NEURAL IMAGE ENHANCEMENT AND ASSESSMENT

Abhishek Sunnak
Department of Computer Science
Simon Fraser University
Burnaby, BC V5A1S6
asunnak@sfu.ca

Sri Gayatri Rachakonda
Department of Computer Science
Simon Fraser University
Burnaby, BC V5A1S6
grachako@sfu.ca

Padmanabhan Rajendrakumar
Department of Computer Science
Simon Fraser University
Burnaby, BC V5A1S6
rajendra@sfu.ca

Vijaytha Vaasan Srinivasan
Department of Computer Science
Simon Fraser University
Burnaby, BC V5A1S6
vvsriniv@sfu.ca

Hang Hu
Department of Computer Science
Simon Fraser University
Burnaby, BC V5A1S6
hha107@sfu.ca

Abstract

In the recent years, there have been many deep neural networks proposed for image enhancement (super-resolution). An image enhancement pipeline typically involves the minimization of a loss function between enhanced and reference images using L1/L2 loss functions. However, most models either target increasing just the PSNR score or the aesthetic quality score of an image but fail at striking a balance between the two. For this project, we have studied the effect of using a hybrid loss function, which tries to enhance an image while also maximizing its aesthetic appeal. The proposed method uses an aesthetic quality predictor network along with pixel by pixel difference between the enhanced image versus the reference image to train deep neural networks for image enhancement. We have used EDSR and MDSR[1] as our baseline image enhancement modules. Training the modules with our modified loss function for just 25 epochs shows similar quality (PSNR) and aesthetic score (NIMA[2]) to the baseline models, which were trained for 300 epochs. Further training of the models would lead to improving the NIMA score of the output images compared to the results of the baseline models.

1 Introduction

Image enhancement deals with improving the quality of images by removing noise, reducing blur, sharpening the edges etc. These problems have been gaining a lot of attention from the research community for years. The desire for image enhancement primarily stems from a desire to improve the pictorial information of images for human interpretation. In this project, we focus on two major image enhancement problems:

- **Super-resolution:** Image resolution describes the details contained in an image: the higher the resolution, the more details in an image. Super-resolution techniques reconstruct a higher-resolution (HR) image from the observed Low-Resolution (LR) images.
- **Image De-blurring:** Image blurring is a common occurrence, which can happen due to various factors such as taking a photo while moving or due to wrong autofocus. De-blurring techniques aim to take a blurry image and achieve as close an approximation to the ground truth image as possible. The information contained in blurry images is insufficient and de-blurring techniques such as deconvolution and unmasking allow as to retrieve this information.

1.1 Related Work:

Deep learning is being extensively used to predict the technical quality of images by extracting high level features using CNN's. Dong et al. [3] defines an image super-resolution model based on deep learning using SRCNN. The model uses an image upscaled by bicubic interpolation and enhances it using two convolutional layers. Recently, residual blocks having shortcut connections are being commonly used for image enhancement. Lim et al. [4] developed two super-resolution models for the NTIRE 2017 single-image super-resolution challenge: EDSR and MDSR. They improved on previous work by removing batch normalization and blending outputs generated from geometrically transformed inputs. Hossein and Peyman [5] developed a Learned Perceptual Image Enhancement architecture which used a hybrid loss function to enhance image quality as well as aesthetic appeal. They trained a multi-scale context aggregation network (CAN) which used dilated convolutions to aggregate global contextual information.

1.2 Our Contribution:

1. Traditionally, image enhancement involves minimization of difference between generated and reference images using L1 and L2 loss functions. We have used a modified cost function [5], which minimizes L2 loss while improving the overall aesthetic quality of an image, to train state of the art image super resolution models (EDSR and MDSR). Our models show comparable performance to baseline models with significantly lesser training epochs.
2. We also trained EDSR and MDSR for image de-blurring using the hybrid cost function to evaluate the performance of these models for image enhancement tasks other than super resolution.
3. The aesthetic quality of an image is scored by a neural network for neural image assessment (NIMA) during the training of the image enhancement modules. The baseline NIMA architecture is implemented on Mobilenet[6]. We trained NIMA on MnasNet [7], a state of the art neural net architecture for mobiles, which is more accurate and faster compared to MobileNet and MobileNetV2. This allows a significant decrease in the training time of our image enhancement models. We achieved a validation loss of 0.07 using MnasNet as compared to a validation loss of 0.08 using MobileNetV2 and MobileNet.

