*Journal of Perceptual Imaging*® 4(1): 010402-1–010402-13, 2021.


# FP-Nets for Blind Image Quality Assessment

**Philipp Grüning**[1] **and Erhardt Barth**[1]

[1] *Institute for Neuro- and Bioinformatics, University of Lübeck, Germany*
*E-mail: gruening@inb.uni-luebeck.de*

**Abstract.** *Feature-Product networks (FP-nets) are a novel deep-network architecture inspired by principles of biological vision. These networks contain the so-called FP-blocks that learn two different filters for each input feature map, the outputs of which are then multiplied. Such an architecture is inspired by models of end-stopped neurons, which are common in cortical areas V1 and especially in V2. The authors here use FP-nets on three image quality assessment (IQA) benchmarks for blind IQA. They show that by using FP-nets, they can obtain networks that deliver state-of-the-art performance while being significantly more compact than competing models. A further improvement that they obtain is due to a simple attention mechanism. The good results that they report may be related to the fact that they employ bio-inspired design principles.* © *2021 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.Percept.Imaging.2021.4.1.010402]

## 1. INTRODUCTION

In recent years, we witnessed a staggering increase in the consumption and distribution of visual content, primarily in social media, streaming platforms, and digital television. Apart from the extension of digital infrastructure, the technology of digital cameras further evolved as well. Nevertheless, distortions coming from all kinds of sources can still deteriorate the image quality. Thus, the need for efficient and effective image quality assessment (IQA) is growing. The desired application could be, for example, a quality score that is computed with every taken image, or a quality ranking of a set of images. Obviously, approaches with high efficiency, low computational cost and memory consumption, would be beneficial for running IQA on mobile devices and embedded systems. Several datasets with artificial [23, 32, 35] or natural distortions [10, 16] and human ratings of image quality are available. These datasets usually pose IQA as a regression problem: given an image, one has to assess the level of distortion either compared to an undistorted reference image or with no-reference (NR-IQA, blind IQA). We focus on the harder task of IQA without a reference image.

Many of today's state-of-the-art IQA approaches use convolutional neural networks (CNNs). CNNs have been derived from models of human vision, which is one reason why they are remarkably successful. Thus, the incorporation of further principles of human vision should be beneficial for CNNs in general and IQA in particular.

Originally, the CNNs had been designed to mimic orientation-selective simple- and complex cells of the primary visual cortex (V1). A typical CNN thus contains a multitude of convolution layers that are followed by a pointwise nonlinearity, most often a rectified linear unit (ReLU). Convolution layers operate on tensors with a certain height, width, and depth. For example, when applying CNNs to images, the first input tensor is the image with a depth of three, since it contains three color channels. These channels are often referred to as feature maps, containing the features that are extracted by the filters of the convolution layers. A filter is a three-dimensional (3D) weight matrix with a small height and width, and a depth that, in most cases, matches the number of incoming channels. It is applied to all image patches of the input tensor. The dot product between each patch and the filter is computed, resulting in the filter's output feature map. This computation simulates a neuron that is excited by certain stimuli yielding large outputs in certain areas of the feature map. It is important to note that the mentioned filter processes a tensor patch with its entire depth and it is not applied to only one image channel or feature map. A convolution layer can contain up to several thousand filters. Hence, the output is another tensor being a stack of several feature maps. Oriented filters, inspired by V1 neurons, reduce the entropy of natural images by encoding oriented straight patterns (1D regions) such as edges of different orientations [45]. Therefore, the filters are used for image compression and IQA, and often emerge in the convolution layers of CNNs when trained with natural images [21].

In cortical area V2, however, many cells are end-stopped to different degrees [15]. These cells are more selective and are thought to detect two-dimensional (2D) regions such as junctions and corners, or, more general, deviations from straight edges and lines. Since 2D regions are unique and sparse in natural images [1, 30, 45], they have the potential of representing images efficiently. A standard way of modeling end-stopped cells is to multiply outputs of orientation-selective cells [43, 44]. This leads to the idea behind Feature-Product networks (FP-nets), which are CNN architectures with additional FP-blocks [11]. In FP-blocks, for each feature map of an input tensor, two filters with a depth of one are applied, i.e., the two filters only operate on one feature map, a single channel. This results in two

Received Oct. 1, 2020; accepted for publication Mar. 17, 2021; published online Apr. 22, 2021. Associate Editor: Ruth Rosenholtz.
2575-8144/2021/4(1)/010402/13/$00.00

Figure 1. Examples for the LIVE in the Wild dataset. The dataset consists of authentic images captured by mainly using mobile devices. The examples show motion blur (top left), underexposure and noise (top right), overexposure (bottom left) and blurring (bottom right).

new feature maps that may, for example, contain edges and lines with certain orientations depending on what kind of two filters are learned. The multiplication of these two feature maps yields a feature map with high activation in areas where both filters yield large activations. An idealized corner detector is a typical example: one filter may react, for example, to horizontal lines and edges, the other only to vertical ones. The multiplication, or an alternative AND operation, leads to a feature map that represents corners consisting of a vertical AND a horizontal edge segment. Regarding IQ, a reasonable argument is that if human vision is focussing on edges and corners, the models used for IQA should also be able to focus on edges and corners. This principle is here applied twice by using (i) a 2D saliency measure and (ii) FP-blocks that explicitly allow for 2D selectivity.

Meanwhile, deep networks are defining the state of the art when it comes to predicting subjective image quality [4, 6, 27, 38]. We therefore expect that if we use such deep networks as reference and include FP-blocks, we could create more efficient networks. Here, we report state-of-the-art IQA results for FP-nets. The benefit of using FP-nets is that they are significantly more compact with less than 40% of the parameters of their deep CNN counterparts, in our case a ResNet-32 and a ResNet-50 [13].

## 2. RELATED WORK

For the FP-nets presented here, the multiplications are a key component. With CNNs, multiplications are also used in reweighting channel distributions [14, 41]. Li et al. [26] presented a bio-inspired architecture called Selective Kernel Networks, where neurons are able to adjust their receptive field size based on the input. Zoumpourlis et al. [49] introduced nonlinear convolution filters and show that augmenting the first layer of a CNN with quadratic forms by using a Volterra kernel [40] can improve generalization. Our FP-block can also be interpreted as a second-order Volterra kernel, but it has far fewer parameters and there are no constraints as to where in the network we can place it. Before the advent of deep learning, second-order terms have been explored in related fields [2, 3].

Multiplicative terms are also used with the so-called bilinear CNNs, first presented by Lin et al. [24], where the outer product of two feature vectors (coming from two separate CNNs) is computed. The resulting combined feature vector is a pooled version of local pairwise interactions, and the approach can yield better performance in fine-grained recognition. Apparent drawbacks are that the number of parameters is doubled at least and that the dimension of the feature vector increases quadratically. Derivations of bilinear models are used in several different applications [7, 9, 37], including IQA [47]. Li et al. [25] presented a factorized version of the approach mentioned above. Different layers, including convolutions, can be extended by their method, to increase the respective layer's capacity. Their results show that bilinear features work best in high-level layers of a CNN. However, using more than one layer of bilinear features decreases performance. In the FP-net versus bilinear CNN Section below, we show that FP-nets can be seen as a special case of a factorized bilinear CNN. However, the architectures differ and so do the results.

Several algorithms for IQA and several benchmarks have already been proposed (for an overview, we refer to Zhai and Min [46], and Kim et al. [19]). We evaluated our algorithms for blind IQA on the LIVE legacy dataset (Legacy) with artificial distortions and the two datasets with natural distortions LIVE in the Wild (LITW), and Kon-IQ (see Figure 1 for LITW examples).

Figure 2. Example from the Legacy dataset: a reference image is distorted by different operations. Image quality is encoded by a number between 1 (high) and 100 (very poor), the ground truth is the mean score of several annotators. Top left: reference; top right: white noise; bottom left: jpeg compression; bottom right: Gaussian blur.

Apart from CNN-based approaches, researchers have used handcrafted or unsupervised features in combination with regression models BRISQUE [29], for example, is based on the statistics of locally normalized luminance coefficients and CORNIA [42] on a dictionary learned from image patches. Tu et al. [36] created an NR-video quality assessment model by carefully selecting a subset of statistical features used in other state-of-the-art algorithms. Pei et al. [33] presented a full-reference IQA model based on difference of Gaussian features paired with a random-forest regressor.

