No-Reference Feature Similarity Index Estimation Using Convolutional Neural Networks For Image Quality Assessment

Niranjan Rudrapatna Nataraja
Nvidia
nnataraja@nvidia.com

Abstract

In this work, we present a convolutional neural network (CNN) which can predict feature similarity index (FSIM) scores and it can be used to assess quality of the images. FSIM by itself needs a reference image to calculate a score; however, our approach is to estimate FSIM score for scenarios where no reference image is available. This task generally falls under the category of image quality assessment (IQA) which measures the visual quality of digital images. The data for CNN is generated by artificially distorting the images using different techniques like blurring, compression and unsharpening. We also experiment with different CNN architectures and report prediction accuracy for evaluating the models. We have been able to achieve 70% accuracy on validation and test datasets.

Keywords: No-reference image quality assessment, Convolutional Neural Networks

1. Introduction

Digital images and videos can be found everywhere today. It is one of the primary modes of communication and entertainment, thus it is very important to ensure high quality image is delivered to the end users. Image quality assessment (IQA) is an important area of research [1] because of its applications in:

- Monitoring Quality of Service (QoS) in internet streaming applications
- To identify level of image degradation which can affect image recognition accuracy
- In medical imaging to help decide compression ratio without loss of information

Human visual system can easily distinguish between good quality images versus bad ones even when a reference image is not available. Feature Similarity Index [2] (FSIM) tries to capture the quality of an image which is a close approximation to human perceived quality. This serves as a motivation for us to be able to predict FSIM color (FSIMc) scores in the absence of reference images. FSIM captures phase congruency (PC) which is a measure of local structure of the image is used as primary measure. The secondary feature is gradient magnitude (GM) and both PC and GM together provide an estimate of local quality of the image with respect to the original image.

Image quality assessment is a challenging task and considerable research has gone into understanding it [3]. It can be broadly divided into subjective and objective assessment. Subjective assessment is very manual and involves obtaining mean opinion scores [MOS] from human subjects. There are quite a few databases which have a collection of IQA data like LIVE [4], CSIQ [5], TID2008 [6] and TID2013 [7]. The number of images in these datasets is limited and also it is expensive to get more data. Objective assessment can be performed three different ways, full-reference (FR) [2, 8], reduced-reference (RR) [9] and no-reference [10] (NR). Full-Reference assessment is possible when a reference image is available for comparison and no-reference as the name indicates there is no reference image available for assessment. FSIM is one of the FR assessment techniques. Other examples of FR assessment are Structural Similarity Index (SSIM) [8], Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Visual Information Fidelity (VIF) [11]. All these methods have high correlation with human perception of quality. In this paper, we have chosen FSIM for no reference image quality assessment (NR-IQA) using CNNs.

Traditional approaches for NR-IQA are usually slow where features are extracted using Natural Scene Statistics approaches [16] by applying wavelet transform [10] or DCT transform [12]. Other approaches like CORNIA [13] and BRISQUE [14] learn features in spatial domain from raw image pixels which perform better in terms of computation time. On similar lines of extracting discriminant features, CNNs have shown extremely good performance for image recognition tasks in computer vision domain. Raw image pixels are used as input to the CNN and as part of the training process, they learn the relevant features. For image recognition tasks, CNNs learn high level representations by making use of deep structures. However, it is not clear whether a shallow network or deep network is preferred for IQA tasks, hence, we experiment with different network architectures for extracting both low and high level feature representations and use those features for image quality assessment.

This paper is further divided into the following sections:

In section 2, we present literature relevant to this work, in section 3 we discuss our data generating process, in section 4 we discuss our approach and sections 5 and 6 we present results and conclusion respectively.

2. Related Work

There are many previous papers on applying CNNs for NR-IQA tasks. Before CNNs, several researchers have used hand crafted features and neural networks for NR-IQA [15]. Hand crafted features included image gradients, phase congruency and entropy. One issue with this approach is that the features learnt are not part of the neural network training process. Deep learning based approaches for NR-IQA can also be found in [21] and [22].