1.3 Architecture:

The proposed method involves feeding input and reference image pairs to the image enhancement module. The loss of the training module is defined by a perceptual loss which comprises of an L2 loss and an aesthetic loss score predicted from the above-mentioned image assessment network. The L2 loss is calculated by comparing pixel-by-pixel values of the output generated from the module and the reference image. The aesthetic loss score is a no-reference quality metric and requires only the output from the enhancement network. The importance of the aesthetic quality score is determined by the ϒ value.

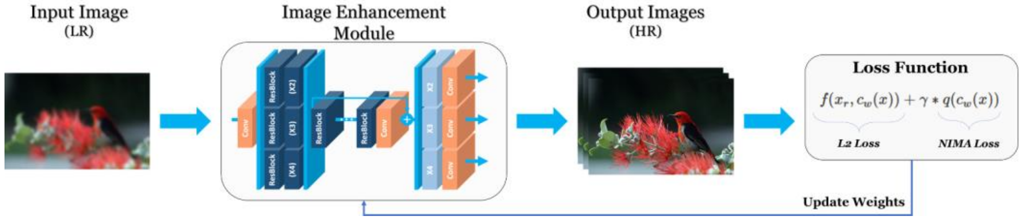

Figure 1: Image Enhancement Framework

2 Approach

2.1 Datasets:

- **Aesthetic Visual Analysis Dataset (AVA[8]):** The AVA dataset was used to train the NIMA module. The dataset consists of 255,000 images which are aesthetically rated by amateur photographers. Each image was scored by an average of 210 people across approximately 900 different challenges. The images were rated between 1 to 10, with 10 being the highest aesthetic score. The annotations have a high intrinsic value because they capture the way hobbyists and professionals understand visual aesthetics. The average ratings of the images are around 5.5 with a mean standard deviation of 1.4.
- **DIV2K Dataset[9]:** The DIV2K dataset was used to train the Image Enhancement models. The DIV2K data is a high-quality collection of 1000 images, which can be used for image enhancement tasks. The data was split into 800 training images, 100 validation images, and 100 test images. The performance of the models is compared on the validation dataset with the ground truth images

(reference images). Before training EDSR and MDSR for image de-blurring, a gaussian blur was applied to the low-resolution images. All the images are saved as binary files to improve the model training speed.

2.2 Image Assessment (NIMA) [2]:

The image assessment module was based on the image quality assessment architecture proposed by Hossein Talebi and Peyman Milanfar in their paper about Neural Image Assessment [2]. Various classifier architectures used for object detection, such as VGG16, Inception-v2, MobileNet were explored in the paper. The paper states that Inception-v2 provides slightly better results among the explored architectures, however it is much slower to train than MobileNet as it has significantly more parameters. As the image assessment module is used in this paper as a loss function for the image enhancement module, efficiency is an important attribute.

MnasNet [7] is recently developed CNN architecture for mobiles with significantly better efficiency than MobileNet and MobileNetV2. When compared to the MobileNetV2 [6] architecture, improves the ImageNet top-1 accuracy by 2% with the same latency on Pixel phone. Also, MNasNet is 1.5× faster than MobileNetV2 and 2.4x faster than NASNet with the same top-1 accuracy.

