Degradation Based Blind Image Quality Evaluation

Ville Ojansivu[1], Leena Lepistö[2], Martti Ilmoniemi[2], and Janne Heikkilä[1]

[1] Machine Vision Group, University of Oulu, Finland
firstname.lastname@ee.oulu.fi
http://www.cse.oulu.fi/MVG
[2] Nokia Corporation, Tampere, Finland
{leena.i.lepisto,martti.ilmoniemi}@nokia.com

Abstract. In this paper, we propose a novel framework for blind image quality evaluation. Unlike the common image quality measures evaluating compression or transmission artifacts this approach analyzes the image properties common to non-ideal image acquisition such as blur, under or over exposure, saturation, and lack of meaningful information. In contrast to methods used for adjusting imaging parameters such as focus and gain this approach does not require any reference image. The proposed method uses seven image degradation features that are extracted and fed to a classifier that decides whether the image has good or bad quality. Most of the features are based on simple image statistics, but we also propose a new feature that proved to be reliable in scene invariant detection of strong blur. For the overall two-class image quality grading, we achieved $\approx 90\%$ accuracy by using the selected features and the classifier. The method was designed to be computationally efficient in order to enable real-time performance in embedded devices.

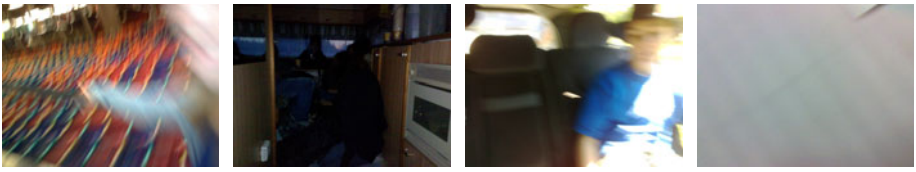
Keywords: image artifacts, blur, exposure, no-reference, quality measurement.

1 Introduction

In this paper, we propose a method for automatic image quality evaluation based on different types of image degradations such as blur, under or over exposure, saturation, or lack of meaningful information which are illustrated in Figure 1. The method does not need the original image as a reference, but the evaluation is done solely based on the features extracted from the degraded image. Our method is designed to be fast to compute so that it can be applied on-line. The method could be applied, for example, to assist photographer by prompting to capture new image, maybe with different camera parameters, if the obtained image quality is poor. Another application could be classifying gallery images based on quality and placing the poor quality images into a trash-folder.

Most of the current image and video quality evaluation methods are aimed for detecting quality reduction due to lossy compression or transmission errors [7]. Part of these methods use original non-degraded image as a reference and

A. Heyden and F. Kahl (Eds.): SCIA 2011, LNCS 6688, pp. 306–316, 2011.



(a) blur (b) under exp. (c) saturation (d) no-info.

Fig. 1. Examples of typical image degradations

perform the evaluation relative to this image. Others are no-reference or blind methods. Typical degradations are blur, noise, and block-based artifacts due to compression. The blind quality evaluation independent of image content is a much more difficult task for a computer although it may be simple for humans. The distinction between image details and impairments may be difficult [4]. For example, the measurements of blur and noise correlate typically heavily with the image content [7]. To our knowledge, there are not many methods for evaluating the overall image quality based on the degradations due to exposure and blurring. The different degradations are measured typically separately before image capture to adjust the imaging parameters. For example, under or over exposure and saturation might be measured for optimal exposure control. On the other hand, there are methods for measuring blur to achieve optimal focus. These methods typically compare the metric between multiple images from the same scene i.e. they make the evaluation relative to a reference. In [5], the authors have taken different approach to image quality evaluation by classifying images as professional vs. snapshots based also on the composition of the image.

We are interested in the overall perceived quality of the image due to multiple factors which include, in addition to blur, also saturation of pixels, incorrect exposure, and information content of the image. The last property means that we consider accidentally captured images representing, for example, floor as poor quality. So, instead of evaluating the technical quality traditionally, we are interested in the overall quality perceived by human. We perform the evaluation without any reference information and independent on the scene content.

