CentraleSupélec

Deep Learning and Contrast Quality Assessment

Supervisor
BRAULT Patrice
Dr, Research Engineer, CNRS (Centre National de la Recherche scientific)
IEE senior member
L2S, Laboratory of signal and system

REHMAN Abdul
M2 Multimedia Networking
École Telecom Paris, University Paris Saclay

Table of Contents

Abstract

In this work we describe a convolutional neural network (CNN) to measure image contrast quality based on human visual systems (HVS) without a reference image. The network consists of four convolutional layer with max pooling, one fully connected layers and the output which is considered as the quality assessment metric after being normalized to a score between 0 and 1 via Sigmoid function. Within the network structure, feature learning and regression are integrated into one optimization process, which leads to a more effective model for estimating image quality based on MOS. For the testing of CNN, we analyzed the test dataset based on the sub-measures defined in CEED2016 as spatial information (SI) [13], colorfulness (CF) [13], and global contrast factor (GCF) [19] because of fact that human observer rates the images w.r.t at least these important image characteristics. Then, three different type of correlations calculated between the predicted values from CNN and the MOS values. We perform different experiments to analyze the effect of the overall correlation between CNN and MOS values w.r.t to network architecture and different parameters within the CNN architecture, like number of layers in the CNN architecture, number of filters for convolutional layer, and the filter size as well. Using this approach achieves state of the art performance on the CEED2016[2] dataset for the sub-measures of spatial information and the global contrast factor(GCF).

Motivational Model

The motivation comes from already proposed model [1] for an image quality assessment tool to measure image enhancement quality with emphasis on contrast. In the tool, which is based on maximizing contrast with minimum artefact (MCMA), local and global properties of an image are measured through pixel-wise and histogram-wise features, respectively. To this aim, three sub measures are introduced, each of which able to detect one contrast-related quality aspect: (i) low dynamic range of image; (ii) histogram shape preservation during image enhancement process; and (iii) local pixel diversity. These sub-measures are combined through a subjective test to adapt them to the mean opinion scores (MOSs) of a diverse set of training contrast enhanced images. A regression algorithm performs the adaptation by fitting the three sub-measures to the MOS values and finding an optimal linear combination by maximizing the Pearson correlation. The experimental results show that MCMA has the highest correlation to the MOS when compared to the existing tested contrast measurement tools. Using MCMA as a reference, we tried to find an ideal model to measure the contrast quality of image. In this work, we proposed a CNN to measure image contrast quality w.r.t the human visual system.

1. Introduction

Image enhancement techniques have been widely used in many applications of image processing where the subjective quality of images is important for human interpretation. The goal of IQA is to enable assessment models to evaluate perceived images automatically and accurately, the results of which provide basis for algorithm parameter optimization and system performance evaluation in image processing filed. With the recent predominance of deep leaning algorithms, IQA has attracted increasing attention because the quality of train images is significant for the performance of learning-based models. Since the metric of IQA is often a continuous-valued score, IQA is always regarded as a regression task to learn the mapping function from images to their corresponding scores. Contrast is an important factor in any subjective evaluation of image quality.

1.1 Contrast enhancement(CE)

Contrast enhancement is a process that makes the image features stand out more clearly by optimal use of colors available on the display or output device. Contrast is created by the difference in luminance reflected from two adjacent surfaces. In other words, contrast is the difference in visual properties that makes an object distinguishable from other objects and the background. In visual perception, contrast is determined by the difference in the color and brightness of the object with other objects. Our visual system is more sensitive to contrast than absolute luminance; therefore, we can perceive the world similarly regardless of the considerable changes in illumination conditions. Many algorithms for accomplishing contrast enhancement have been developed and applied to problems in image processing.

1.1.1 Contrast enhancement with human visual system(HVS)

Contrast quality is a very complex yet inherent characteristic of an image. In principle, it is the measure of the contrast level compared with an ideal imaging model or perfect reference image. IQA provides an objective index or real value to measure the quality of the specified image. Since human beings are the ultimate receivers of visual information in practical applications, the most reliable IQA is to build a computational model to mimic the HVS. HVS-based image enhancement aims to emulate the way in which the HVS discriminates between useful and useless data [14]. The HVS system uses an objective evaluation measure for selection of parameters. This allows for more consistent results while reducing the time required for the enhancement process. The performance measure utilizes established methods of measuring contrast and processes these values to assess the useful information contained in the image.

1.1 Deep learning models for contrast quality measurement

Most deep learning methods use neural network architectures, which is why deep learning models are often referred to as deep neural networks. The term “deep” usually refers to the number of hidden layers in the neural network. Traditional neural networks only contain 2-3 hidden layers, while deep networks can have as many as 150. Deep learning models are trained by using large sets of labeled data and neural network architectures that learn features directly from the data without the need for manual feature extraction. Deep learning is an active area of research, in the aspect of every field level application. Deep learning is a machine learning approach which is currently revolutionizing a number of disciplines including image processing and computer vision. The work [4] proposes a novel blind IQA model evolved from Vgg16, a state-of the-art classification deep convolutional neural network(CNN). It modifies the classification network to apply it to regression task by keeping only one neuron in the last fully-connected layer. Vgg16 in [4] can evaluate image quality by giving an assessment score ranging from 0 to 1, inspired

by the recent trend that modified classification models based on deep learning theory can be applied into regression tasks because of the fact that humans prefer qualitative assessments instead of quantitative ones. An efficient cross correlation coefficient between the predictive assessment score of this model and the label score of TID-2013. But this work not considering contrast in particular, it's taking into the distortion

2. Implementation and testing of models

2.1 Tools and their support packages

2.1.2 MATLAB

MATLAB [12] is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include:

- Create, modify, and analyze deep learning architectures using apps and visualization tools.
- Preprocess data and automate ground-truth labeling of image, video, and audio data using apps.
- Accelerate algorithms on NVIDIA® GPUs, cloud, and datacenter resources without specialized programming.
- Collaborate with peers using frameworks like TensorFlow, PyTorch, and MxNet.
- Simulate and train dynamic system behavior with reinforcement learning.
- Generate simulation-based training and test data from MATLAB and Simulink® models of physical systems.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar no interactive language such as C or Fortran. MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB, toolboxes allow you to *learn* and *apply* specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control systems, neural networks, fuzzy logic, wavelets, simulation, and many others.

Deep learning support package

Deep Learning Toolbox provides a framework for designing and implementing deep neural networks with algorithms, pre-trained models, and apps. We can use convolutional neural networks (ConvNets, CNNs) and long short-term memory (LSTM) networks to perform classification and regression on image, time-series, and text data. We can build network architectures such as generative adversarial networks (GANs) and Siamese networks using automatic differentiation, custom training loops, and shared weights. Create new deep networks for image classification and regression tasks by defining the network architecture and training the network from scratch. We can also use transfer learning to take advantage of the knowledge provided by a pre-trained network to learn new patterns in new data. Fine-tuning a pre-trained image classification network with transfer learning is typically much faster and easier than training from scratch.

Functions:

trainingoptions	Options for training of neural network
Trainnetwork	Train a neural network for deep learning
analyzenetwork	Analyze a deep learning network architecture

Use *trainNetwork* [15] to train a convolutional neural network (ConvNet, CNN), a long short-term memory (LSTM) network, or a bidirectional LSTM (BiLSTM) network for deep learning classification and regression

problems. You can train a network on either a CPU or a GPU. For image classification and image regression, you can train using multiple GPUs or in parallel. *analyzeNetwork(layers)* analyzes the deep learning network architecture specified by layers. The *analyzeNetwork* [16] function displays an interactive visualization of the network architecture, detects errors and issues in the network, and provides detailed information about the network layers. The layer information includes the sizes of layer activations and learnable parameters, the total number of learnable parameters, and the sizes of state parameters of recurrent layers.

2.2 Dataset and Convolutional Neural Network(CNN)

2.2.1 TID2013

TID2013, [6] intended for evaluation of full-reference visual quality assessment metrics. This new database contains a larger number (3000) of test images obtained from 25 reference images, 24 types of distortions for each reference image, and 5 levels for each type of distortion. Mean opinion scores (MOS) for the new database have been collected by performing 985 subjective experiments with volunteers (observers) from five countries (Finland, France, Italy, Ukraine, and USA). The availability of MOS allows the use of the designed database as a fundamental tool for assessing the effectiveness of visual quality. Furthermore, existing visual quality metrics have been tested with the proposed database and the collected results have been analyzed using rank order correlation coefficients between MOS and considered metrics. These correlation indices have been obtained both considering the full set of distorted images and specific image subsets, for highlighting advantages and drawbacks of existing, state of the art, quality metrics. Since, we interested in the contrast enhancement, from this given dataset [6], we extracted the contrast distortion from TID2013 dataset images alongside the MOS values and then used to train the vgg16 deep convolutional neural network. This dataset with contrast as distortion, we have 25 reference images and then 5 different level of distortion for each reference images, so for the training of vgg16 deep convolutional neural network, we have 5 x 25 = 125 images in the database.