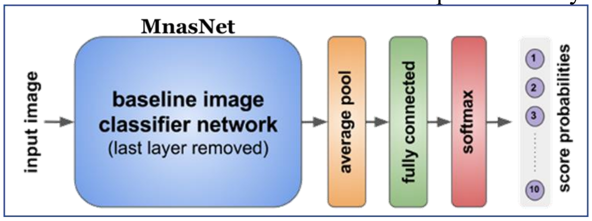

Figure 2: Neural Image Assessment (NIMA) framework

A pre-trained MnasNet model [7] was used as the baseline architecture for NIMA with the last layer of the baseline CNN replaced with an average pooling layer followed by a fully connected layer with 10 neurons. The Fully-connected layers in the baseline CNN are implemented by convolutional layers. We can feed images of arbitrary dimensions to the baseline CNN due to the addition of the fully convolutional layers along with average pooling before the final layer. This design allows backpropagating from quality score to input pixels. The baseline CNN weights are initialized by training on the ImageNet dataset, and the last fully-connected layer is initialized randomly. All NIMA weights are found by retraining on the AVA dataset.

2.2.1 Training NIMA

The main objective during training NIMA is to predict the distribution of the quality ratings for a given image in the AVA dataset. The ground truth distribution of human ratings of a given image is expressed as an empirical probability mass function p = [p_{s1} , … ,p_{si},.. , p_{sN}] with s_1 <= s_i <=s_N, where s_i represents the ith score bucket, and N denotes the total number of score buckets. The probability of ratings are given as $\sum_{i=1}^{N} p_{si} = 1$ where, p_{si} represents the probability of a quality score falling in the i^{th} bucket. In the AVA dataset, N = 10, s_1 = 1 and s_N = 10. Each AVA example image is assigned a set of ground truth (user) ratings p. The objective of the training is to find an accurate estimate of the probability mass function p.

A squared Earth Mover's Distance-based (EMD) loss was used to train NIMA. EMD penalizes mis-classifications according to class distances in contrast to cross-entropy loss, which ignores inter-class relationships. EMD is defined as the minimum cost to move the mass of one distribution (ground truth probability mass function **p**) to another (estimated probability mass function $\hat{\mathbf{p}}$). With N ordered classes of distance ||si -sj||l, the normalized Earth Mover's Distance can be defined as:

$$EMD(\boldsymbol{p},\hat{\mathbf{p}}) = \left(\frac{1}{N} * \sum_{k=1}^{N} \left|CDF_p(k) - CDF_{\hat{\mathbf{p}}}(k)\right|^l\right)^{1/l}$$

The mean NIMA score of an image is given by:

$$NIMA\ Score = \sum_{i=1}^{10} i * p_i$$

Where, i denotes a score bucket between 1 and 10 and pi is the probability of the quality score falling in the i^{th} score bucket.

The AVA dataset is divided to 80% training and 20% test data with a batch size of 128 and trained for 25 epochs. All images are scaled to 256 * 256 pixels, then the images are randomly cropped by 192 *

192 pixels. The baseline MnasNet is initialized by training on ImageNet, and the last fully connected layer is randomly initialized. An Adam optimizer is used with an initial learning rate of 0.0001 and an exponential decay to find the optimal learning rate.

2.3 Image Enhancement

Image enhancement deals with improving the quality of images by removing noise, reducing blur, sharpening the edges etc. It encompasses various techniques such as image super-resolution, deblurring, denoising and contrast improvement. Super-resolution aims to construct a high resolution image from a low-resolution one. Generally, the relationship between ILR and the original high-resolution image I^{HR} can vary depending on the situation. Many studies assume that ILR is a bicubic down-sampled version of I^{HR}, but other degrading factors such as blur, decimation, or noise can also be considered for practical applications.

