applied sciences

MDPI

Article

Multi-Pooled Inception Features for No-Reference Image Quality Assessment

Domonkos Varga *

Department of Networked Systems and Services, Budapest University of Technology and Economics, Budapest H-1111, Hungary

* Correspondence: varga.domonkos7@upcmail.hu

Received: 5 February 2020; Accepted: 18 March 2020; Published: 23 March 2020

Abstract: Image quality assessment (IQA) is an important element of a broad spectrum of applications ranging from automatic video streaming to display technology. Furthermore, the measurement of image quality requires a balanced investigation of image content and features. Our proposed approach extracts visual features by attaching global average pooling (GAP) layers to multiple Inception modules of on an ImageNet database pretrained convolutional neural network (CNN). In contrast to previous methods, we do not take patches from the input image. Instead, the input image is treated as a whole and is run through a pretrained CNN body to extract resolution-independent, multi-level deep features. As a consequence, our method can be easily generalized to any input image size and pretrained CNNs. Thus, we present a detailed parameter study with respect to the CNN base architectures and the effectiveness of different deep features. We demonstrate that our best proposal—called MultiGAP-NRIQA—is able to outperform the state-of-the-art on three benchmark IQA databases. Furthermore, these results were also confirmed in a cross database test using the LIVE In the Wild Image Quality Challenge database.

Keywords: no-reference image quality assessment; deep learning; convolutional neural networks

1. Introduction

With the increasing popularity of imaging devices as well as the rapid spread of social media and multimedia sharing websites, digital images and videos have become an essential part of daily life, especially in everyday communication. Consequently, there is a growing need for effective systems that are able to monitor the quality of visual signals. Obviously, the most reliable way of assessing image quality is to perform subjective user studies, which involves the gathering of individual quality scores. However, the compilation and evaluation of a subjective user study are very slow and laborious processes. Furthermore, their application in a real-time system is impossible. In contrast, objective image quality assessment (IQA) involves the development of quantitative measures and algorithms for estimating image quality.

Objective IQA is classified based on the availability of the reference image. Full-reference image quality assessment (FR-IQA) methods have full access to the reference image, whereas no-reference image quality assessment (NR-IQA) algorithms possess only the distorted digital image. In contrast, reduced-reference image quality assessment (RR-IQA) methods have partial information about the reference image; for example, as a set of extracted features. Objective IQA algorithms are evaluated on benchmark databases containing the distorted images and their corresponding mean opinion scores (MOSs), which were collected during subjective user studies. The MOS is a real number, typically in the range 1.0–5.0, where 1.0 represents the lowest quality and 5.0 denotes the best quality. Furthermore, the MOS of an image is its arithmetic mean over all collected individual quality ratings. As already mentioned, publicly available IQA databases help researchers to devise and evaluate IQA algorithms and metrics. Existing IQA datasets can be grouped into two categories with respect to the introduced image distortion types. The first category contains images with artificial distortions, while the images of the second category are taken from sources with "natural" degradation without any additional artificial distortions.

The rest of this section is organized as follows. In Section 1.1, we review related work in NR-IQA with a special attention on deep learning based methods. Section 1.2 introduces the contributions made in this study.

1.1. Related Work

Many traditional NR-IQA algorithms rely on the so-called natural scene statistics (NSS) [1] model. These methods assume that natural images possess a particular regularity that is modified by visual distortion. Further, by quantifying the deviation from the natural statistics, perceptual image quality can be determined. NSS-based feature vectors usually rely on the wavelet transform [2], discrete cosine transform [3], curvelet transform [4], shearlet transform [5], or transforms to other spatial domains [6]. DIIVINE [2] (Distortion Identification-based Image Verity and INtegrity Evaluation) exploits NSS using wavelet transform and consists of two steps. Namely, a probabilistic distortion identification stage is followed by a distortion-specific quality assessment one. In contrast, He et al. [7] presented a sparse feature representation of NSS using also the wavelet transform. Saad et al. [3] built a feature vector from DCT coefficients. Subsequently, a Bayesian inference approach was applied for the prediction of perceptual quality scores. In [8], the authors presented a detailed review about the use of local binary pattern texture descriptors in NR-IQA.

Another line of work focuses on opinion-unaware algorithms that require neither training samples nor human subjective scores. Zhang et al. [9] introduced the integrated local natural image quality evaluator (IL-NIQE), which combines features of NSS with multivariate Gaussian models of image patches. This evaluator uses several quality-aware NSS features, i.e., the statistics of normalized luminance, mean subtracted and contrast-normalized products of pairs of adjacent coefficients, gradient, log-Gabor filter responses, and color (after the transformation into a logarithmic-scale opponent color space).

Kim et al. [10] introduced a no-reference image quality predictor called the blind image evaluator based on a convolutional neural network (BIECON), in which the training process is carried out in two steps. First, local metric score regression and then subjective score regression are conducted. During the local metric score regression, non-overlapping image patches are trained independently; FR-IQA methods such as SSIM or GMS are used for the target patches. Then, the CNN trained on image patches is refined by targeting the subjective image score of the complete image. Similarly, the training of a multi-task end-to-end optimized deep neural network [11] is carried out in two steps. Namely, this architecture contains two sub-networks: a distortion identification network and a quality prediction network. Furthermore, a biologically inspired generalized divisive normalization [12] is applied as the activation function in the network instead of rectified linear units (ReLUs). Similarly, Fan et al. [13] introduced a two-stage framework. First, a distortion type classifier identifies the distortion type then a fusion algorithm is applied to aggregate the results of expert networks and produce a perceptual quality score.

In recent years, many algorithms relying on deep learning have been proposed. Because of the small size of many existing image quality benchmark databases, most deep learning based methods employ CNNs as feature extractors or take patches from the training images to increase the database size. The CNN framework of Kang et al. [14] is trained on non-overlapping image patches extracted from the training images. Furthermore, these patches inherit the MOS of their source images. For preprocessing, local contrast normalization is employed. The applied CNN consists of conventional building blocks, such as convolutional, pooling, and fully connected layers. Bosse et al. [15] introduced a similar method. Namely, they developed a 12-layer CNN that is trained on 32 × 32 image patches. Furthermore, a weighted average patch aggregation method was introduced in which weights representing the relative importance of image patches in quality assessment are learned by a subnetwork. In contrast, Li et al. [16] combined a CNN trained on image patches with the Prewitt magnitudes of segmented images to predict perceptual quality.