The authors Bosse et al [17] and Kang et al [18] show that CNNs can be used for NR-IQA and they use LIVE dataset which has the MOS from subjective evaluations. The database has less than 1000 images and hence the authors in [17] use a patch-wise training process and aggregate the scores by taking the mean from each patch to estimate the overall score of the image. The authors in [18] apply local contrast normalization to images and claim that there is a 3% drop in performance if applied to the entire image. The main difference between [17] and [18] is the depth of the networks used where [17] uses a 10 convolutional layer network plus two fully connected layers whereas [18] uses a signle convolutional layer network with 2 fully connected layers. For this reason, we decided to investigate both shallow and deep network architectures. Both papers claim that the correlation with MOS is greater than 0.95.

Bianco et al [19] show that pre-trained CNNs on image classification can be tuned for IQA tasks. They also adopt a similar prediction pooling strategy where patches are used to extract five different grades of features. These features are then used in SVM for predicting the scores which are similar to MOS. This approach results in correlation of greater than 0.90 with MOS and the advantage of not having to train the CNN from scratch.

It is also worth mentioning that deep architectures are used not only in image domain but also for video quality assessment [20]. For this project, we will restrict ourselves to images.

In our work, we mainly want to address the problem of getting larger datasets for NR-IQA by artificially generating the data instead of depending on human subjective evaluations. We use FSIMc as a proxy for MOS since FSIMc is highly correlated with human perception. Due to time and resource constraints we will not be able to evaluate our model against LIVE or any of standard databases and report correlation with MOS. We also prepare different kinds of datasets with variations in distortions and number of classes.

3. Dataset

In this section, we will describe our data generation process. We used data from Imagenet database [25] which has images of varying sizes. All the images were scaled to 256x256 size to be consistent. Also, we used up to a maximum of 200 different class of images out of 1000 classes and each class has approximately 1300 images. This subset was chosen randomly. We did not require the class labels available for images in Imagenet. A total of 12 different types of distortions are applied randomly to each image. We also included images without distorting it. The distortion types include: MinFilter, MaxFilter, ModeFilter, MedianFilter, RankFilter, GaussianBlur, UnsharpMask, Kernel, JpegCompression, LocalisedBlur, Color2Gray2Color and WhiteBox. These distortions can be found in [23]. The parameters for the distortions were randomly drawn from a uniform distribution from 1 to 9 depending on the distortion type. After applying the distortions, we calculate the FSIM and FSIMc scores based on the original (or reference) image. Authors of FSIMc paper [2] have generously shared their MATLAB implementation for FSIM calculations. Since we are using color images, we only use FSIMc for training purposes in this paper. Once FSIMc is calculated we can discard the reference image and use distorted image for training, validation and testing. Note that the goal of this project is to predict the FSIMc scores. Due to time constraints, we simplify the prediction process to predict the integer value of FSIMc instead of continuous values. This will change our model formulation from using a regression model to classification model.

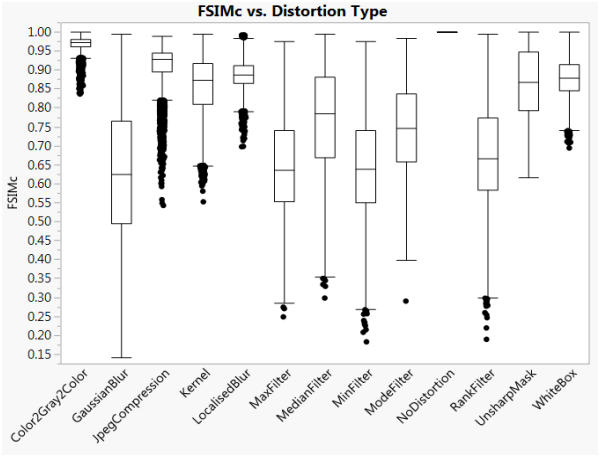

Figure 1: Distribution of FSIMc by Distortion Type

Data was generated in three different batches for analyses. The first batch was using 200 randomly chosen image classes. Different distortions were applied and after applying the distortion the images were broken down in 32x32 64 non-overlapping patches using image slicer library in python [24]. All 64 patches were assigned the

same label from the 256x256 image. Fig. 1 shows the distribution of FSIMc scores across each of the distortion types. This resulted in total sample size of 250,000 images.

