. In that scenario, we trained the classifiers with clean, undistorted images and removed the distortions present in the test data prior to classification. This emulates the case in which we know what distortions to expect in evaluated data, but we are not able to obtain sufficient amounts of distorted data for training. We considered all of

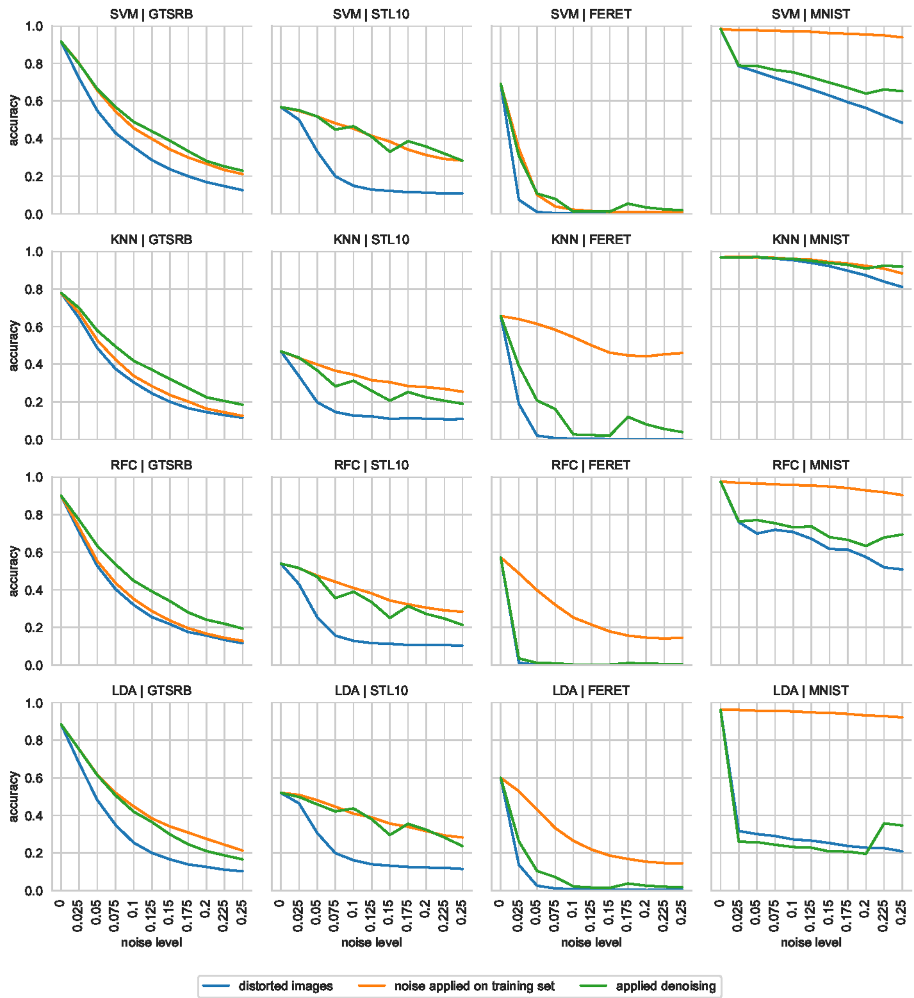

Figure 7: Classification after applying Gaussian noise, with denoising algorithms.

the previously used noise models, that is Gaussian, quantization and salt-and-pepper noise, as well as three denoising algorithms: median filtering, bilateral filtering and BM3D algorithm [Dabov et al., 2008]. We performed a grid-search on training part of each dataset to find the optimal denoising algorithm along with its parameters. For the median filtering we tested the following kernels 3×3, 5×5, 7×7, 9×9, 11×11 and 13×13, respectively. For the bilateral filtering we tested $\sigma_s \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ and $\sigma_r \in \{3, 5, 7\}$, respectively. Lastly, for the BM3D algorithm values of $\sigma \in \{0.05, 0.1, 0.2, 0.4, 0.5\}$ were used. Tested