Li et al. [17] trained a CNN on 32 × 32 image patches and employed it as a feature extractor. In this method, a feature vector of length 800 represents each image patch of an input image and the sum of image patches' feature vectors is associated with the original input image. Finally, a support vector regressor (SVR) is trained to evaluate the image quality using the feature vector representing the input image. In contrast, Bianco et al. [18] utilized a fine-tuned AlexNet [19] as a feature extractor on the target database. Specifically, image quality is predicted by averaging the quality ratings on multiple randomly sampled image patches. Further, the perceptual quality of each patch is predicted by an SVR trained on deep features extracted with the help of a fine-tuned AlexNet [19]. Similarly, Gao et al. [20] employed a pretrained CNN as a feature extractor, but they generate one feature vector for each CNN layer. Furthermore, a quality score is predicted for each feature vector using an SVR. Finally, the overall perceptual quality of the image is determined by averaging these quality scores. In contrast, Zhang et al. [21] trained first a CNN to identify image distortion types and levels. Furthermore, the authors took another CNN, which was trained on ImageNet, to deal with authentic distortions. To predict perceptual image quality, the features of the last convolutional layers were pooled bi-linearly and mapped onto perceptual quality scores with a fully-connected layer. He et al. [22] proposed a method containing two steps. In the first step, a sequence of image patches is created from the input image. Subsequently, features are extracted with the help of a CNN and long short-term memory (LSTM) is utilized to evaluate the level of image distortion. In the second stage, the model is trained to predict the patches' quality score. Finally, a saliency weighted procedure is applied to determine the whole image's quality from the patch-wise scores. Similarly, Ji et al. [23] utilized a CNN and LSTM for NR-IQA, but the deep features were extracted from the convolutional layers of a VGG16 [24] network. In contrast to other algorithms, Zhang et al. [25] proposed an opinion-unaware deep method. Namely, high-contrast image patches were selected using deep convolutional maps from pristine images which were used to train a multi-variate Gaussian model.

1.2. Contributions

Convolutional neural networks (CNNs) have demonstrated great success in a wide range of computer vision tasks [26–28], including NR-IQA [14–16,29]. Furthermore, pretrained CNNs can also provide a useful feature representation for a variety of tasks [30]. In contrast, employing pretrained CNNs is not straightforward. One major challenge is that CNNs require a fixed input size. To overcome this constraint, previous methods for NR-IQA [14–16,18] take patches from the input image. Furthermore, the evaluation of perceptual quality was based on these image patches or on the features extracted from them. In this paper, we make the following contributions. We introduce a unified and content-preserving architecture that relies on the Inception modules of pretrained CNNs, such as GoogLeNet [31] or Inception-V3 [32]. Specifically, this novel architecture applies visual features extracted from multiple Inception modules of pretrained CNNs and pooled by global average pooling (GAP) layers. In this manner, we obtain both intermediate-level and high-level representation from CNNs and each level of representation is considered to predict image quality. Due to this architecture, we do not take patches from the input image as in previous methods [14–16,18]. Unlike previous deep architectures [15,18,22], we do not utilize only the deep features of the last layer of a pretrained CNN. Instead, we carefully examine the effect of different features extracted from different layers on the prediction performance and we point out that the combination of deep features from mid- and high-level layers results in significant prediction performance increase. With experiments on three publicly available benchmark databases, we demonstrate that the proposed method is able to outperform other state-of-the-art methods. Specifically, we utilized KonIQ-10k [33], KADID-10k [34], and TID2013 [35] databases. KonIQ-10k [33] is the largest publicly available database containing 10,073 images with authentic distortions, while KADID-10k [34] consists of 81 reference images and 10,125 distorted ones (81 reference images × 25 types of distortions × 5 levels of distortions). TID2013 [35] is significantly smaller than KonIQ-10k [33] or KADID-10k [34] because it contains 25 reference images and 3000 distorted ones (25 reference images × 24 types of distortions × 5 levels of distortions). For a cross database test, LIVE In the Wild Image Quality Challenge Database [36] was applied, which contains 1162 images with authentic distortions evaluated by over 8100 unique human observers.

The remainder of this paper is organized as follows. Section 2 introduces our proposed approach. In Section 3, the experimental results and analysis are presented. A conclusion is drawn in Section 4.

2. Methodology

To extract visual features, GoogLeNet [31] or Inception-V3 [32] were applied as base models. GoogLeNet [31] is a 22-layer deep CNN and was the winner of ILSVRC 2014 with a Top 5 error rate of 6.7%. The depth and width of the network were increased but not simply following the general method of stacking the layers on each other. A new level of organization was introduced codenamed Inception module (see Figure 1). In GoogLeNet [31] ,not everything happens sequentially as in previous CNN models; pieces of the network work in parallel. We were inspired by a neuroscience model in [37] where for handling multiple scales a series of Gabor filters were used with a two layer deep model. However, contrary to the above-mentioned model, all layers are learned and not fixed. In GoogLeNet [31], architecture Inception layers are introduced and repeated many times. Subsequent improvements of GoogLeNet [31] have been called Inception-v*N* where *N* refers to the version number put out by Google. Inception-V2 [32] was refined by the introduction of batch normalization [38]. Inception-V3 [32] was improved by factorization ideas. Factorization into smaller convolutions means for example replacing a 5 × 5 convolution by a multi-layer network with fewer parameters but with the same input size and output depth.

We chose the features of Inception modules for the following reasons. The main motivation behind the construction of Inception modules is that salient parts of images may very extremely. This means that the region of interest can occupy very different image regions in terms of both size and location. That is why determining the convolutional kernel size in a CNN is very difficult. Namely, a larger kernel size is required for visual information that is distributed rather globally. On the other hand, a smaller kernel size is better for visual information that is distributed more locally. As already mentioned, the creators of Inception modules reacted to this challenge by the introduction of multiple filters with multiple sizes on the same level (Figure 1). Visual distortions have a similar nature. Namely, the distortion distribution is strongly influenced by image content [39].

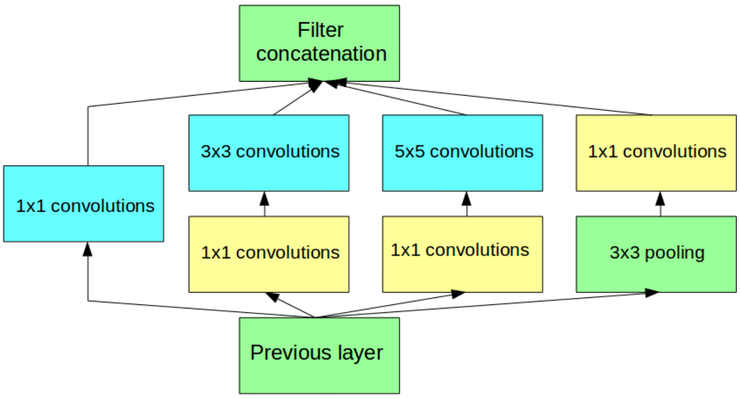


Figure 1. Illustration of Inception module. It was restricted to filter sizes 1 × 1, 3 × 3, and 5 × 5. Subsequently, the outputs were concatenated into a single vector that is the input for the next stage. Adding of an alternative parallel pooling path was found to be beneficial. Applying filters of 1 × 1 convolution makes it possible to reduce the volume before the expensive 3 × 3 and 5 × 5 convolutions [31].