The challenging aspect of this data generating process is it is not possible to generate data for a given FSIMc score. Hence, we cannot generate equal samples in the entire FSIMc range. FSIMc score of 1 indicates undistorted image and anything lower than 1 indicates some degree of distortion. Also, FSIMc scores below 0.8 results in heavily distorted images and all the colors are washed out. Fig. 2 shows the original at the left and JPEG compressed image on the right who's FSIMc score is 0.8. For all the experiments, the number of examples available for FSIMc scores less than 0.5 was very less. So, we discarded all the examples less than 0.5.

Figure 2: Original Image (left) vs. Jpeg compressed image (right)

The second batch of data was generated by first slicing the 256x256 images to 64 32x32 patches and then applying a subset of the distortions excluding the localized blurring and white box distortions. This was primarily generated to test how IQA works when CNNs are used on the entire image instead of patches. Also, this will help decrease the model run times since the size of the images is smaller. Based on the results from the experiments on the first two batches, we generated another batch of data to confirm the understanding of the results. Another reason for third dataset is because the LIVE dataset consists of JPEG compression, Gaussian blur, fast fading and white noise distortions. We applied two of the four distortions JPEG compression and Gaussian compression in a controlled manner so the FSIMc scores are in the range of 0.8 to 1. The other two were left out of this study in the interest of time. The second and third batches were created only using randomly chosen 10 classes in Imagenet. This yielded around 820,000 examples.

For data preprocessing, we used the per channel mean from the training dataset. This was subtracted from all the three training, validation and test datasets. The labels are rounded down to the nearest integer, scaled by 100 and subtracted by 100 so they have a nice interpretation starting from 0 to 100 with 0 being no distortion. A balanced data would therefore consist of equal number of sample in each class for each distortion type. All the three datasets were split into 80:10:10 ratios for training, validation and testing.

4. Convolutional Neural Network Model

A generic architecture of our CNN model is shown in Fig. 3. It starts with an input image or a patch extracted from an image and passed through convolutional layer(s), pooling layer(s) and fully connected layer(s). The last layer is an n-node fully connected layer which feeds into softmax cross-entropy loss function where n is the number of output classes. In the following subsections we define some of the architecture details that are applicable to across all the experiments.

4.1 ReLU Non-linearity

All the convolutional layers and fully connected layers except the last layer use ReLU nonlinearity function. This is the standard non-linearity function in most CNNs for two reasons: network trains faster and another is it reduces the effect of saturation of the neurons provided the learning rate is not set too high.

$$f(x) = \max(0, x)$$

4.2 Loss optimization

The optimization of loss function is done through ADAM with a learning rate of 0.0001. This worked well across all the experiments. Other optimizers increased training times but were able to yield similar results, so we decided to use ADAM for faster training times.

4.3 Regularization using Dropout

For the fully connected layers, we used a dropout probability of 0.5 during the training process. This has a regularization effect by preventing co-adaptation of neurons and also reduces the need for training an ensemble of models to some extent. We tried other dropout probabilities but 0.5 worked the best in all scenarios.

4.4 Pooling Layer

We experimented with and without pooling layers. Network with pooling layers trained faster by at least 2X and also provided better accuracy by at least 2% depending on the network architecture. In all experiments we used max pooling with 2x2 kernels and a stride of 2. Other kernel sizes and strides resulted in poor performance. We also tried average pooling but it did not yield good performance as compared to max pooling and the difference was around 2% in accuracy depending on the architecture.

4.5 Patch-wise training

We used a patch-wise training approach for the first batch of data as described in the previous section. The predicted result for each image consists of 64 values for