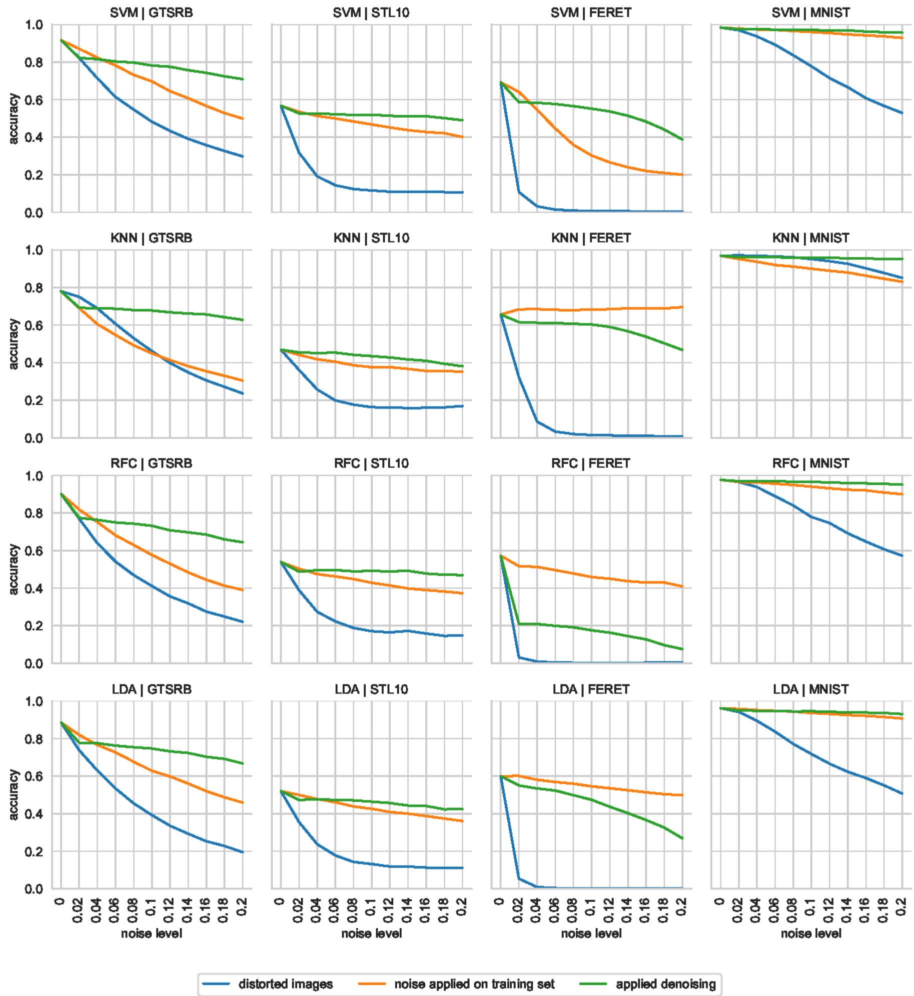

Figure 8: Classification after applying Salt-and-Pepper noise, with denoising algorithms.

types of noise were discussed in section 5.1.

Results for this stage are presented in Figures7 through 13. For the reference both the case in which no denoising algorithm was applied, as well as strategy in which distortions are applied on the training data are presented. In this case only classification with HOG descriptors is considered. In almost all cases applying denoising algorithms resulted in as-good-as or better performance than in the baseline scenario. Only 4 out of 48 considered cases turned out otherwise: classification on MNIST distorted with Gaussian noise using LDA, classification on MNIST and GTSRB distorted with salt-and-pepper noise using KNN, and classification on MNIST distorted with quantization noise using LDA. However, in none of these cases did the baseline case outperform both scenario using denoising algorithms, and the one in which noise was applied to the training data. For both Gaussian and quantization noise, applying training noise led to at least as-good-as or better classification accuracy than denoising for STL-10, FERET and MNIST datasets for all of the classification algorithms, while for GTSRB results varied depending on the classifier. Considering medium to high intensity salt-and-pepper distortions denoising improved performance for GTSRB, STL-10 and MNIST, while decreasing accuracies for FERET.

5.4 Handling mixtures of distortions

In the final stage of experimental study we considered the scenario of unknown distortion type and intensity. This was divided into two cases:

- noise type with unknown intensity and
- unknown noise type and intensity.

This setting is the closest to a real conditions, in which both the type and intensity of the distortions can change on a case-by-case basis. Once again, similar strategies were examined - the baseline case, applying the same, in this case random, distortions on the training data, as well as applying denoising when possible. Classification accuracy for undistorted data is presented for the reference.

The results of these experiments were presented in Tables 2 through 5. As can be seen, applying one of the strategies for dealing with distortions leads to an improved performance in almost all of the cases, with the exception of KNN classifier used to classify the occluded images from the GTSRB and FERET datasets, as well as the MNIST with applied noise of random type. Applying denoising on the training data also proved to be beneficial in 11 out of 12 considered examples. For other types of noise no clear trends regarding the choice of strategy of dealing with distortions were observed, with classification accuracy varying depending on the dataset and the used classifier.