2.1. Pipeline of the Proposed Method

The pipeline of the proposed framework is depicted in Figure 2. A given input image to be evaluated is run through a pretrained CNN body (GoogLeNet [31] and Inception-V3 [32] were considered in this study) which carries out all its defined operations. Specifically, global average pooling (GAP) layers are attached to the output of each Inception module. Similar to max- or min-pooling layers, GAP layers are applied in CNNs to reduce the spatial dimensions of convolutional layers. However, a GAP layer carries out a more extreme type of dimensional reduction than a max- or min-pooling layer. Namely, an $h \times w \times d$ block is reduced to $1 \times 1 \times d$. In other words, a GAP layer reduces a feature map to a single value by taking the average of this feature map. By adding GAP layers to each Inception module, we are able to extract resolution independent features at different levels of abstraction. Namely, the feature maps produced by neuroscience models inspired [37] Inception modules have been shown representative for object categories [31,32] and correlate well with human perceptual quality judgments [40]. The motivation behind the application of GAP layers was the following. By attaching GAP layers to the Inception modules, we gain an architecture which can be easily generalized to any input image resolution and base CNN architecture. Furthermore, this way the decomposition of the input image into smaller patches can be avoided, which means that parameter settings related to the database properties (patch size, number of patches, sampling strategy, etc.) can be ignored. Moreover, some kinds of image distortions are not uniformly distributed in the image. These kinds of distortions could be better captured in an aspect-ratio and content preserving architecture.

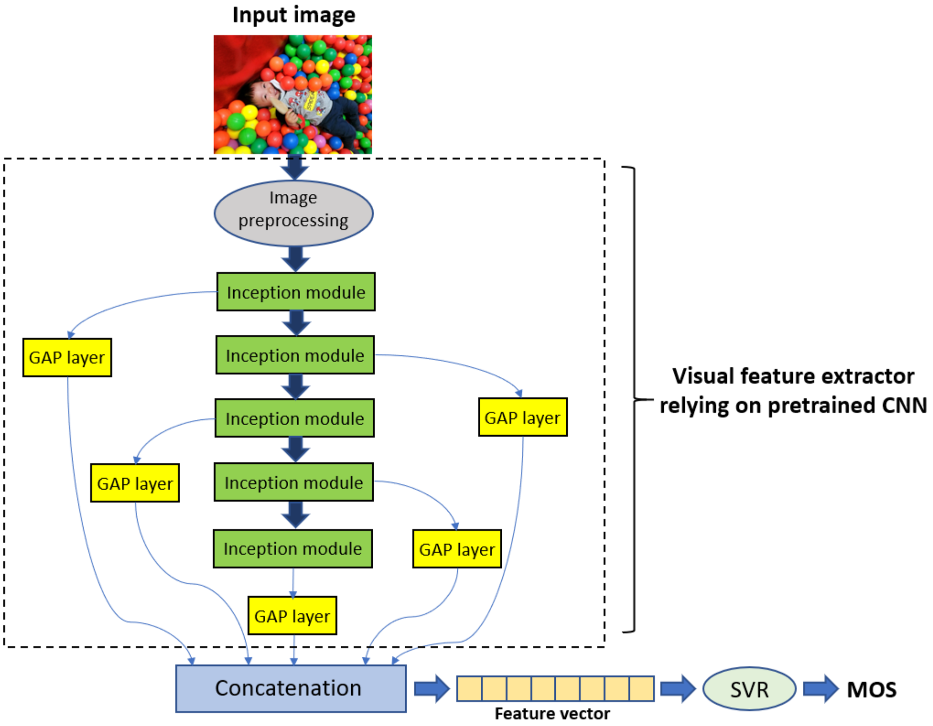


Figure 2. The pipeline of the proposed method. An input image is run through on an ImageNet [41] database pretrained CNN body (GoogLeNet [31] and Inception-V3 [32] were considered in this study) which carries out all its defined operations. Furthermore, global average pooling (GAP) layers are attached to each Inception module to extract resolution independent deep features at different abstraction levels. The feature vectors obtained from the Inception modules are concatenated and an SVR with radial basis function is applied to predict perceptual image quality.

As already mentioned, a feature vector is extracted over each Inception module using a GAP layer. Let $\mathbf{f}_k$ denote the feature vector extracted from the kth Inception module. The input image's feature vector is obtained by concatenating the respective feature vectors produced by the Inception modules. Formally, we can write $\mathbf{F} = \mathbf{f}_1 \oplus \mathbf{f}_2 \oplus ... \oplus \mathbf{f}_N$, where N denotes the number of Inception modules in the base CNN and $\oplus$ stands for the concatenation operator. In Section 3.3, we present a detailed analysis about the effectiveness of different Inception modules' deep features as a perceptual metric. Furthermore, we point out the prediction performance increase due to the concatenation of deep features extracted from different abstraction levels.

Subsequently, an SVR [42] with radial basis function (RBF) kernel is trained to learn the mapping between feature vectors and corresponding perceptual quality scores. Formally, it can be written:

$$y_i = \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b, \tag{1}$$

where $\mathbf{w}$ is the separating hyper-plane and b is the bias. They are learned from the training data. Furthermore, the function $\Phi(\cdot)$ maps the original feature space into a higher dimensional one and its inner product is a RBF kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\{-\frac{1}{2\sigma^2} \|\mathbf{x}_i - \mathbf{x}_j\|\}, \tag{2}$$

where σ is a hyper-parameter and corresponds to the scale of the RBF kernel; $\mathbf{x}_i$ and $\mathbf{x}_j$ are the feature vectors belonging to the ith and jth training images, respectively.

2.2. Database Compilation and Transfer Learning

Many image quality assessment databases are available online, such as TID2013 [35] or LIVE In the Wild [36], for research purposes. In this study, we selected the recently published KonIQ-10k [33] database to train and test our system, because it is the largest available database containing digital images with authentic distortions. Furthermore, we present a parameter study on KonIQ-10k [33] to find the best design choices. Our best proposal is compared against the state-of-the-art on KonIQ-10k [33] and also on other publicly available databases.