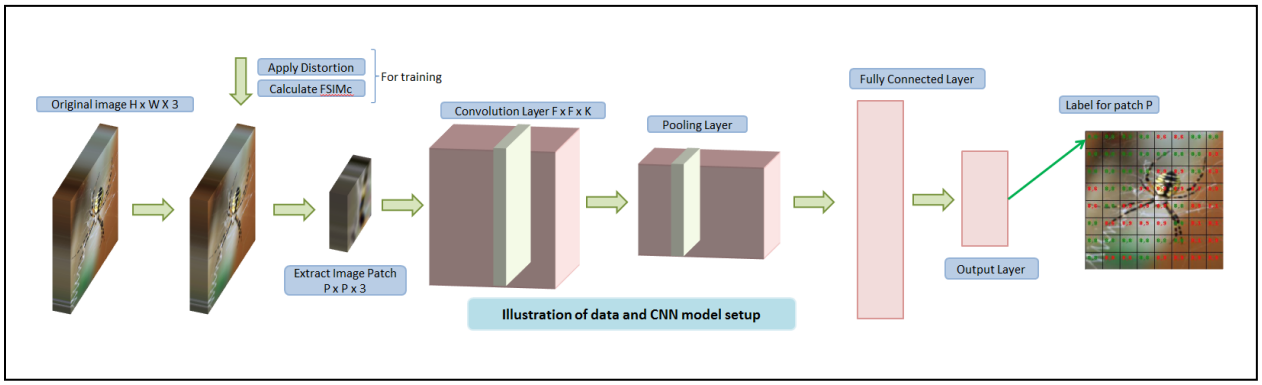

Figure 3: Generic CNN architecture for No Reference Image Quality Assessment

each of 64 32x32 patches. We used both mean and mode of the 64 values to assign the final label for the image. For second and third batches, we used the entire image and no aggregation of results was necessary.

5. Results

In this section, we will discuss the results from the experiments. We tried different approaches for NR-IQA by varying the datasets, network architecture and model hyperparameters. Depending on the dataset we restrict the number of epochs to a maximum of 100, batch size maximum of 512, number of filters to 64 and number of fully connected layers to 2. We also use batch normalization after the pooling layers. The models are implemented in Tensorflow [26]. After running several combinations of architectures described in Table 1, we finalized on conv5-64, maxpool, batch norm, FC-4096, FC-4096, FC-n where n is the number of classes. We used a padding of 2 for the convolutional layer. Unless specified the same architecture is used in all experiments.

Parameters	Value Range
Number of Filters	32, 64, 128
Filter Size (F)	3, 5, 7
Number of Conv Layers	1, 2, 4, 6
Number of FC Layers	1, 2, 3
Number of nodes in FC	256, 512, 1024, 248, 4096
Pooling	Max, Avg - 2x2, stride 2
Batch Normalization	Yes, No
Activation Functions	ReLU, Leaky ReLU
Dropout	0, 0.4, 0.5, 0.6
Padding	1, 2, 3
Stride	1, F
Learning Rate	0.1 - 0.00001
Batch Size	64 - 1024
Epochs	10 - 100

Table 1: Network parameters

5.1 Patch-wise Training with FSIMc 0.51-1 with 12 Distortion Types

Our preliminary runs indicated that the CNN model was not able to accurately predict FSIMc scores below 0.9 (we use inverted FSIMc scores, so this corresponds to greater than 9). The validation and test accuracy was at 5.5% but this is not relevant here because we are training on the patches. After aggregating the results from the patches using mode, the accuracy was around 19% and using mean, the accuracy was around 34%. The predictions from the test set Pearson correlation co-efficient to FSIMc scores is 0.35. Fig. 4 shows two examples where the model failed to predict the label for any patch as compared to another image where it got majority of the patches correct. The difference between the two examples is that one of them has higher distortion than the other.

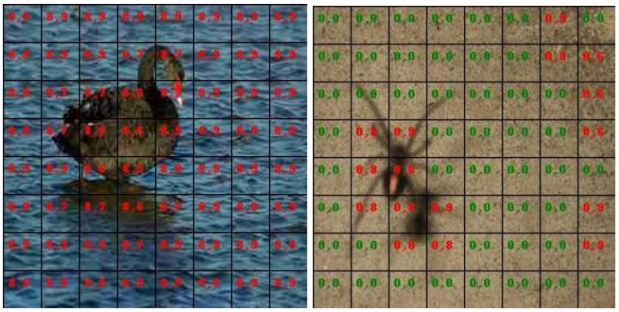

Figure 4: FSIMc = 0.92 JPEG Compression vs. FSIMc = 0.99 Unsharpened Image

Fig. 5 shows accuracy by distortion type. Fig. 6 shows the accuracy and loss of the training and validation datasets. The stopping criterion was based on when validation accuracy does not change over certain epochs but training accuracy increases. This will ensure that there is no over-fitting. Based on results in Fig. 5, our conclusion from this experiment is that CNN was not able to do a good job on localized distortions. This arises two questions: at what degradation values do accuracy decreases and the other is if we use local contrast normalization, will the model be able to predict localized distortions with higher accuracy. We will try to address the first question in this project and we will address the second one in our future work.

