BTP II - Stereo-Image Super Resolution

Pulkit Mahajan (2019UCS0073)
Shreyas Chatterjee (2019UCS0097)

Supervisor:
Dr. Vinit Jakhetiya

Objective and Detailed Problem Statement

To make a SOTA model for Stereo Image Resolution

Problem Statement - To construct high-resolution (HR) stereo images from their low resolution (LR) counterparts. We aim to obtain SR results with high fidelity (PSNR) and high perceptual score with promising stereo consistency under standard bicubic degradation.

Work Done in BTP-I (7th Semester) - Pulkit

- Supervisor: Dr. Vinit Jakhetiya
- 3D Novel View Synthesis using Neural Radiance Fields
  - Literature Review: Mildenhall NeRF
  - Construction of 3D volumes using multiple images of a scene
  - Use Cases in Augmented Reality, 3D modelling, VR Gaming
  - Representation of a scene in form of an overfitted fully connected neural network architecture.
- Reproduced Results from paper.
- Experimented on the code with different hyper parameters
- Lack of Computational Resources

Work Done in BTP-I (7th Semester) - Shreyas

Supervisor - Dr. Shaifu Gupta

Topic Name - Detection of PII leakage from app traffic using binary features.

What is PII? - Personal Identifiable Information Leakage

Why is detection of PII important?

- Little to no visibility
- Limited control over network flow
- Lack of knowledge in users about how to protect their data.

Work Done in BTP-I Continued - Shreyas

Work Done:

- Literature review of existing papers
  - ReCon: Revealing and Controlling PII Leaks in Mobile Network Traffic
  - AntShield: On-Device Detection of Personal Information Exposure
- Implementation and replication of results of ReCon.
- Both papers had ML algorithm based solution, but there was no DL based solution publicly available.
- Applied a heuristic and filtered important features using the confidence score achieved for the features.
- Implemented a NN based model
- Achieved better results than both ReCon and AntShield

How much development was done in BTP-I?

All the above points were completed along with trying various different models and architectures including the industry grade Microsoft MEB model architecture.

BTP-II - Work and Activities for this semester

- Literature Review
  - NAFSSR
  - ESR-GANs
- Implementation and reproduction of results of the base model
- Understanding and experimenting with various loss functions
  - L1 Loss
  - Perceptual Loss
  - Texture Loss
  - MANIQA (No Reference Loss)
- Adding an independent novel Image2Image translation model for post processing of SR images, on top of existing architecture.
- Submission of results to NTIRE 2023 CVPR Challenge
- Publishing a paper.
- Making certain sections of code reusable and publishing to PyPI for future usage.

BTP-II - Status and Progress until Mid Sem (11.04.23)

- Literature Review
  - NAFSSR
  - ESR-GANs
- Implementation and reproduction of results of the base model
- Understanding and experimenting with various loss functions
  - L1 Loss
  - Perceptual Loss
  - Texture Loss
  - MANIQA (No Reference Loss)
  - SISR Quality Loss Extented to StereoISR Quality Loss
- Adding an independent novel Image2Image translation model for post processing of SR images, on top of existing architecture.
- Submission of results to NTIRE 2023 CVPR Challenge
  - Ranked 7 out of 93 Participants in Track 1
  - Ranked 13 out of 176 Participants in Track 2
  - Received Co-Authorship in Challenge Paper to be submitted for CVPR 2023
- Making certain sections of code reusable and publishing to PyPI for future usage.

BTP-II - Progress after Mid Sem

- Literature Review
  - NAFSSR
  - ESR-GANs
- Implementation and reproduction of results of the base model
- Understanding and experimenting with various loss functions
  - L1 Loss
  - Perceptual Loss
  - Texture Loss
  - MANIQA (No Reference Loss)
  - SISR Quality Loss
- Adding an independent novel Image2Image translation model for post processing of SR images, on top of existing architecture.
- Submission of results to NTIRE 2023 CVPR Challenge
  - Ranked 7 out of 93 Participants in Track 1
  - Ranked 13 out of 176 Participants in Track 2
  - Received Co-Authorship in Challenge Paper to be submitted for CVPR 2023
- Using Student-Teacher Distillation for improvement in accuracy

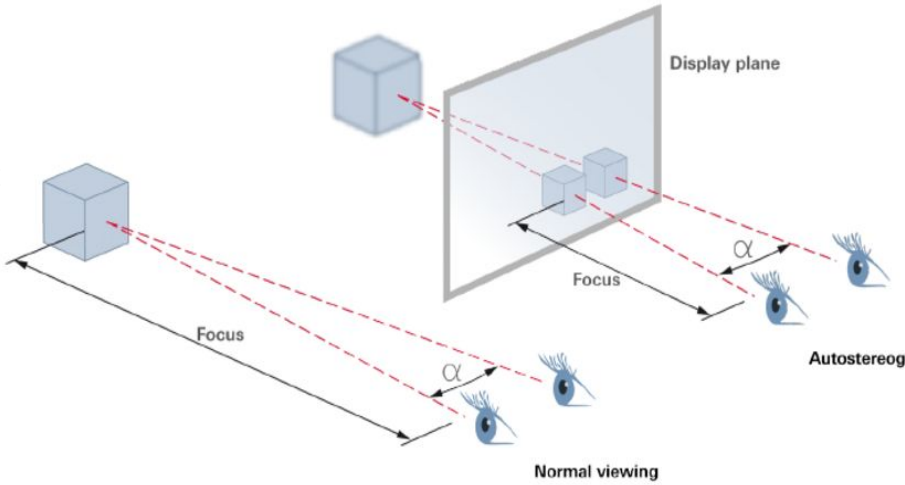
Dataset for the Challenge

Main Reference Dataset - Flickr 1024

- 800 Left-Right Stereo Image Pairs (HR) for training
- Corresponding Left-Right Stereo Image Pairs (LR) (produced by bicubic downsampling) were also provided.
- Validation dataset : 112 Left-Right Image Pairs (LR only)
- Testing dataset: 100 Left-Right Image Pairs (LR only)
    - HR Images for validation and testing were unseen and evaluation metrics like PSNR and SSIM were evaluated by submitting image results on challenge organizers' online platform.