KonIQ-10k [33] consists of 10,073 digital images with the corresponding MOS values. To ensure the fairness of the experimental setup, we selected randomly 6073 images ($\sim$60% for training, 2000 images ($\sim$20%) for validation, and 2000 images ($\sim$20%) for testing purposes. First, the base CNN was fine-tuned on target database KonIQ-10k [33] using the above-mentioned training and the validation subsets. To this end, regularly the base CNN's last 1000-way softmax layer was removed and replaced by a five-way one in previous methods [18], because the training and validation subsets were reorganized into five classes with respect to the MOS values: Class A for excellent image quality $(5.0 > MOS \geq 4.2)$, Class B for good image quality $(4.2 > MOS \geq 3.4)$, Class C for fair image quality $(3.4 > MOS \geq 2.6)$, Class D for poor image quality $(2.6 > MOS \geq 1.8)$, and Class E for very poor image quality $(1.8 > MOS \geq 1.0)$. Subsequently, the base CNN was further trained to classify the images into quality categories. Since the MOS distribution in KonIQ-10k [33] is strongly imbalanced (see Figure 3), there would be very few images in the class for excellent images. That is why we took a regression-based approach instead of classification-based approach for fine-tuning. Namely, we removed the base CNN's last 1000-way softmax layer and we replaced it by a regression layer containing only one neuron. Since GoogLeNet [31] and Inception-V3 [32] accept images with input size of 224 × 224 and 299 × 299, respectively, *twenty* 224 × 224-sized or 299 × 299-sized patches were cropped randomly from each training and validation images. Furthermore, these patches inherit the perceptual quality score of their source images and the fine-tuning is carried out on these patches. Specifically, we trained the base CNN further for regression to predict the images patches MOS values which are inherited from their source images. During fine-tuning, Adam optimizer [43] was used; the initial learning rate was set to 0.0001 and divided by 10 when the validation error stopped improving. Further, the batch size was set to 28 and the momentum was 0.9 during fine-tuning.

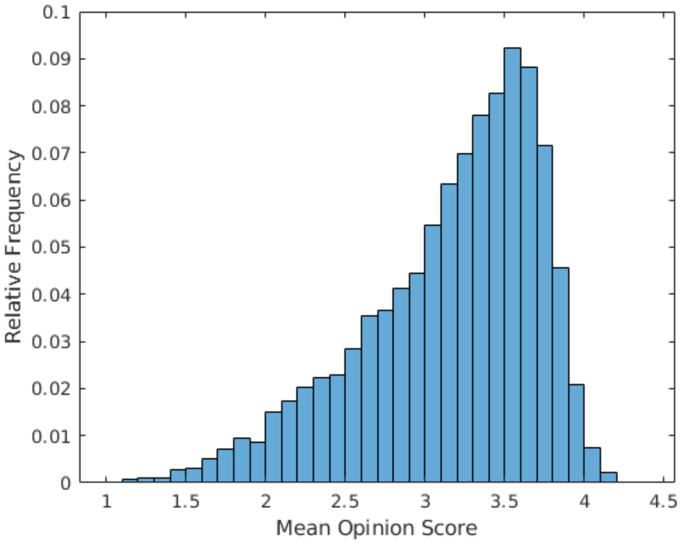


Figure 3. MOS distribution in KonIQ-10k [33] database. It contains 10,073 RGB images with authentic distortions and the corresponding MOS values ,which are on a scale from 1.0 (worst image quality) to 5.0 (best image quality).

3. Experimental Results and Analysis

In this section, we demonstrate our experimental results. First, we give the definition of the evaluation metrics in Section 3.1. Second, we describe the experimental setup and the implementation details in Section 3.2. In Section 3.3, we give a detailed parameter study to find the best design choices of the proposed method using KonIQ-10k [33] database. Subsequently, we carried out a comparison to other state-of-the-art methods using KonIQ-10k [33], KADID-10k [34], and TID2013 [35] publicly available IQA databases. Finally, we present a so-called cross database test using LIVE In the Wild Image Quality Challenge database [36].

3.1. Evaluation Metrics

The performance of NR-IQA algorithms are characterized by the correlation calculated between the ground-truth scores of a benchmark database and the predicted scores. To this end, Pearson's linear correlation coefficient (PLCC) and Spearman's rank order correlation coefficient (SROCC) are widely used in the literature [44]. PLCC between datasets A and B is defined as

$$PLCC(A,B) = \frac{\sum_{i=1}^{m}(A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum_{i=1}^{m}(A_i - \bar{A})^2}\sqrt{\sum_{i=1}^{m}(B_i - \bar{B})^2}}, \tag{3}$$

where $\bar{A}$ and $\bar{B}$ denote the average of sets A and B, and A_i and B_i denote the ith elements of sets A and B, respectively. SROCC, it can be expressed as

$$SROCC(A,B) = \frac{\sum_{i=1}^{m}(A_i - \hat{A})(B_i - \hat{B})}{\sqrt{\sum_{i=1}^{m}(A_i - \hat{A})^2}\sqrt{\sum_{i=1}^{m}(B_i - \hat{B})^2}}, \tag{4}$$

where $\hat{A}$ and $\hat{B}$ stand for the middle ranks.

3.2. Experimental Setup and Implementation Details

As already mentioned, a detailed parameter study was carried out on the recently published KonIQ-10k [33], which is the currently largest available IQA database with authentic distortions, to determine the optimal design choices. Subsequently, our best proposal was compared to the state-of-the-art using other publicly available databases as well.

The proposed method was implemented in MATLAB R2019a mainly relying on the functions of the Deep Learning Toolbox (formerly Neural Network Toolbox), Image Processing Toolbox, and Statistics and Machine Learning Toolbox. Thus, the parameter study was also carried out in MATLAB environment. More specifically, it was evaluated by 20 random train–validation–test splits of the applied database and we report on the average of the PLCC and SROCC values. As usual in machine learning, ∼60% of the images were used for training, ∼20% for validation, and ∼20% for testing purposes. Further, the models were trained and tested on a personal computer with 8-core i7-7700K CPU two NVidia Geforce GTX 1080 GPUs.

3.3. Parameter Study

First, we conducted experiments to determine which Inception module in GoogLeNet [31] or in Inception-V3 [32] is the most appropriate for visual feature extraction to predict perceptual image quality. Second, we answer the question whether the concatenation of different Inception modules' feature vectors improves the prediction's performance or not. Third, we demonstrate that fine-tuning of the base CNN architecture results in significant performance increase. In this parameter study, we used KonIQ-10k database to answer the above-mentioned questions and to find the most effective design choices. As presented in the next subsection, our best proposal was used to carry out a comparison to the state-of-the-art using other databases as well.