For the Legacy dataset (see Figure 2 for an example), Kang et al. [18] first presented promising results using CNNs with only two convolution layers. They estimated the quality on small $32 \times 32$ patches and combined the resulting scores (patch-based fusing). Although it was assumed that high-level features do not contribute much to the visibility of image distortions, Bosse et al. [5] showed that deeper networks can yield better results. Even deeper models [19, 38] and the use of pre-training [4] were explored further. Liu et al. [27] used unlabeled data to pre-train a VGG16 [34] network on a large IQA-related dataset. Cheng et al. [6] showed that the prediction errors of a model are higher in patches of homogeneous areas (see Figure 3). Therefore, they used saliency to weight each patch. Similarly, we use the more salient patches with a simple saliency measure based on the structure tensor [17]. For datasets with authentic distortions, like LITW and Kon-IQ, successful CNN approaches use deeper and wider networks with larger input patches, and CNNs pre-trained on the ImageNet [8] dataset. Kim et al. [19] used pre-trained networks such as the ResNet-50 [13] and fused scores of several random patches of typical ImageNet inputs sizes ($224 \times 224$ or $227 \times 227$). They argue that the features learned (on ImageNet) to represent natural images are also viable in characterizing natural distortions but not artificial distortions. Bianco et al. [4] increased their model's performance by first training it on ImageNet and the Places [48] dataset. This pre-trained network was then fine-tuned on a different version of the LITW database by using a classification task and support vector regression on the resulting feature vector. Varga et al. [38] used a similar fine-tuning step with a spatial pyramid pooling layer and a multilayer perceptron. Zhang et al. [47] introduced the use of bilinear CNNs for IQA. Ma et al. [28] trained a multi-task CNN to compute a quality score by assigning each input image to a specific distortion class.

## 3. METHODS

In this section, we first explain the patch-based fusing approach that is typically used when employing CNNs in IQA. To determine the quality of one image, the network (used for regression) is presented with several image patches. For each patch, a score is computed and these values are combined to one global score, in our case by averaging. Selecting specific patches that are more salient can further enhance the results. To this end, we include an attention-based patch selection. After explaining how the FP-block is realized within a deep-learning framework, we compare FP-blocks with bilinear CNNs. Finally, we explain how FP-net architectures are created based on FP-blocks.

In blind IQA, the desired output for an input image is a score for subjective quality. In our case, a CNN is trained to directly predict the score. The network is presented with a subset of images, a batch, and predicts the scores. Each network has a number of parameters, for example, the weights for convolution. These parameters are updated to fit the labels of each batch via backpropagation. As

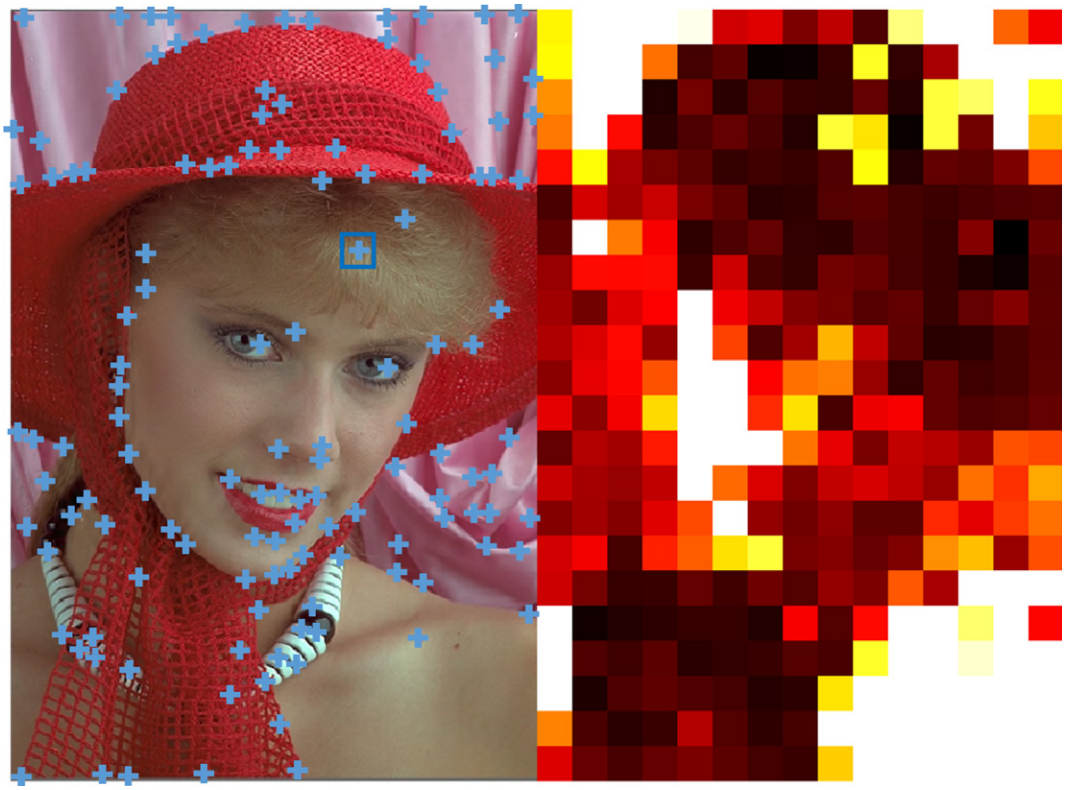

Figure 3. Illustration of saliency-based sampling: for the image on the left, the right panel shows the differences between the prediction and the ground truth for local patches (for white pixels the absolute difference is > 25). Note that predicting the image quality in a homogeneous area is more difficult. Therefore, we sample patches (see example blue box in the left image) with high structural information (blue crosses show patch centers) using the structure tensor.

loss function, we use the absolute difference of the CNN's prediction and the ground truth. To ensure that all images of a batch have the same dimension, only patches with constant height and width are processed.

In the testing phase, we follow the typical approach of fusing patch-based predictions as illustrated in Figure 4: to determine an image's quality, the trained network processes several image patches, yielding one score for each patch. These scores are then averaged to create the final output. When using this patch-based approach, one important design choice is the actual patch size: how much context is needed to predict the overall quality? Research indicates that smaller patches are needed for artificial distortions, while natural distortions require a larger input size [19]. Moreover, good results could only be achieved with sufficient pre-training on large-scale datasets such as ImageNet. This shows that the quality assessment of natural distortions is a more demanding task that needs a larger amount of data. So far, no IQA dataset exists that comes close to a million or more samples, which is not surprising, since label acquisition for IQA is a more intricate task compared to, for example, label acquisition for classification.

The fusing of patch-based predictions relies on the assumption that the scores of the individual patches capture the score of the whole image. This assumption is convenient since one can use several hundred crops per image to train a model, but it is not necessarily valid for all patches. Accordingly, some patches are better suited for prediction. We enhanced our results by selecting patches using an attention model based on the structure tensor.