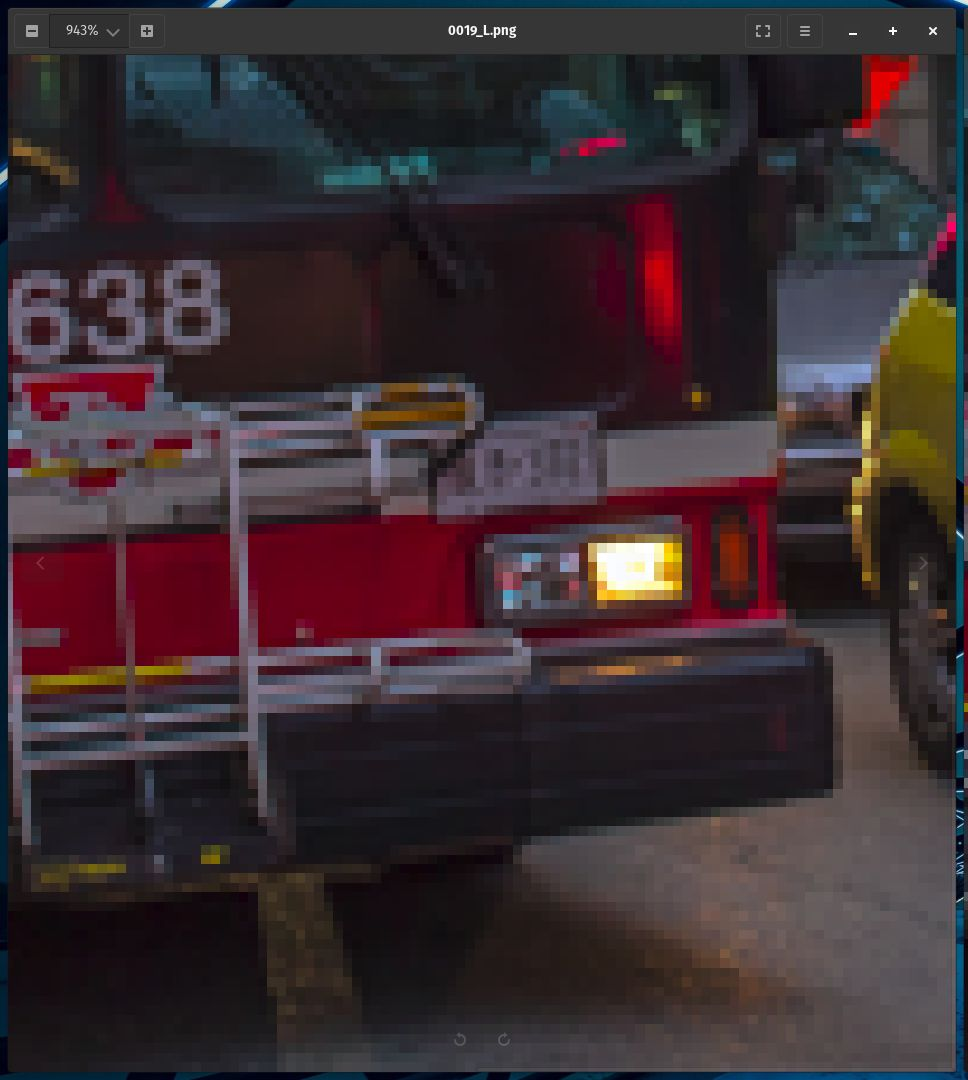
Stereo Images and their Importance in Research

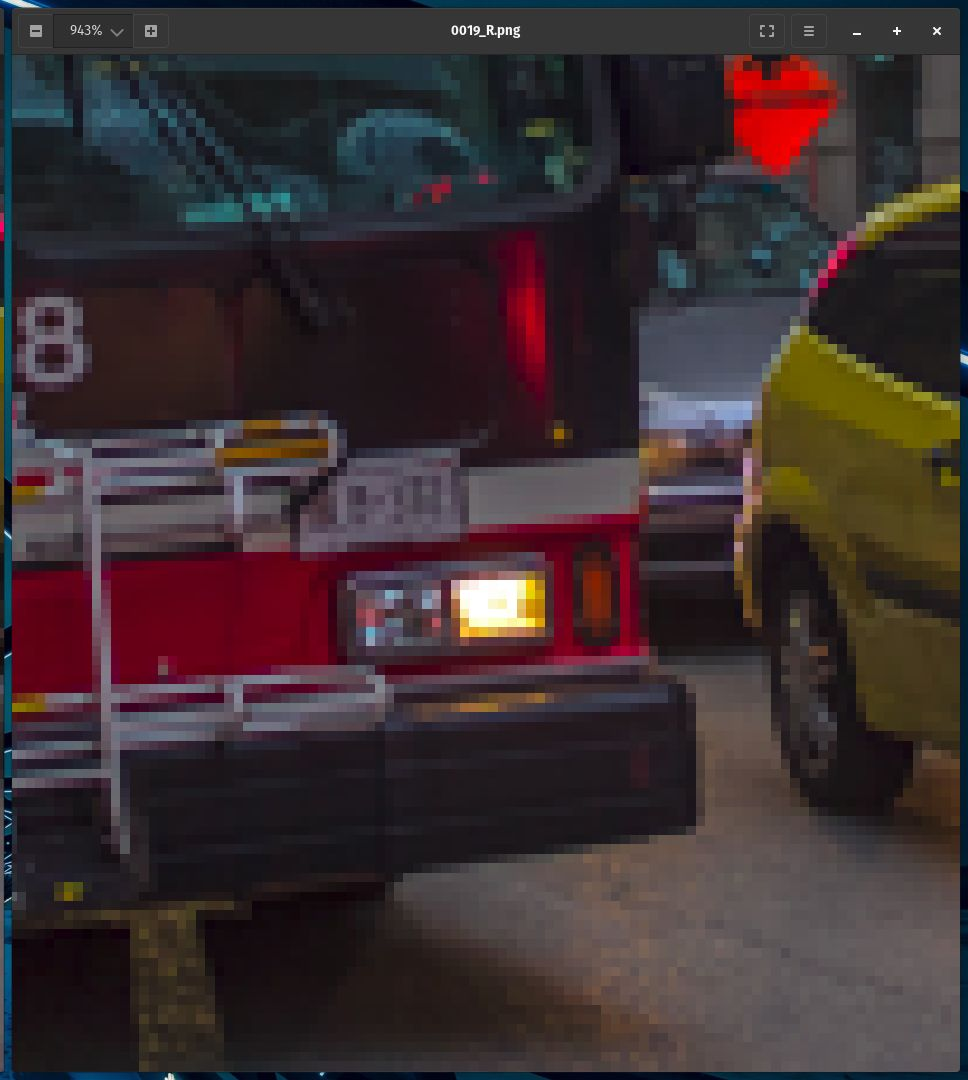
- Stereo images capture two views of the same scene from slightly different viewpoints, simulating the human visual system binocular vision.
- This difference in viewpoints allows for the calculation of depth information, or disparity, between corresponding pixels in the left and right images.
- This disparity information is used to create a depth map that can be used for a variety of computer vision tasks, such as estimating the position and orientation of objects in 3D space.
- Stereo images provide depth information that is critical for many applications, including scene reconstruction, object detection, tracking, and recognition.
- More desired with the popularity of dual cameras in mobile phones.
- Stereo images are also used for robotics, autonomous driving, and virtual and augmented reality applications, making them a critical part of computer vision research.