We also looked at heat map of each of the patches that were used in training and aggregated it based on

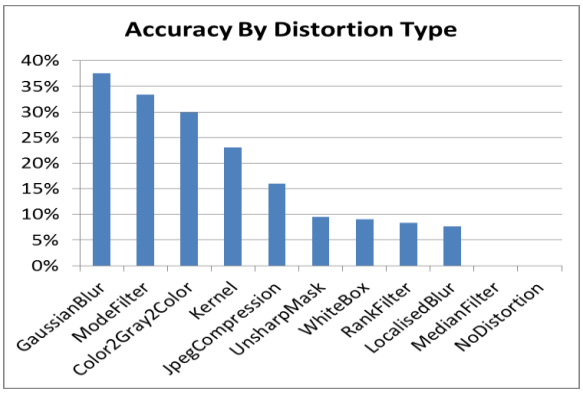

Figure 5: Accuracy by distortion type

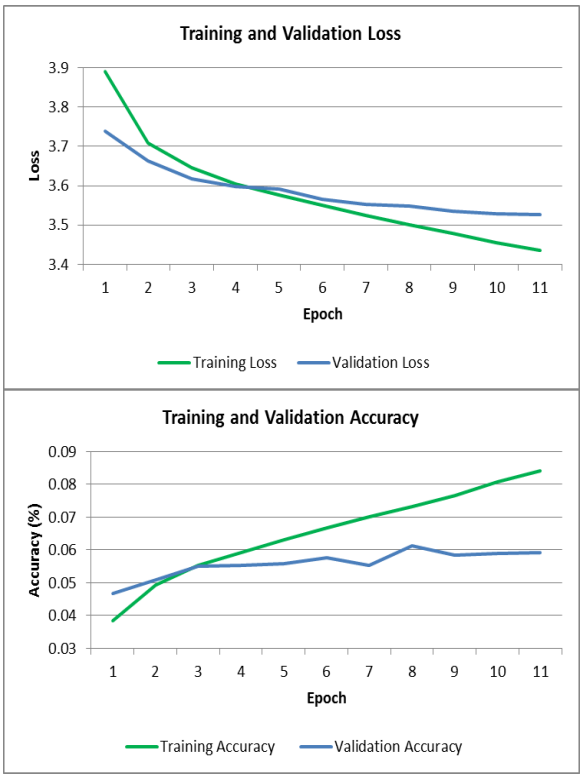

Figure 6: Traing and validation loss and accuracy for patch-wise training with FSIMc 0.51-1

percentage accuracy. It can be seen Fig. 7. This was to ensure that all patches had at least non-zero accuracy and CNN did not give preference to one part of the image versus another. The white regions show lowest accuracy percentage.

At this point, using the entire dataset would not have yielded any new insights. Hence, we decided to run all further experiments which show CNN model accuracy decreases as FSIMc decreases. This also helps run more experiments within the time constraints. We also resorted to using 32x32 images instead of 256x256 so data loading times are faster.

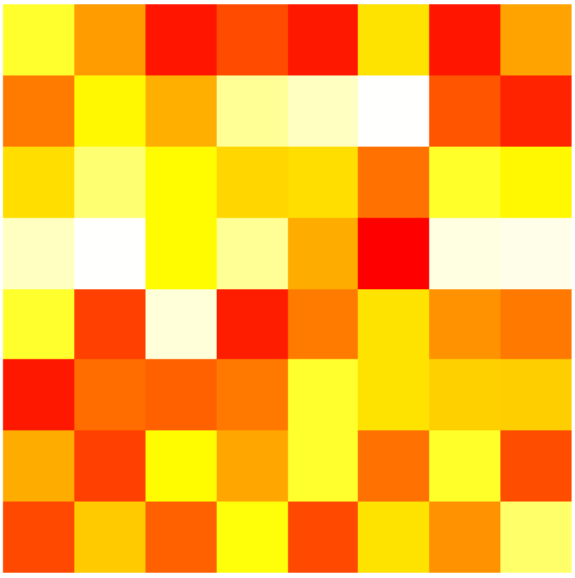

Figure 7: Heatmap of accuracy of 64 32x32 patches

5.2 Training on Whole Image with FSIMc 0.9-1 with 7 Distortion Types

In this set of experiments, we excluded the localized distortions since we were not looking at local contrast normalization method. This will allow us to study the effects of global distortions namely GaussianBlur, JpegCompression, Kernel, MedianFilter, ModeFilter, RankFilter and UnsharpMask.