## 4. ATTENTION MODEL

For artificial distortions, we can assume that small patches already contain sufficient information about perceptual quality. However, the predictive quality of a particular patch heavily depends on the image structure in that patch. The left side of Fig. 3 shows an example of how the predictive quality of patches can vary for jpeg 2000 compression. The right side of Fig. 3 shows the patchwise absolute distance between the ground truth and the local prediction of a CNN. One problem that can be observed is that blurring and compression effects do not alter homogeneous areas, and therefore, no information can be gained there. Hence, one should focus on patches that contain more structure. To this end, we present an effective strategy to sample such patches by using the structure tensor [17]

$$J = \int_{\Omega} \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix} d\Omega. \tag{1}$$

$I_x$ and $I_y$ are the image derivatives in horizontal- and vertical directions; $d\Omega$ is a local region. High values of

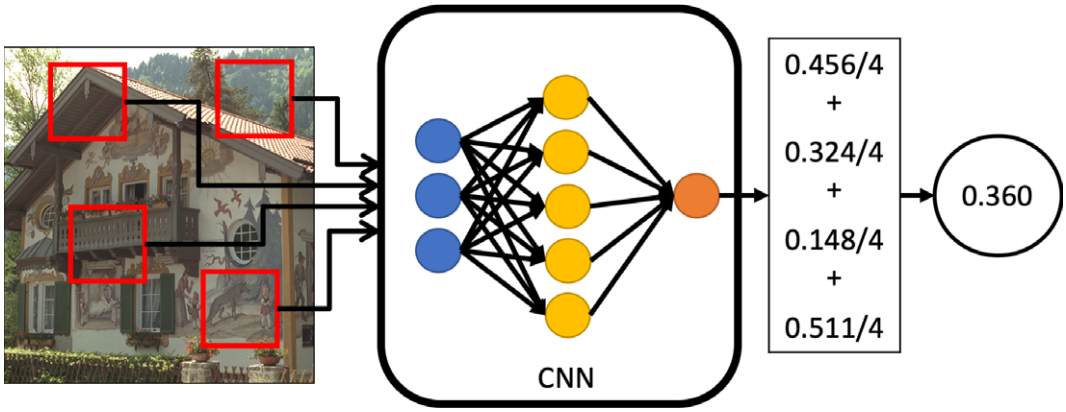


Figure 4. Patch-based processing and fusing for IQA. To infer the score for one image, the CNN is presented with a number of image patches. For each patch a score is predicted. The final score for the whole image is the mean value of all individual scores. For the datasets with authentic distortions, we select patches randomly. For the Legacy dataset, we compare an attention-based patch selection to random selection.

**Table I.** Pseudo-code of the attention-based inference method. For random cropping, steps 1 to 4 are substituted by 'crop $n$ patches randomly'.

| | |
|---|---|
| (1) | For the input image $I$, compute the determinant of $J$ for each pixel $(x, y)$ |
| (2) | Apply non-maximum suppression |
| (3) | Find the top $n$ pixels $(x_i, y_i)$, $i = 1, \ldots, n$ with the highest saliency value |
| (4) | Crop patches $p_i$ with $(x_i, y_i)$ in the center |
| (5) | For each $p_i$ compute the network prediction $z_i = f(p_i)$ |
| (6) | Return the score of $I$ as $F(I) = \frac{1}{n}\sum_{i=1}^{n} f(p_i)$ |

the determinant of $J$ indicate areas with 2D structure. In our implementation, we first convolved the input image with a Gaussian filter with $\sigma_{pre} = 3$. We computed the derivatives using Sobel filters. After computing the product terms $I_x I_x$, $I_x I_y$, and $I_y I_y$, we filtered the product terms with a Gaussian filter with $\sigma_{post} = 5$ (integration over a local region). We computed the determinant and applied non-maximum suppression in a window of 15 pixels to obtain a feature map, where each pixel encodes the amount of structure (an example is shown in Figure 5). To select the best $n$ patches, we use the top $n$ saliency values and crop patches around the respective center pixel. The pseudo-code for the method is given in Table I. The invariants of the structure tensor have already been used successfully to model human attention and saliency [39], i.e., areas where humans tend to look in an image.

## 5. FP-BLOCKS

Multiplications of oriented filters, and in general AND operations, can lead to more efficient representation of images [43–45]. Accordingly, we investigated whether AND operations can be helpful in CNN architectures and introduced FP-blocks as additional building blocks for CNNs [11]. Similar to a convolution layer, the input of an FP-block is a 3D tensor with a certain height, width, and depth (number of feature maps). The output is another tensor with possibly a smaller height and width, and usually, an increased depth. A schema of an FP-block is shown in Figure 6 on the right. The input depth $d_{in}$ is expanded to $q \cdot d_{out}$, the desired output depth of the block times an expansion factor $q$. These new feature maps are created by weighted sums over the input feature maps, an operation implemented by the so-called $1 \times 1$ convolutions that actually convolve with kernels of size $1 \times 1 \times d_{in}$.

Subsequently, the expanded tensor is convolved by two depthwise separable (DWS) convolutions that are then multiplied. In deep-learning jargon, DWS convolutions are convolutions that operate only on one feature map (channel) and not across feature maps. After multiplication, the signal is $z$-scored (batch normalization without re-scaling) to reduce the risk of vanishing or exploding values. An additional $1 \times 1$ convolution (weighted summations over feature maps) creates the final output tensor of the FP-block with $d_{out}$ feature maps.

To illustrate how the multiplication of two linear filters can lead to end-stopping, consider an image of a rectangle with horizontal and vertical edges, and two linear filters, one selective to horizontal edges and one to vertical edges. At the corners, both filters would be activated and thus the product would differ from zero. The horizontal edges, however, would only activate the horizontal filter, the vertical edges only the vertical filter, and thus the product would be zero for all straight edges. For a full account on how to model end-stopped neurons based on multiplications of oriented filters see [43].

## 6. FP-NETS VERSUS FACTORIZED BILINEAR CNNS

The essential function of an FP-block is that it multiplies two filtered versions of the same input. In vectorized form, a patch $\vec{p}$, and a filter-pair $(\vec{f}_a, \vec{f}_b) \in \mathbb{R}^{k^2}$, $k$ being the kernel size, determine the value $g(\vec{p})$ of the patch's center pixel as

$$g(\vec{p}) = (\vec{f}_a{}^T \vec{p}) \left(\vec{f}_b{}^T \vec{p}\right) \tag{2}$$

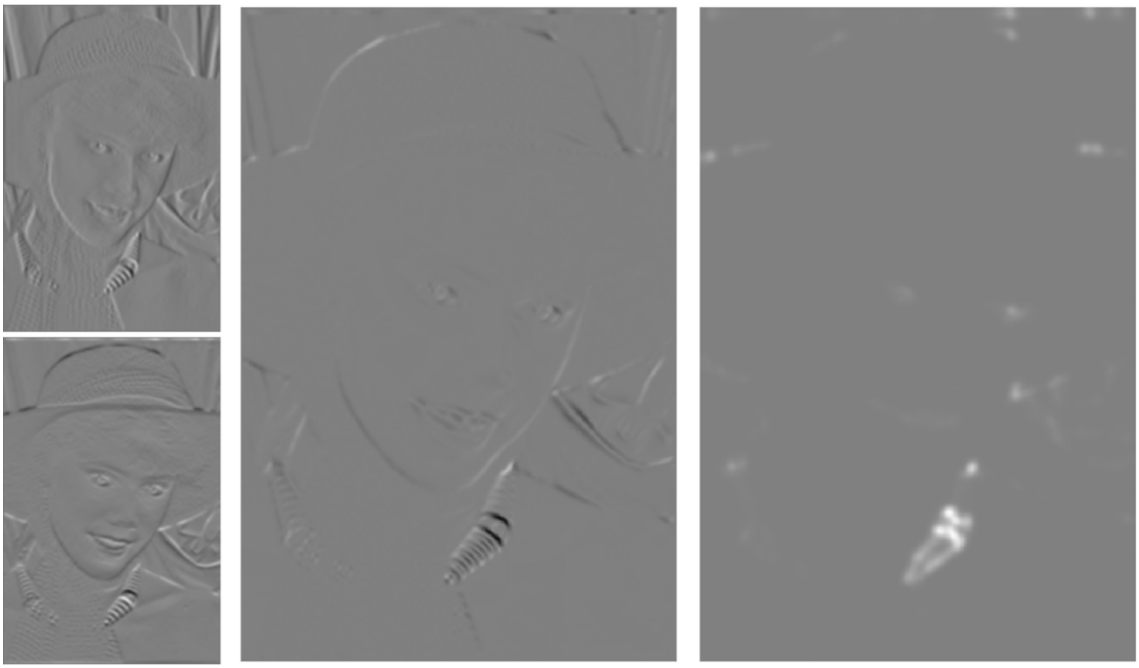

Figure 5. Left top and bottom panels show the horizontal and vertical derivatives of an image and the middle panel the pixelwise multiplication of the two derivatives. The determinant of the structure tensor (right) is used to encode the amount of structure in a local area.

$$g(\vec{p}) = \left(\sum_{i=1}^{k^2} f_a{}^i\, p^i\right)\left(\sum_{j=1}^{k^2} f_b{}^j\, p^j\right) \tag{3}$$

$$g(\vec{p}) = \sum_{i=1}^{k^2}\sum_{j=1}^{k^2} w^{ij} p^i p^j = \vec{p}^{\,T}\boldsymbol{W}\vec{p}. \tag{4}$$