2.2.2 Vgg16

The proposed blind IQA model [7] is VGG-16 follows the typical CNN structure that stacked convolutional layers optional with pooling layers serve as feature extractor while following fully-connected layers serve as classifier, and it is optimized to accelerate training process and improve assessment performance from the following 4 aspects. Firstly, As Fig. 1 [7] shows, the 13 convolutional layers are divided into 5 groups, within-group layers share the same size of feature map and groups are connected with max-pooling layers to enhance presentation ability and robustness. Secondly, to effectively reduce the number of training parameters while maintaining satisfying model performance, 5×5 and 7×7 convolutional kernels are broken up into cascaded 3×3 ones, resulting in deeper network structure and higher non-linear level, and local response normalization is also removed. Thirdly, unlike general classification network in which the

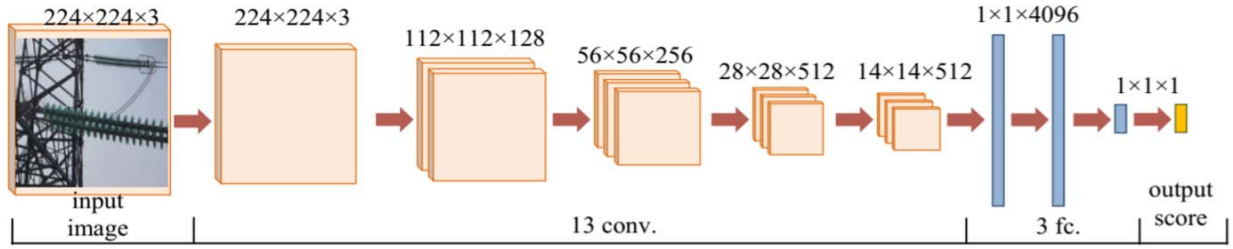

Figure 1: the main Structure of blind IQA model. Conv. is short for convolutional layers, and fc. is short for fully connected layers

number of neurons in the last fully-connected layer is the same as that of categories and the output of each neuron indicates the probability of its corresponding category, we retain only one neuron in the last layer, and because its output can be positive or negative with equal probability, we adopt Sigmoid function to normalize it into the assessment score of input image between 0 and 1 rather than the ReLU (Rectified linear unit) activation function used in previous layers.

2.2.3 Testing of Vgg16 with TID2013

We take some test images to test the trained vgg16 and then we compare the values of MCMA, MOS and predicted values of vgg16, we seen that, a similar curve follows up by MCMA and MOS for the images of five different contrast level as shown in figure 2, but on the other hand, a negative correlation between the MOS and predicted values of vgg16 as shown in figure 3.

2.2.3.1 Comparison between MCMA, MOS and Vgg16

We have reference image and 5 different contrast enhanced images, and then we have MCMA, MOS values for each of five images as shown in figure 2. We have also plot a graph between the values of MCMA, MOS and vgg16 predicted values as shown in figure 3. We have seen that the values predicted from vgg16 and the original values of MOS for these five images are not closely related to each other.

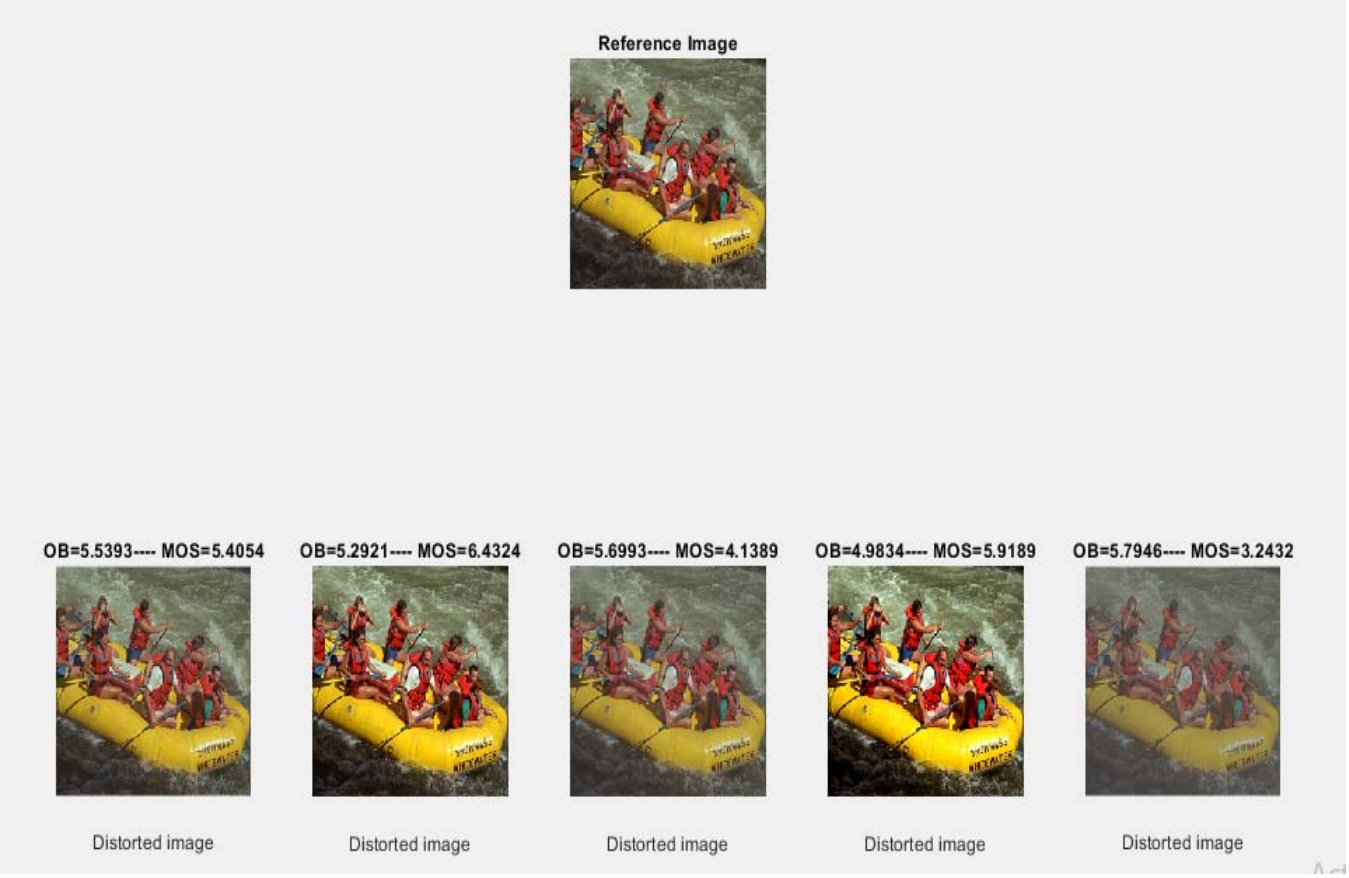

Figure 2: Reference image with five different enhanced image and values of Vgg16 with original MOS

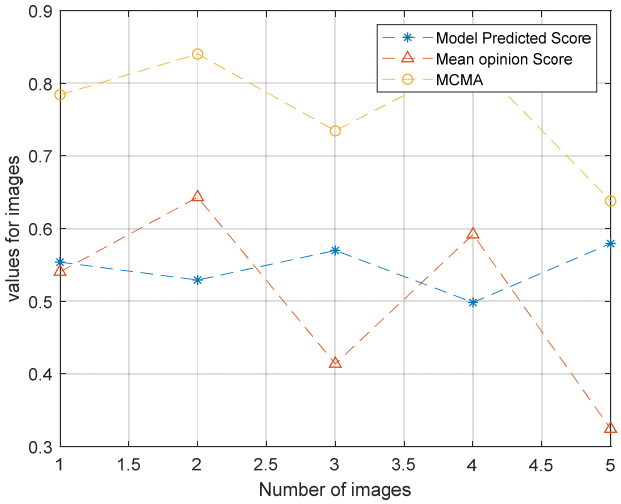

Figure 3: Predicted Score from Vgg16 and MCMA vs original MOS

2.2.3.2 Correlation between MCMA, MOS and Vgg16

Correlation is an analysis that measures the strength of association between two variables and the direction of the relationship. In terms of the strength of relationship, the value of the correlation coefficient varies between +1 and -1. As testing different images shows that the relationship between MOS and predicted values from vgg16 are negatively correlated as shown in figure 4.

Pearson	spearman	kendall
-0.23178	-0.25212	-0.17349

Figure 4: Correlation between vgg16 and MOS values

2.2.4 Drawbacks for using TID2013 with Vgg16

There are multiple drawbacks for using TID2013 database with Vgg16 for image contrast quality assessment. In proposed blind IQA model [7] VGG-16 with TID2013 dataset, they are taking whole TID2013 dataset including all type of distortion (like blurring, noise and compression etc.) for training vgg16. In this application, we are interested in contrast as distortion more particularly but TID2013 dataset is not much aligned with image contrast enhancement assessment, it's considering the noise and blurring as distortion in images (and all other types of distortion like compression), also It doesn't specify the CE (contrast enhancement) methods. Training of Vgg16 deep CNN with 41 layers is computationally expensive for such a small dataset that we able to obtained from TID2013 dataset considering contrast as a distortion.