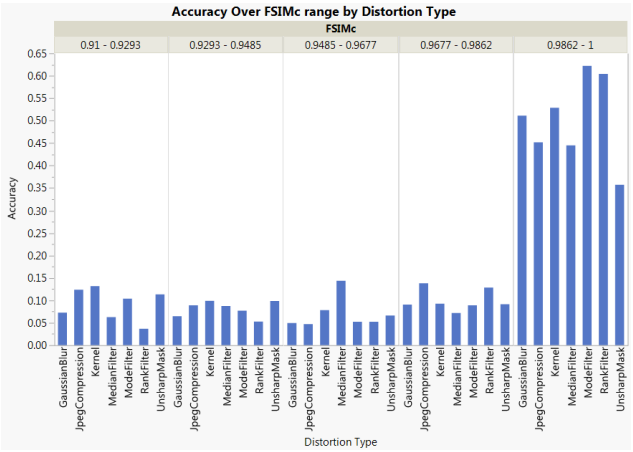

Figure 8: Accuracy over FSIMc range for different distortion types

Based on Fig. 8, we can notice that the accuracy of all FSIMc scores greater than 0.90 and less than 0.98 is around 10%. However, accuracy is around 55% for higher FSIMc scores. With this we can conclude that higher distortions in images are difficult to predict and does not depend on the type of distortion. Since LIVE database has Gaussian blur and JPEG compression distortions we wanted to investigate if it is possible to predict those

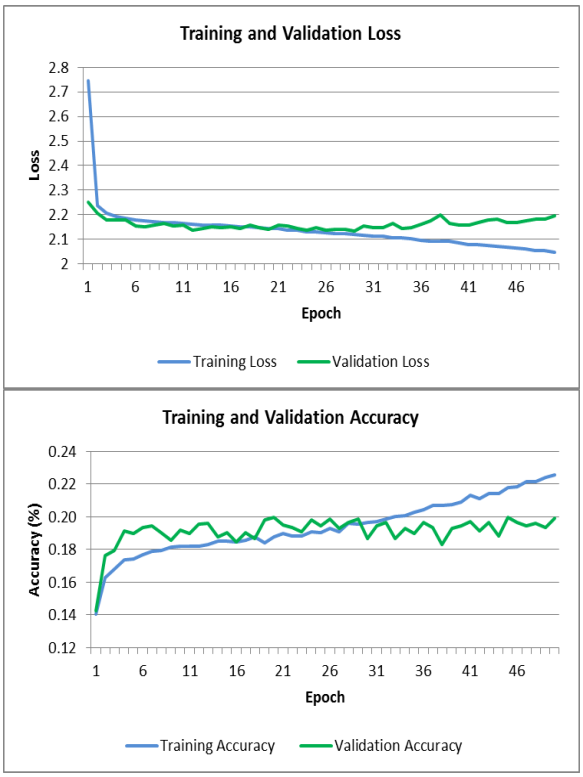

Figure 9: Training and Validation Loss and Accuracy for Whole Image with FSIMc 0.91-1 with 7 Distortion Types

distortions accurately. We will look into this in the next section. Fig. 9 shows the accuracy and loss curves for training and validation datasets. This model was easier to train than the previous dataset in section 5.1. The accuracy of validation and test set is around 20%.