The output is a weighted sum of $k^4$ possible pixel pairs. In a factorized bilinear layer, as presented by Li et al. [25], the output of each neuron depends on a linear part (not shown here) and a weighted combination of pairwise terms across feature maps. Let $\vec{x} \in \mathbb{R}^n$ be the vectorization of the $d$-dimensional input patch $\mathbf{P} \in \mathbb{R}^{k\,x\,k\,x\,d}$, and $n = k^2 d$. The quadratic part of an output neuron is computed by a scalar product of a vector $\vec{m}_0 \in \mathbb{R}^{n^2}$ and the vectorized version of the outer product of $x$ with itself

$$y = \vec{m}_0^{\,T}\, vec(\vec{x}\vec{x}^{\,T}) = \vec{x}^{\,T}\boldsymbol{M}\vec{x}. \tag{5}$$

$\boldsymbol{M} \in \mathbb{R}^{n\times n}$ is a reshaped version of $\vec{m}_0$ that can be expressed by a factorized matrix $\boldsymbol{V} \in \mathbb{R}^{c\times n}$

$$y = \vec{x}^{\,T}\boldsymbol{V}^{T}\boldsymbol{V}\vec{x} = \sum_{i=1}^{n}\sum_{j=1}^{n}(\vec{v}_i^{\,T}\vec{v}_j)x^i x^j \tag{6}$$

$$y = \sum_{i=1}^{n}\sum_{j=1}^{n} m^{ij} x^i x^j, \tag{7}$$

with $\vec{v}_i$ being the $i$th column of $\boldsymbol{V}$. Arguably, Eq. (7) can be seen as a general case of Eq. (4). However, instead of computing a weighted sum across all feature maps and summing over $k^4 d^2$ values, we restrict the generation of multiplicative terms to only one feature map at a time. Thus, FP-nets differ from bilinear CNNs in a way similar to how CNNs differ from multilayer perceptrons (MLPs). In addition, the combination of two different filters can result in matrixes $\boldsymbol{W}$ that cannot be reproduced by factorized matrixes $\boldsymbol{M}$. For example, $\vec{f}_a = (1, 0)^T$ and $\vec{f}_b = (0, 1)^T$ yield the symmetric $2 \times 2$ matrix $\boldsymbol{W} = \frac{1}{2}(\vec{f}_a\vec{f}_b^{\,T} + \vec{f}_b\vec{f}_a^{\,T})$ that has zero entries on the main diagonal and $\frac{1}{2}$ on the top-right and bottom-left positions. No combination of $\boldsymbol{V}^T\boldsymbol{V}$ can create such a matrix.

### 6.1 *FP-net Architectures*

FP-nets are CNNs that contain one or more FP-blocks. In principle, CNNs can approximate AND terms required for more efficient representations with a succession of convolutions and ReLUs. However, explicit incorporation of AND terms via multiplication should lead to more efficient networks. Accordingly, FP-nets should require fewer layers and parameters at equal performance. To test this hypothesis, we compared a baseline ResNet-32 [13] with two modified ResNets that contain FP-blocks (FP-net I and FP-net II). Fig. 7 shows the ResNet-32 architecture that was used by He et al. [13] on the Cifar-10 [22] dataset. We trained our networks on patches of size $32 \times 32$, which is the size of Cifar-10 images. For the ResNet-32 (see Fig. 7), the input was first processed by a $3 \times 3$ convolution and then by three stacks each consisting of five basic blocks. For the second and third stack, the first block downsampled the spatial tensor dimensions by using a stride of two. Finally, the tensor was reduced to a 64-dimensional vector by global average pooling, and the vector was then linearly combined to a scalar value that was normalized to the range [0, 1] by a sigmoid

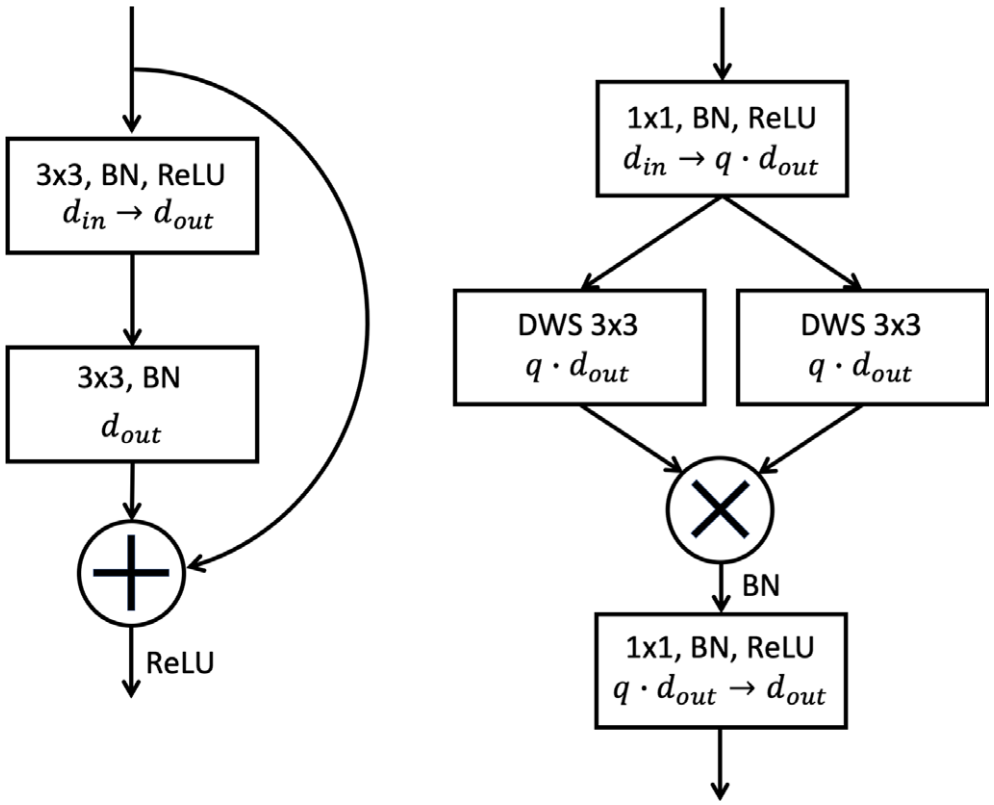


Figure 6. Comparison of the Cifar-10 ResNet basic block and our FP-block [44]. The left panel shows the architecture of the main building block used in the baseline model (see Figure 7). The right panel shows the novel FP-block based on feature products. Rectangles and layers depict the operations that are applied. Each rectangle's first line describes which operations are used (in sequence from left to right). BN: batch normalization; DWS: depthwise separable convolution; 3 × 3: convolution with a kernel of size 3 × 3; $q$: expansion factor. The second line denotes the input and output depth.

function. Fig. 6 shows, on the left, the ResNet's basic block that contains a residual connection. For our FP-net I, we substituted the last stack, i.e., five basic blocks, with three FP-blocks (see Figure 8). The expansion factor was $q = 2$. After the first FP-block, the signal was downsampled using max-pooling with a stride of 2 and a kernel size of 2.

Simply replacing an entire stack with a set of FP-blocks is just one variant of enriching a typical CNN architecture. Accordingly, this approach may not always be optimal, and combinations of FP-blocks within a stack may yield better results. In order to find a suitable architecture, we ran several experiments with different FP-nets on Cifar-10 and evaluated them. We widened the search space to different convolutional blocks and adopted the basic block and bottleneck block structure presented in Han et al.'s [12] Pyramid Residual Networks. Finally, we found one architecture (displayed in Figure 9) that performed particularly well with even fewer layers and parameters. We compare this architecture, denoted FP-net II, to the original FP-net I. For the FP-net II, we found that pre-training on Cifar-10 was not beneficial. Furthermore, each FP-block was augmented with shortcut connections, which were not used in the FP-net I. A shortcut connection adds the input to the output of a block. Again, an expansion factor $q = 2$ was used.