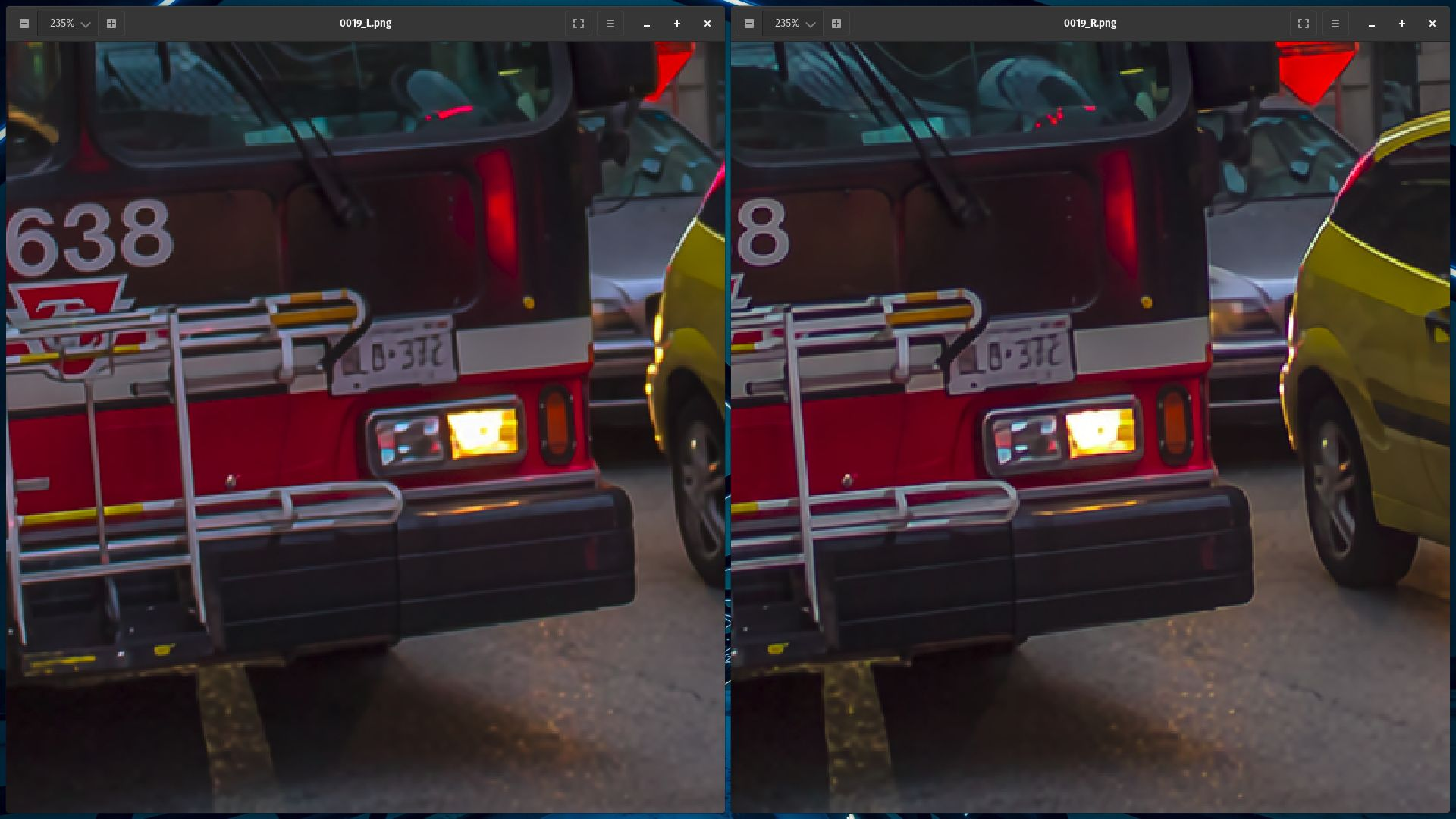


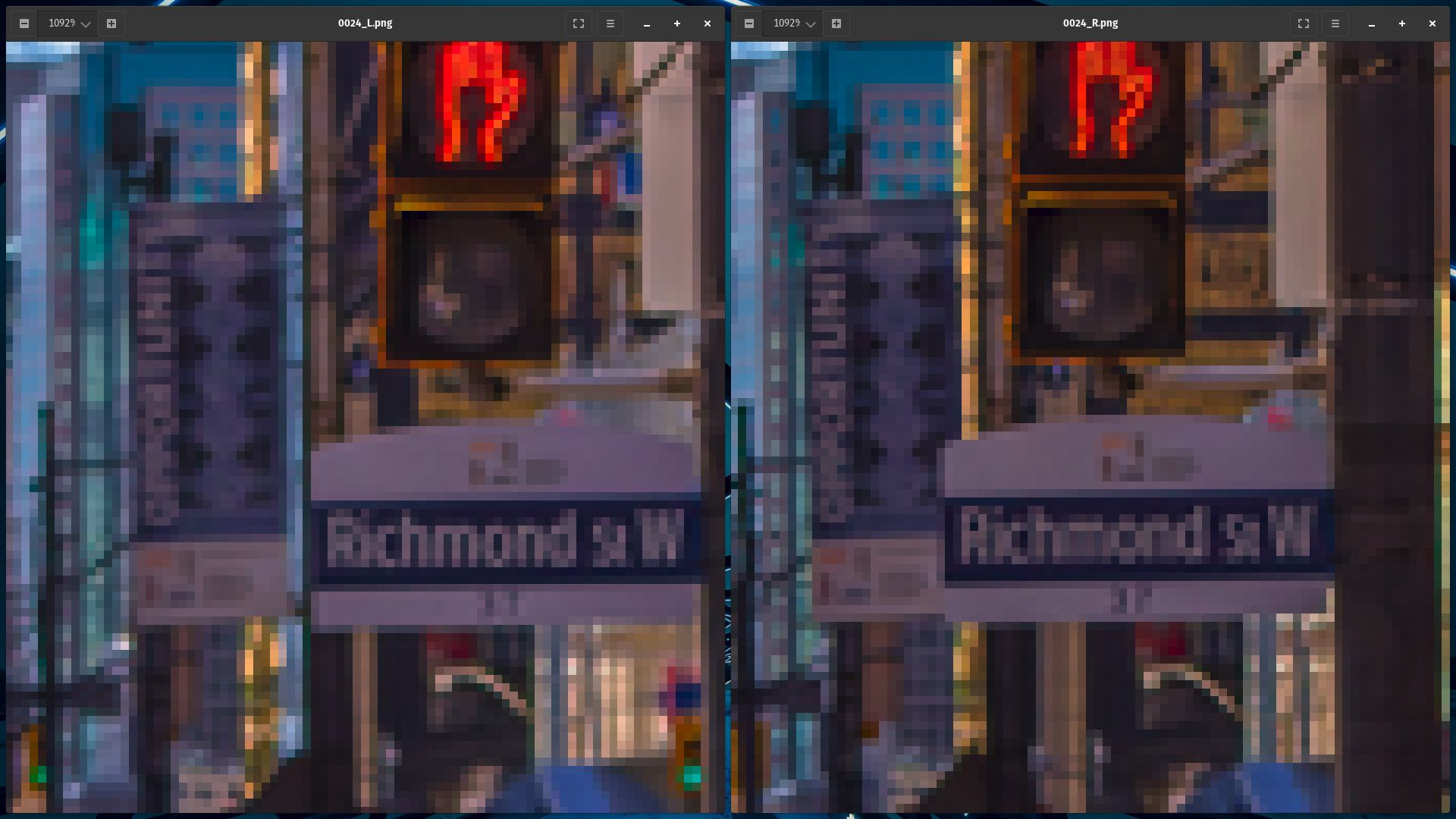
Some of the results achieved before...