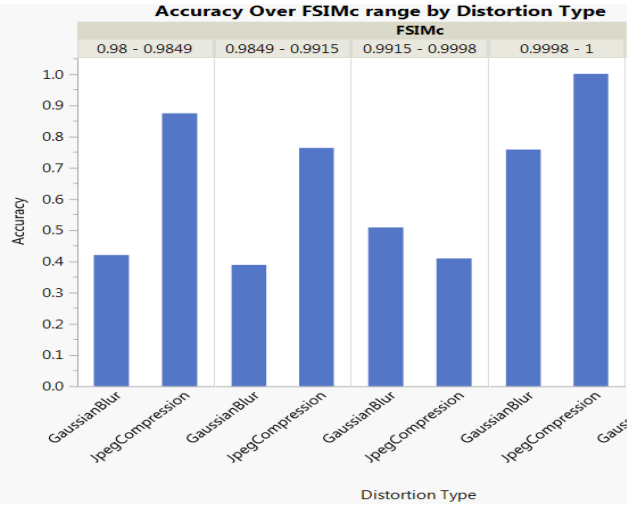

Figure 10: Accuracy over FSIMc range for JPEG compression and Gaussian blur

5.3 Training on Whole Image with FSIMc 0.98-1 with 2 Distortion Types

In this experiment we are trying to understand if the model accuracy improves if we have minimal distortion subjected to Gaussain blur and JPEG compression.

Figures 10 and 11 show that accuracy substantially improves to around 70% at lower distortion levels. This model can be trained further to improve the accuracy but we decided stop here to focus on other analysis tasks.

We also tried different grouping of FSIMc scores like increasing the range of FSIMc score in one group from 1 to 2 so that the algorithm can find better discrimintative features. But this improved the accuracy only by 2% on a 10 class dataset. Another experiement to confirm this behavior was by excluding all the examples with FSIMc scores greater than 0.98 because this is the bucket driving accuracy higher. This lowered the accuracy of the models by 50%. This confirms our hypothesis that this model does not perform well when images are heavily distorted.

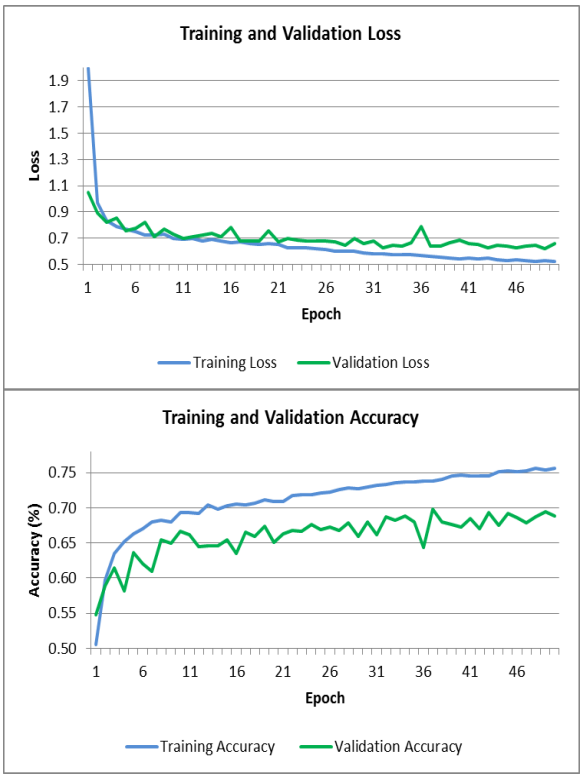

Figure 11: Training and Validation Loss and Accuracy for Whole Image with FSIMc 0.98 - 1 with 2 Distortion Types

6. Conclusion

In this paper, we applied CNN for NR IQA and explored sevral different CNN architectures. We were able to successfully apply CNN to demonstrate it is possible to predict FSIMc scores for distorted images without a reference image provided the level distortion is not too high. One of the questions that arose from literature was

whether we need a deep arhitecure versus a shallow architecture of CNN for IQA and we found that shallow architecture performs better than deeper architectures. This is intuitive in some sense that deeper architectures start learning the features which includes the objects in the image and the initial layers just learn edges and blobs of colors. For highly distorted images, we still need to explore the CNN architecutre so that discriminatory features can be created for classification task. Another approach that we did not explore in this study was using local constrast normalization which may be beneficial if the distortions are localised to a certain area of the image. We also want to test this model on some of the standard IQA databases like LIVE and TID2013 and evaluate the performance of the model.

References

[1] Z. Wang, "Applications of Objective Image Quality Assessment Methods [Applications Corner]," in IEEE Signal Processing Magazine, vol. 28, no. 6, pp. 137-142, Nov. 2011

[2] Zhang et al, "FSIM: A feature similarity index for image quality assessment", IEEE Transactions on Image Processing, 2011.