Models dealing with authentic distortions (LITW and Kon-IQ dataset) require larger input sizes and pre-training on larger datasets such as ImageNet. To evaluate FP-nets for this more challenging task, we compared a ResNet-50 to a corresponding FP-net, both pre-trained on ImageNet (for more information see Grüning et al. [11]). The architecture of the FP-net is similar to the one presented in Fig. 7. However, the input size is now 224 × 224, and four stacks are used. Basic blocks are substituted with bottleneck blocks, and the network uses more feature maps. A comparison of the ResNet-50 and FP-net architectures is given in Table II. Here, the expansion factor was equal to one.

## 7. EXPERIMENTS

We tested different approaches using FP-nets for three blind IQA datasets. The Legacy dataset contains artificial distortions of high-quality reference images. The LITW and Kon-IQ dataset contain images with natural distortions. For all three benchmarks, the FP-nets were trained on image patches. To infer the overall quality of an image, the values obtained for different patches were averaged. The approaches used for the artificial and authentic datasets differ in the size of a patch, the way patches are selected, the use of pre-training, and the network size. An overview is given in Table III. All experiments were conducted using PyTorch [31].

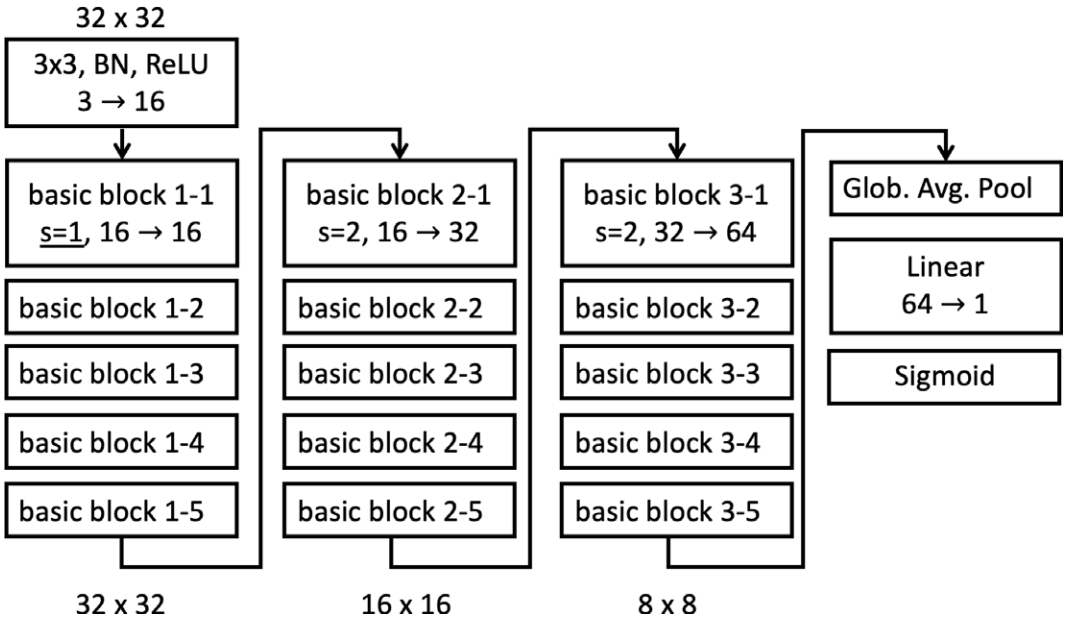


Figure 7. Schema of the ResNet-32 architecture used on the Cifar-10 dataset (output adapted for IQA-regression). The input size of the images is 32 × 32; $s = 2$ denotes a convolution stride of 2, 16 → 32 denotes a change in the number of channels from 16 to 32 feature maps. The input is first processed by a 3 × 3 convolution layer with batch normalization (BN) and ReLU. The subsequent operations are grouped in 3 × 5 basic blocks (see Fig. 6, left). Global average pooling transforms the processed input into a 64-dimensional feature vector. The output score is a weighted sum of the vector's entries that is normalized by a sigmoid function.

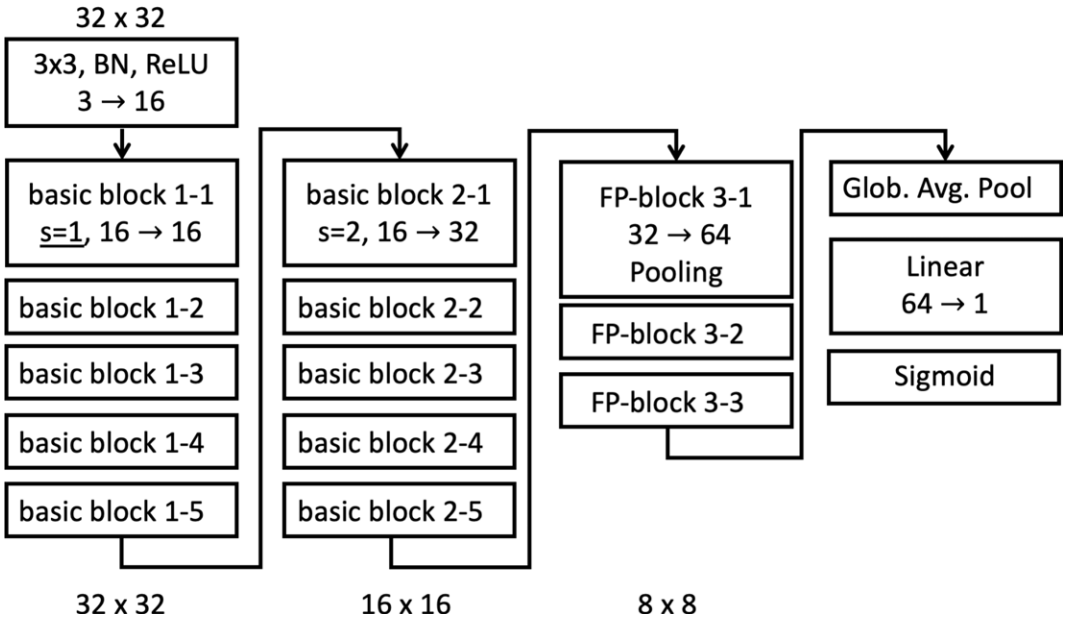


Figure 8. Schema of a FP-net I architecture based on the ResNet-32. For the third stack, we substituted five basic blocks of the ResNet with three FP-blocks. The number of parameters is reduced to less than 40% of the original network.

### 7.1 *LIVE Legacy Dataset*

We report results for 10 random 80/20 splits of the LIVE legacy database [35], which contains 981 distorted images obtained from 29 reference images (see Fig. 2 for an example). Each image was rated by up to 29 subjects on a continuous scale ranging from 'poor' to 'excellent' quality. With these ratings, a score in the range [1, 100] was calculated for each image with 1 for best quality and 100 for bad quality. As in Bosse et al. [5], we used 17 images for training, 6 images for validation and early stopping, and 6 images for testing.

For training, we randomly cropped patches of size 32 × 32 from the images that were furthermore flipped along the vertical axis with a probability of 50%. Each patch was divided by 255. Subsequently, each color channel was

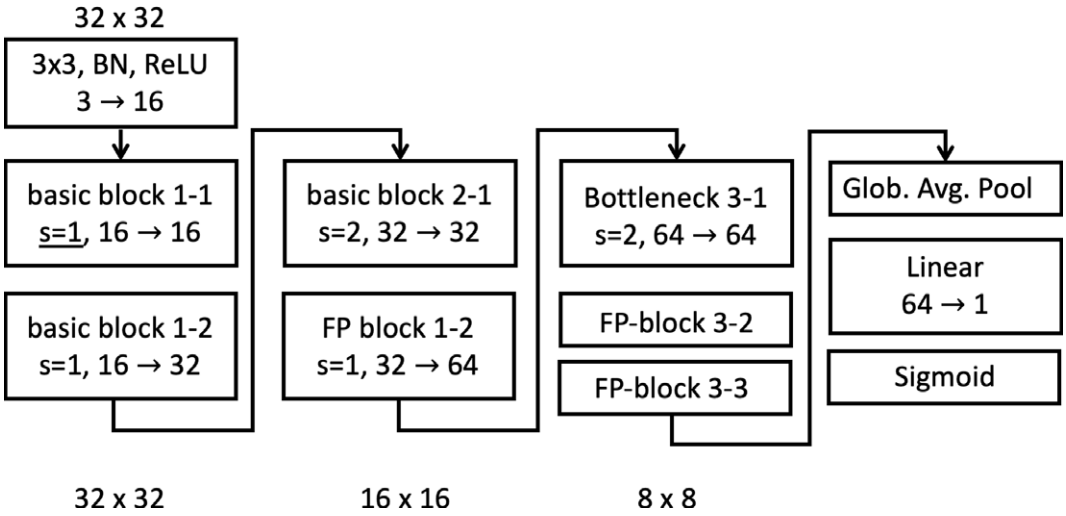


Figure 9. Schema of the FP-net II architecture: using compositions of convolutional blocks and FP-blocks, we can further reduce the number of blocks and parameters while improving the performance of the network. Here, we used the basic blocks and bottleneck blocks of PyramidNets.