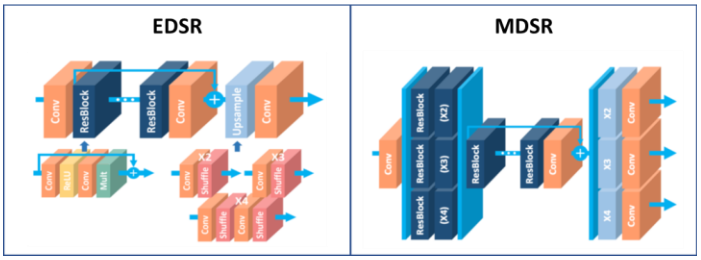

Figure 3: Image Enhancement Architecture

Enhanced deep super-resolution (EDSR) network is a deep neural network used for increasing the resolution of an image by a single scale. The baseline EDSR model uses 16 residual blocks. EDSR does not have any batch normalization layers and uses an activation function (ReLU) only within the residual layer. The overall architecture of the EDSR is given in Figure 3. EDSR only models for 1 scale at a time. It has 3 models for x2, x3, and x4 scaling. MDSR improves the EDSR architecture by allowing 1 model to be able to generate 3 different upscaled images (x2, x3, x4).

MDSR architecture can been seen in Figure 3 It consists of scale-specific preprocessing that is required before following a similar structure to EDSR. There are scale-specific up-sampling layers which are present, which handle the super-resolution of the image by the corresponding scale.

2.3.1 Training Image Enhancement Module (EDSR/MDSR):

To study the effects of perceptual loss, our training was done on two image enhancement modules: EDSR and MDSR. The models are trained on the DIV2K dataset using a perceptual loss for super-resolution and de-blurring.

During training, every image is divided into patches of size 24*24. Each patch is up-sampled to 48*48 and then put together to generate the output image. The perceptual loss is calculated on the final output by comparing with the reference image. The perceptual loss consists of 2 components:

- **L2 Loss**: The component represents the deviation of the output from the given reference. This represents the main part of the image enhancement task.
- **Aesthetic Score Loss**: The aesthetic score loss represents the aesthetic quality of the image. This is the additional component we propose to traditional loss functions.

The perceptual loss can be represented by the following equation:

$$\mathrm{l(W)} = \mathrm{f}\big(x_r, c_W(\mathrm{x})\big) + \gamma * \left(10 - \sum_{i=1}^{10} i * p_i\right)$$

where γ represents the importance of the aesthetic score loss with respect to the given task, f(.) represents the L2 loss, x_r represents the reference or ground truth image and $c_W(\mathrm{x})$ represents the output generated by the image enhancement module.

Both EDSR and MDSR are trained using an Adam optimizer with a learning rate of 0.0001 with no weight decay. All the models were only trained for 20 epochs due to computing constraints.

3 User Interface

A user interface for image super-resolution was built using ReactJS to enhance and access the input images. There are 2 main components to the user interface - the upload page and the results page. The upload page allows a user to upload images. On uploading, it is stored in the local filesystem where the image is enhanced using the baseline and modified models. All the images are then assessed using NIMA and the scores are displayed to the user on the results page. On clicking the image, the higher-resolution image is shown on a new tab.

4 Experiments

We trained EDSR and MDSR models for image super-resolution and image de-blurring using our hybrid cost function. The models were trained for only 20 epochs as the training time for the models was significantly large. Training for 20 epochs for EDSR models took around 7 hours while MDSR took 5 hours. We had to experiment with various hyperparameters and learning rates to improve accuracy, so we could not increase the number of epochs per model. However, we were able to achieve similar NIMA scores to the baseline models which were trained for 300 epochs. We can improve performance by increasing the number of training epochs.

4.1 Image Super-resolution

The enhancement models (MDSR) were trained with various upsilon values to find the optimal value to be used. The PSNR value for γ=0.4 was slightly better after 20 epochs. We expect this difference to increase with more training.