The results of the parameter study are summarized in Tables 1–4. Specifically, Tables 1 and 3 contain the results with GoogLeNet [31] and Inception-V3 [32] base architectures without fine-tuning, respectively. On the other hand, Tables 2 and 4 summarize the results when fine-tuning is applied. In these tables, we report the average, median, and standard deviation of the PLCC and SROCC values obtained after 20 random train–validation–test splits using KonIQ-10k database. Furthermore, we report the effectiveness of deep features extracted from different Inception modules. Moreover, the tables also contain the prediction performance of the concatenated deep feature vector. From these results, it can be concluded that the deep features extracted from the early Inception modules perform slightly poorer than those of intermediate and last Inception modules. Although most state-of-the-art methods [15,18,22] utilize the features of the last CNN layers, it is worth examining earlier layers as well, because the tables indicate that the middle layers encode the information that is the most powerful for perceptual quality prediction. We can also assert that feature vectors containing both mid-level and high-level deep representations are significantly more efficient than those of containing only one level's feature representation. Finally, it can be clearly seen that fine-tuning the base CNN architectures also improves the effectiveness of the extracted deep features. Overall, the deeper Inception-V3 [32] provides more effective features than GoogLeNet [31]. Our best proposal relies on Inception-V3 and concatenates the features of all Inception modules. In the following, we call this architecture *MultiGAP-NRIQA* and compare it to other state-of-the-art in the next subsection.

Another contribution of this parameter study may be the following. It is worth studying the features of different layers separately because the features of intermediate layers may provide a better representation of the given task than high-level features. Furthermore, the proposed feature extraction method may be also superior in other problems where the task is to predict one value only from the image data itself relying on a large enough database.

In our environment (MATLAB R2019a, PC with 8-core i7700K CPU and two NVidia Geforce GTX 1080), the computational times of the proposed MultiGAP-NRIQA method are the following. The loading of the base CNN and the 1024 × 768-sized or the 512 × 384 input image takes about 1.8s. Furthermore, the feature extraction from multiple Inception modules of Inception-V3 [32] and concatenation takes on average 1.355s or 0.976s on the GPU, respectively. Furthermore, the SVR regression takes 2.976s on average computing on the CPU.

Table 1. Performance comparison of deep features extracted from GoogLeNet's [31] Inception modules without fine-tuning measured on KonIQ-10k [33]. Average/median (±*std*) values are reported over 20 random train-test splits. The best results are in **bold**.

Layer	Dimension	PLCC	SROCC
inception_3a-output	256	0.845/0.845(±0.006)	0.842/0.841(±0.007)
inception_3b-output	480	0.861/0.861(±0.007)	0.856/0.858(±0.007)
inception_4a-output	512	0.876/0.876(±0.004)	0.872/0.872(±0.006)
inception_4b-output	512	0.874/0.874(±0.005)	0.865/0.864(±0.008)
inception_4c-output	512	0.875/0.877(±0.006)	0.865/0.865(±0.006)
inception_4d-output	528	0.876/0.875(±0.007)	0.864/0.864(±0.007)
inception_4e-output	832	0.872/0.871(±0.006)	0.861/0.862(±0.005)
inception_5a-output	832	0.873/0.874(±0.005)	0.859/0.860(±0.005)
inception_5b-output	1024	0.861/0.861(±0.008)	0.851/0.850(±0.008)
All concatenated	5488	**0.889/0.889**(±0.007)	**0.879/0.877**(±0.006)

Table 2. Performance comparison of deep features extracted from GoogLeNet's [31] Inception modules with fine-tuning measured on KonIQ-10k [33]. Average/median (±*std*) values are reported over 20 random train–validation–test splits. The best results are in **bold**.

Layer	Dimension	PLCC	SROCC
inception_3a-output	256	0.850/0.849(±0.007)	0.846/0.846(±0.007)
inception_3b-output	480	0.866/0.866(±0.006)	0.861/0.862(±0.007)
inception_4a-output	512	0.881/0.881(±0.005)	0.877/0.876(±0.006)
inception_4b-output	512	0.877/0.876(±0.005)	0.870/0.870(±0.006)
inception_4c-output	512	0.879/0.880(±0.005)	0.869/0.868(±0.005)
inception_4d-output	528	0.880/0.880(±0.006)	0.869/0.868(±0.005)
inception_4e-output	832	0.877/0.877(±0.005)	0.867/0.867(±0.007)
inception_5a-output	832	0.878/0.878(±0.007)	0.864/0.864(±0.007)
inception_5b-output	1024	0.865/0.865(±0.007)	0.856/0.856(±0.008)
All concatenated	5488	**0.894/0.894**(±0.006)	**0.884/0.884**(±0.006)

Table 3. Performance comparison of deep features extracted from Inception-V3's [32] Inception modules without fine-tuning measured on KonIQ-10k [33]. Average/median (±*std*) values are reported over 20 random train–test splits. The best results are in **bold**.

Layer	Dimension	PLCC	SROCC
mixed0	256	0.843/0.843(±0.006)	0.839/0.839(±0.006)
mixed1	288	0.848/0.848(±0.005)	0.844/0.844(±0.005)
mixed2	288	0.849/0.849(±0.006)	0.844/0.844(±0.007)
mixed3	768	0.861/0.860(±0.005)	0.858/0.855(±0.006)
mixed4	768	0.897/0.897(±0.005)	0.889/0.889(±0.006)
mixed5	768	0.906/0.906(±0.004)	0.898/0.898(±0.005)
mixed6	768	0.902/0.901(±0.004)	0.890/0.891(±0.006)
mixed7	768	0.884/0.884(±0.004)	0.870/0.870(±0.006)
mixed8	1280	0.892/0.891(±0.004)	0.879/0.879(±0.006)
mixed9	2048	0.871/0.871(±0.005)	0.859/0.859(±0.006)
mixed10	2048	0.842/0.844(±0.006)	0.828/0.829(±0.008)
All concatenated	10,048	**0.910/0.911**(±0.005)	**0.901/0.901**(±0.005)

Table 4. Performance comparison of deep features extracted from Inception-V3's [32] Inception modules with fine-tuning measured on KonIQ-10k [33]. Average/median ($\pm std$) values are reported over 20 random train–validation–test splits. The best results are in **bold**.

Layer	Dimension	PLCC	SROCC
mixed0	256	0.848/0.848(±0.008)	0.848/0.848(±0.007)
mixed1	288	0.853/0.853(±0.007)	0.853/0.853(±0.006)
mixed2	288	0.854/0.853(±0.007)	0.853/0.853(±0.006)
mixed3	768	0.866/0.865(±0.006)	0.867/0.867(±0.007)
mixed4	768	0.902/0.902(±0.007)	0.898/0.897(±0.006)
mixed5	768	0.911/0.910(±0.005)	0.908/0.908(±0.006)
mixed6	768	0.907/0.906(±0.005)	0.900/0.900(±0.006)
mixed7	768	0.889/0.889(±0.005)	0.880/0.880(±0.006)
mixed8	1280	0.897/0.897(±0.006)	0.888/0.887(±0.008)
mixed9	2048	0.876/0.876(±0.005)	0.869/0.870(±0.007)
mixed10	2048	0.847/0.847(±0.005)	0.837/0.836(±0.008)
All concatenated	10048	**0.915/0.914**(±0.005)	**0.911/0.911**(±0.005)

3.4. Comparison to the State-of-the-Art

To compare our proposed method to other state-of-the-art algorithms, we collected *seven* traditional learning-based NR-IQA metrics (BIQI [45], BLIINDS-II [3], BRISQUE [6], CORNIA [46], DIIVINE [2], HOSA [47], and SSEQ [4]), *four* deep learning based methods (BosICIP [15], CNN [14], DIQaM-NR [48], and WaDIQaM-NR [48]), and *one* opinion-unaware method (PIQE [49]) whose original source code are available. Moreover, we reimplemented the deep learning based DeepBIQ [18] method. Furthermore, we added the FR-IQA metric SSIM to our comparison. Overall, we compared our proposed method—*MultiGAP-NRIQA*—to 14 other state-of-the-art IQA algorithms or metrics. The results can be seen in Table 5.