**Table II.** Comparison of the ResNet-50 and FP-net architectures. For rows 2 to 4, the first convolution downsamples the input with a stride of 2. If a FP-block is involved, the output of the FP-block is downsampled by using max-pooling with a kernel size of 2 and a stride of 2.

| Output size | ResNet-50 | FP-net |
|---|---|---|
| 112 | $7 \times 7$, 64, stride 2 | |
| 56 | $3 \times 3$ max pool, stride 2 | |
| | $\begin{bmatrix} 1\times1 & 64 \\ 3\times3 & 64 \\ 1\times1 & 256 \end{bmatrix} \times 3$ | |
| 28 | $\begin{bmatrix} 1\times1 & 128 \\ 3\times3 & 128 \\ 1\times1 & 512 \end{bmatrix} \times 4$ | 1 × FP-block<br>$(256 \rightarrow 512)$, $q = 1$<br>$2 \times 2$ max pool, stride 2 |
| 14 | $\begin{bmatrix} 1\times1 & 256 \\ 3\times3 & 256 \\ 1\times1 & 1024 \end{bmatrix} \times 6$ | |
| 7 | $\begin{bmatrix} 1\times1 & 512 \\ 3\times3 & 512 \\ 1\times1 & 2048 \end{bmatrix} \times 3$ | 1 × FP-block<br>$(1024 \rightarrow 2048)$, $q = 1$<br>$2 \times 2$ max pool, stride 2 |
| 1 | Global average pooling, linear, sigmoid | |

subtracted with the ImageNet mean (0.485, 0.456, 0.406) and divided by the ImageNet standard deviation (0.229, 0.224, 0.225) for RGB. The presented networks were trained for 100 epochs with a learning rate of 0.001, a batch size of 128, and a weight decay of 0.001, using the Adam optimizer. The training loss function was the absolute error between the sigmoid output of the network and the target score divided by 100. The ResNet-32 and the FP-net I were pre-trained on Cifar-10, see He et al. [13]. The FP-net II was trained from scratch. For validation and early stopping, we randomly sampled 32 patches of each validation image.

**Table III.** Overview of the performed experiments.

| Name | Patch size | Patch selection | Dataset | Distortions | Pre-Training |
|---|---|---|---|---|---|
| ResNet-32 | 32 × 32 | Random | Legacy | Simulated | Cifar-10 |
| FP-net I | 32 × 32 | Random | Legacy | Simulated | Cifar-10 |
| FP-net II | 32 × 32 | Random | Legacy | Simulated | None |
| ResNet-32 (J) | 32 × 32 | Attention-based | Legacy | Simulated | Cifar-10 |
| FP-net I (J) | 32 × 32 | Attention-based | Legacy | Simulated | Cifar-10 |
| FP-net II (J) | 32 × 32 | Attention-based | Legacy | Simulated | None |
| ResNet-50 | 224 × 224 | Random | LITW/ Kon-IQ | Authentic | ImageNet |
| FP-net-50 | 224 × 224 | Random | LITW/Kon-IQ | Authentic | ImageNet |

The scores obtained for the patches were averaged, and then the Pearson linear correlation coefficient (PLCC) and the Spearman's rank order correlation coefficient (SROCC) between the prediction scores and the validation scores were computed. For testing, we used the model with the highest validation PLCC. We compared the baseline ResNet-32 to the FP-nets for two sampling strategies: random sampling and attention-based sampling, i.e., for each test image, we either sampled 128 samples randomly, or we selected the 128 samples with the highest saliency (defined by the determinant of $J$).

## 8. DATASETS WITH AUTHENTIC DISTORTIONS

With LITW, Ghadiyaram and Bovik [10] compiled a demanding benchmark containing 1,162 images with 350,000 scores from over 8,100 human observers. The images were captured mainly by using mobile devices in everyday life situations. Thus, apart from distortions due to the device's processing chain, image quality can be impaired by the photographer himself, due to over- and underexposure, motion blur, and other sources; see Fig. 1 for examples.

**Table IV.** Results for the LIVE legacy dataset (mean values). *J* denotes the use of the attention model. *N*. Param. (*M*) denotes the number of parameters in millions.

| Model | PLCC | SROCC | N. Param. (M) |
|---|---|---|---|
| Ref. [29] | 0.942 | 0.940 | – |
| Ref. [42] | 0.935 | 0.942 | – |
| Ref. [18] | 0.953 | 0.956 | 0.72 |
| Ref. [5] | 0.972 | 0.960 | 4.97 |
| Ref. [20] | 0.977 | 0.975 | 0.40 |
| Ref. [6] | 0.978 | 0.974 | 4.82 |
| Ref. [4] | 0.98 | 0.97 | 21.58 |
| Ref. [38] | 0.98 | 0.98 | >44.7 |
| Ref. [47] | 0.971 | 0.968 | >138 |
| Ref. [27] | 0.982 | 0.981 | 138 |
| ResNet-32 | 0.971 | 0.959 | 0.46 |
| ResNet-32 J | 0.975 | 0.963 | 0.46 |
| FP-net I | 0.973 | 0.962 | 0.17 |
| FP-net I J | 0.976 | 0.965 | 0.17 |
| FP II | 0.973 | 0.961 | 0.14 |
| FP II J | 0.977 | 0.965 | 0.14 |

**Table V.** Comparison of different approaches on the LIVE in the Wild (LITW) dataset. Results for [29] and [42] are reported from Kim et al. [19].

| Model | PLCC | SROCC | N. Param. (M) |
|---|---|---|---|
| Ref. [29] | 0.607 | 0.585 | – |
| Ref. [42] | 0.618 | 0.662 | – |
| Ref. [4] | 0.908 | 0.889 | 21.6 |
| Ref. [47] | 0.869 | 0.851 | >138 |
| Ref. [38] | 0.93 | 0.91 | >44.7 |
| Ref. [19] | 0.849 | 0.819 | 23.5 |
| ResNet-50 | 0.859 | 0.843 | 23.5 |
| FP-net | 0.856 | 0.839 | 14.1 |

**Table VI.** Comparison of different approaches on the Kon-IQ dataset. All results, except for ResNet-50, FP-net, and [38], are reported from Hosu et al. [16].

| Model | PLCC | SROCC | N. Param. (M) |
|---|---|---|---|
| Ref. [29] | 0.707 | 0.705 | – |
| Ref. [42] | 0.808 | 0.780 | – |
| ResNet-50 | 0.920 | 0.909 | 23.5 |
| FP-net | 0.918 | 0.907 | 14.1 |
| Ref. [38] | 0.95 | 0.92 | >44.7 |
| DeepBIQ (VGG16) | 0.886 | 0.872 | >18.1 |
| DeepBIQ (Inception) | 0.911 | 0.907 | >60 |
| KonCept512 | 0.937 | 0.921 | >60 |

The currently largest labeled dataset with authentic distortions is Kon-IQ [16], containing 10,073 images with 1,2M quality ratings from 1,459 annotators. We trained and evaluated on the small image sizes ($512 \times 384$).