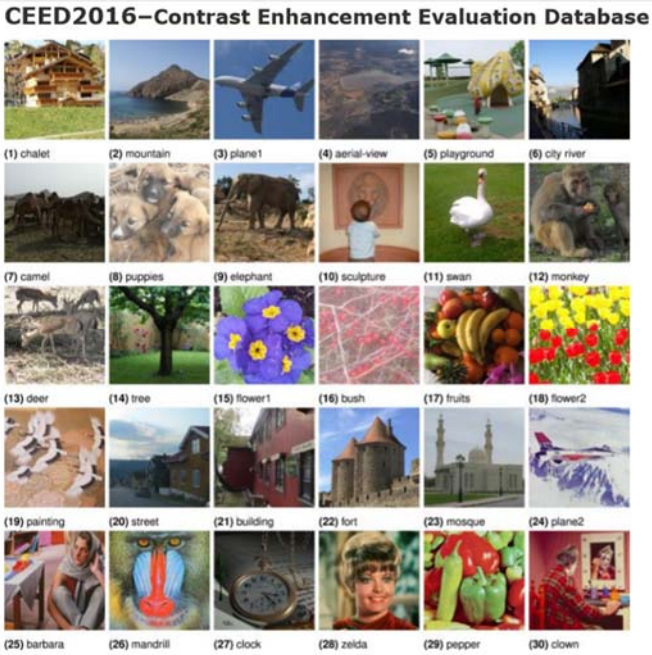

Figure 5: All 30 original Color images in CEED2016

2.3 CEED2016 and EVgg16

2.3.1 CEED2016

The CEED2016 [2] is newly developed image database dedicated for image contrast enhancement evaluation. The database [2] contains 30 original color images (as seen in figure 5) and 180 enhanced images obtained using six different Contrast Enhancement (CE) methods. The six CE methods in CEED2016 analysis are Adaptive Edge Based Contrast Enhancement (AEBCE), Contrast Limited Adaptive Histogram Equalization (CLAHE), Discrete Cosine Transform based (DCT), Global Histogram Equalization (GHE), Top Hat Transformation based (TOPHAT), and Multiscale Retinex (MRETINEX). The CEED2016 database was built with our own captured images and some common pictures widely used by the image processing community. The reason for using the old images is that these images are widely used in research related to image contrast enhancement and contrast enhancement evaluation. For consistency, they opted to include some of these images in our proposed database. Initially, they constructed a database consisting of 30 original color images with size of 512 × 512 pixels. Most of the images are captured by their lab members, while they have also included some classical images commonly used by the signal processing community. It is well understood that the human perception of image quality is highly dependent upon the scene under observation. For this reason, they selected images with different contents, color distribution, and contrast variations. They [2] have used three quantitative measures for the selection of images. These measures are Spatial Information (SI) [13], Colorfulness (CF) [13], and Global Contrast Factor (GCF). From the subjective experiments, they have derived the preferences scores (the number of times the algorithm is preferred over others) of the six CE algorithms. From the preference scores, it is clear that CLAHE is highly preferred whereas TOPHAT are not preferred most of the time. The contrast enhancement evaluation is also complex due to the effect of many parameters, such as image pleasantness, naturalness, and colorfulness.

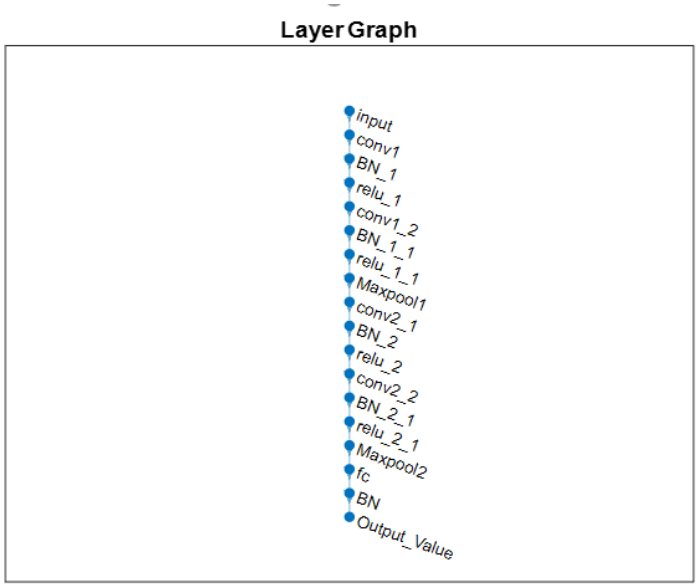

Figure 6: Layers included in EVgg16

2.3.2 EVgg16

The proposed network consists of four convolutional layers. Figure 6 shows the architecture of our network, which is a 224×224x3− 3× 3 × 64 − 3 × 3 x 128 − 6027392 − 1 structure. The input is locally normalized 224 x 224 x 3 image. The first layer is a convolutional layer which filters the input with 64 kernels each of size 3x3 with a stride of 1 pixel. The convolutional layer produces 64 feature maps each of size 3x3, followed by a pooling operation that reduces each feature map to one max and one min. one fully connected layers of 6027392 nodes each come after the pooling. The last layer is a simple linear regression with a one dimensional output that gives the score. The network architecture can vary depending on the types and numbers of layers included. The types and number of layers included depends

on the particular application or data. A smaller network with only one or two convolutional layers might be sufficient to learn on a small number of grayscale image data. On the other hand, for more complex data with millions of colored images, you might need a more complicated network with multiple convolutional and fully connected layers. To specify the architecture of a deep network with all layers connected sequentially, create an array of layers directly, starting by creating an image input layer for 224-by-224 color images with name 'input'. By default, the layer performs data normalization by subtracting the mean image of the training set from every input image. A 2-D convolutional layer applies sliding convolutional filters to the input. The layer convolves the input by moving the filters along the input vertically and horizontally and computing the dot product of the weights and the input, and then adding a bias term. Create a convolutional layer with 96,128 filters, each with a height and width of 3. Use a stride (step size) of 1 in the horizontal and vertical directions. A ReLU layer performs a threshold operation to each element of the input, where any value less than zero is set to zero. This operation is equivalent to equation 1.

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x \leq 0 \end{cases} \dots\dots\dots eq.1$$

A batch normalization (BN) layer normalizes each input channel across a mini-batch. To speed up training of convolutional neural networks and reduce the sensitivity to network initialization, use batch normalization layers between convolutional layers and nonlinearities, such as ReLU layers. The layer first normalizes the activations of each channel by subtracting the mini-batch mean and dividing by the mini-batch standard deviation. A dropout layer randomly sets input elements to zero with a given probability. At training time, the layer randomly sets input elements to zero given by the dropout mask $rand\left(size(x)\right) < Probability$, where x is the layer input and then scales the remaining elements by 1/(1-Probability). This operation effectively changes the underlying network architecture between iterations and helps prevent the network from overfitting. A higher number results in more elements being dropped during training. At prediction time, the output of the layer is equal to its input. A max pooling layer performs down-sampling by dividing the input into rectangular pooling regions, and computing the maximum of each region. The height and the width of the rectangular regions (pool size) are both 2. The pooling regions do not overlap because the step size for traversing the images vertically and horizontally (stride) is also [2 2]. Include a max pooling layer with non-overlapping regions in a Layer array. A sigmoid layer [17] applies a sigmoid function to the input such that the output is bounded in the interval (0,1). Create a fully connected layer using *fullyConnectedLayer* [18]. A fully connected layer multiplies the input by a weight matrix and then adds a bias vector. The convolutional (and down-sampling) layers are followed by one or more fully connected layers. As the name suggests, all neurons in a fully connected layer connect to all the neurons in the previous layer. This layer combines all of the features (local information) learned by the previous layers across the image to identify the larger patterns. For classification problems, the last fully connected layer combines the features to classify the images. This is the reason that the output Size argument of the last fully connected layer of the network is equal to the number of classes of the data set. For regression problems, the output size must be equal to the number of response variables. create a regression layer using *regressionLayer* [18]. A regression layer computes the half-mean-squared-error loss for regression problems. For typical regression problems, a regression layer must follow the final fully connected layer. EVgg16 is a reduced form of vgg16 in term of number of convolutional layers, and it has four convolutional layers with same bias and weights as defined in vgg16.

2.3.3 Training of EVgg16 with CEED2016

Training of EVgg16 based in MATLAB, using *trainNetwork* command to train a convolutional neural network (ConvNet, CNN), for deep learning classification and regression problems. A set of options for training a network using stochastic gradient descent with momentum. Reduce the learning rate by a factor of 0.2 every 5 epochs. Set the maximum number of epochs for training to 5, and use a mini-batch with 32 observations at each iteration. Turn on the training progress plot. When you train networks for deep learning, it is often useful to monitor the training progress as shown in figure 7. For regression networks, the figure plots the root mean square error (RMSE) instead of the accuracy. The figure marks each