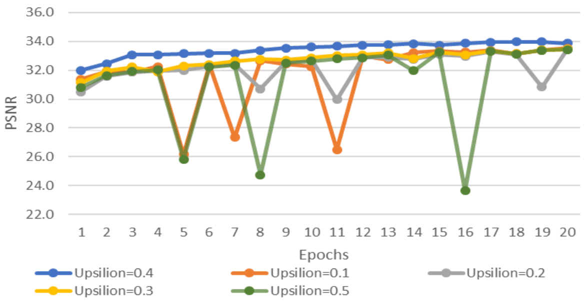

Figure 4: PSNR value – by Upsilon

Then we trained EDSR and MDSR for the same upsilon values to ensure that the loss was comparable across the models. Even though EDSR was slower to train, it outperformed MDSR for both loss and PSNR scores. The PSNR score for EDSR after 20 epochs was 33.87 on the validation dataset as compared to a PSNR score of 33.23 for MDSR

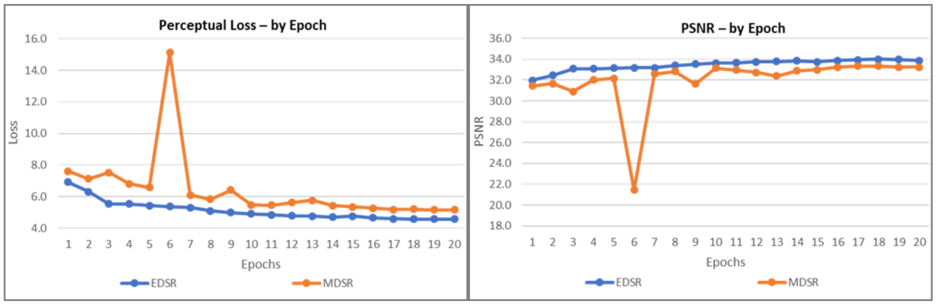

Figure 5: Model Training (EDSR vs MDSR) – by Epochs

The modified EDSR model performs significantly better than modified MDSR as well as the baseline models. A comparison of the NIMA and output images can be seen below:

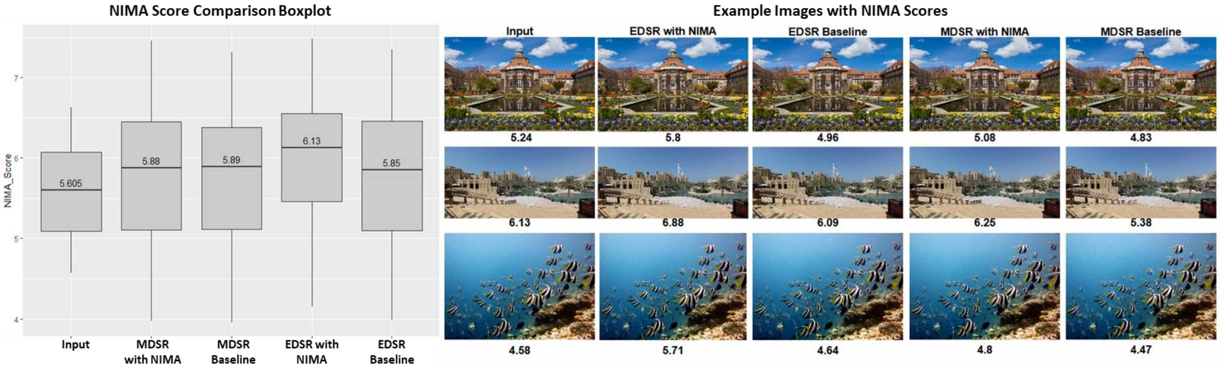

Figure 6: Model Output Assessment and examples (super-resolution)

4.2 Image De-blurring

We generated blurred images by applying a gaussian blur to the LR images. Then we trained the enhancement models by following a similar strategy to image super resolution. We observed that lower values of upsilon gave better results for de-blurring as compared to super resolution. The PSNR value for γ=0.1 was the best after 20 epochs