Figure 2 presents our framework for image quality evaluation which consists of three steps: preprocessing, feature extraction, and classification. In the preprocessing step, the images are low-pass filtered and resized to the VGA size. This is followed by feature extraction. We used seven scalar features each reflecting the amount of single degradation present in the image. These features are described in more detail in Section 2 and summarized in Table 1. The features are fed to a binary classifier which has been trained based on subjective evaluations of the training images. Classification is described in more detail in Section 3. The features as well as the classifier are selected carefully so that they can be implemented efficiently on-line. For this purpose we tested various methods for feature extraction as well as for classification. We also developed a completely new feature for detecting strong blur.

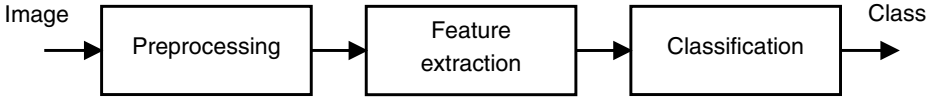


Fig. 2. Framework for blind image quality evaluation

2 Features for Detecting Degradations

We used seven separate features for detecting different impairments: blur, under or over exposure, saturation, and lack of information. These features are presented in the following with possible discussion of alternatives. Notice that the amount of noise can be typically predicted based on the exposure parameters and therefore it is not considered in this work.

Blur. In this work, the aim was to measure global blur caused by sudden camera motion or defocus of the lens system without any reference information. Different approaches for blur measurement are shortly reviewed next, before presenting the method we used.

A lot of earlier research exists on blur measurement in few different contexts. Traditionally, blur has been measured using metrics based on variance of image pixels, autocorrelation, image derivatives, estimation of edge widths, investigation of frequency spectrum, or histograms of pixels values or DCT coefficients. All these methods are based on the fact that blurring fades out image details and edges which corresponds to attenuation of the high frequency components of the image spectrum [4]. Image noise often disturbs these measures as it brings more variation to image which may be interpreted as sharp details.

Many of the existing blur measurement methods are targeted for autofocusing systems. In these systems, blur of the same image is measured with different focus settings. These methods can also work with motion blur. The only criterion for the measure is that it behaves monotonically when the amount of blur changes. If these blur measures are applied to images of different scenes, such as in Figure 4(a)-4(c), the results are not comparable as the amount of details in the image also affects to the measure. There are also methods which are targeted for quality evaluation of JPEG coded images. These methods measure the blurring caused by quantization or deblocking filter and are not suitable for our purpose [7].

Another group of blur metrics, which attempts to measure the amount of blur independent of the image content, is based on edge detection followed by estimation of the average edge width in the gradient direction or just horizontally [4]. These methods divide images into blocks and use only blocks containing edges. When the blur is strong [9] or images noisy [3] it may be however difficult to find edges reliably. A bigger obstacle is that these methods are suitable only for defocus blur. In motion blurred images, the sharpest edges are in opposite direction of the motion which makes the results incorrect. There is also a method which estimates partially blurred images with different scenes [6]. The method divides images into blocks and compares blur metrics between these blocks and

the whole image to detect blurred/sharp blocks corresponding to foreground objects. The method does not work for global blur.

Absolute blurriness between completely different scenes is very difficult to measure reliably because the image content affects sometimes even more to the metric than the blurring. From our tested methods, the most consistent blur measurements between different scenes gave a method proposed by Crete et al. [1]. The method is based on comparison of the x and y gradients of a blurred and re-blurred image. The method uses the assumption that re-blurring already blurred image does not change the image derivatives as much as blurring of a sharp image.