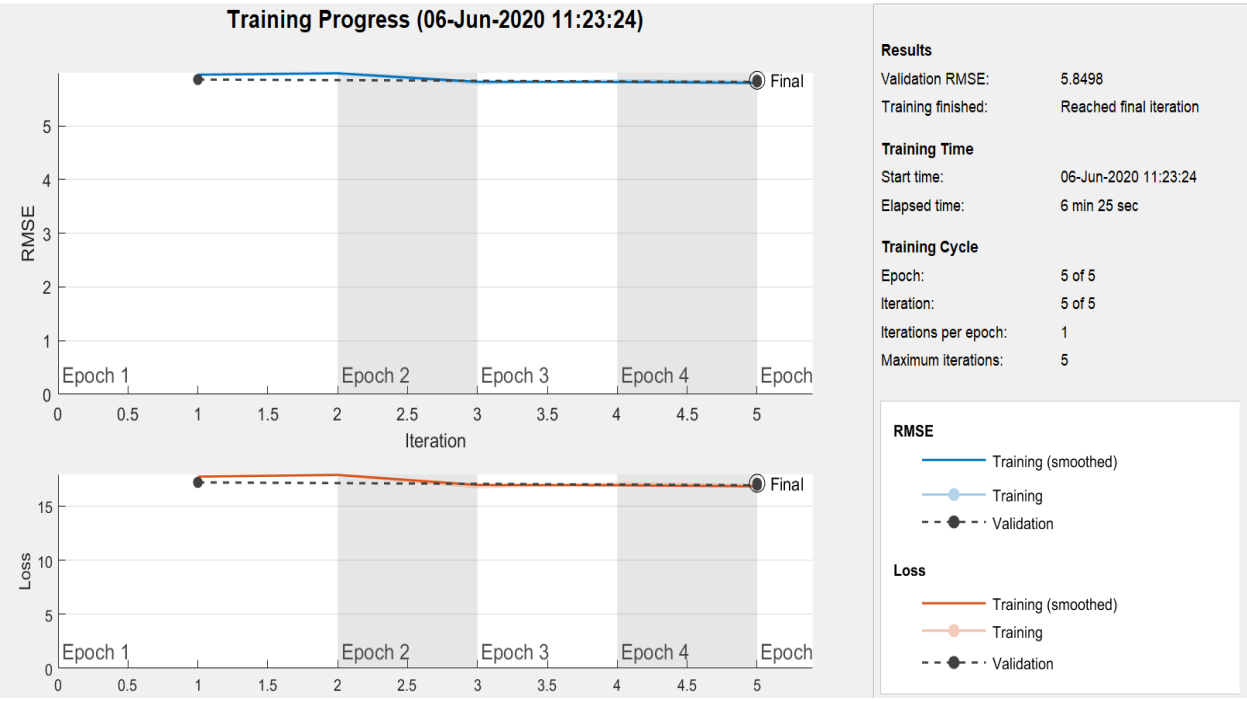

Figure 7: Training Process of EVgg16 for regression task

training Epoch using a shaded background. An epoch is a full pass through the entire data set. By plotting various metrics during training, you can learn how the training is progressing. For example, you can determine if and how quickly the network accuracy is improving, and whether the network is starting to over-fit the training data as shown in figure 7. When you specify 'training-progress' as the 'Plots' value in *trainingOptions* and start network training, *trainNetwork* creates a figure and displays training metrics at every iteration. Each iteration is an estimation of the gradient and an update of the network parameters. The figure plots the following:

- **Training accuracy** [13] — Classification accuracy on each individual mini-batch for classification problem.
- **Smoothed training accuracy** [13] — Smoothed training accuracy, obtained by applying a smoothing algorithm to the training accuracy. It is less noisy than the unsmoothed accuracy, making it easier to spot trends.
- **Training loss**, **smoothed training loss**, and **validation loss** [13] — The loss on each mini-batch, its smoothed version, and the loss on the validation set, respectively.

For regression networks, the figure plots the root mean square error (RMSE) instead of the accuracy. The figure marks each training epoch using a shaded background. An epoch is a full pass through the entire data set. Once training is complete, *trainNetwork* returns the trained network.

2.3.4 Testing of EVgg16

For the testing of Evgg16, we have test dataset of six classical research images that's defined in CEED2016, we had to take a look into our results based on the measures that defined as spatial information (SI) [13], colorfulness (CF) [13], and global contrast factor (GCF) [19] because of fact that human observer rates the images on these three measures. Contrast in image processing is usually defined as a ratio between the darkest and the brightest spots of an image. c [2]. GCF measures richness of detail as perceived by a human observer, and as such can be used in various application areas like rendering, tone mapping, volume visualization, and lighting design [19]. Colorfulness (CF) as a perceptual indicator of the variety and intensity of colors in the image [4]. Using $r_g = R - G$ and $y_b = 0.5(R + G) - B$ as a simple opponent color space, colorfulness is defined as in equation 2.

$$CF = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} + 0.3\sqrt{\mu_{rg}^2 + \mu_{by}^2} \dots\dots\dots eq.2$$

Spatial information (SI) as an indicator of edge energy. Let s_h and s_v denote the gray-scale images filtered with horizontal and vertical Sobel kernels, respectively. $s_r = \sqrt{s_v + s_h}$ then represents the edge magnitude at every pixel [4]. The SI as shown in eq.3 [4] value used here is the root mean square of the edge magnitude over the image or frame. SI can take in the means of texture analysis as well, in image

Figure 8: Showing the colorfulness measure values for all test dataset images

areas with smooth texture, the range of values in the neighborhood around a pixel is a small value; in areas of rough texture, the range is larger as shown in figure 10.

$$SI = \sqrt{\frac{L}{1080}} \sqrt{\sum \frac{s_r^2}{P}} \dots\dots\dots eq.3$$

Here is the number of pixels in the filtered image. The normalization factor $\sqrt{L/1080}$ (L is the number of lines, i.e., vertical resolution) is a somewhat crude but necessary step to reduce the scale/resolution-dependence of SI. By studying three sub-measures and testing these images for EVgg16 as shown in figure 8, we are able categories our test dataset in to three sub-categories as Best, Moderate, and worst predictions. So we have analyzed that Pepper and Mandrill images with sub-measure of a colorfulness(CF) characterized as best category, Clock and Plane images with sub-measure of a global contrast factor(GCF) goes into moderate, and Clown and Barbara images with sub-measure of spatial information(SI) distinguish as worst category as shown in figure 9 for the evaluation of contrast quality in regard with MOS and EVgg16.

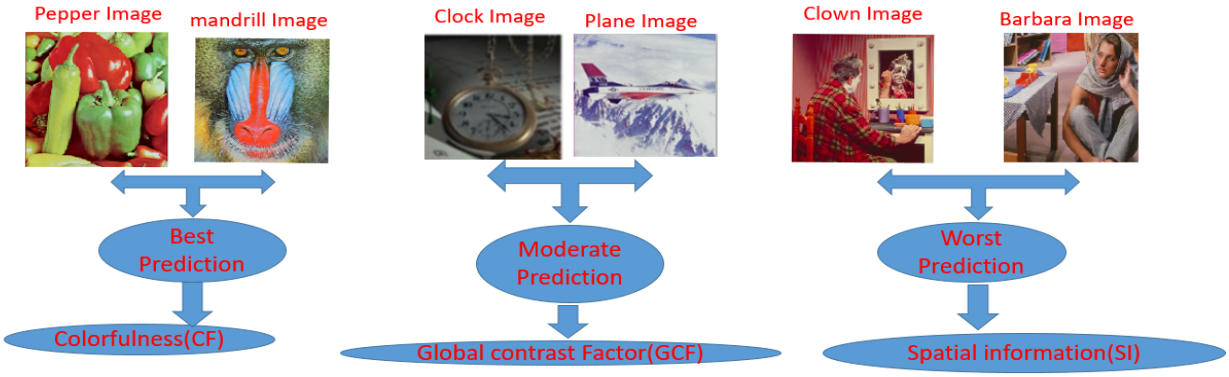

Figure 9: characterization of test images w.r.t categories on the basis of prediction and sub-measures

Figure 10: Texture analysis of all test dataset images using rangefilt in matlab

2.3.4.1 Comparison between MCMA, MOS and EVgg16

In this section, three test images are shown on the basis of sub-measures and categories of prediction, best, moderate and worst category of prediction.

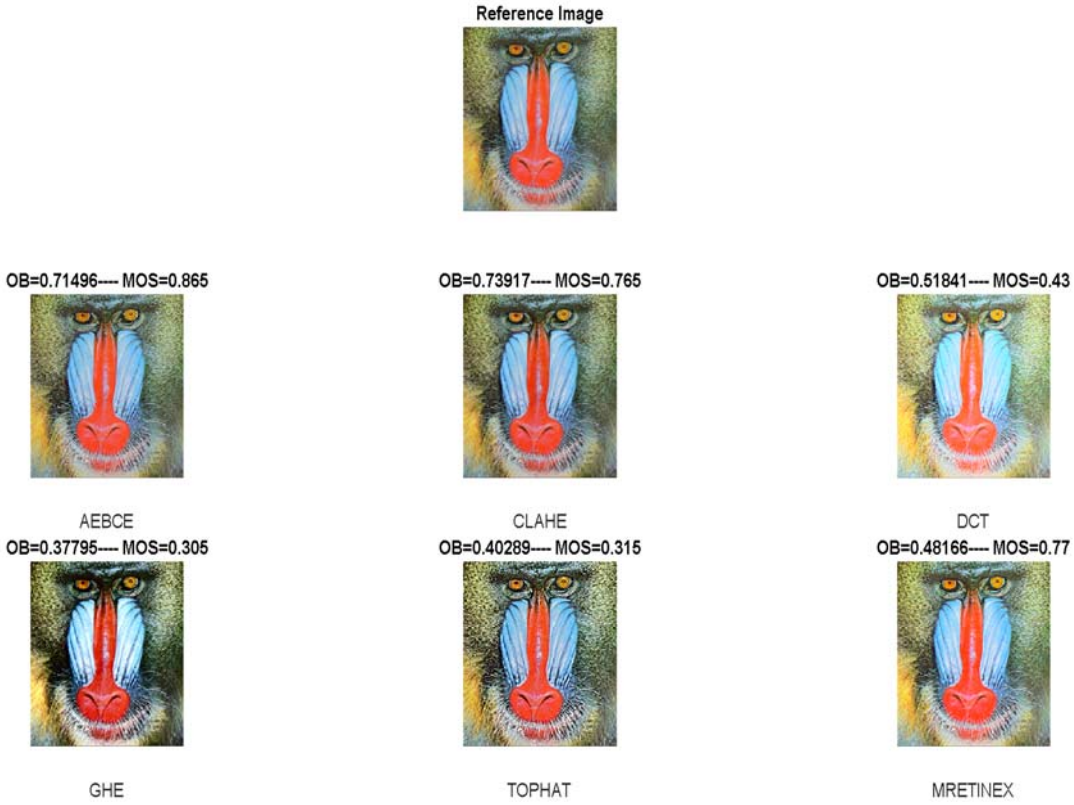

Figure 11: Reference image with six different enhanced image and the values of EVgg16 with original MOS for mandrill image with SI, best