6 Conclusions

In the paper the experimental evaluation of the impact of different types of distortions on image recognition with the HOG features is presented. Four different classification methods and on four reference databases have been used. For reference the classification with vectorized images with pure intensity has been used. Moreover two strategies of dealing with distortions were evaluated. First relies on applying similar distortions on the training data; In the second case, one of the filtering (denoising) method is applied to the nosiy data. Our main conclusions can be summaraized as follows:

– Distortions significantly affects classification performance when using HOG

Table 2: Results of classification with random distortion intensity using HOG features combined with the SVM classifier. Two strategies of dealing with distortions were tested: The same distortions on the training data (TD) and, when applicable, denoising (DN)

Distortion type	Dataset	Baseline	TD	DN
Gauss. noise	GTSRB	0.32	0.40	**0.41**
	STL-10	0.19	**0.38**	0.32
	MNIST	0.64	**0.96**	0.68
	FERET	0.01	0.02	**0.03**
S&P noise	GTSRB	0.50	0.65	**0.77**
	STL-10	0.14	0.45	**0.51**
	MNIST	0.75	0.95	**0.97**
	FERET	0.02	0.27	**0.53**
Quant. noise	GTSRB	0.45	**0.54**	**0.54**
	STL-10	0.25	**0.45**	0.37
	MNIST	0.70	**0.97**	0.73
	FERET	0.03	0.06	**0.16**
Random noise	GTSRB	0.43	0.51	**0.58**
	STL-10	0.19	**0.40**	0.39
	MNIST	0.70	**0.96**	0.70
	FERET	0.02	0.05	**0.23**
None	GTSRB	0.92		
	STL-10	0.57		
	MNIST	0.98		
	FERET	0.69		

features, as well as when classifying on pure intensities of the vectorized images. However, HOG descriptors are more susceptible to distortions than the pure intensity signals. As a result, despite better performance observed for HOG feature descriptors in case on undistorted images, classification with vectorized images can be a potentially useful alternative in case of highly distorted evaluation data. This can be explained by the well-known effect of noise when computing derivatives of the signal. Hence, consecutive computation of orientation histograms leads to poor results.

– The strategy of applying a similar distortion on the training data, as well as the strategy of using denoising algorithms, led to significant improvements in the classification accuracy. Particularly, in the case of the salt-and-pepper noise, denoising turned out to be the preferable approach. For other types of distortions no universal trends for all datasets were observed with results varying depending on data and classification algorithms.

– In the case of unknown distortions similar trends were observed. Most noticeably, both considered strategies of dealing with distortions led to improved performance compared to the baseline scenario.

Nevertheless it is important to notice that while both of the aforementioned strategies of dealing with distortions led to significant improvements in classification accuracy on distorted images, complete restoration of the baseline performance on undistorted data was rarely possible. A negative impact of distortions on image recognition tasks could be further reduced by developing better image restoration algorithms. This leaves us with the room for improvement especially considering classification with HOG features.