When the approach of Crete is used, there is another problem in the case of strong blur: noise added into the smooth image after blurring appears as false texture lowering measured blur level. This can be alleviated by suppressing noise using low pass filtering. In addition, we propose another method for detecting especially strong blur, which is presented next.

Strong Blur. The feature for strong blur detection is computed by average normalized difference d_α between observed image $g_{\mathbf{n}}$ and an artificially blurred image $b^\alpha_{\mathbf{n}}$, namely

$$d_\alpha = \sum_{\mathbf{n}} \frac{|g_{\mathbf{n}} - b^\alpha_{\mathbf{n}}|}{g_{\mathbf{n}} + \delta} \ , \tag{1}$$

where α is the motion blur angle used to blur the observed image, $\mathbf{n}$ denotes pixel location, and δ is a small real number. For an observed image $g_{\mathbf{n}}$, which already contains defocus blur d_α is small for all angles α of artificial blur, and for an image containing motion blur, d_α will be small for angle α corresponding to the motion blur direction in observed image $g_{\mathbf{n}}$. For this reason, minimum of results d_α is selected as the final blur feature d, namely

$$d = \min\{d_{\alpha_i}\} \ . \tag{2}$$

Blur to the observed image is generated by a 1×9 averaging filter which is rotated into angles $\alpha = \{0, 45, 90, 135\}$ degrees. The main difference between the proposed and Crete's [1] method is that we do not use image gradient for computing the feature.

Under or Over Exposure. Under and over exposure is measured using the mean of the image pixel values, which ranges from 0 to 255. It is assumed that value 128 corresponds to a well exposed image. Smaller values correspond to under exposure and larger values over exposure. We used separate features for under and over exposure. Using two separate features enables to weight them differently in the classification step to better correspond to the subjective evaluation of the image quality.

Saturation. The saturation features are based on amount of saturated pixels in saturated areas which are larger than 50 pixels. These are supposed to correspond disturbing highlighted areas in image. So, single saturated pixels are not counted. Saturation is detected separately in 1/3 top image and 2/3 bottom image. This

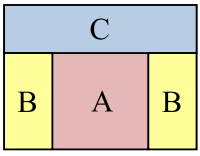


Fig. 3. Saturation is detected separately for areas A, B, and C

Table 1. Features used to measure artifacts for image quality evaluation

Feature	Range	Based on	Time sec.
1. blur	0…1	Difference of derivates of image and blurred image	0.18
2. strong blur	0…1	Difference of image and blurred image	0.10
3. under exposure	0…1	Mean value of image pixels	0.0010
4. over exposure	0…1	Mean value of image pixels	incl. to 3.
5. no-information	0…1	Entropy of image	0.011
6. top saturation	0…1	Saturated pixels in top 1/3 area of image	0.023
7. bottom saturation	0…1	Saturated pixels in bottom 2/3 area of image	0.024

Range: 0 = no artifact
1 = strong artifact

is due to the fact that most of the saturation in natural images appears in top area including sky, sun, lights etc. This top image saturation cannot be avoided in many situations, and on the other hand, top image saturation is perceived as more natural and not so disturbing. Top-saturation feature is computed from area C in Figure 3.

Bottom image saturation feature is computed from areas A and B. It is assumed that saturation of image is most disturbing in the central area A. For this reason the area A has double weight compared to area B in computation of the bottom-saturation. The algorithm assumes that the image orientation is known.

No-Information. Some images do not contain any meaningful information. These images may be captured accidentally, for example, toward the floor. The image entropy is used as a feature to measure the lack of information in the image.

All the features are normalized logarithmically into scale [0,1] so that feature value 0.5 corresponds approximately to the threshold between good and bad images in subjective quality. The features with their ranges, basis techniques, and approximate computation times are summarized in Table 1. Times are based on computation of the features for a VGA image using non-optimized Matlab implementations and 3 GHz Intel Core 2 Duo E8400 CPU with 4 GB RAM.