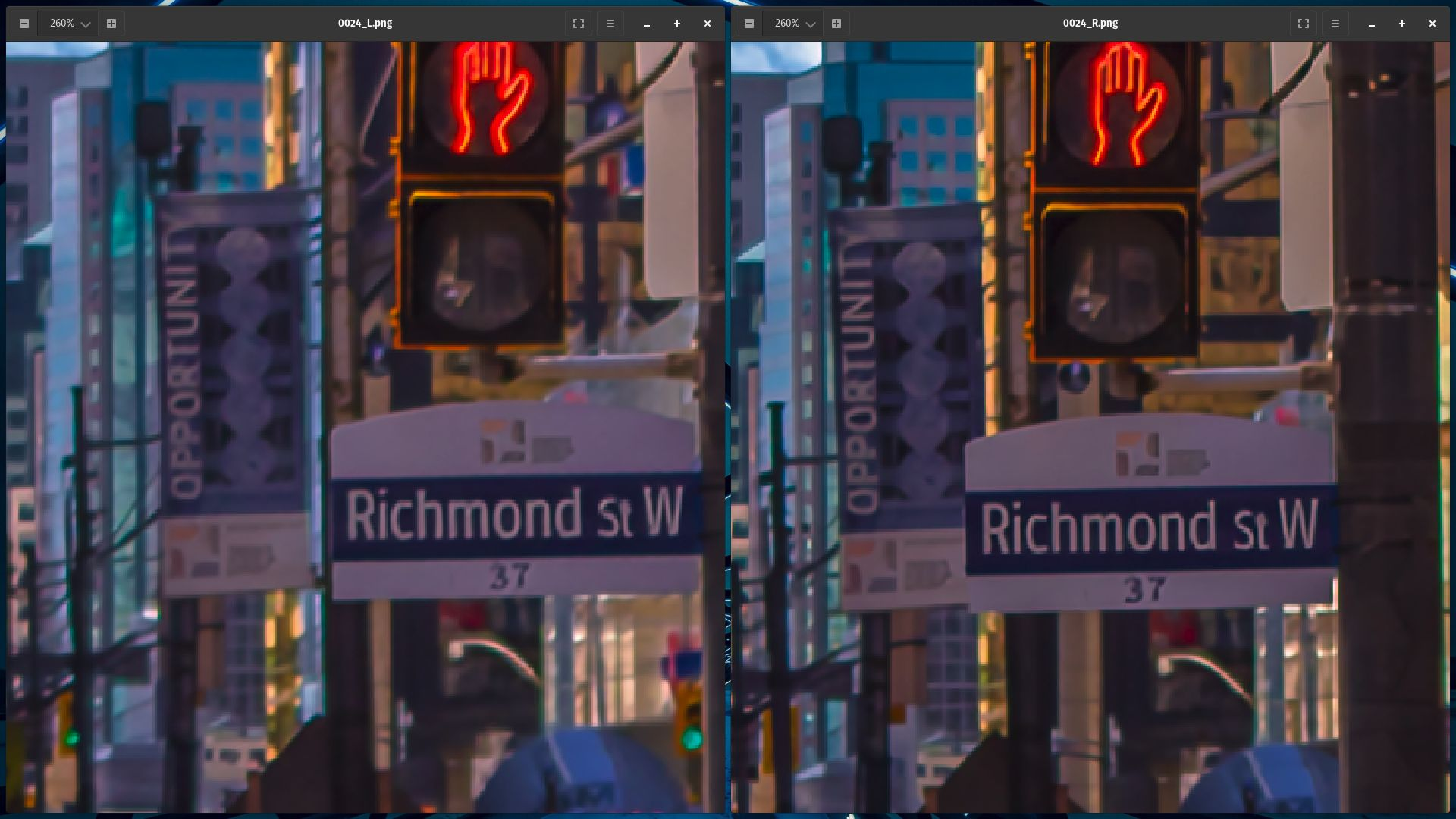
1st Pair - Left and Right LR Images with resolution (H,W)
2nd Pair - Left and Right HR Images with resolution (4H,4W)