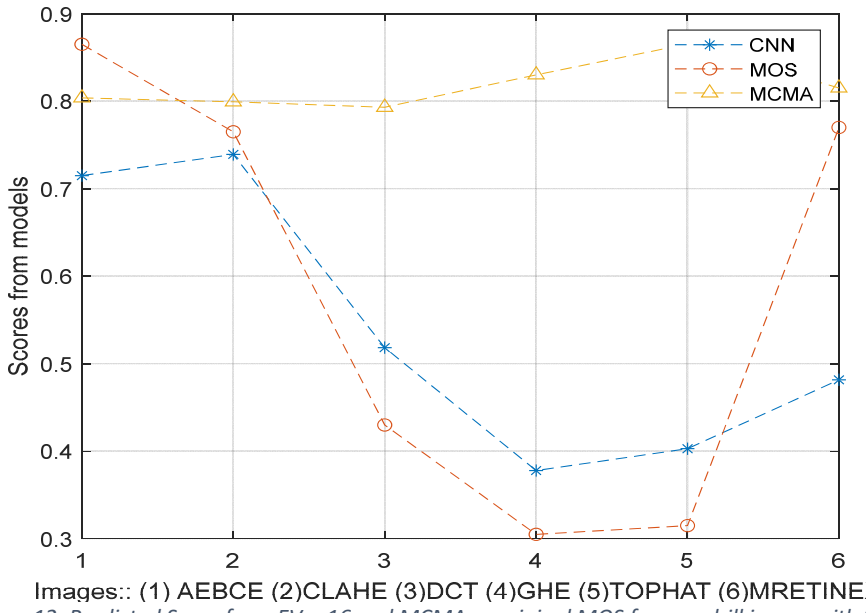

Figure 12: Predicted Score from EVgg16 and MCMA vs original MOS for mandrill image with SI, best

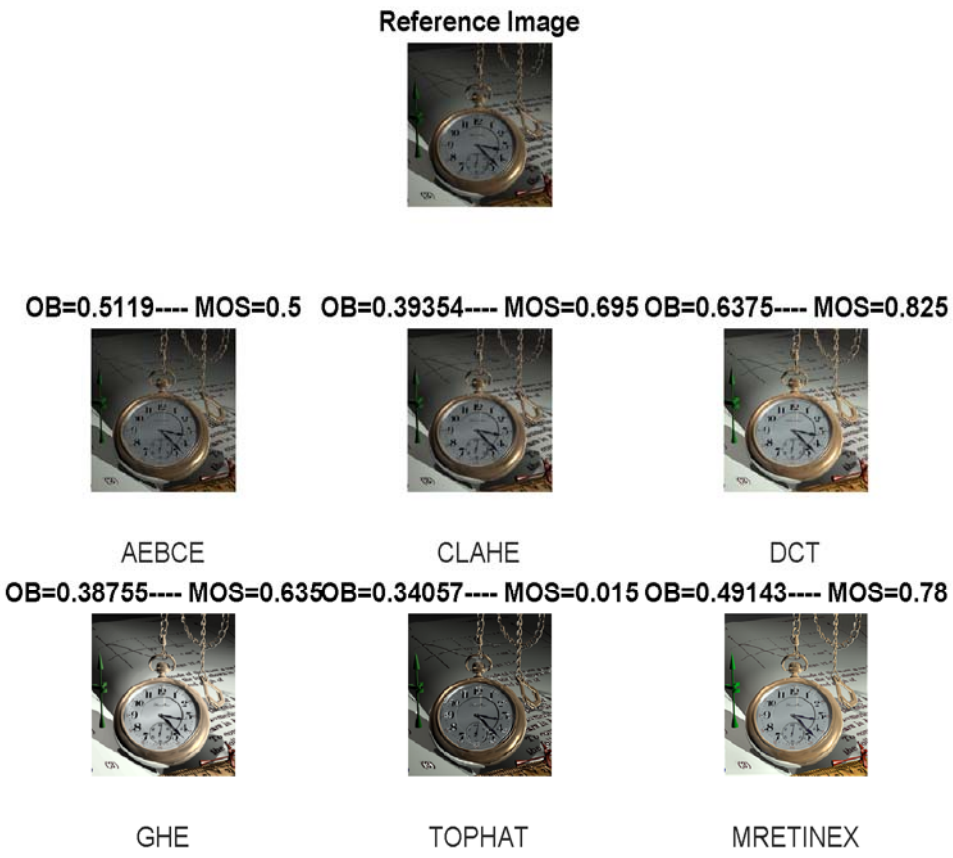

Figure 13: Reference image with six different enhanced image and the values of EVgg16 with original MOS for clock image with GCF, Moderate

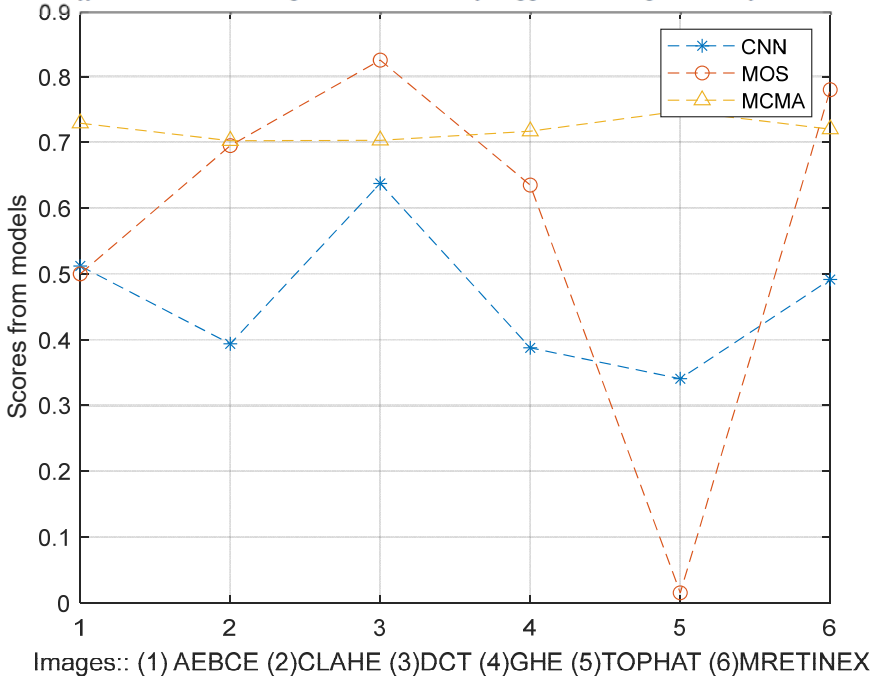

Figure 14: Predicted Score from EVgg16 and MCMA vs original MOS for clock image with GCF, Moderate

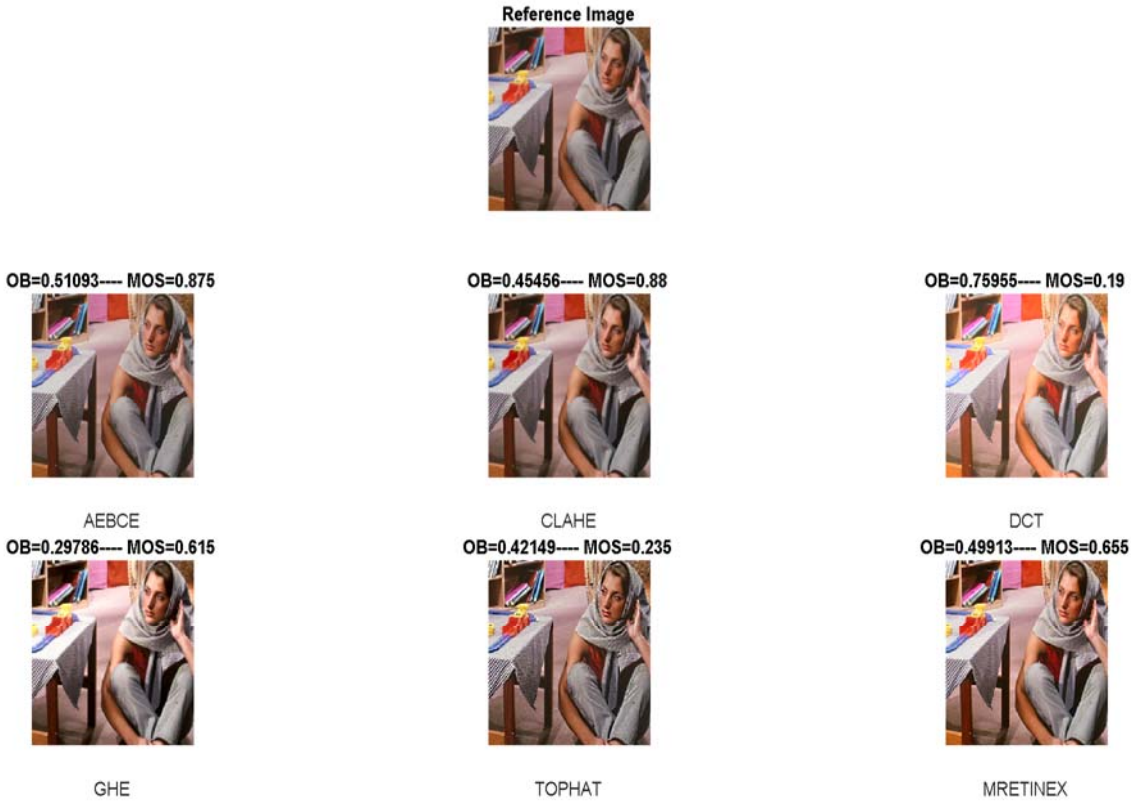

Figure 15: Reference image with six different enhanced image and the values of EVgg16 with original MOS for Barbara image with Colorfulness, worst