We used the same approach for both datasets: as with the Legacy benchmark, we trained and tested CNNs for 10 different 80/20 splits. On LITW, Kim et al. [19] obtained good results with a pre-trained ResNet-50 and we therefore compared a ResNet-50 with a corresponding FP-net, both pre-trained on ImageNet. We trained both networks for 100 epochs with the Adam optimizer and with learning rate 0.0001 and weight decay 0.001. After the 40th and 80th epoch, the learning rate was reduced by a factor of 0.1. For the 2048-dimensional feature vector right before the linear mapping to the quality score, we used a dropout layer with $p = 0.5$. Each training mini batch contained 32 patches of size $224 \times 224$, roughly a quarter of the $500 \times 500$ dimensional input images. As before, we used the L1-norm as the loss function, vertical flips for data augmentation, and the same normalization as for the Legacy dataset. Due to the large patch size, attention-based patch selection yielded no benefit. Instead of 128 patches, each test image score was predicted with 25 random patches.

To better evaluate the generalization capabilities of the models, all networks trained on LITW were tested on the full Kon-IQ dataset and vice versa.

In addition to the performance scores, we report the number of parameters in millions (M) to compare for efficiency. If possible, we determined the number of parameters from the respective paper's description. In many cases, the CNNs used in the literature are large and were combined with even larger MLPs. In those cases, we report only the CNN's standard size as a lower bound. A ">" denotes that the size is even larger due to, for example, an additional MLP. Note that with the FP-nets, no additional regression models were used. We do not report the number of parameters for BRISQUE [29] and CORNIA [42].

Figure 10 presents results for all configurations tested on the Legacy dataset. First, note that our simple attention model (denoted by *J*) increased the median PLCC by 0.2% for the ResNet, and by 0.4% for the FP-nets I and II (with % we denote an absolute increase of 1/100, for example, 0.975 with an 0.2% increase yields 0.977). Also, note that the FP-nets with attention model outperformed the ResNet by 0.2%. As shown in Table IV, the fact that FP-nets yield results comparable to the state of the art is remarkable since the number of parameters is reduced by a substantial amount. Furthermore, we were able to outperform the FP-net I in terms of mean PLCC by 0.1% using the FP-net II architecture (the minimum and maximum PLCC values also increased) with even fewer convolution layers and parameters. Note that the SROCC scores of both FP-nets are better than the ResNet-32's and that the attention mechanism increased the SROCC scores of all three architectures. The results for the ten splits of the LITW dataset are shown in Figure 11. Here, the FP-net has a higher variance (0.021 versus 0.015) with a lower minimum and a higher

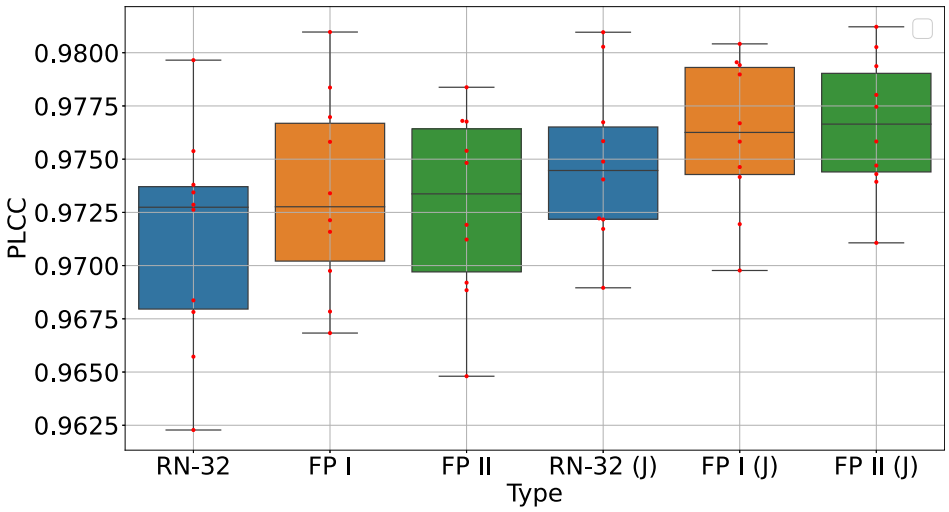


Figure 10. Results for the Legacy dataset obtained for ten different 80/20 splits. The first three boxes are obtained with the random sampling method for the ResNet-32 (blue, architecture shown in Fig. 7), the FP-net I (orange, architecture shown in Fig. 8), and the FP-net II (green, architecture shown in Fig. 9). The results for attention-based patch selection, marked with a (J), are illustrated with the last three boxes. Red dots indicate the actual test values obtained for each split.

**Table VII.** Cross-database results. For the LITW column, the networks were trained on Kon-IQ and tested on LITW, and vice versa. All results, except for ResNet-50 and FP-net, are reported from Hosu et al. [16].

| Model | LITW (PLCC/SROCC) | Kon-IQ |
|---|---|---|
| Ref. [29] | 0.598/0.561 | – |
| Ref. [42] | 0.644/0.621 | – |
| ResNet-50 | 0.790/0.786 | 0.799/0.744 |
| FP-net | 0.786/0.782 | 0.793/0.738 |
| DeepBIQ (VGG16) | 0.747/0.742 | – |
| DeepBIQ (Inception) | 0.821/0.804 | – |
| KonCept512 | 0.848/0.825 | – |

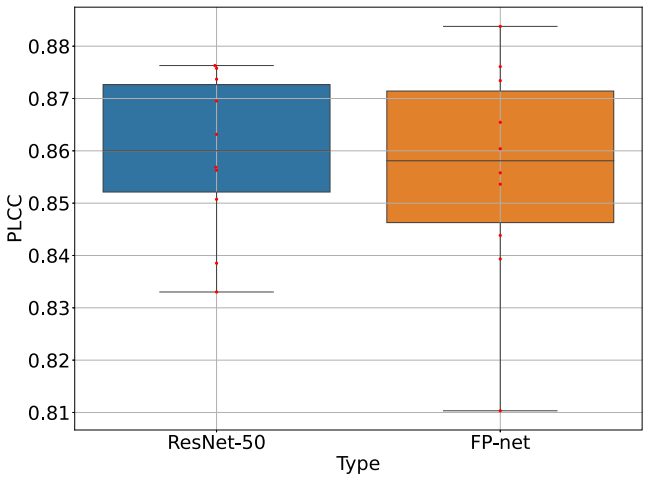


Figure 11. Results for the ResNet-50 and the corresponding FP-net on the LITW dataset.

maximum score. The mean value is decreased by 0.3% (0.859 versus 0.856), the median value is decreased by 0.2% (0.860 versus 0.858). Nevertheless, these results show that a comparable performance is possible with fewer convolution layers and significantly fewer parameters. Table V shows results for further state-of-the-art approaches on the LITW dataset.

Regarding the Kon-IQ results shown in Table VI, the FP-net performed on par with the ResNet-50 baseline. A similar outcome can be observed for the cross-database results in Table VII. Here, the differences between the mean PLCC and SROCC values are within one standard deviation (that was equal for the ResNet-50 and the FP-net). When trained on LITW and evaluated on Kon-IQ, the standard deviation was 0.01. When trained on Kon-IQ and evaluated on LITW, the standard deviations were 0.005 and 0.004 for PLCC and SROCC, respectively.

## 9. DISCUSSION

We used FP-nets to model perceptual image quality and obtained state-of-the-art results with fewer parameters and fewer layers. This increased efficiency was obtained by substituting generic convolutional blocks with the so-called FP-blocks that employ multiplications of feature maps.

FP-blocks are inspired by characteristics of end-stopped cells in biological visual systems; these cells can be modeled by multiplications of orientation-selective filters and provide efficient 2D representations. Since image quality assessment is based on human perception, models based on principles of human vision should be beneficial. The effectiveness of multiplicative terms and salient 2D representations is further illustrated by the attention model that we use for patch sampling.

An additional explanation of why FP-nets work well may be that they provide a second-order polynomial kernel that increases the layer's capacity and therefore reduces the number of layers needed.

We conclude that inspiration from research on human vision can still provide useful ideas for the design of deep networks beyond just convolutions and pooling. The FP-nets in particular seem well-suited for predicting subjective image quality with rather compact network models.

## REFERENCES


[1] E. Barth and A. B. Watson, "A geometric framework for nonlinear visual coding," Opt. Express **7**, 155–165 (2000).

[2] P. Berkes and L. Wiskott, "On the analysis and interpretation of inhomogeneous quadratic forms as receptive fields," Neural Comput. **18**, 1868–1895 (2006).