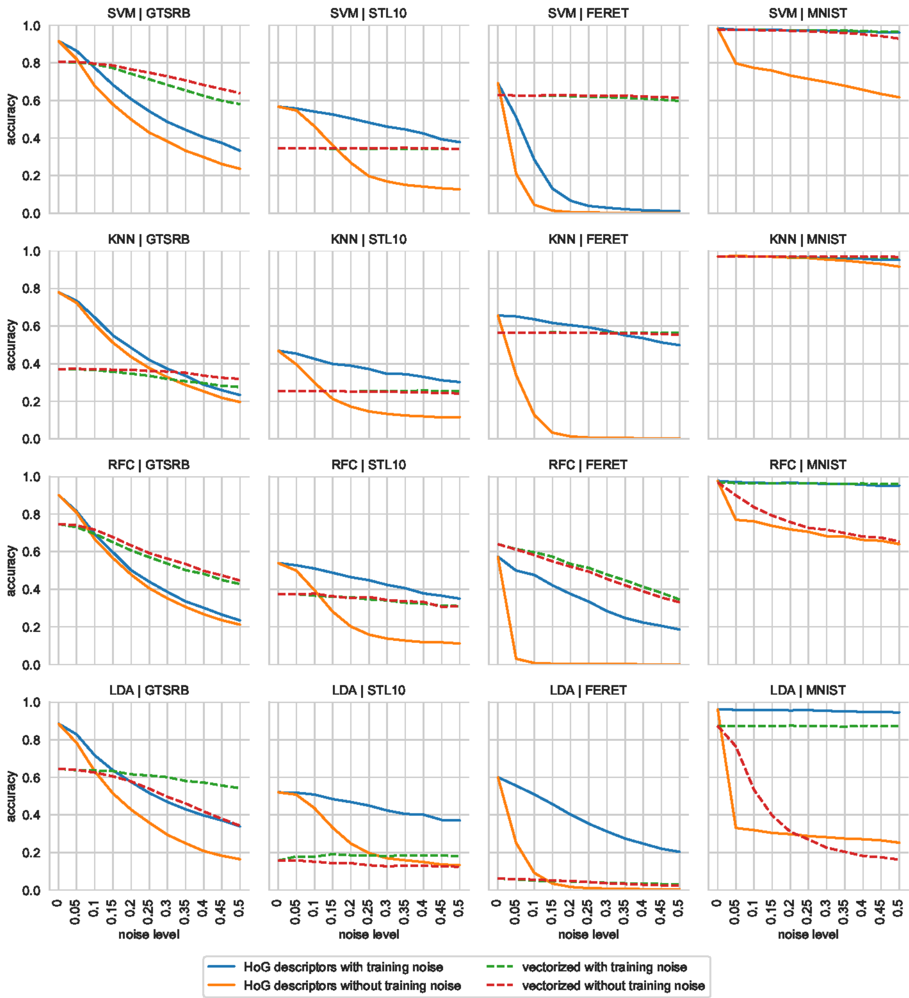

Figure 9: Results of classification after applying quantization noise.

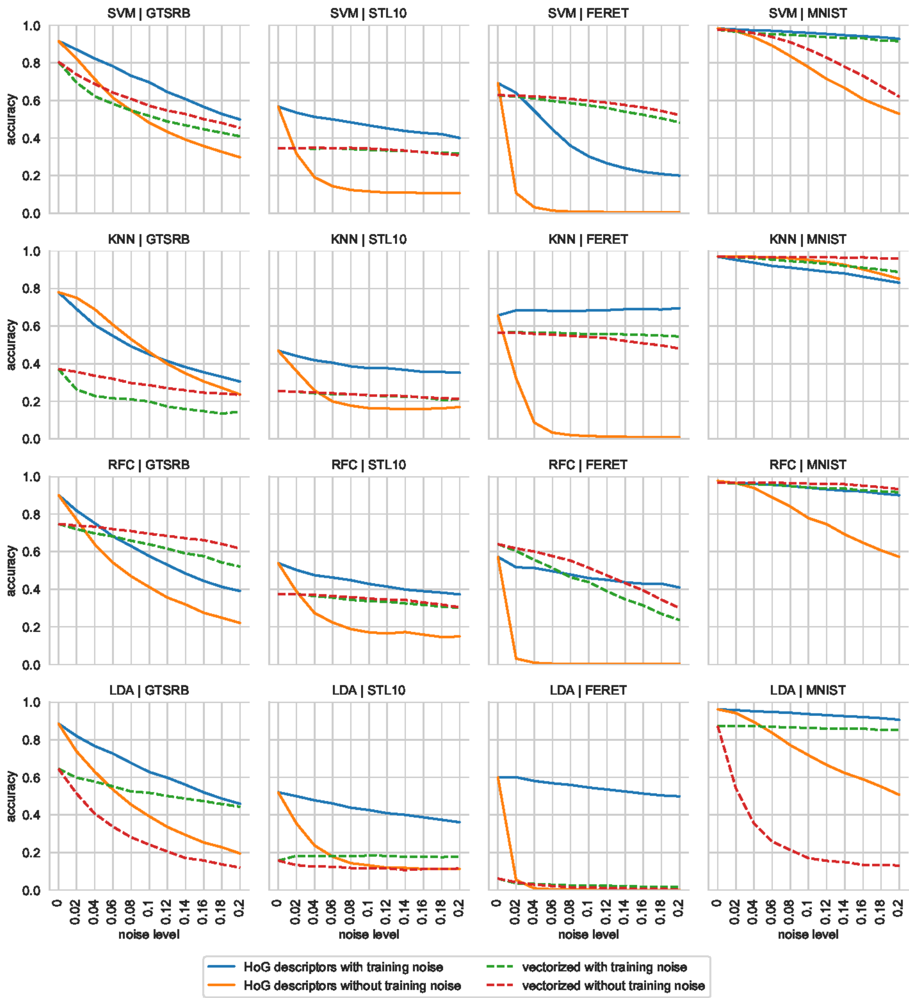

Figure 10: Classification results after applying Salt-and-Pepper noise.

