Image Distortion Classification With Deep CNN Final Report

Mike Hsieh - Roland Duffau - Alex He
{mhsieh33, rduffau, yuzehe}@stanford.edu

1 Introduction

Image distortion is a well known challenge in image processing and computer vision. Today, billions of images are uploaded and exchanged over the Internet daily, but the quality of these images vary significantly. Depending on the application and context, distorted images can be useless.

One of the largest trend of experiments related to image distortion is called Image Quality Assessment (IQA). It refers to the process of measuring the weighted combination of all of the visually significant attributes of an image, and particularly the distortions that affect it. Almost all related methods tend to compute a score for the distortion, reflecting the perceptual quality of digital images in a manner that is consistent with human subjective opinions.

Yet this approach has two main limitations: first, it requires collecting human perceptions of image quality which often adds a strong bias to the experiment. Second, it doesn't specifically name the distortions which affect the image.

Being able to identify precisely the type of distortion which affects an image would have several benefits: simplifying the restoration process by advising the right method, helping develop novel denoising schemes specific to one type of noise, and better classifying image datasets to avoid bias introduced by some distortions.

We thus propose an approach to classify the types of distortions present in an image. To that extent, we trained different deep Convolutional Neural Networks (CNNs) over dedicated datasets. We assessed their performance over commonly used public IQA dataset, and performed error analysis to improve our models. Finally, we identified next steps to scale our project.

2 Related work

Over the past 10 years, several algorithmic methods have been developed to score image quality [1] [2]. Some of the most recent approaches leverage deep CNN [3] [4] [5] [6].

Many of these past experiments have been conducted to build full-reference (FR) or no reference (NR) image quality assessment (IQA) algorithms based on deep CNN. The CNN takes image patches as an input and estimates the quality with (for FR) or without (for NR) the help of pristine reference images. All of these methods output a score, measuring either the distortion level (PSNR, MSE, SSIM, MS-SSIM, IFC, VIF, VGG2.2), the perceptual quality (NIQE, BRISQUE, Ma et al), or even a trade-off between both [7] for the latest ones. None of them outputs the exact list of distortions that affect the image.

Aside, several image reconstruction techniques exist. The most recent experiments tend to leverage Generative Adversarial Networks or similar techniques to recreate the original high quality image, whatever distortion affect it. But the most generalized ones are specific to one type distortion: image deblurring [8] [9] and image denoising [10].

CS230: Deep Learning, Winter 2020, Stanford University, CA.

Such methods use well know spatial filters (Wiener filtering [11], Bilateral filtering [12], PCA method [13], Wavelet transform method [14], BM3D [15], LRA-SVD [16], WNNM [17]) or more recently CNNs (DnCNN [18] and FFDNet [19]) with variable efficiency depending on the type of noise affecting the image. It could thus be useful to know precisely which types of noise are present.

3 Dataset and Features

We created two datasets for training our models - one to identify various different distortions and one focused on only two distortions which generalizes to external datasets. In the first dataset, we downloaded ImageNet [20] images and applied six different distortions with constant distortion parameters on each image as shown in Figure 1. In our second dataset, for each ImageNet image, we generated four Gaussian blur and four non-monochrome Gaussian noise images using variable distortion parameters. This variety of distortion levels allowed our model to learn a wider variety of blurry and noisy images. We wrote a java program utilizing image processing filters from JH Labs [21]. Next, we standardize the dimension of each image to 224 x 224 pixels. We followed a similar approach to image standardization as [22]. We first rescaled the image so that the shorter side was 224 pixels and then cropped out the central 224 x 224 segment. Through this process we generate 200k images in each dataset which we randomly shuffle and split 95% into the training, 2.5% into the dev, and 2.5% into the test set.