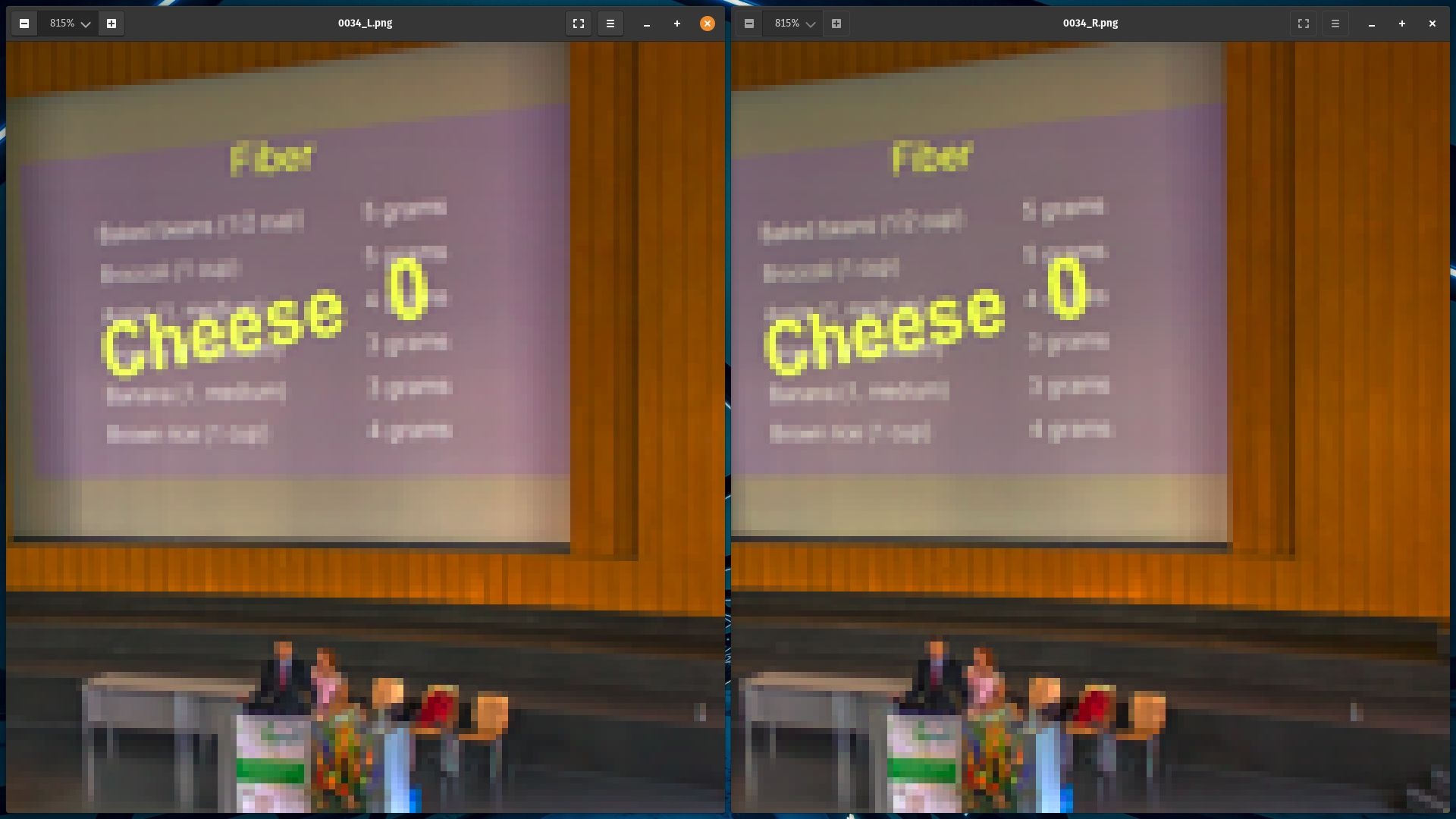

943%
0019_L.png
638

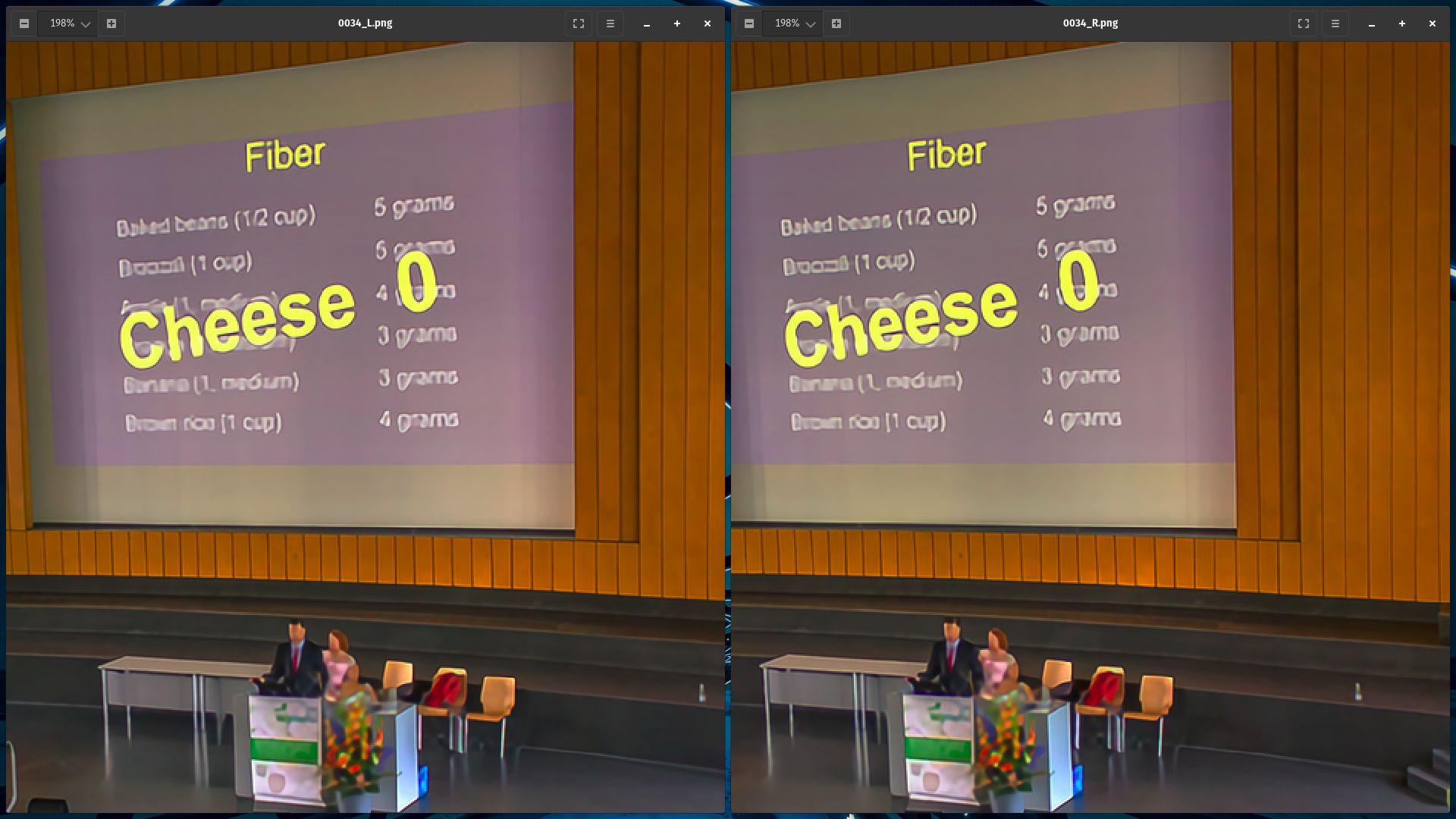

943%
0019_R.png
8


235%
0019_L.png
638
235%
0019_R.png
8

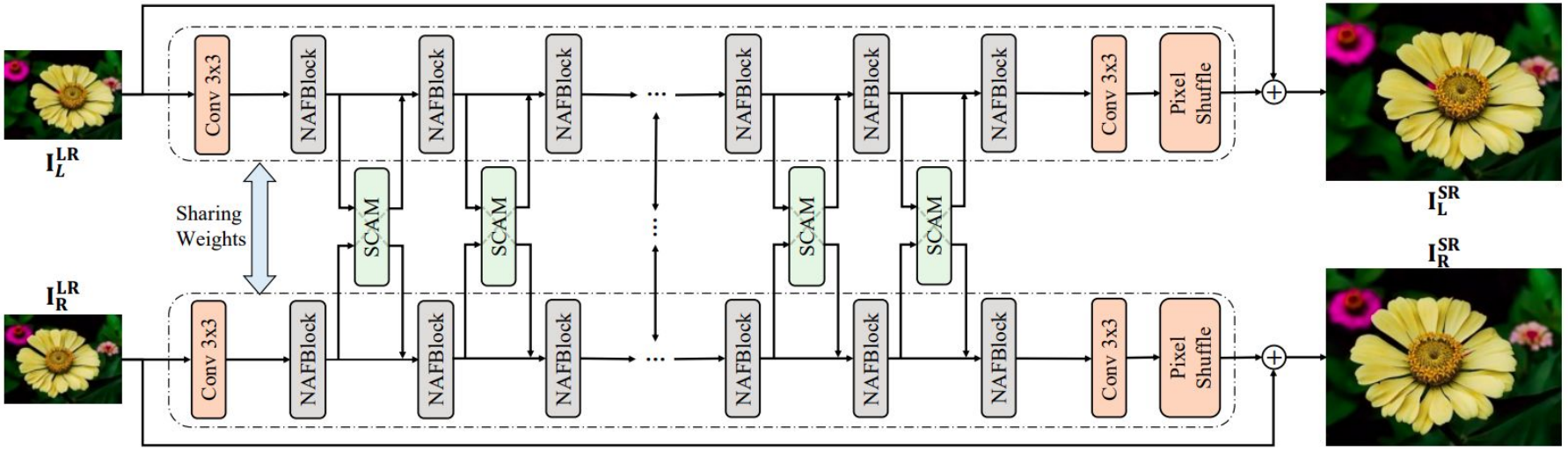

0024_L.png
0024_R.png
Richmond St W
Richmond St W


260%
0024_L.png
OPPORTUNITY
Richmond St W
37
260%
0024_R.png
OPPORTUNITY
Richmond St W
37


815%
0034_L.png
Fiber
Cheese 0
815%
0034_R.png
Fiber
Cheese 0


198%
0034_L.png
198%
0034_R.png
Fiber
Cheese 0
Fiber
Cheese 0

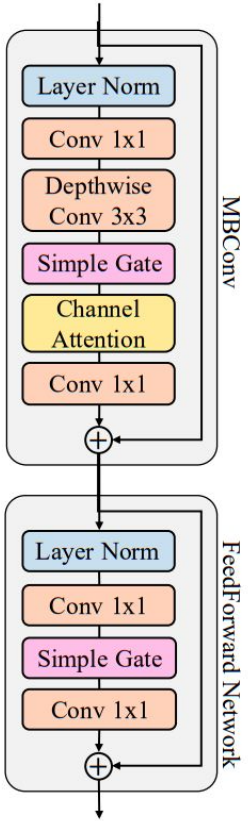
NAFSSR Model Architecture



NAFSSR Model Explanation

- NAFNet achieves competitive performance on single image restoration tasks with low system complexity and NAFSSR is a modification of the same.
- NAFNet blocks (NAFBlocks) extract intraview features for both views in a weight-sharing manner.
- NAFSSR adds simple cross attention modules to NAFNet so that it can fully utilize both intra-view information and cross-view information
- Stereo Cross-Attention Modules (SCAMs) are provided to fuse features extracted from the left and the right images.
- Training Strategies Used:
    - Data Augmentation to Reduce Overfitting: Models were trained with small patches cropped from full-resolution images. These patches were randomly flipped horizontally and vertically
    - Color Augmentation: RGB channels were shuffled randomly
    - Regularizer: Stochastic Depth i.e. a subset of layers is randomly dropped and bypass them with the identity function.
    - Loss: Pixel-wise L1 distance between the super-resolution and ground-truth stereo images

NAFBlock Module



SCAM Module

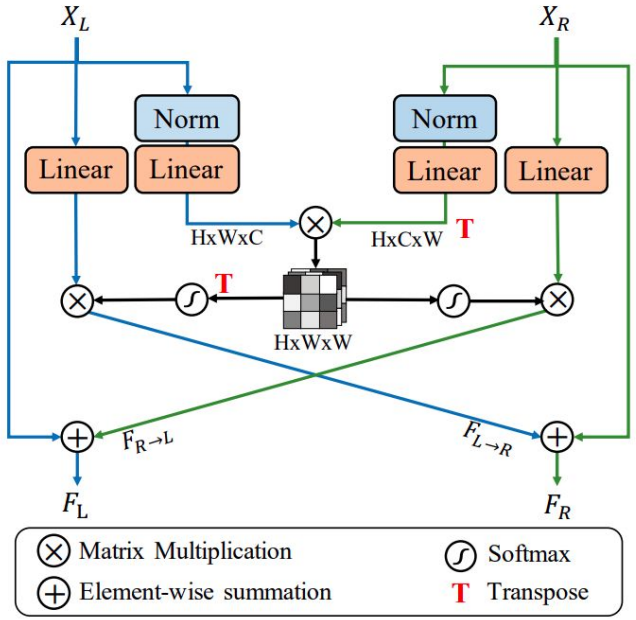


Figure 4. Stereo Cross Attention Module (SCAM). It fuses the features of the left and right views.