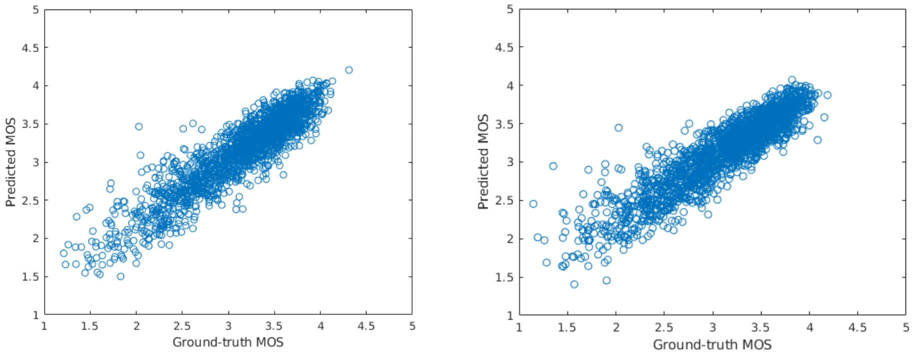
To ensure a fair comparison, these traditional and deep methods were trained, tested, and evaluated exactly the same as our proposed method. Specifically, ∼60% of the images were used for training, ∼20% for validation, and ∼20% for testing purposes. If a validation set was not required, the training set contained ∼80% of the images. To compare our method to the state-of-the-art, we report the average PLCC and SROCC values of 20 random train–validation–test splits of our method and those of other algorithms. As already mentioned, the results are summarized in Table 5. More specifically, this table illustrates the measured average PLCC and SROCC on the three largest available IQA databases (Table 6 summarizes the major parameters of the IQA databases used in this paper). As one can see from the results, our proposed approach outperformed other state-of-the-art methods on all three benchmark IQA databases. On the large KonIQ-10k [33] and KADID-10k [34] databases, the difference between our results and those of other methods is more apparent than on the smaller TID2013 [35] databases. More specifically, our architecture is able to outperform the state-of-the-art on KonIQ-10k [33] and KADID-10k [34] even without fine-tuning. Furthermore, the difference between our method and other state-of-the-art algorithms is somewhat larger on KADID-10k than on KonIQ-10k because the applied fine-tuning process is more effective on artificial distortions than on authentic ones. We attribute this improvement to the fact that our method extracts resolution independent features from images at different levels of abstraction in contrast to previous deep approaches. Furthermore, test images are not broken into patches in our approach during the prediction phase in contrast to other deep methods, such as DeepBIQ [18], DIQaM-NR [48], or WaDIQaM-NR [48]. Although the advantage of the proposed feature extraction method is less significant on TID2013 [35], our method still preserves its first place. Furthermore, the performance of the proposed method without fine-tuning achieves the state-of-the-art on TID2013 [35] as well. Specifically, our method without fine-tuning outperforms two deep methods (DIQaM-NR [48] and WaDIQaM-NR [48]) and seven traditional methods which generally perform significantly better on the smaller TID2013 than on KonIQ-10k or KADID-10k. The scatter plots showing the predicted MOS values against the ground-truth MOS values on KonIQ-10k and KADID-10k test sets are depicted in Figures 4 and 5, respectively.

Table 5. Comparison of *MultiGAP-NRIQA* with state-of-the-art NR-IQA and FR-IQA algorithms trained and tested on KonIQ-10k [33], KADID-10k [34], and TID2013 [35] databases. The average PLCC and SROCC values are reported measured over 20 random train–validation–test split. The best results are shown in **bold** and the second best results in *italic*.

	KonIQ-10k [33]		**KADID-10k [34]**		**TID2013 [35]**	
Method	**PLCC**	**SROCC**	**PLCC**	**SROCC**	**PLCC**	**SROCC**
SSIM [50]	-	-	0.645	0.718	0.578	0.616
BIQI [45]	0.619	0.550	0.453	0.437	0.840	0.816
BLIINDS-II [3]	0.602	0.590	0.565	0.520	0.909	0.884
BRISQUE [6]	0.711	0.696	0.554	0.527	0.913	0.897
CORNIA [46]	0.814	0.776	0.569	0.534	0.866	0.831
DIIVINE [2]	0.611	0.591	0.540	0.492	0.896	0.885
HOSA [47]	0.814	0.792	0.651	0.613	*0.949*	0.946
SSEQ [4]	0.605	0.600	0.454	0.424	**0.950**	*0.947*
PIQE [49]	0.206	0.245	0.289	0.289	0.462	0.364
BosICIP [15]	0.604	0.609	0.628	0.630	0.929	0.926
CNN [14]	0.589	0.574	0.619	0.603	0.934	0.934
DeepBIQ [18]	0.881	0.865	0.912	0.896	*0.949*	**0.951**
DIQaM-NR [48]	0.553	0.557	0.882	0.890	0.850	0.839
WaDIQaM-NR [48]	0.587	0.591	0.888	0.895	0.786	0.758
MultiGAP-NRIQA (without fine-tuning)	*0.910*	*0.901*	*0.939*	*0.944*	0.910	0.926
MultiGAP-NRIQA	**0.915**	**0.911**	**0.966**	**0.965**	**0.950**	**0.951**

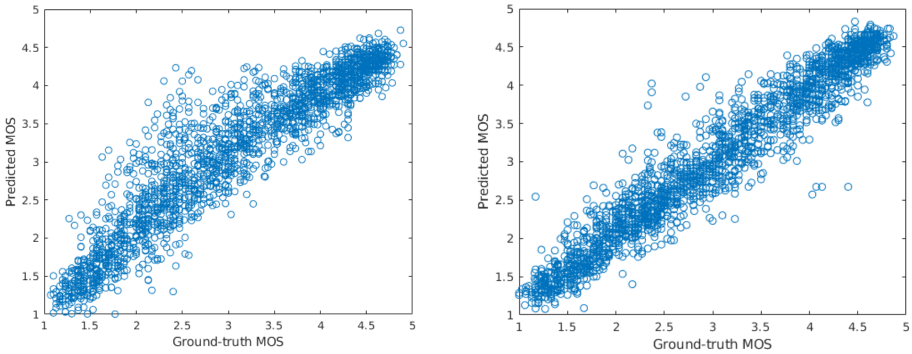
Table 6. Publicly available IQA databases used in this study. Publicly available IQA databases can be divided into two groups. The first one contains a smaller set of reference images and artificially distorted images are derived from them using different noise types at different intensity levels, while the second one contains images with "natural" degradation without any additional artificial distortions.