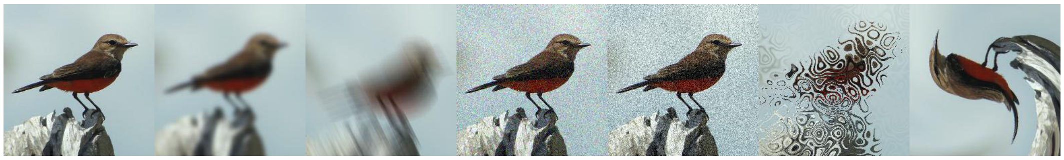


Figure 1: Original image, Gaussian blur, motion blur, non-monochrome Gaussian noise, monochrome Gaussian noise, marble, and twirl

Additionally, we referenced 4 public datasets commonly used to evaluate and benchmark IQA models [23] [24] [25] [26]. These datasets have a limited number of images for each distortion (less than 1000), hence can't be used for model training. One of our objectives was to get our trained classifier to generalize to these external images. As we focused mostly on two types of distortions (blur and white noise), we used the LIVE dataset as the reference, given it includes these 2 distortion types and it is the most quoted in the IQA literature.

In this (unbalanced) LIVE dataset, 29 reference images are used to generate distorted images with the following distortion types: JPEG2000, JPEG, White noise in the RGB components (145 images), Gaussian blur in the RGB components (145 images), and bit errors in JPEG2000 bitstream when transmitted over a simulated fast-fading Rayleigh channel.

4 Method

As our initial baseline, we applied optimization to a simple one layer softmax regression model. This model represents the classification potential of a shallow architecture.

Next, we trained three separate deep CNNs on our generated datasets.

The first model was trained from scratch and utilizes an architecture based on the VGG16 neural network [27]. The VGG16 model consists of 5 blocks of a series of convolutional then max-pooling layers stacked sequentially. The first two blocks each contain two convolutional layers followed by a max-pooling layer, and the next three blocks each contain three convolutional layers followed by a max-pooling layer. We take the network up to and including the fourth max pooling layer from the original model, and stack one convolutional layer, one max-pooling layer, and one fully connected layer before the final output layer. The final layer uses softmax activations for predictions. The weights for all layers in the baseline model are randomly initialized and trained on the training dataset.

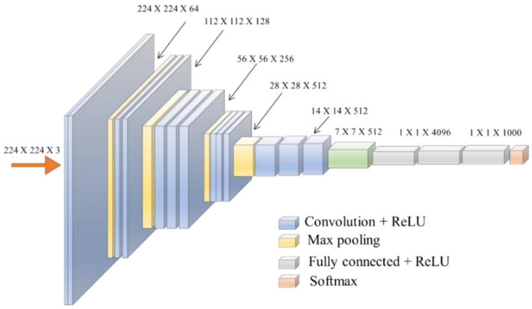


Figure 2: VGG16 Architecture

The second model is a transfer learning model. The architecture is shown in Figure 2. We have used the pre-trained weights from ImageNet in a VGG16 neural network. The VGG16 model is the same as described above, and we also add three layers at the end: one convolutional layer, one max-pooling layer, and one fully connected layer before the final output softmax layer. Only the final three layers at the end is fine-tuned (105,136,643 trainable parameters) while keeping pre-trained weights for the retained VGG16 layers fixed. With both of these models trained, we are able to compare the performance of transfer learning using ImageNet pre-trained weights versus our model trained from scratch.

The third model also resorts to transfer learning, but is based on a model pre-trained specifically for IQA. We chose the SGDNet model [6] because it is a recent model, it has high accuracy as compared to the state-of-the-art approaches, and it is complementary to VGG16. The architecture is shown in Figure 3: it is based on a ResNet50 neural network reinforced by a saliency prediction sub-network, both sharing a feature extractor for final score prediction. We tried several updates to this original architecture, and the one that showed the best results is the following: we removed the last four layers of this network (2 fully connected layers, 1 dropout and the final softmax), and added a final output softmax-3 layer in place. Only the final layer was fine-tuned (1,539 trainable parameters), while keeping pre-trained weights for the retained SGDNet layers fixed.

As we trained our models, the hyperparameters we tuned were the optimizer algorithm, learning rate decay, batch size, and the number of layers to append to our transfer learning original model. We tried RMSProp and Adam optimizer and settled on using RMSProp. Additionally we had to increase the learning rate decay to ensure the training remained stable. We trained the models using mini-batch sizes ranging from 128 to 256 depending on GPU memory. Lastly, we experimented with adding more convolution layers and dropout layers in our transfer learning appended layers though we found that having less layers was more stable during training.