More about NAFBlocks

- The main mechanism in the NAFBlock is: where X is the input image

$$\mathbf{X} = \mathrm{MBConv}(\mathrm{LN}(\mathbf{X})) + \mathbf{X}$$
$$\mathbf{X} = \mathrm{FFN}(\mathrm{LN}(\mathbf{X})) + \mathbf{X}$$

- Also what helps remove the use of nonlinear activations (e.g., ReLU, GELU) is the use of SimpleGate units.

$$\mathrm{SimpleGate}(\mathbf{X}) = \mathbf{X_1} \odot \mathbf{X_2}, \qquad (2)$$

where $\odot$ represents element-wise multiplication. The Sim-

- The unit first splits the input into two features arrays along the channel dimension and then computes the output via the linear gate.

More about SCAMs

- SCAM - Stereo Cross Attention Module
- It computes the dot products of the query with all keys and applies a softmax function to obtain the weights on the values

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\mathbf{Q}\mathbf{K}^T/\sqrt{C}\right)\mathbf{V} \quad (3)$$

where $\mathbf{Q} \in \mathbb{R}^{H\times W\times C}$ is *query* matrix projected by source intra-view feature (*e.g.*, left-view), and $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{H\times W\times C}$ are *key*, *value* matrices projected by target intra-view feature (*e.g.*, right-view). Here, H, W, C represent height, width and number of channels of feature map. Since stereo images

$$\begin{aligned}\mathbf{F}_{\mathbf{R}\to\mathbf{L}} &= \text{Attention}(\mathbf{W}_1^{\mathbf{L}}\bar{\mathbf{X}}_{\mathbf{L}}, \mathbf{W}_1^{\mathbf{R}}\bar{\mathbf{X}}_{\mathbf{R}}, \mathbf{W}_2^{\mathbf{R}}\mathbf{X}_{\mathbf{R}}),\\ \mathbf{F}_{\mathbf{L}\to\mathbf{R}} &= \text{Attention}(\mathbf{W}_1^{\mathbf{R}}\bar{\mathbf{X}}_{\mathbf{R}}, \mathbf{W}_1^{\mathbf{L}}\bar{\mathbf{X}}_{\mathbf{L}}, \mathbf{W}_2^{\mathbf{L}}\mathbf{X}_{\mathbf{L}}),\end{aligned} \quad (4)$$

where $\mathbf{W}_1^{\mathbf{L}}, \mathbf{W}_1^{\mathbf{R}}, \mathbf{W}_2^{\mathbf{L}}$ and $\mathbf{W}_2^{\mathbf{R}}$ are projection matrices.

$$\begin{aligned}\mathbf{F}_{\mathbf{L}} &= \gamma_L\mathbf{F}_{\mathbf{R}\to\mathbf{L}} + \mathbf{X}_{\mathbf{L}},\\ \mathbf{F}_{\mathbf{R}} &= \gamma_R\mathbf{F}_{\mathbf{L}\to\mathbf{R}} + \mathbf{X}_{\mathbf{R}},\end{aligned} \quad (5)$$

where γ_L and γ_R are trainable channel-wise scale and initialized with zeros for stabilizing training.

Novelty introduced - Perceptual Loss

L1 Loss alone over-smoothens the output images, which is why better loss functions were required.

- Perceptual Loss: Used a pre-trained network VGG-16 to extract **high-level features** from SR and HR Images and compare them using some distance metric, like MSE.
- High-Level Features capture Perceptual Content of an Image (shape, composition, edges, colors, etc).

$$L_{Percep} = ||\phi(I) - \phi(I')||^2$$

- I and I' are ground truth image and Generated Image respectively
- $\boldsymbol{\Phi}$ is a pre-trained VGG-16 that extracts high-level features

Novelty introduced - Texture Loss

- Texture loss using the gram matrix is a method for measuring the similarity of textures in deep learning-based image synthesis tasks.
- Texture loss is good at finding low level information such as texture features.
- We used a pre-trained VGG-19 and used outputs from mid-layers of the architecture.

$$G^I_{\{ij\}} = \frac{1}{C_I H_I W_I} \sum_{k=1}^{C_I} F^I_{\{i,k\}} * F^I_{\{j,k\}}$$

C_l, H_l, and W_l are the number of feature maps, height, and width of layer l, respectively.
F_{ik}^l is the activation of the k-th feature map at position i of layer l

Flattening the feature map into a vector and computing the outer product of the vector with itself.

$$L_{\{texture\}} = \frac{1}{N_I} \sum_{i=1}^{N_I} \left(G^I_{\{i,j\}} - A^I_{\{i,j\}} \right)^2$$

N_l is the number of elements in the gram matrix of layer l
A_{ij}^l is the corresponding element of the gram matrix of the input image at layer l

Taking MSE loss of gram matrices

Evaluation Metrics

PSNR - Peak signal-to-noise ratio

The PSNR (in dB) is defined as

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$$

SSIM - Structural Similarity

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

with:

- μ_x the pixel sample mean of x;
- the pixel sample mean of y;
- σ_x^2 the variance of x;
- σ_y^2 the variance of y;
- σ_{xy} the covariance of x and y;
- $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ two variables to stabilize the division with weak denominator;
- L the dynamic range of the pixel-values (typically this is $2^{\#bits\ per\ pixel} - 1$);
- $k_1 = 0.01$ and $k_2 = 0.03$ by default.

Results from Experiments

Model Type	Final PSNR
Small Model with L1 loss only	22.763
Small Model with L1 + Perceptual loss	23.27
Small Model with L1 + Perceptual + Textual loss	**23.46**
Large Model with L1 loss only	23.63
Large Model with L1 + Perceptual loss	23.8998
Large Model with L1 + Perceptual + Textual loss	**24.0696**

Some comparisons with different losses

1st Picture - Result with L1 loss only
2nd Picture - Result with L1 + Perceptual Loss
3rd Picture - Result with L1 + Perceptual + Texture Loss