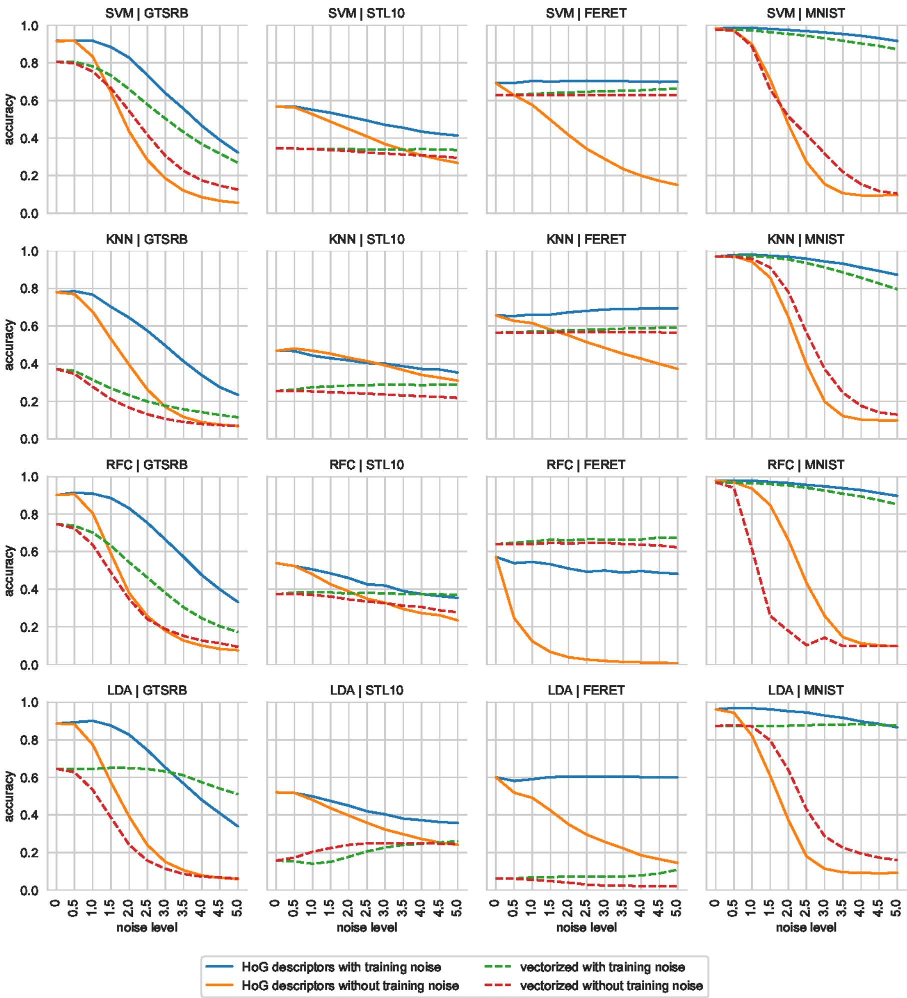

Figure 11: Classification results after applying Gaussian Blur.