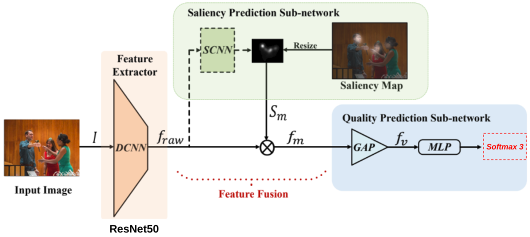


Figure 3: SGDNet Transfer Learning Architecture

We used the categorical cross entropy loss function $-\sum_{k=1}^{K} y_k \log(\hat{y}_k)$ which is consistent with our multi-class classification task. We tried input data normalization but it didn't show major impact.

5 Results

The table below shows our results using our first generated dataset across 6 different distortion types with constant distortion paramaters:

Model	Validation Acc.	Validation Loss	Test Acc.	LIVE Acc.
Softmax	0.243	5464372.5	0.243	-
VGG16 scratch	0.989	0.0348	0.988	0.484
VGG16 transfer learning	0.991	0.255	0.992	0.170
SGDNet transfer learning	0.872	3.15	0.887	0.102

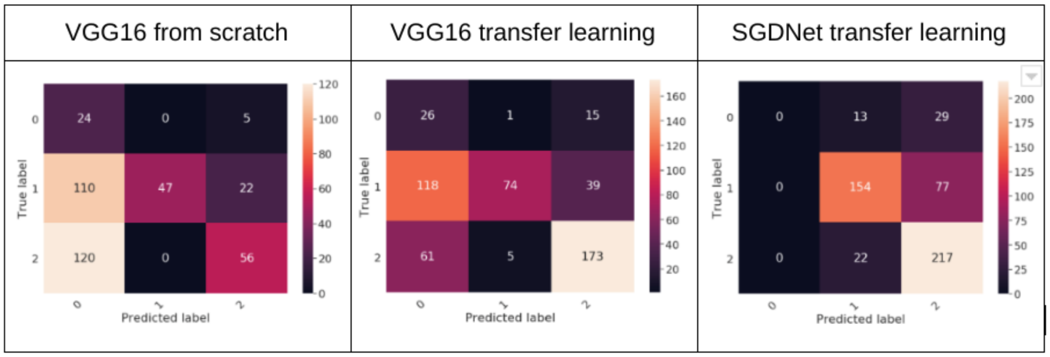
It is evident that the softmax model does not classify the distortions well. Our deep CNN model trained from scratch performed similarly to the fine-tuned network pre-trained on ImageNet images. Both models achieved around 99% accuracy. The IQA model fine tuned for our dataset performed quite efficiently as well.

However, when evaluating our 4 models on the LIVE dataset, we could notice that test accuracy dropped dramatically. This reflects the fact that our models are overfitting to our generated distortions, which don't reflect correctly the real world distribution (data mismatch).

We thus continued with our second dataset, which had variable distortion levels for only two distortion types. This more closely reflected the distribution of real data (both IQA datasets, and real world images since they present diverse levels of distortion). The table below shows our results.

Model	Validation Acc.	Validation Loss	Test Acc.	LIVE Acc.
Softmax	0.450	9997.60	0.455	-
VGG16 scratch	0.864	0.37	0.856	0.661
VGG16 transfer learning	0.852	0.79	0.850	0.40
SGDNet transfer learning	0.869	0.54	0.837	0.724

Our models performed slightly worse on this second generated dataset, but generalized much better to the LIVE dataset. Figure 4 shows the corresponding confusion matrix for our 3 CNNs (label 0 corresponds to reference images, label 1 to blurred version, and label 2 to white noise distortion) :


VGG16 from scratch

VGG16 transfer learning

SGDNet transfer learning

Figure 4: Confusion matrix of our three CNNs on LIVE dataset

We then proceeded to error analysis to better understand how performance might be improved further. Computing the Precision, Recall and F1 Score for each class and each model showed that the models performed the worst on class 0 (reference images.) As this class is relatively less represented in the dataset, this weak performance may have been due to imbalanced data. Additionally, it is possible that the reference images themselves had innate blurriness or noise. The following table shows the results for the SGDNet transferred model:

Class	Precision	Recall	F1 Score
0 - Reference Images	0	0	0
1 - Uniform Blur	0.815	0.667	0.733
2 - White Noise	0.672	0.908	0.772

We could also check that, unsurprisingly, predictions were correlated to the level of distortion severity. The higher the distortion severity, the better was the prediction. Wrong predictions occurred mostly on images with low distortion level, harder to distinguish from reference images or from scarcely distorted images of the other class. The following table illustrates this correlation for the VGG16 scratch model, by showing the relative distortion level of predicted images compared to the dataset average for this same distortion type (for example, correctly predicted classes are > 100% distortion vs the dataset average):

True class	Predicted Class 0	Predicted Class 1	Predicted Class 2
1 - Uniform Blur (Radius)	22.16%	105.56%	28.32%
2 - White Noise (Amount)	23.11%	32.45%	104.06%
2 - White Noise (Density)	79.07%	78.48%	102.44%

This is also clearly illustrated by calculating the structural similarity index metric. This allows us to examine how our models perform when the extent of image distortion becomes more severe, which would check the robustness of our model against various distortions.

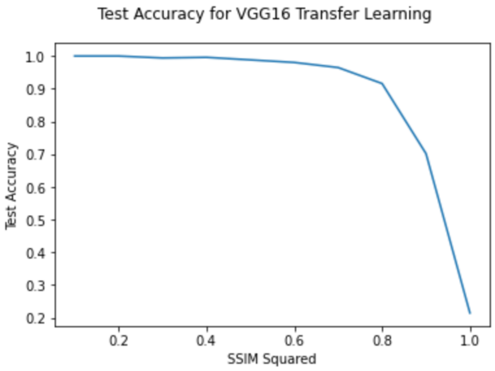


Figure 5

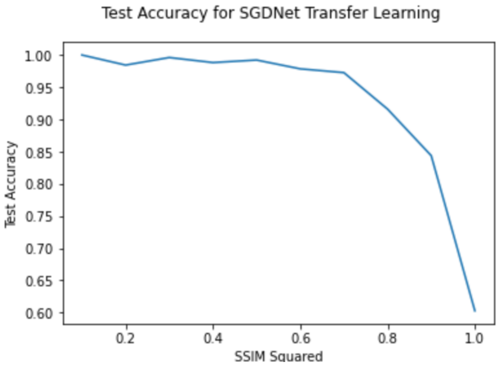


Figure 6

As shown in figures above, the accuracy of our predictions clearly worsened as images become more similar as measured by SSIM. A larger SSIM indicates more similar with the reference image and a SSIM of 1 indicates identical image.

6 Conclusion

Overall we managed to build classifiers predicting the main type of distortion (starting with Blur and White Noise) affecting an image.

Yet as real world images may present multiple types of distortion at a time, the multiclass classifiers we built may not be enough. As a next step, we'd be keen to extend our model to a multi-label classifier, being able to predict properties of a sample that are not mutually exclusive. To that extent, we may start by extending our classifier with the Sklearn MultiOutputClassifier [28].

It is also more difficult to make accurate predictions when the image distortions are not severe as shown by our previous analysis. Images with only slight distortions require disproportionately more attention in order to achieve better accuracy. It may be helpful to fine-tune further with more images with only slight distortions (and even potentially create dedicated classes).

If we had more time, we would also assess the human level performance on our datasets. To do so, we would ask a wide diversity of persons to provide their own perception of the level of distortion affecting image samples. This would allow us to assess if our test accuracy is acceptable or not.

Finally, we need to test our models more extensively on the distorted images from external datasets. We'd like to ensure that our models are robust and that the distribution of our generated dataset is indeed a good proxy for the distribution of broader real world datasets.

7 Contributions

Michael Hsieh generated the two datasets, developed the training environment, trained the baseline softmax model, the VGG16 transfer learning model, and the VGG16 scratch model. Alex He maintained AWS, trained VGG16 baseline model, implemented the structural similarity index metric for evaluation and contributed to all report write-ups. Roland Duffau studied the related work, selected SGDNet model among multiple other IQA references, transferred it, evaluated and performed error analysis of all 3 models.