256%
0008_L.png
ph centre


257%
0008_L.png
ph centre


257%
0008_L.png
ph centre

One of Results with Various Losses

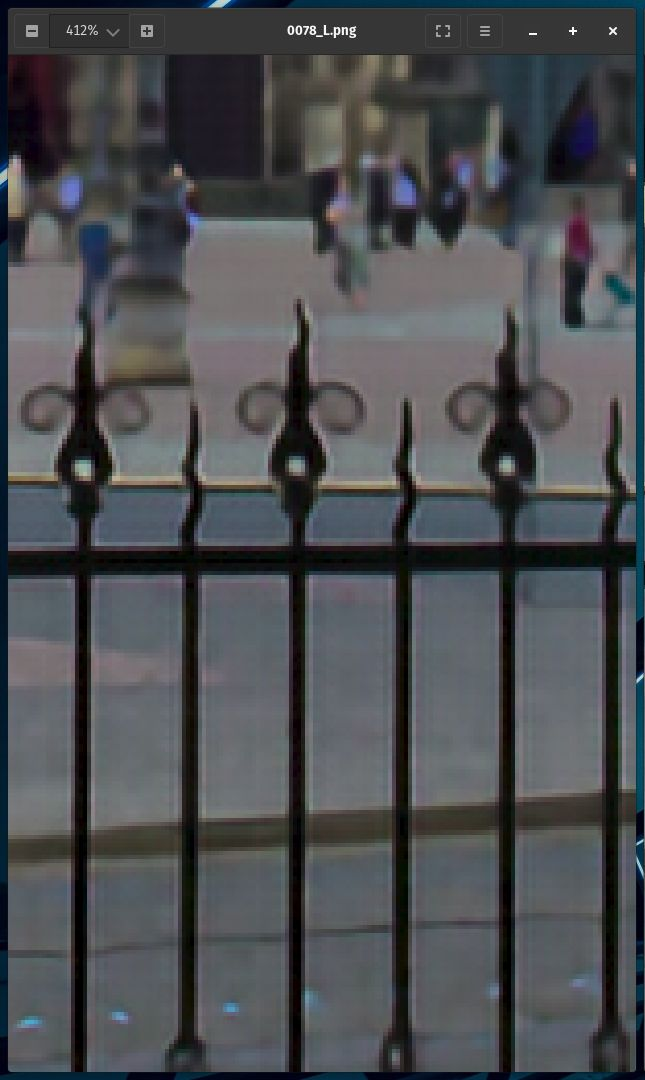
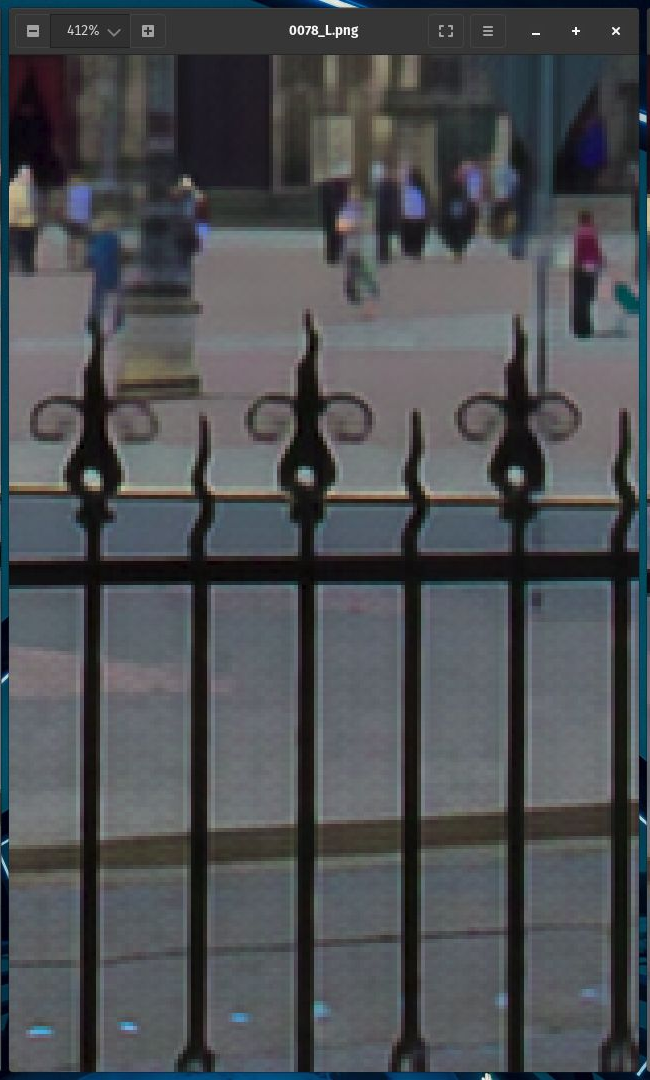
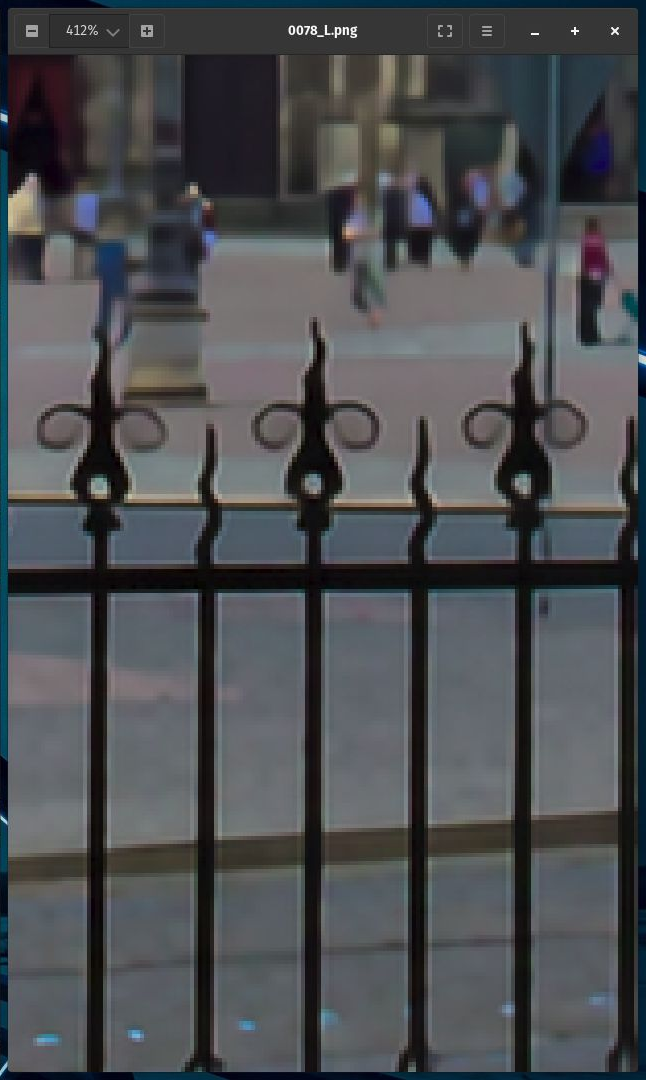
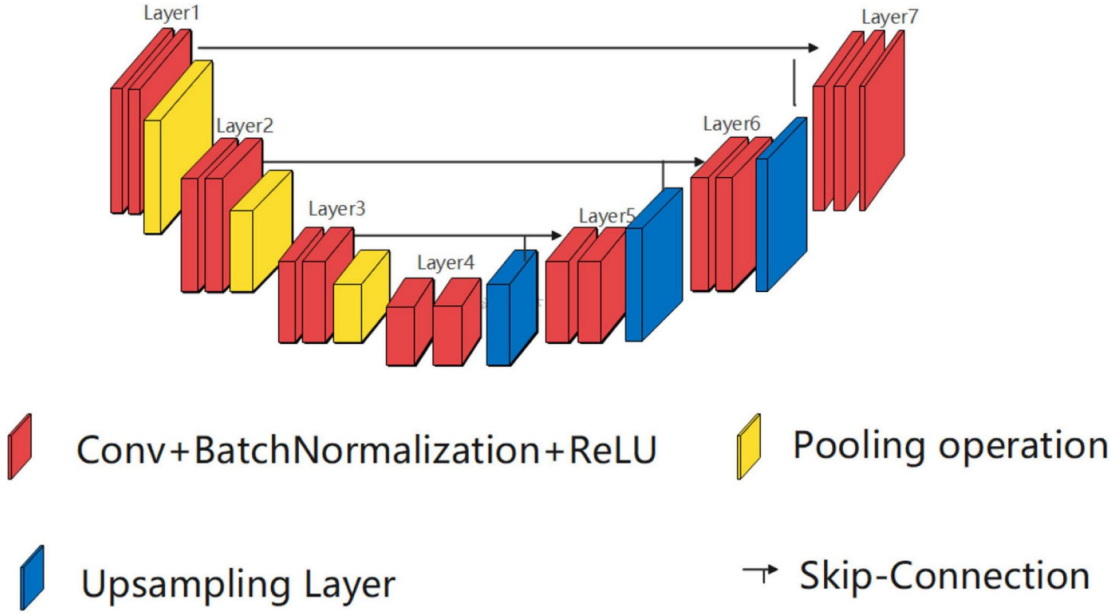

412%
0078_L.png

412%
0078_L.png

412%
0078_L.png

Image-2-Image Translation - UNet On Top of NAFSSR



UNet Model Architecture

Why Image2Image Translation and UNet?