Database	**Year**	**Reference Images**	**Test Images**	**Distortion Type**	**Resolution**	**Subjective Score**
TID2013 [35]	2013	25	3,000	artificial	512 × 384	DMOS (0-9)
LIVE In the Wild [36]	2015	-	1,162	authentic	500 × 500	MOS (1-5)
KonIQ-10k [33]	2018	-	10,073	authentic	1024 × 768	MOS (1-5)
KADID-10k [34]	2019	81	10,125	artificial	512 × 384	DMOS (1-5)



(a) Without fine-tuning. **(b)** With fine-tuning.

Figure 4. Scatter plots showing the ground-truth MOS values against the predicted MOS values on KonIQ-10k [33] test set.



(a) Without fine-tuning. **(b)** With fine-tuning.

Figure 5. Scatter plots showing the ground-truth MOS values against the predicted MOS values on KADID-10k [34] test set.

3.5. Cross Database Test

To prove the generalization capability of our proposed MultiGAP-NRIQA method, we carried out a so-called cross database test. This means that our model was trained on the whole KonIQ-10k [33] database and tested on LIVE In the Wild Image Quality Challenge Database [36]. Moreover, the other learning-based NR-IQA methods were also tested this way. The results are summarized in Table 7. From the results, it can be clearly seen that all learning-based methods performed significantly more poorly in the cross database test than in the previous tests. It should be emphasized that our MultiGAP-NRIQA method generalized better than the state-of-the-art traditional or deep learning based algorithms even without fine-tuning. The performance drop occurs owing to the fact that images are treated slightly differently in each publicly available IQA database. For example, in LIVE In The Wild [36] database, the images were rescaled. In contrast, the images of KonIQ-10k [33] were cropped from their original counterparts.

Table 7. Cross database test. The learning-based NR-IQA methods were trained on the whole KonIQ-10k [33] database and tested on LIVE In the Wild [36] database. The measured PLCC and SROCC values are reported. The best results are shown in **bold** and the second best results are in *italic*.

	LIVE In The Wild [36]	
Method	**PLCC**	**SROCC**
BIQI [45]	0.455	0.374
BLIINDS-II [3]	0.110	0.088
BRISQUE [6]	0.585	0.577
CORNIA [46]	0.660	0.638
DIIVINE [2]	0.482	0.435
HOSA [47]	0.652	0.639
SSEQ [4]	0.282	0.240
BosICIP [15]	0.483	0.491
CNN [14]	0.462	0.477
DeepBIQ [18]	0.738	0.738
DIQaM-NR [48]	0.308	0.316
WaDIQaM-NR [48]	0.339	0.372
MultiGAP-NRIQA (without fine-tuning)	*0.760*	*0.764*
MultiGAP-NRIQA	**0.772**	**0.769**

4. Conclusions

In this paper, we introduce a deep framework for NR-IQA, which constructs a feature space relying on multi-level Inception features extracted from pretrained CNNs via GAP layers. Unlike previous deep methods, the proposed approach does not take patches from the input image, but instead treats the image as a whole and extracts image resolution independent features. As a result, the proposed approach can be easily generalized to any input image size and CNN base architecture. Unlike previous deep methods, we extract multi-level features from the CNN to incorporate both mid-level and high-level deep representations into the feature vector. Furthermore, we pointed out in a detailed parameter study that mid-level features provide significantly more effective descriptors for NR-IQA. Another important observation was that the feature vector containing both mid-level and high-level representations outperforms all feature vectors containing the representation of one level. We also carried out a comparison with other state-of-the-art methods and our approach outperformed the state-of-the-art on the largest available benchmark IQA databases. Moreover, the results were also confirmed in a cross database test. There are many directions for future research. Specifically, we would like to improve the fine-tuning process in order to transfer quality-aware features more effectively into the base CNN. Another direction of future research could be the generalization of the applied feature extraction method to other CNN architectures, such as residual networks.

Funding: This research received no external funding.

Acknowledgments: The authors thanks the anonymous reviewers for their careful reading of the manuscript and their many insightful comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Reinagel, P.; Zador, A.M. Natural scene statistics at the centre of gaze. *Netw. Comput. Neural Syst.* **1999**, *10*, 341–350. [CrossRef]
2. Moorthy, A.K.; Bovik, A.C. Blind image quality assessment: From natural scene statistics to perceptual quality. *IEEE Trans. Image Process.* **2011**, *20*, 3350–3364. [CrossRef] [PubMed]
3. Saad, M.A.; Bovik, A.C.; Charrier, C. DCT statistics model-based blind image quality assessment. In Proceedings of the 2011 18th IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 3093–3096.
4. Liu, L.; Liu, B.; Huang, H.; Bovik, A.C. No-reference image quality assessment based on spatial and spectral entropies. *Signal Process. Image Commun.* **2014**, *29*, 856–863. [CrossRef]
5. Li, Y.; Po, L.M.; Xu, X.; Feng, L. No-reference image quality assessment using statistical characterization in the shearlet domain. *Signal Process. Image Commun.* **2014**, *29*, 748–759. [CrossRef]
6. Mittal, A.; Moorthy, A.K.; Bovik, A.C. No-reference image quality assessment in the spatial domain. *IEEE Trans. Image Process.* **2012**, *21*, 4695–4708. [CrossRef]
7. He, L.; Tao, D.; Li, X.; Gao, X. Sparse representation for blind image quality assessment. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1146–1153.
8. Garcia Freitas, P.; Da Eira, L.P.; Santos, S.S.; Farias, M.C.Q.D. On the Application LBP Texture Descriptors and Its Variants for No-Reference Image Quality Assessment. *J. Imaging* **2018**, *4*, 114. [CrossRef]
9. Zhang, L.; Zhang, L.; Bovik, A.C. A feature-enriched completely blind image quality evaluator. *IEEE Trans. Image Process.* **2015**, *24*, 2579–2591. [CrossRef]
10. Kim, J.; Lee, S. Fully deep blind image quality predictor. *IEEE J. Sel. Top. Signal Process.* **2016**, *11*, 206–220. [CrossRef]
11. Ma, K.; Liu, W.; Zhang, K.; Duanmu, Z.; Wang, Z.; Zuo, W. End-to-end blind image quality assessment using deep neural networks. *IEEE Trans. Image Process.* **2017**, *27*, 1202–1213. [CrossRef]
12. Li, Q.; Wang, Z. Reduced-reference image quality assessment using divisive normalization-based image representation. *IEEE J. Sel. Top. Signal Process.* **2009**, *3*, 202–211. [CrossRef]
13. Fan, C.; Zhang, Y.; Feng, L.; Jiang, Q. No reference image quality assessment based on multi-expert convolutional neural networks. *IEEE Access* **2018**, *6*, 8934–8943. [CrossRef]
14. Kang, L.; Ye, P.; Li, Y.; Doermann, D. Convolutional neural networks for no-reference image quality assessment. In Proceedings of the IEEE Conference on Computer Vision and pattern Recognition, Columbus, OH, USA, 23–27 June 2014; pp. 1733–1740.