3 Classifier for Quality Evaluation

We compared different classifiers for quality evaluation of the images. What we need is a binary classifier which takes the seven features characterizing the degradations as input and gives the class good/bad quality as output. The quality

cannot be classified simply by using a concatenation the features, because single strong artifact destroys the image quality even if the other features indicate good quality. It seems that the dominant degradation in the image will indicate quite well the subjective image quality.

Based on the previous discussion we first tried to use classifier which bases the classification only on single dominant degradation. This means that the classifier selects the largest artifact feature value to represent image quality. This value is compared to a threshold. During training with subjectively labeled data the relative weights of the features are selected by increasing or decreasing them iteratively to best reflect the subjective evaluations. This method, referred hereafter as *MaxFeature* classifier, produced relatively good results as shown in the experiment section and the method is also very fast to compute.

We tested also AdaBoost and support vector machine (SVM) classifiers. Both of these methods are well known classifiers for a two-class classification problem. For SVM we used the radial basis function (RBF) kernel, which is in general a good choice when the relation between the classes and features is nonlinear.

4 Experimental Results

Test Images and Preprocessing. As test images, we used 508 5 Mpix images photographed using Nokia N95 mobile phone. These images contain degradations caused by real imaging situation including blur, noise, under or over exposure, saturation of pixel values caused by over exposure or bright sky, sun, lights etc., and also accidentally captured images with random content. Figures 1, 6, and 8 show examples of the test images.

This data set is challenging since the images are photographed in various situations resulting also in images containing no meaningful information ("no-information"). Many of the images contain multiple degradations at the same time. Most common artifact is blur due to motion or out of focus. All images contain also substantial amount of noise. Many images are saturated in part but at the same time in part under exposed. Saturation appears especially in the images depicting gray sky.

Before computing the artifact features we preprocessed the 5 Mpix image as follows. We first low-pass filtered the images to suppress noise, which is essential for the blur detection, although the image is at the same time blurred slightly. Filtering was done using a 5-by-5 uniform filter, which can be implemented very efficiently but still resulted in similar results as a Gaussian filter. Subsequently the images were resized to the VGA size (640×480) to reduce the computation in the feature extraction step.

Subjective Evaluation. For training and testing of the classifiers, the image quality of the 508 images was evaluated subjectively. This was done by inspecting the original images on a 19 inch screen. The images were given one of the grades $\{0, 1, 2, 3\}$, ranging from good to useless quality, with respect to the attainable quality range of a mobile phone camera. For the blur measurement experiment

the images were evaluated similarly but taking into account only the perceived blur artifact.

4.1 Features for Detecting Blur

In this section, we present results with blur measurement algorithms. The selection of blur features for overall image quality evaluation was based on these results. In the first experiment, we compared methods for detecting blur in images containing completely different types of scenes, shown in Figure 4(a)-4(c), because blur feature should be invariant to the scene content. As blurring appears as smoothing of the image, the blur detection algorithms often rate image containing more texture, such as Figure 4(a), as sharper than image containing smoother regions, such in Figure 4(c).

We compared our proposed method and the Crete's method [1] for which there is a Matlab implementation available online[1]. Additionally we compared the following methods. Method by Erasmus and Smith [2] is based on the variance of the image. This method is targeted for autofocusing and illustrates how this kind of methods are dependent on the image content. The method of Tsomko et al. [8] computes variances of horizontal derivative image blocks and uses the maximum variance as a measure of blurriness. The method by Zhu and Milanfar [10] is a more complicated method which attempts to measure noise and blur simultaneously. It is included only for demonstration because the Matlab implementation is available online[2].

Diagrams in Figures 4(d)-4(f) and 4(g)-4(i) illustrate the results of blur estimation with different methods in case of increasing the extent of artificial circular or horizontal motion blur, respectively. In both cases, it can be noticed that the Crete's method and the proposed method behave most consistently between the three different scenes. The least consistent results are obtained with Erasmus' method, which mainly reflects the amount of texture in the image instead of blur. Zhu's method is nearly as inconsistent between the scenes and in addition does not behave monotonically. Tsomko's method is better, but not among the best. Based on these results we selected the proposed and Crete's method for further experiments.