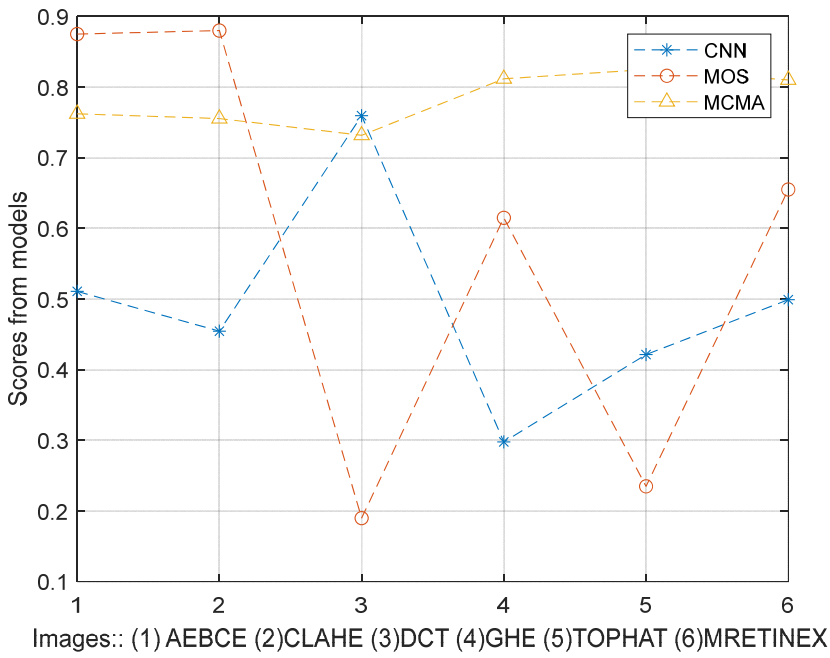

Figure 16: Predicted Score from EVgg16 and MCMA vs original MOS for Barbara image with Colorfulness, worst

2.3.4.2 Correlation between MCMA, MOS and EVgg16

A value of ± 1 indicates a perfect degree of association between the two variables. Here we have taken three different types of correlation as shown in figure 17, Pearson correlation is the most widely used correlation statistic to measure the degree of the relationship between linearly related variables. Kendall rank correlation is a non-parametric test that measures the strength of dependence between two variables. Spearman rank correlation is a non-parametric test that is used to measure the degree of association between two variables. The Spearman rank correlation test does not carry any assumptions about the distribution of the data and is the appropriate correlation analysis when the variables are measured on a scale that is at least ordinal. The experimental results show that the EVgg16 measure is well correlated to the human quality perception, whereas other existing contrast measurement deep learning networks are less reliable at properly reflecting image contrast quality.

	Image 1			Image 2			Image 3		
MOS VS EVgg16	Pearson	spearman	kendall	Pearson	spearman	kendall	Pearson	spearman	kendall
	0.81579	0.88571	0.73333	0.40305	0.54286	0.46667	0.24048	0.42857	0.33333
MOS VS MCMA	Pearson	spearman	kendall	Pearson	spearman	kendall	Pearson	spearman	kendall
	0.1165	0.028571	0.066667	-0.04507	-0.25714	-0.2	-0.10209	-0.14286	-0.33333

Figure 17: Correlation between MOS and Vgg16 also MCMA and original MOS

2.3.4.3 Experiments

CNN architecture involves different parameters, number of convolutional layers in architecture as shown in figure 18, number of filters, size of filter as shown in figure 19, and different training options. Different experiments and performed to analyze the effect of the parameter and then we concluded that EVgg16 with four convolutional layers and filter size is 3x3 and number of filter is 64,128 for convolutional layer would be the better choice to achieve the prediction of CNN in corresponds with HVS.

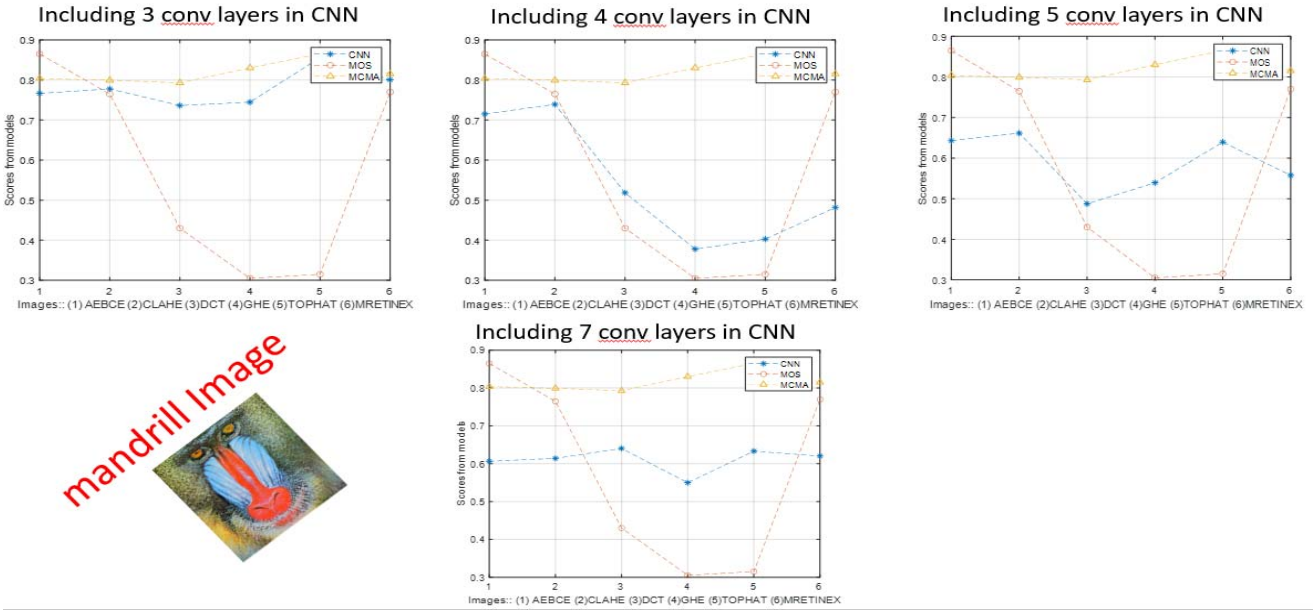

Figure 18: Comparison of overall prediction of CNN by changing number convolutional layers in EVgg16 architecture

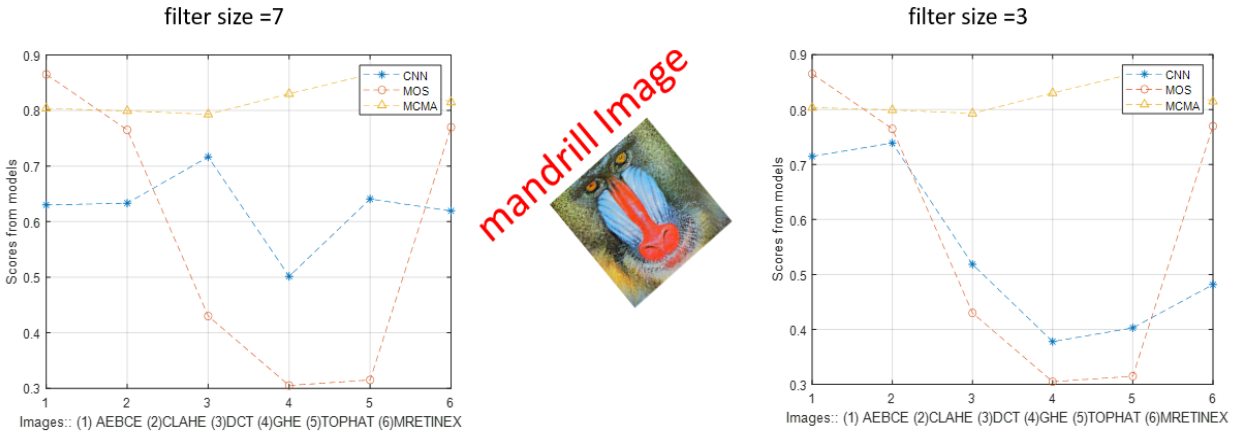

Figure 19: Comparison of overall prediction of CNN by changing filter size for convolutional layers in EVgg16

2.3.5 BRISQUE (Blind/Reference less Image Spatial Quality Evaluator)

A BRISQUE model is trained on a database of images with known distortions, and BRISQUE is limited to evaluating the quality of images with the same type of distortion. BRISQUE [21] is opinion-aware, which means subjective quality scores accompany the training images. No distortion specific features such as ringing, blur or blocking were modeled in the algorithm. The algorithm only quantifies the 'naturalness' in the image due to presence of distortions as shown in figure 20. We detailed the algorithm and the statistical features extracted, and demonstrated how each of these features correlate with human perception. The distribution of pixel intensities of natural images differs from that of distorted images [20]. This difference in distributions is much more pronounced when we normalize pixel intensities and calculate the distribution over these normalized intensities. In particular, after normalization pixel intensities of natural images follow a Gaussian Distribution (Bell Curve) while pixel intensities of unnatural or distorted images do not. The deviation of the distribution from an ideal bell curve is therefore a measure of the amount of distortion in the image.