We'd also like to thank Advay Pal for his guidance to help us plan and execute the project.

8 Code

The code is available at https://github.com/mhsieh33/distortionClassfication .

References

[1] Alan Conrad Bovik Anish Mittal, Anush Krishna Moorthy. No-reference image quality assessment in the spatial domain (brisque). 08 2012.

[2] Guopu Zhu Fu-Zhao Ou, Yuan-Gen Wang. A novel blind image quality assessment method based on refined natural scene statistics. 08 2019.

[3] Klaus-Robert Muller Thomas Wiegand Wojciech Samek Sebastian Bosse, Dominique Maniry. Deep neural networks for no-reference and full-reference image quality assessment. 12 2017.

[4] Kai Zhang Zhengfang Duanmu Zhou Wang Wangmeng Zuo Kede Ma, Wentao Liu. End-to-end blind image quality assessment using deep neural networks. 11 2017.

[5] Sanghoon Lee Jongyoo Kim. Deep learning of human visual sensitivity in image quality assessment framework (deepqa). 11 2017.

[6] Weisi Lin Yongtao Wang Sheng Yang, Jiang Qiuping. Sgdnet: An end-to-end saliency-guided deep neural network for no-reference image quality assessment. 10 2019.

[7] Yochai Blau and Tomer Michaeli. Perception - distortion tradeoff. *ArXiv e-prints*, 2019.

[8] Leida Li, Ya Yan, Yuming Fang, Shiqi Wang, Lu Tang, and Jiansheng Qian. Perceptual quality evaluation for image defocus deblurring. 09 2016.

[9] Wei-Sheng Lai, Jia-Bin Huang, Zhe Hu, Narendra Ahuja, and Ming-Hsuan Yang. A comparative study for single image blind deblurring. 06 2016.

[10] Zhang F. Fan H. et al. Fan, L. Brief review of image denoising techniques. 7 2019.

[11] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., USA, 1989.

[12] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. pages 839–846, 1998.

[13] D.D. Muresan and T.W. Parks. Adaptive principal components and image denoising. volume 1, pages I – 101, 10 2003.

[14] Stéphane Mallat. Mallat, s.g.: A theory of multiresolution signal decomposition: The wavelet representation. ieee trans. pattern anal. machine intell. 11, 674-693. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11:674 – 693, 08 1989.

[15] Katkovnik V Egiazarian K. Dabov K, Foi A. Image denoising by sparse 3-d transform-domain collaborative filtering. 08 2007.

[16] Qiang Guo, Caiming Zhang, Zhang Yunfeng, and Hui Liu. An efficient svd-based method for image denoising. *IEEE Transactions on Circuits and Systems for Video Technology*, 26:1–1, 01 2015.

[17] Shuhang Gu, Qi Xie, Deyu Meng, Wangmeng Zuo, Xiangchu Feng, and Lei Zhang. Weighted nuclear norm minimization and its applications to low level vision. *International Journal of Computer Vision*, 121, 07 2016.

[18] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, PP, 08 2016.

[19] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn based image denoising. *IEEE Transactions on Image Processing*, PP, 10 2017.

[20] Imagenet. `http://www.image-net.org/`.

[21] Jh labs java image processing. `http://www.jhlabs.com/ip/distorting.html`.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[23] Laboratory for image and video engineering. `https://live.ece.utexas.edu/research/Quality/subjective.htm`.

[24] Csiq image quality database. `http://vision.eng.shizuoka.ac.jp/mod/page/view.php?id=23`.

[25] Tampere image database 2008 tid2008, version 1.0. `http://www.ponomarenko.info/tid2008.htm`.

[26] Tampere image database 2013 tid2013, version 1.0. `http://www.ponomarenko.info/tid2013.htm`.

[27] Andrew Zisserman Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *IEEE Transactions on Image Processing*, PP, 09 2014.

[28] Multioutputclassifier. `https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputClassifier.html#sklearn.multioutput.MultiOutputClassifier`.