15. Bosse, S.; Maniry, D.; Wiegand, T.; Samek, W. A deep neural network for image quality assessment. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, Arizona, 25–28 September 2016; pp. 3773–3777.
16. Li, J.; Zou, L.; Yan, J.; Deng, D.; Qu, T.; Xie, G. No-reference image quality assessment using Prewitt magnitude based on convolutional neural networks. *Signal Image Video Process.* **2016**, *10*, 609–616. [CrossRef]
17. Li, J.; Yan, J.; Deng, D.; Shi, W.; Deng, S. No-reference image quality assessment based on hybrid model. *Signal Image Video Process.* **2017**, *11*, 985–992. [CrossRef]
18. Bianco, S.; Celona, L.; Napoletano, P.; Schettini, R. On the use of deep learning for blind image quality assessment. *Signal Image Video Process.* **2018**, *12*, 355–362. [CrossRef]
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2012; pp. 1097–1105.
20. Gao, F.; Yu, J.; Zhu, S.; Huang, Q.; Tian, Q. Blind image quality prediction by exploiting multi-level deep representations. *Pattern Recognit.* **2018**, *81*, 432–442. [CrossRef]
21. Zhang, W.; Ma, K.; Yan, J.; Deng, D.; Wang, Z. Blind image quality assessment using a deep bilinear convolutional neural network. *IEEE Trans. Circuits Syst. Video Technol.* **2018**. [CrossRef]
22. He, L.; Zhong, Y.; Lu, W.; Gao, X. A Visual Residual Perception Optimized Network for Blind Image Quality Assessment. *IEEE Access* **2019**, *7*, 176087–176098. [CrossRef]
23. Ji, W.; Wu, J.; Shi, G.; Wan, W.; Xie, X. Blind image quality assessment with semantic information. *J. Vis. Commun. Image Represent.* **2019**, *58*, 195–204. [CrossRef]
24. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
25. Zhang, Z.; Wang, H.; Liu, S.; Durrani, T.S. Deep activation pooling for blind image quality assessment. *Appl. Sci.* **2018**, *8*, 478. [CrossRef]
26. Varga, D. No-Reference Video Quality Assessment Based on the Temporal Pooling of Deep Features. *Neural Process. Lett.* **2019**, *50*, 2595–2608. [CrossRef]
27. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
28. Gordo, A.; Almazán, J.; Revaud, J.; Larlus, D. Deep image retrieval: Learning global representations for image search. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 241–257.
29. Alaql, O.; Lu, C.C. No-reference image quality metric based on multiple deep belief networks. *IET Image Process.* **2019**, *13*, 1321–1327. [CrossRef]
30. Sharif Razavian, A.; Azizpour, H.; Sullivan, J.; Carlsson, S. CNN features off-the-shelf: An astounding baseline for recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 24–27 June 2014; pp. 806–813.
31. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 1–9.
32. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
33. Lin, H.; Hosu, V.; Saupe, D. KonIQ-10k: Towards an ecologically valid and large-scale IQA database. *arXiv* **2018**, arXiv:1803.08489.
34. Lin, H.; Hosu, V.; Saupe, D. KADID-10k: A large-scale artificially distorted IQA database. In Proceedings of the 2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX), Berlin, Germany, 5–7 June 2019; pp. 1–3.
35. Ponomarenko, N.; Jin, L.; Ieremeiev, O.; Lukin, V.; Egiazarian, K.; Astola, J.; Vozel, B.; Chehdi, K.; Carli, M.; Battisti, F.; et al. Image database TID2013: Peculiarities, results and perspectives. Signal Process. *Image Commun.* **2015**, *30*, 57–77. [CrossRef]
36. Ghadiyaram, D.; Bovik, A.C. Massive online crowdsourced study of subjective and objective picture quality. *IEEE Trans. Image Process.* **2015**, *25*, 372–387. [CrossRef] [PubMed]

37. Serre, T.; Wolf, L.; Bileschi, S.; Riesenhuber, M.; Poggio, T. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, 411–426. [CrossRef]
38. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv* **2015**, arXiv:1502.03167.
39. Gu, K.; Wang, S.; Zhai, G.; Lin, W.; Yang, X.; Zhang, W. Analysis of distortion distribution for pooling in image quality prediction. *IEEE Trans. Broadcast.* **2016**, *62*, 446–456. [CrossRef]
40. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 586–595.
41. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE conference on computer vision and pattern recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
42. Drucker, H.; Burges, C.J.; Kaufman, L.; Smola, A.J.; Vapnik, V. Support vector regression machines. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 1997; pp. 155–161.
43. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
44. Xu, L.; Lin, W.; Kuo, C.C.J. *Visual Quality Assessment by Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2015.
45. Moorthy, A.K.; Bovik, A.C. A two-step framework for constructing blind image quality indices. *IEEE Signal Process. Lett.* **2010**, *17*, 513–516. [CrossRef]
46. Ye, P.; Kumar, J.; Kang, L.; Doermann, D. Unsupervised feature learning framework for no-reference image quality assessment. In Proceedings of the 2012 IEEE conference on computer vision and pattern recognition, Providence, RI, USA, 16–21 June 2012; pp. 1098–1105.
47. Xu, J.; Ye, P.; Li, Q.; Du, H.; Liu, Y.; Doermann, D. Blind image quality assessment based on high order statistics aggregation. *IEEE Trans. Image Process.* **2016**, *25*, 4444–4457. [CrossRef] [PubMed]
48. Bosse, S.; Maniry, D.; Müller, K.R.; Wiegand, T.; Samek, W. Deep neural networks for no-reference and full-reference image quality assessment. *IEEE Trans. Image Process.* **2017**, *27*, 206–219. [CrossRef] [PubMed]
49. Venkatanath, N.; Praneeth, D.; Bh, M.C.; Channappayya, S.S.; Medasani, S.S. Blind image quality evaluation using perception based features. In Proceedings of the 2015 Twenty First National Conference on Communications (NCC), Mumbai, India, 27 February–1 March 2015; pp. 1–6.
50. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]