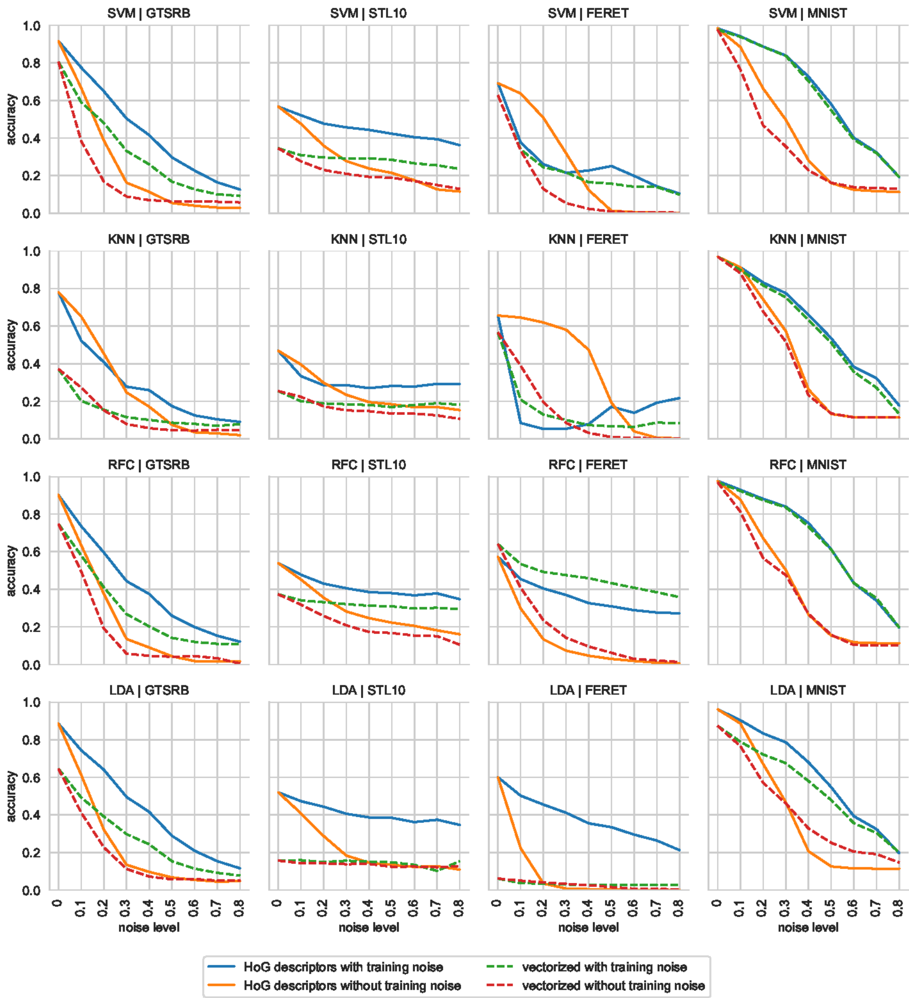

Figure 12: Classification results after applying occlusions.

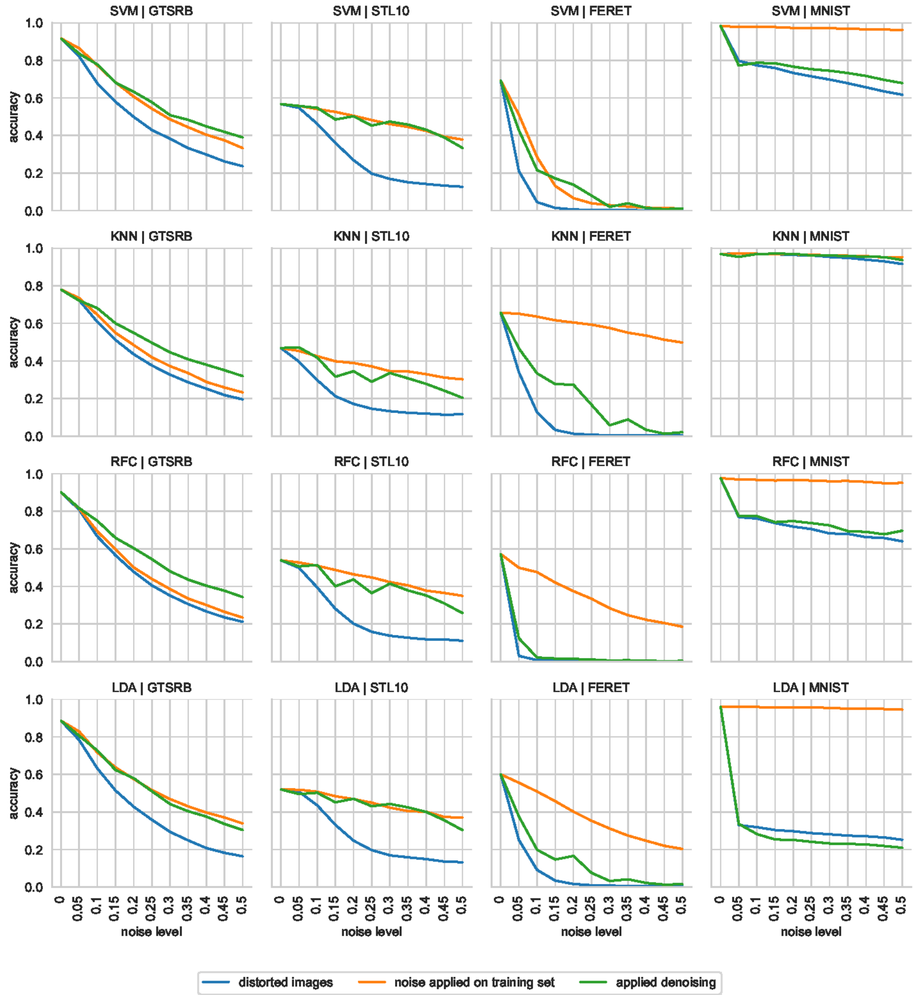

Figure 13: Classification after applying quantization noise, and with denoising algorithms.