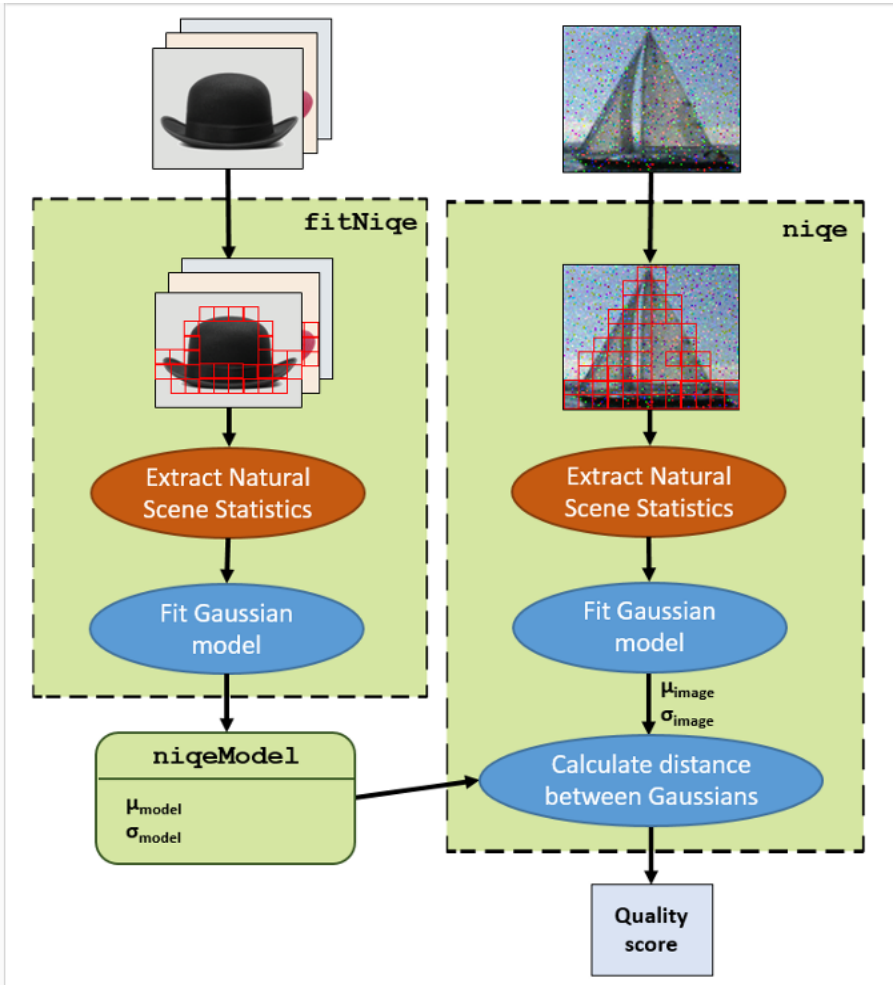

Figure 20: Workflow of BRISQUE

There are a few different ways to normalize an image. One such normalization is called Mean Subtracted Contrast Normalization (MSCN) [21]. To calculate the MSCN Coefficients, the image intensity $I(i,j)$at pixel $(i,j)$is transformed to the luminance $\hat{I}(i,j)$.

$$\hat{I}(i,j) = \frac{I(i,j) - \mu(i,j)}{\sigma(i,j) + C}$$

Where $i \in 1,2,,,,,,,M$,$J \in 1,2,,,,,,N$ (M and N are height and width respectively). Functions $\mu(i,j)$and $\sigma(i,j)$ are local mean field and local variance field respectively. Local Mean Field (μ) is nothing but the Gaussian Blur of the original image, while Local Variance Field (σ) is the Gaussian Blur of the square of the difference of original image and μ . In the equation below W is the Gaussian Blur Window function.

$$\mu = W \times I$$

$$\sigma = \sqrt{W \times (I - \mu)^2}$$

Then, the feature vector is calculated by fitting the MSCN image to a Generalized Gaussian Distribution (GGD). A GGD has two parameters — one for shape and one for variance. In a typical Machine Learning application, an image is first converted to a feature vector. Then the feature vectors and outputs (in this case the quality score) of all images in the training dataset are fed to a learning algorithm like Support Vector Machine (SVM). In this implementation, a support vector machine (SVM) regressor (SVR) [22] is used. SVR has previously been applied to image quality assessment problems. For example, a learning driven feature pooling approach using SVR was proposed in [23]. SVR is generally noted for being able to handle high dimensional data. Then evaluation of the BRISQUE index in terms of correlation with human perception and demonstrated that BRISQUE is statistically better than FR PSNR and SSIM as well as highly competitive to all NR algorithms compared with. BRISQUE performance is independent of database content and BRISQUE features may be used for distortion-identification as well. Further, BRISQUE is computationally efficient and that its efficiency is superior to other distortion-generic approaches to NR IQA [21], thus making BRISQUE an attractive option for practical applications like image de-noising, etc.

2.3.6 Testing and evalution of BRISQUE with MCMA, EVgg16 and MOS

The implementation and evaluation of BRISQUE of based on MATLAB with predefined function, A *brisqueModel* object encapsulates a model used to calculate the blind/referenceless image spatial quality evaluator (BRISQUE) perceptual quality score of an image. Using *fitbrisque* train a BRISQUE model containing a custom trained support vector regressor (SVR) model. In this section we analyze the prediction of BRISQUE w.r.t the MCMA, EVgg16 and MOS, from the analysis of correlation between the MCMA and BRISQUE values we have concluded that the BRISQUE and MCMA is more closely related to each other. But, the correlation between the values of MOS and BRISQUE shows that, they both are negatively correlated to each other. In the section 2.3.6.1 and 2.3.6.2 we have to test images from CEED2016 and showing the plot of prediction and correlation between the values of BRISQUE, MCMA, MOS, and EVgg16.