Next we applied the selected features for the 508 test images. For detecting two highest blur levels $\{2, 3\}$ out of possible levels $\{0, 1, 2, 3\}$ we obtain the receiver operating characteristics (ROC) curves illustrated in Figure 5(a). Curves show the true positive rate (TPR) of detecting blur as a function of false positive rate (FPR) when the thresholds for the different features are lowered. As can be seen, Crete's method gives slightly better results than the proposed method. However, a combination of the methods, which selects the larger of the single features, gives clearly the best results. (Area under curve (AUC): Crete 0.902, proposed 0.870, and combined 0.943.) In the other case, illustrated by ROC curves in Figure 5(b), we investigated detection of the strongest blur level $\{3\}$.

[1] www.mathworks.com/matlabcentral/fileexchange/24676-image-blur-metric
[2] http://users.soe.ucsc.edu/~xzhu/doc/metricq.html

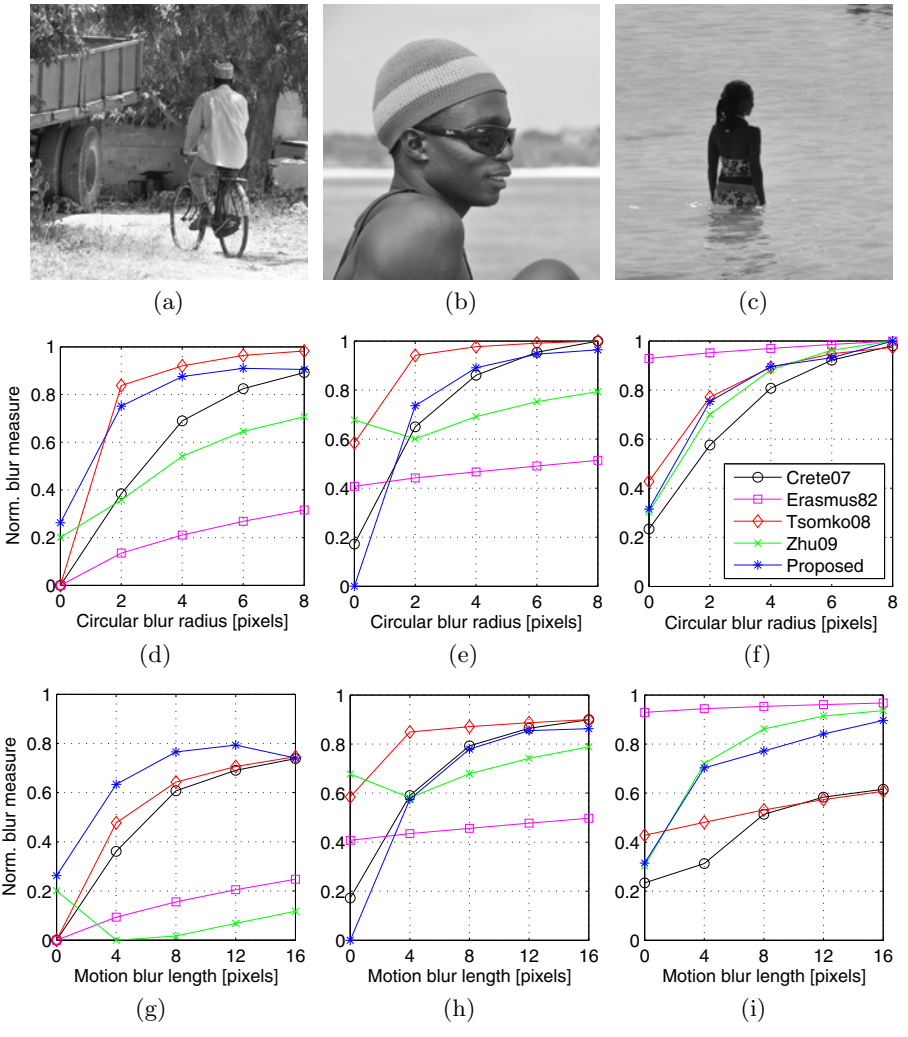


Fig. 4. Blur estimation results using different methods for different types of scenes (a-c) when blur level is increased: circular blur (d-f) and horizontal motion blur (g-i)