[3] Z. Wang, A. C. Bovik and L. Lu, "Why is image quality assessment so difficult?," 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, Orlando, FL, USA, 2002, pp. IV-3313-IV-3316.

[4] H. R. Sheikh, Z. Wang, L. Cormack, and A. C. Bovik, "LIVE image quality assessment Database Release 2," [Online]. Available: -http://live.ece.utexas.edu/research/quality

[5] E. C. Larson and D. M. Chandler, "Categorical image quality (CSIQ) database," [Online], Available: http://vision.okstate.edu/csiq

[6] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti, "TID2008-A database for evaluation of full-reference visual quality assessment metrics," Advances of Modern Radioelectronics, vol.10, pp. 30-45, 2009.

[7] N. Ponomarenko, O. Ieremeiev, V. Lukin, K. Egiazarian, L. Jin, J.Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti, and C.-C. Jay Kuo,"Color image database TID2013: Peculiarities and preliminary results,"4th European Workshop on Visual Information Processing EUVIP2013, pp.106-111, June 2013.

[8] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Trans. Image Process., vol. 13, no. 4, pp. 600-612, April 2004.

[9] G. Zhai, X. Wu, X. Yang, W. Lin, and W. Zhang, "A psychovisual quality metric in free-energy principle," IEEE Trans. Image Process., vol. 21, no. 1, pp. 41-52, January 2012.

[10] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: From scene statistics to perceptual quality," IEEE Trans. Image Process., vol. 20, no. 12, pp. 3350-3364, December 2011.

[11] H. R. Sheikh, A. C. Bovik, and G. de Veciana. An information fidelity criterion for image quality assessment using natural scene statistics. IEEE Transactions on Image Processing, 14(12):2117–2128, Dec. 2005.

[12] M. Saad, A. Bovik, and C. Charrier. Blind image quality assessment: A natural scene statistics approach in the DCT domain. IEEE Transactions on Image Processing, 21(8):3339–3352, Aug. 2012.

[13] P. Ye, J. Kumar, L. Kang and D. Doermann, "Unsupervised feature learning framework for no-reference image quality assessment," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 1098-1105.

[14] A. Mittal, A. Moorthy, and A. Bovik. No-reference image quality assessment in the spatial domain. IEEE Transactions on Image Processing, 21(12):4695–4708, 2012.

[15] C. Li, A. C. Bovik and X. Wu, "Blind Image Quality Assessment Using a General Regression Neural Network," in IEEE Transactions on Neural Networks, vol. 22, no. 5, pp. 793-799, May 2011.

[16] H. R. Sheikh, A. C. Bovik, and L. Cormack, "No-reference quality assessment using natural scene statistics: JPEG2000," IEEE Trans. Image Process. , vol. 14, no. 11, pp. 1918–1927, Nov. 2005.

[17] S. Bosse, D. Maniry, T. Wiegand and W. Samek, "A deep neural network for image quality assessment," 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, 2016, pp. 3773-3777.

[18] L. Kang, P. Ye, Y. Li and D. Doermann, "Convolutional Neural Networks for No-Reference Image Quality Assessment," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 1733-1740.

[19] Bianco et al, "On the Use of Deep Learning for Blind Image Quality Assessment", arXiv 2016.

[20] M. T. Vega, D. C. Mocanu, J. Famaey, S. Stavrou and A. Liotta, "Deep Learning for Quality Assessment in Live Video Streaming," in IEEE Signal Processing Letters, vol. 24, no. 6, pp. 736-740, June 2017. doi: 10.1109/LSP.2017.2691160

[21] Gu et al, "Deep learning network for blind image quality assessment", IEEE International Conference on Image Processing, 2014.

[22] Hou et al, "Blind Image Quality Assessment via Deep Learning", IEEE Transactions on Neural Networks and Learning Systems, 2015.

[23] https://pillow.readthedocs.io/en/4.1.x/reference/ImageFilter.html

[24] http://image-slicer.readthedocs.io/en/latest/

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database.In CVPR, 2009

[26] https://www.tensorflow.org/, Tensorflow implementation adapted from CS231n assignment 2