Table 3: Classification results with random distortion intensity using HOG features and the KNN classifier. Two strategies of dealing with distortions were used: Applying the same distortions on training data (TD) and, when applicable, denoising (DN))

Distortion type	Dataset	Baseline	TD	DN
Gauss. noise	GTSRB	0.28	0.29	**0.34**
	STL-10	0.15	**0.31**	0.26
	MNIST	0.91	**0.94**	0.92
	FERET	0.03	**0.32**	0.15
S&P noise	GTSRB	0.47	0.41	**0.67**
	STL-10	0.19	0.36	**0.43**
	MNIST	0.93	0.88	**0.96**
	FERET	0.06	0.52	**0.58**
Quant. noise	GTSRB	0.40	0.43	**0.46**
	STL-10	0.18	**0.38**	0.27
	MNIST	0.95	**0.96**	0.95
	FERET	0.06	**0.41**	0.22
Random noise	GTSRB	0.38	0.37	**0.50**
	STL-10	0.18	**0.34**	0.29
	MNIST	**0.93**	0.91	0.92
	FERET	0.04	**0.26**	**0.26**
Occlusion	GTSRB	**0.21**	0.18	
	STL-10	0.23	**0.26**	
	MNIST	0.37	**0.54**	
	FERET	**0.32**	0.04	
Blur	GTSRB	0.32	**0.48**	
	STL-10	**0.40**	0.37	
	MNIST	0.45	**0.91**	
	FERET	0.50	**0.54**	
None	GTSRB	0.78		
	STL-10	0.47		
	MNIST	0.97		
	FERET	0.66		

Table 4: Results of HOG based classification with random distortion intensity and the LDA classifier. Two strategies of dealing with distortions were presented: The same distortions on training data (TD) and, when applicable, denoising (DN)

Distortion type	Dataset	Baseline	TD	DN
Gauss. noise	GTSRB	0.26	**0.40**	0.34
	STL-10	0.19	**0.33**	0.32
	MNIST	0.64	**0.96**	0.68
	FERET	0.01	0.02	**0.03**
S&P noise	GTSRB	0.41	0.61	**0.74**
	STL-10	0.17	0.40	**0.45**
	MNIST	0.71	0.93	**0.94**
	FERET	0.01	0.42	**0.44**
Quant. noise	GTSRB	0.38	**0.51**	0.47
	STL-10	0.25	**0.40**	0.34
	MNIST	0.29	**0.95**	0.24
	FERET	0.05	**0.23**	0.15
Random noise	GTSRB	0.35	0.48	**0.53**
	STL-10	0.20	0.35	**0.35**
	MNIST	0.42	**0.94**	0.27
	FERET	0.03	0.19	**0.20**
Occlusion	GTSRB	0.17	**0.34**	
	STL-10	0.19	**0.45**	
	MNIST	0.34	**0.55**	
	FERET	0.03	**0.28**	
Blur	GTSRB	0.33	**0.59**	
	STL-10	0.36	**0.40**	
	MNIST	0.34	**0.87**	
	FERET	0.31	**0.49**	
None	GTSRB	0.89		
	STL-10	0.52		
	MNIST	0.96		
	FERET	0.60		

Table 5: Classification results with random distortion intensity using HOG features and the RFC classifier. Two strategies of dealing with distortions were presented: Applying the same distortions on training data (TD) and, when applicable, denoising (DN)

Distortion type	Dataset	Baseline	TD	DN
Gauss. noise	GTSRB	0.30	0.31	**0.37**
	STL-10	0.16	**0.34**	0.28
	MNIST	0.66	**0.94**	0.68
	FERET	0.00	**0.12**	0.02
S&P noise	GTSRB	0.43	0.55	**0.72**
	STL-10	0.21	0.40	**0.49**
	MNIST	0.77	0.94	**0.96**
	FERET	0.01	**0.18**	0.03
Quant. noise	GTSRB	0.42	0.45	**0.50**
	STL-10	0.21	**0.40**	0.31
	MNIST	0.70	**0.96**	0.71
	FERET	0.01	**0.23**	0.15
Random noise	GTSRB	0.39	0.40	**0.54**
	STL-10	0.20	**0.36**	0.35
	MNIST	0.71	**0.93**	0.69
	FERET	0.01	**0.10**	0.06
Occlusion	GTSRB	0.17	**0.31**	
	STL-10	0.25	**0.36**	
	MNIST	0.35	**0.60**	
	FERET	0.07	**0.27**	
Blur	GTSRB	0.35	**0.61**	
	STL-10	0.35	**0.40**	
	MNIST	0.46	**0.91**	
	FERET	0.06	**0.37**	
None	GTSRB	0.90		
	STL-10	0.54		
	MNIST	0.98		
	FERET	0.57		

Acknowledgment

This work was supported by the National Science Centre, Poland, under the grant NCN no. 2016/21/B/ST6/01461.

References

[Bai et al., 2009] Bai, Y., Guo, L., Jin, L., and Huang, Q. (2009). A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 3305–3308. IEEE.

[Bay et al., 2008] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110:346–359.

[Bukala et al., 2019] Bukala, A., Koziarski, M., Cyganek, B., Koc, O., and Kara, A. (2019). The impact of distortions on the image recognition with histograms of oriented gradients. *Image Processing and Communications. IPC 2019. Advances in Intelligent Systems and Computing*, 1062:166–178.