As can be seen, the proposed method can detect the strongest blur better than Crete's method while the combined method is superior in this sense. (AUC: Crete 0.826, proposed 0.911, and combined 0.974.) It seems that Crete's method, based on image gradients, is more sensitive to remaining noise in the image which appears as false texture in strongly blurred, smooth, images. Figure 6 illustrates some examples of images, containing strong blur, which can be detected by the combination of the proposed and Crete's feature but are missed with Crete's feature alone.

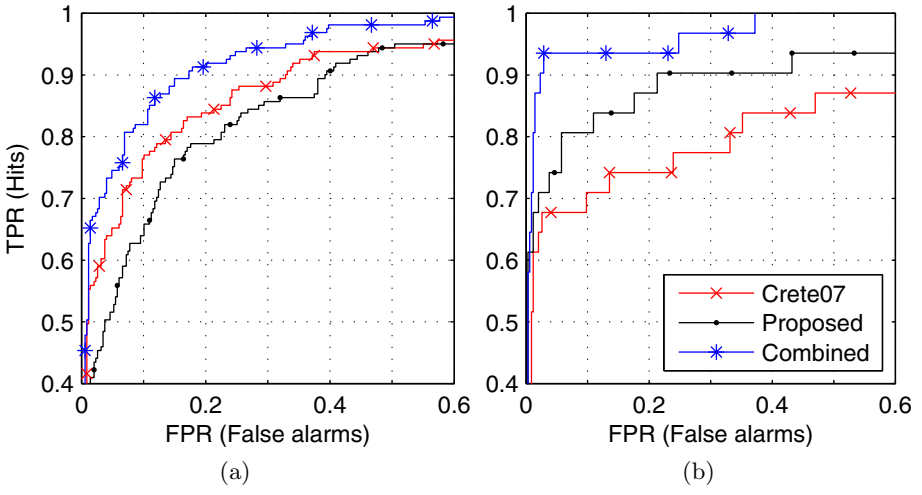


Fig. 5. ROC curves for detecting blurred images with different methods: detection of blur levels {2,3} (a) and only strongest level {3} (b)

Fig. 6. Examples of strongly blurred images which can be detected with the combination of the proposed and Crete's feature but not by Crete's feature alone. (Corresponds to operating point FPR=0.09 in Figure 5(a).)

4.2 Image Quality Classification

For the classification step, we tested three methods: Support Vector Machine (SVM) with the RBF kernel[3], Real AdaBoost[4] with single branch weak learners, and our own MaxFeature classifier using only single dominant artifact feature. The classifiers are trained/tested using leave-one-out cross validation. This means that one image at time is picked for testing and the classifier is trained using the remaining 507 images. This gives largest amount of training data without using the test sample for training.