[3] J. Bergstra, G. Desjardins, P. Lamblin, and Y. Bengio, "Quadratic polynomials learn better image features," Tech. report **1337** (2009).

[4] S. Bianco, L. Celona, P. Napoletano, and R. Schettini, "On the use of deep learning for blind image quality assessment," Signal, Image Video Process. **12**, 355–362 (2018).

[5] S. Bosse, D. Maniry, T. Wiegand, and W. Samek, "A deep neural network for image quality assessment," *2016 IEEE Int'l. Conf. on Image Processing (ICIP)* (IEEE, Piscataway, NJ, 2016), pp. 3773–3777.

[6] Z. Cheng, M. Takeuchi, and J. Katto, "A pre-saliency map based blind image quality assessment via convolutional neural networks," *2017 IEEE Int'l. Symposium on Multimedia (ISM)* (IEEE, Piscataway, NJ, 2017), pp. 77–82.

[7] A. R. Chowdhury, T.-Y. Lin, S. Maji, and E. Learned-Miller, "One-to-many face recognition with bilinear cnns," *2016 IEEE Winter Conf. on Applications of Computer Vision (WACV)* (IEEE, Piscataway, NJ, 2016), pp. 1–9.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," *2009 IEEE Conf. Comput. Vis. Pattern Recognit.* (IEEE, Piscataway, NJ, 2009), pp. 248–255.

[9] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, "Compact bilinear pooling," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2016), pp. 317–326.

[10] D. Ghadiyaram and A. C. Bovik, "Massive online crowdsourced study of subjective and objective picture quality," IEEE Trans. Image Process. **25**, 372–387 (2015).

[11] P. Grüning, T. Martinetz, and E. Barth, "Feature products yield efficient networks," Preprint arXiv:2008.07930 (2020).

[12] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2017), pp. 5927–5935.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2016), pp. 770–778.

[14] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2018), pp. 7132–7141.

[15] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat," J. Neurophysiol. **28**, 229–289 (1965).

[16] V. Hosu, H. Lin, T. Sziranyi, and D. Saupe, "KonIQ-10k: An ecologically valid database for deep learning of blind image quality assessment," IEEE Trans. Image Process. **29**, 4041–4056 (2020).

[17] B. Jähne, *Spatio-Temporal Image Processing: Theory and Scientific Applications* (Springer, Berlin, 1993), Vol. 751.

[18] L. Kang, P. Ye, Y. Li, and D. Doermann, "Convolutional neural networks for no-reference image quality assessment," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2014), pp. 1733–1740.

[19] J. Kim, H. Zeng, D. Ghadiyaram, S. Lee, L. Zhang, and A. C. Bovik, "Deep convolutional neural models for picture quality prediction," IEEE Signal Process. Mag. **34**, 130–141 (2017).

[20] J. Kim, A.-D. Nguyen, and S. Lee, "Deep CNN-based blind image quality predictor," IEEE Trans. Neural Netw. Learn. Syst. **30**, 11–24 (2018).

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Adv. Neural Inf. Process. Syst. **25**, 1097–1105 (2012).

[22] A. Krizhevsky, V. Nair, and G. Hinton, "CIFAR-10 (Canadian Institute for Advanced Research)," (2020).

[23] E. C. Larson and D. M. Chandler, "Most apparent distortion: full-reference image quality assessment and the role of strategy," J. Electron. Imaging **19**, 11006 (2010).

[24] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear CNN models for fine-grained visual recognition," *Proc. IEEE Int'l. Conf. on Computer Vision* (IEEE, Piscataway, NJ, 2015), pp. 1449–1457.

[25] Y. Li, N. Wang, J. Liu, and X. Hou, "Factorized bilinear models for image recognition," *Proc. IEEE Int'l. Conf. on Computer Vision* (IEEE, Piscataway, NJ, 2017), pp. 2079–2087.

[26] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2019), pp. 510–519.

[27] X. Liu, J. van de Weijer, and A. D. Bagdanov, "Rankiqa: Learning from rankings for no-reference image quality assessment," *Proc. IEEE Int'l. Conf. on Computer Vision* (IEEE, Piscataway, NJ, 2017), pp. 1040–1049.

[28] K. Ma, W. Liu, K. Zhang, Z. Duanmu, Z. Wang, and W. Zuo, "End-To-end blind image quality assessment using deep neural networks," IEEE Trans. Image Process. **27**, 1202–1213 (2018).

[29] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," IEEE Trans. image Process. **21**, 4695–4708 (2012).

[30] C. Mota and E. Barth, "On the uniqueness of curvature features," Dynamische Perzeption **9**, 175–178 (2000).

[31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Red Hook, NY, USA, 2019), pp. 8024–8035.

[32] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, and C.-C. J. Kuo, "Image database TID2013: Peculiarities, results and perspectives," Signal Process. Image Commun. **30**, 57–77 (2015).

[33] S. C. Pei and L. H. Chen, "Image quality assessment using human visual DOG model fused with random forest," IEEE Trans. Image Process. **24**, 3282–3292 (2015).

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1 (2014) arXiv preprint arXiv:1409.1556 pp. 1–14.

[35] H. R. Sheikh, M. F. Sabir, and A. C. Bovik, "A statistical evaluation of recent full reference image quality assessment algorithms," IEEE Trans. image Process. **15**, 3440–3451 (2006).

[36] Z. Tu, Y. Wang, N. Birkbeck, B. Adsumilli, and A. C. Bovik, "UGC-VQA: Benchmarking blind video quality assessment for user generated content," *arXiv*, pp. 1–13, (2020).

[37] E. Ustinova, Y. Ganin, and V. Lempitsky, "Multi-region bilinear convolutional neural networks for person re-identification," *2017 14th IEEE Int'l. Conf. on Advanced Video and Signal Based Surveillance (AVSS)* (IEEE, Piscataway, NJ, 2017), pp. 1–6.

[38] D. Varga, D. Saupe, and T. Szirányi, "DeepRN: A content preserving deep architecture for blind image quality assessment," *2018 IEEE Int'l. Conf. on Multimedia and Expo (ICME)* (IEEE, Piscataway, NJ, 2018), pp. 1–6.

[39] E. Vig, M. Dorr, T. Martinetz, and E. Barth, "Intrinsic dimensionality predicts the saliency of natural dynamic scenes," IEEE Trans. Pattern Anal. Mach. Intell. **34**, 1080–1091 (2012).

[40] V. Volterra, *Theory of Functionals and of Integral and Integro-differential Equations* (Dover, New York, 1959).

[41] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-net: Efficient channel attention for deep convolutional neural networks," Preprint arXiv:1910.03151 (2019).

[42] P. Ye, J. Kumar, L. Kang, and D. Doermann, "Unsupervised feature learning framework for no-reference image quality assessment," *2012 IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2012), pp. 1098–1105.

[43] C. Zetzsche and E. Barth, "Fundamental limits of linear filters in the visual processing of two-dimensional signals," Vis. Res. **30**, 1111–1117 (1990).

[44] C. Zetzsche and E. Barth, "Image surface predicates and the neural encoding of two-dimensional signal variation," *Human Vision and Electronic Imaging: Models, Methods, and Applications* (SPIE, Bellingham, WA, 1990), Vol. 1249, pp. 160–177.

[45] C. Zetzsche, E. Barth, and B. Wegmann, "The importance of intrinsically two-dimensional image features in biological vision and picture coding," *Digital Images and Human Vision*, edited by A. B. Watson (MIT Press, Cambridge, 1993), pp. 109–138.

[46] G. Zhai and X. Min, "Perceptual image quality assessment: a survey," Sci. CHINA Inf. Sci. **63**, 211301 (2020).

[47] W. Zhang, K. Ma, J. Yan, D. Deng, and Z. Wang, "Blind image quality assessment using a deep bilinear convolutional neural network", IEEE Trans. Circuits Syst. Video Technol. **30**, 36–47 (2018).

[48] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," IEEE Trans. Pattern Anal. Mach. Intell. **40**, 1452–1464 (2017).

[49] G. Zoumpourlis, A. Doumanoglou, N. Vretos, and P. Daras, "Non-linear convolution filters for CNN-based learning," *Proc. IEEE Int'l. Conf. on Computer Vision* (IEEE, Piscataway, NJ, 2017), pp. 4761–4769.