[Coates et al., 2011] Coates, A., Ng, A., and Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223.

[Cyganek, 2007] Cyganek, B. (2007). Road signs recognition by the scale-space template matching in the log-polar domain. *Lecture Notes in Computer Science*, 4477:330–337.

[Cyganek, 2009] Cyganek, B. (2009). Recognition of solid objects in images invariant to conformal transformations. *Conference on Computer Recognition Systems CORES'09, Advances in Soft Computing*, 57:247–255.

[Cyganek and Gongola, 2018] Cyganek, B. and Gongola, K. (2018). Real-time marine snow noise removal from underwater video sequences. *Journal of Electronic Imaging*, 27(4):1–24.

[Cyganek and Siebert, 2009] Cyganek, B. and Siebert, P. (2009). *An Introduction to 3D Computer Vision Techniques and Algorithms*. Wiley, Hoboken, New Jersey.

[Dabov et al., 2008] Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. (2008). Image restoration by sparse 3D transform-domain collaborative filtering. In *Image Processing: Algorithms and Systems VI*, volume 6812, page 681207. International Society for Optics and Photonics.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE.

[Déniz et al., 2011] Déniz, O., Bueno, G., Salido, J., and De la Torre, F. (2011). Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12):1598–1603.

[Do and Kijak, 2012] Do, T.-T. and Kijak, E. (2012). Face recognition using co-occurrence histograms of oriented gradients. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 1301–1304. IEEE.

[Dodge and Karam, 2016] Dodge, S. and Karam, L. (2016). Understanding how image quality affects deep neural networks. In *Quality of Multimedia Experience (QoMEX), 2016 Eighth International Conference on*, pages 1–6. IEEE.

[Dutta et al., 2012] Dutta, A., Veldhuis, R. N., and Spreeuwers, L. J. (2012). The impact of image quality on the performance of face recognition. In *33rd WIC Symposium on Information Theory in the Benelux*. Centre for Telematics and Information Technology (CTIT).

[Ebrahimzadeh and Jampour, 2014] Ebrahimzadeh, R. and Jampour, M. (2014). Efficient handwritten digit recognition based on histogram of oriented gradients and SVM. *International Journal of Computer Applications*, 104(9).

[Freeman and Roth, 1995] Freeman, W. T. and Roth, M. (1995). Orientation histograms for hand gesture recognition. In *International Workshop on Automatic Face and Gesture Recognition*, volume 12, pages 296–301.

[Grabek and Cyganek, 2019] Grabek, J. and Cyganek, B. (2019). Speckle noise filtering in side-scan sonar images based on the tucker tensor decomposition. *Sensors*, 19(13)(2903):1–21.

[Karahan et al., 2016] Karahan, S., Yildirum, M. K., Kirtac, K., Rende, F. S., Butun, G., and Ekenel, H. K. (2016). How image degradations affect deep CNN-based face recognition? In *Biometrics Special Interest Group (BIOSIG), 2016 International Conference of the*, pages 1–5. IEEE.

[Karami et al., 2017] Karami, E., Prasad, S., and Shehata, M. (2017). Image matching using SIFT, SURF, BRIEF and ORB: Performance comparison for distorted images. *arXiv preprint arXiv:1710.02726.*

[Khan et al., 2011] Khan, N. Y., McCane, B., and Wyvill, G. (2011). SIFT and SURF performance evaluation against various image deformations on benchmark dataset. In *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on*, pages 501–506. IEEE.

[Koziarski and Cyganek, 2017] Koziarski, M. and Cyganek, B. (2017). Image recognition with deep neural networks in presence of noise–dealing with and taking advantage of distortions. *Integrated Computer-Aided Engineering*, 24(4):337–349.

[LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

[Phillips et al., 2000] Phillips, P. J., Moon, H., Rizvi, S. A., and Rauss, P. J. (2000). The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 22(10):1090–1104.

[Rublee, 2011] Rublee, Ethan; Rabaud, V. K. K. B. G. (2011). Orb: an efficient alternative to sift or surf. *IEEE International Conference on Computer Vision (ICCV).*

[Stallkamp et al., 2012] Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332.

[Tan et al., 2013] Tan, H., Yang, B., and Ma, Z. (2013). Face recognition based on the fusion of global and local HOG features of face images. *IET computer vision*, 8(3):224–234.

[Vasiljevic et al., 2016] Vasiljevic, I., Chakrabarti, A., and Shakhnarovich, G. (2016). Examining the impact of blur on recognition by convolutional networks. *arXiv preprint arXiv:1611.05760.*