Figure 7 shows the ROC curves for classification using different classifiers. True/false (T/F) correspond to bad/good quality images with labels {3,4}/{0,1}, respectively. The ROC curves show TPR and FPR when the threshold for the score of the classifier is lowered gradually. As can be seen in Figure 7, SVM gives the best result followed by quite similar MaxFeature and AdaBoost

[3] `www.csie.ntu.edu.tw/~cjlin/libsvm`
[4] `graphics.cs.msu.ru/ru/science/research/machinelearning/adaboosttoolbox`

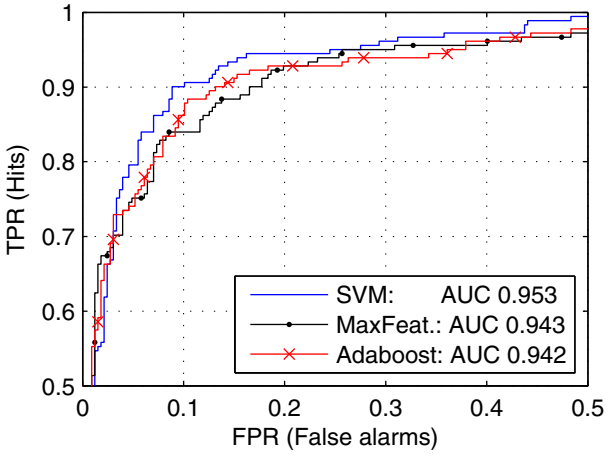


Fig. 7. ROC curves for detecting bad quality images with different classifiers

Fig. 8. Examples of classification results. Bad quality (top row) and good quality (bottom row).

classifiers. Although, the MaxFeature classifier is fastest to compute, we chose to use SVM classifier in operating point corresponding to the threshold 0: TPR 0.812 (147/181), FPR 0.055 (18/327), and total accuracy 89.76. Figure 8 shows some examples of classification results. It is noteworthy that in this operating point there was no {0} labels in FP samples and only four {3} labels among FN samples. So, none of the best quality images would be thrown away, which is important.

5 Conclusions

In this study, we proposed a method for blind image quality evaluation based on different types of image degradations. Evaluation was done using features extracted from the image which are subsequently fed to a SVM classifier. Also a completely new feature for detecting strong blur was proposed. According to the

experiments, the most reliable detection of the blur is achieved by using both an existing and the proposed blur measurement features. For the overall two-class image quality grading, we achieved $\approx 90\%$ accuracy by using the selected features and the classifier.

The proposed method is designed to be fast to compute so that it can be applied on-line and also with mobile devices. The applications could include, for example, assisting photographer by warning about low quality results or removing low quality gallery images.

References

1. Crete, F., Dolmiere, T., Ladret, P., Nicolas, M.: The blur effect: Perception and estimation with a new no-reference perceptual blur metric. In: Proc. SPIE, vol. 6492 (2007)
2. Erasmus, S.J., Smith, K.C.A.: An automatic focusing and astigmatism correction system for the sem and ctem. Journal of Microscopy 27, 185–199 (1982)
3. Ferzli, R., Karam, L.J.: No-reference objective wavelet based noise immune image sharpness metric. In: IEEE International Conference on Image Processing, pp. 405–408 (2005)
4. Ferzli, R., Karam, L.J.: A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB). IEEE Trans. Image Processing 18(4), 717–728 (2009)
5. Ke, Y., Tang, X., Jing, F.: The design of high-level features for photo quality assessment. In: IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, pp. 419–426 (June 2006)
6. Liu, R.T., Li, Z.R., Jia, J.Y.: Image partial blur detection and classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
7. Sheikh, H.R., Bovik, A.C., Cormack, L.: No-reference quality assessment using natural scene statistics: JPEG 2000. IEEE Trans. Image Processing 14(11), 1918–1927 (2005)
8. Tsomko, E., Kim, H.J., Paik, J., Yeo, I.K.: Efficient method of detecting blurry images. Journal of Ubiquitous Convergence Technology 2(1), 27–39 (2008)
9. Varadarajan, S., Karam, L.J.: An improved perception-based no-reference objective image sharpness metric using iterative edge refinement. In: IEEE International Conference on Image Processing, pp. 401–404 (2008)
10. Zhu, X., Milanfar, P.: A no-reference sharpness metric sensitive to blur and noise. In: International Workshop on Quality of Multimedia Experience, pp. 64–69 (2009)