2.3.6.1 Comparison between MCMA, MOS, BRISQUE and EVgg16

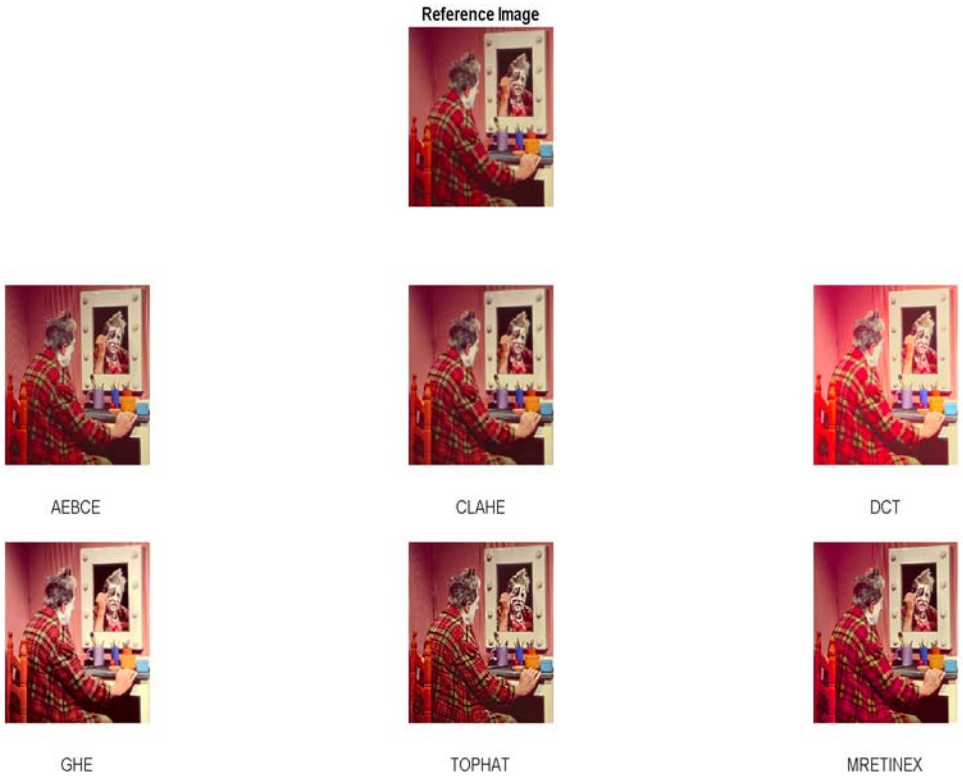

Figure 21: Reference image with six different enhanced image

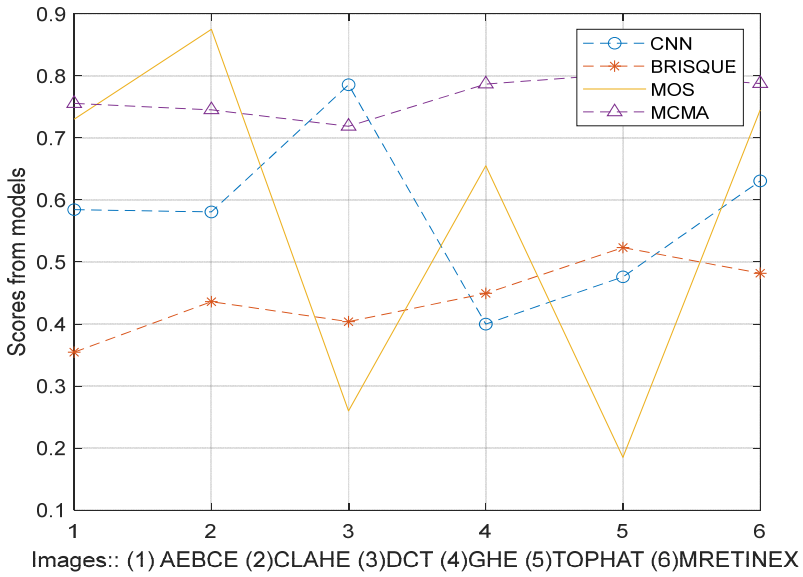

Figure 22: Predicted Score from EVgg16, BRISQUE and MCMA vs original MOS

Figure 23: Reference image with six different enhanced image

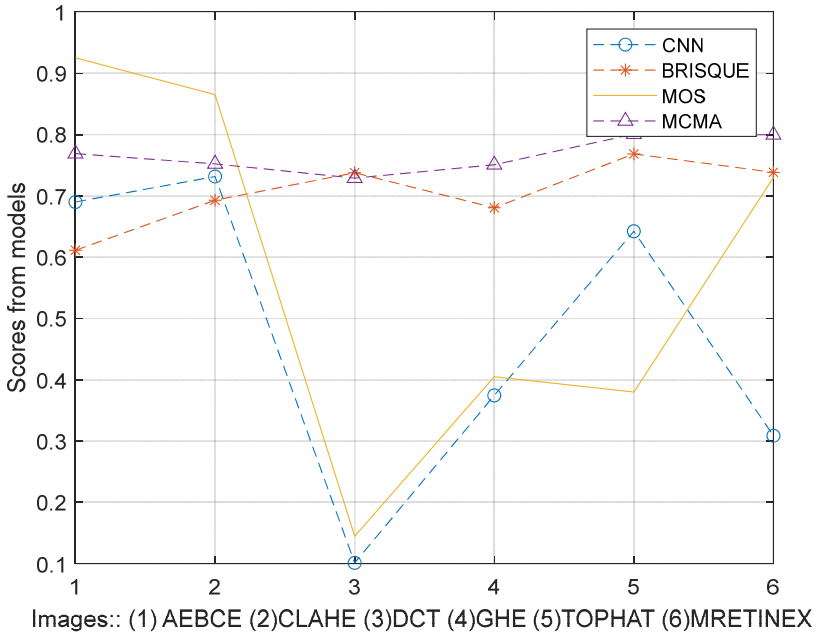

Figure 24: Predicted Score from EVgg16, BRISQUE and MCMA vs original MOS

2.3.6.2 Correlation between BRISQUE and MOS

Correlation is an analysis that measures the strength of association between two variables and the direction of the relationship. The experimental results show that the BRISQUE is well correlated to the MCMA instead of human quality perception, whereas MOS score is negatively correlated to the BRISQUE as shown in figure 25.

	Image 1			Image 2		
	Pearson	spearman	kendall	Pearson	spearman	kendall
BRISQUE VS MOS	-0.33099	-0.25714	-0.2	-0.61357	-0.6	-0.46667

Figure 25: Correlation between the BRISQUE and original MOS

Conclusion

In conclusion, we describe a Convolutional Neural Network (CNN) that's evolved from vgg16 deep neural network to measure image contrast quality based on human visual systems (HVS) values without the reference image. Three different type of correlation calculated between the predicted values from CNN to the MOS values to find out the relation. This approach achieves state of the art performance on the CEED2016 dataset. we understood that the human perception of image quality is highly depend upon the scene under observation. We had to analyzed the performance of EVgg16 w.r.t these three sub-measures that defined as spatial information (SI), colorfulness (CF), and global contrast factor (GCF) in CEED2016 dataset. As we know that, the contrast enhancement evaluation is complex task when we're taking into account the different parameters like naturalness, colorfulness and the pleasantness. In future of this work direction, to come up with a state-of-art contrast enhancement evaluation network, the parameter like spatial information(SI) should take into account for this task.

References

[1] M. Abdoli, F. Nasiri, P. Brault and M. Ghanbari, "Quality assessment tool for performance measurement of image contrast enhancement methods," in *IET Image Processing*, vol. 13, no. 5, pp. 833-842, 18 4 2019, doi: 10.1049/iet-ipr.2018.5520.

[2] M.A. Qureshi, A. Beghdadi, M. Deriche, "Towards the design of a consistent image contrast enhancement evaluation measure", Signal Processing: Image Communication, Volume 58, Part C, October 2017, pp.212-227

[3] M. Qureshi, A. Beghdadi, B. Sdiri, M. Deriche, F. A. Cheikh, "A Comprehensive Performance Evaluation of Objective Quality Metrics For Contrast Enhancement Techniques", in: European Workshop on Visual Information Processing (EUVIP), Marseille, France, 25-27 October 2016, pp. 1–5,http://dx.doi.org/10.1109/EUVIP.2016.7764589

[4] W. Zhang, K. Ma, J. Yan, D. Deng and Z. Wang, "Blind Image Quality Assessment Using a Deep Bilinear Convolutional Neural Network," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 36-47, Jan. 2020, doi: 10.1109/TCSVT.2018.2886771.

[5] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Comput. Sci., 2014.

[6] A. Mikhailiuk, M. Pérez-Ortiz and R. Mantiuk, "Psychometric scaling of TID2013 dataset," *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*, Cagliari, 2018, pp. 1-6, doi: 10.1109/QoMEX.2018.8463376.

[7] Lili Gao Jiangsu Wiscom Technologies Co., Ltd Nanjing, "Blind Image Quality Assessment Model Based on Deep Convolutional Neural Network".

[8] https://www.sciencedirect.com/topics/computer-science/human-visual-system

[9] Moon, Young & Han, Bok & Yang, Hyeon & Lee, Ho. (2019). Low Contrast Image Enhancement Using Convolutional Neural Network with Simple Reflection Model. Advances in Science, Technology and Engineering Systems Journal. 4. 10.25046/aj040115.

[10] Li, Yinhao & Iwamoto, Yutaro & Chen, Yen-Wei. (2019). Medical Image Enhancement Using Deep Learning. 10.1007/978-3-030-32606-7_4.

[11] Sarode, Surbhi & Sarade, Shunottara & Meshram, Sanjana & Kokande, Twinkle & Palandurkar, Ashish. (2020). IMAGE RESTORATION AND ENHANCEMENT USING DEEP LEARNING. International Journal of Engineering Applied Sciences and Technology. 04. 109-112. 10.33564/IJEAST.2020.v04i11.021.

[12] https://www.mathworks.com/products/matlab.html

[13] S. Winkler, "Analysis of Public Image and Video Databases for Quality Assessment," IEEE Journal of Selected Topics in Signal Processing, vol. 6, no. 6, pp. 616–625, oct 2012

[14] E. Wharton, S. Agaian, and K. Panetta, "A logarithmic measure of image enhancement," in Proc. SPIE, Defense Security Symp., Orlando, FL, Apr. 20, 2006, p. 62 500P

[15] https://www.mathworks.com/help/deeplearning/ref/trainnetwork.html

[16] https://www.mathworks.com/help/deeplearning/ref/analyzenetwork.html

[17] https://www.mathworks.com/help/deeplearning/ug/list-of-deep-learning-layers.html

[18] https://www.mathworks.com/help/deeplearning/ug/layers-of-a-convolutional-neural-network.html

[19] K. Matkovic, L. Neumann, A. Neumann, T. Psik, and W. Purgatholer, "Global Contrast Factor - a New Approach to Image Contrast," Computational Aesthetics in Graphics, Visualization and Imaging, pp. 159–168, 2005

[20] https://www.learnopencv.com/image-quality-assessment-brisque/

[21] A. Mittal, A. K. Moorthy and A. C. Bovik, "No-Reference Image Quality Assessment in the Spatial Domain," in *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695-4708, Dec. 2012, doi: 10.1109/TIP.2012.2214050.

[22] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," Neural Comput., vol. 12, no. 5, pp. 1207– 1245, 2000.

[23] M. Narwaria and W. Lin, "SVD-based quality metric for image and video using machine learning," IEEE Trans. Syst., Man, Cybern., B, Cybern., vol. 42, no. 2, pp. 347–364, Apr. 2012