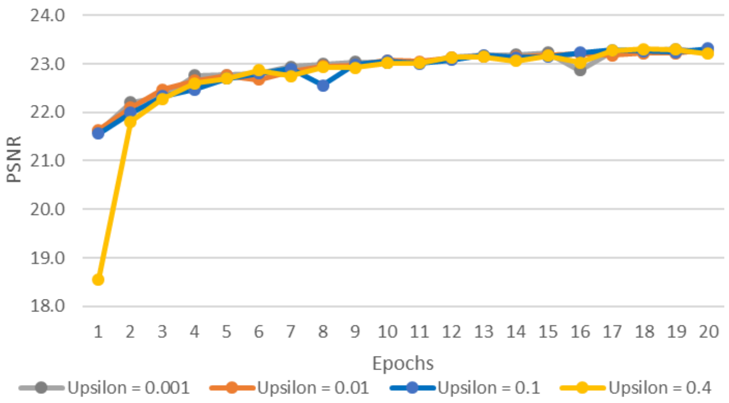

Figure 7: PSNR value – by Upsilon

EDSR outperformed MDSR for image deblurring as well, giving a PSNR score of 23.5 on the validation dataset as compared to a PSNR score of 23.3 for MDSR. The charts for PSNR values and perceptual loss are given below:

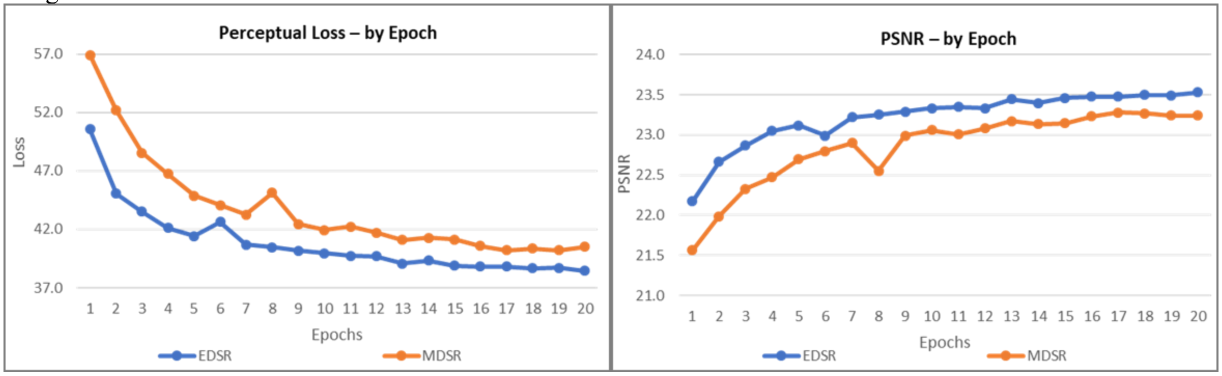

Figure 8: Model Training (EDSR vs MDSR) – by Epochs

The modified enhancement models perform on par with the baseline models. A comparison of the NIMA and output images can be seen below:

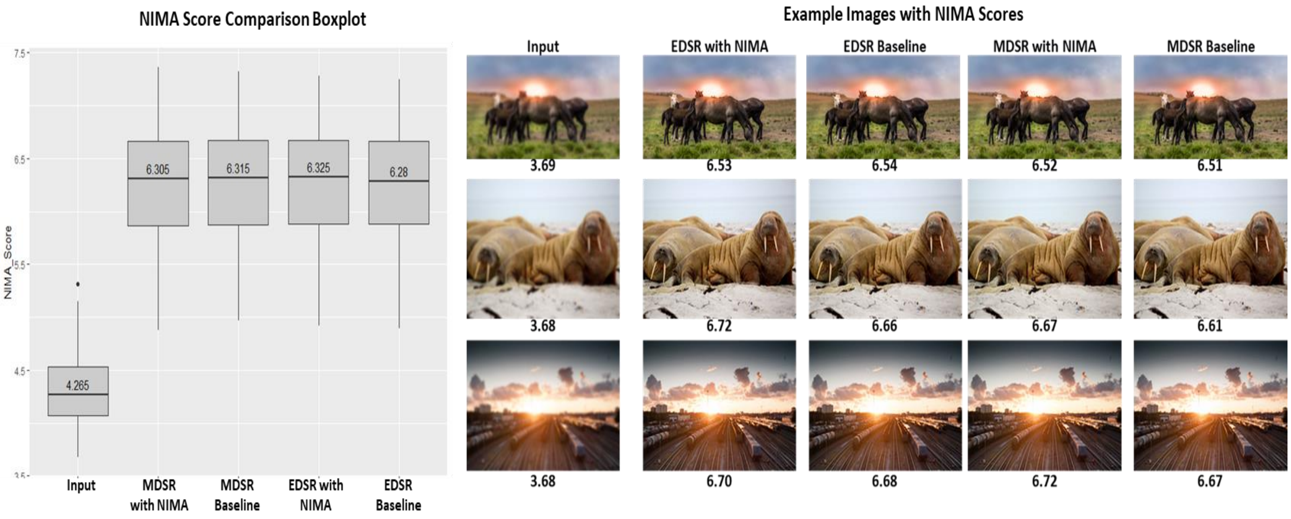

Figure 9: Model Output Assessment and examples (De-blurring)

5 Contributions

- **Sri Gayatri Rachakonda**: Developed the perceptual cost function, integrated the perceptual loss with the image enhancement modules (EDSR and MDSR), trained the MDSR models for deblurring, contributed to the writing the report
- **Abhishek Sunnak**: Trained the NIMA model on the MNasNet acrchitecture, generated the input images for deblurring and trained EDSR models for deblurring, contributed to the writing of the report, generated all the graphs required for the report, set up GCP instance for training models
- **Hang Hu**: Developed the user interface for the demo, trained and fine-tuned the MDSR & EDSR models for super-resolution by experimenting with various hyper-parameters
- **Padmanabhan Rajendrakumar**: Trained the MDSR models for deblurring, trained the MDSR models for super-resolution, developed the inference script for image assessment
- **Vijaytha Vaasan Srinivasan**: Trained the EDSR enhancement model for super-resolution, wrote the scripts to generate enhanced models using the baseline and modified models, designed the poster.

6 Conclusion

Throughout We observed that our models with the modified cost function performed on par with the baseline models with significantly lower training epochs. We also observed that our EDSR models significantly outperformed all the other models for image super resolution. If we train our models for more epochs, we would definitely see them perform better than the baseline models. This helps us show that using a perceptual loss gives better results with lesser training epochs than models trained using just L1 or L2 loss.

7 References

[1] Lim, Bee et al. “Enhanced Deep Residual Networks for Single Image Super-Resolution.” 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2017)

[2] Talebi, Hossein and Peyman Milanfar. “NIMA: Neural Image Assessment.” IEEE Transactions on Image Processing 27 (2018)

[3] L. Kang, P. Ye, Y. Li, and D. Doermann, “Convolutional neural networks for no-reference image quality assessment,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014

[4] H. Zeng, L. Zhang, and A. C. Bovik, “A probabilistic quality representation approach to deep blind image quality prediction,” arXiv preprint arXiv:1708.08190, 2017

[5] Hossein Talebi, Peyman Milanfar: “Learned Perceptual Image Enhancement” IEEE International Conference on Computational Photography (ICCP),May 2018.

[6] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam: “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications” arXiv:1704.04861

[7] Tan, Mingxing et al. “MnasNet: Platform-Aware Neural Architecture Search for Mobile.” CoRR abs/1807.11626 (2018): n. pag.

[8] N. Murray, L. Marchesotti, and F. Perronnin, “AVA: A large-scale database for aesthetic visual analysis,” in Computer Vision and Pattern Recognition (CVPR)

[9] Agustsson, Eirikur and Timofte, Radu: “NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops 2017

[10] Codebases used:

a. https://github.com/thstkdgus35/EDSR-PyTorch
b. https://github.com/truskovskiyk/nima.pytorch
c. https://github.com/facebook/create-react-app
d. https://github.com/billhhh/MnasNet-pytorch-pretrained