- We expected a translation model do better the accuracy after NAFSSR has super-resoluted the LR images.
- UNet uses the encoder to capture high-level features from the image and the decoder to recover the spatial information lost during compression.
- UNet is a sort of autoencoder type model and it compares two masks and tries to resolve the difference between the two.
- We tried to replace the mask with SR images (model output) and actual HR images.

Changes made in UNet Architecture to incorporate Stereo Information

- Experiment 1: Independent Dual UNet
- Experiment 2: SCAM Modules introduced after every layer

Issues faced with UNet

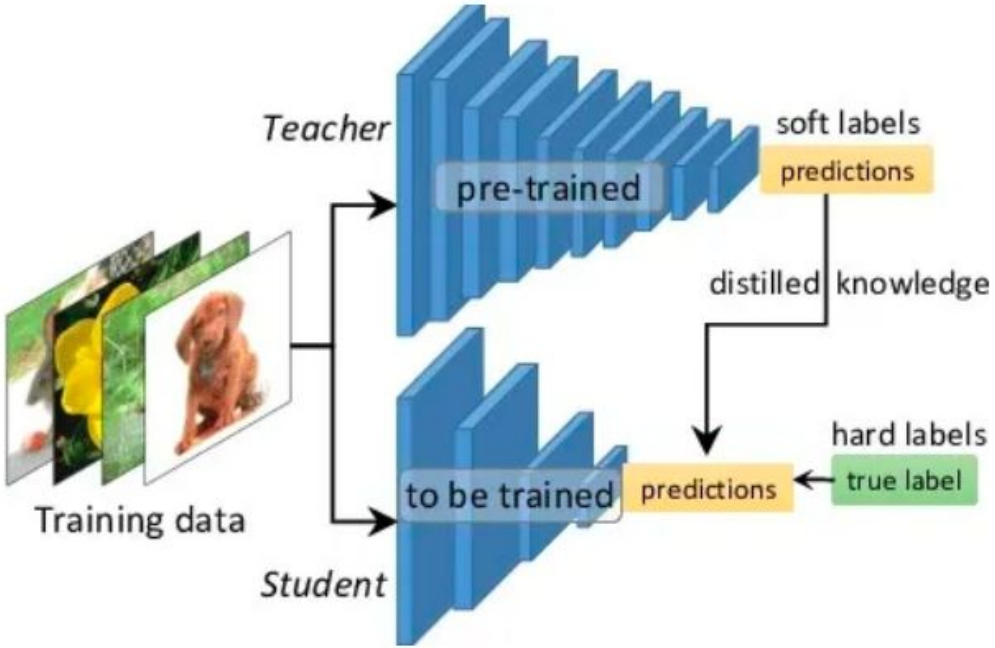
Poorer results i.e. giving lower PSNR than before.

Possible Reasons:

- Small size of the model i.e. lower trainable parameters, might be causing poorer results.
- Might be causing over-smoothening due to L1 Loss.
- Model unable to represent features properly because of smaller bottleneck layer
- Errors propagated through the NAFSSR might be getting magnified further by the UNet Model, instead of getting removed.

Student Teacher Model

Knowledge Distillation



- Large Models can't be deployed much easily, limitations in size of model that can be loaded onto the GPU.
- But we also know that larger model of same architecture would clearly outperform a smaller model.
- Teacher acts as a supervisor and adds an additional loss term to enable student to learn faster.
- The Student is then able to learn better than what he was learning without the teacher.
- Performance is now somewhat in between the performance of lone student and lone pre-trained teacher.

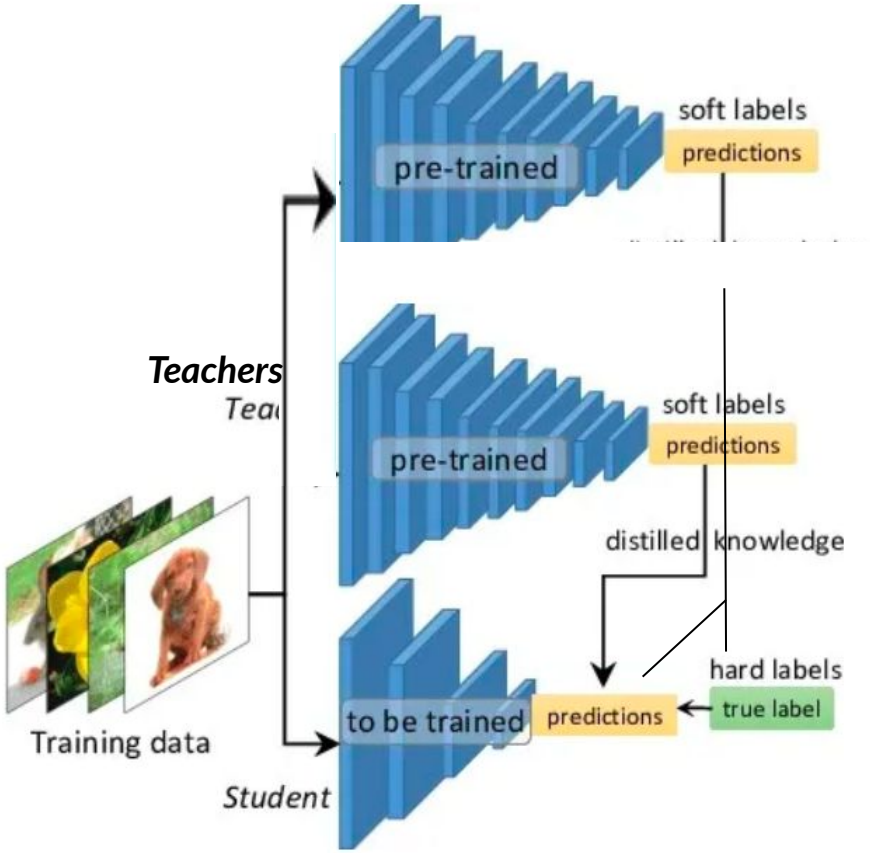
Note:

- Back Propagation, weight update only occurs in student, Teachers' weights are left untouched
- Soft labels: predictions from pre-trained teacher model.
- Hard Labels: Ground Truths

Motivating Knowledge Distillation

MOTIVATION:

- We have two lead performers among all the model's we tried:
  - Pre-Trained Large Model, trained only on L1 Loss.
  - Our Trained Large Model, trained on combination of L1 Loss + Perceptual Loss + Texture Loss.
  - The Training proceeds by taking patches of the train data.
  - Testing Results showed that In 90% of the patches selected randomly from data our model outperformed better than Pre-Trained model.
  - Selective Patches, showed better performance when processed with Pre-Trained Model.



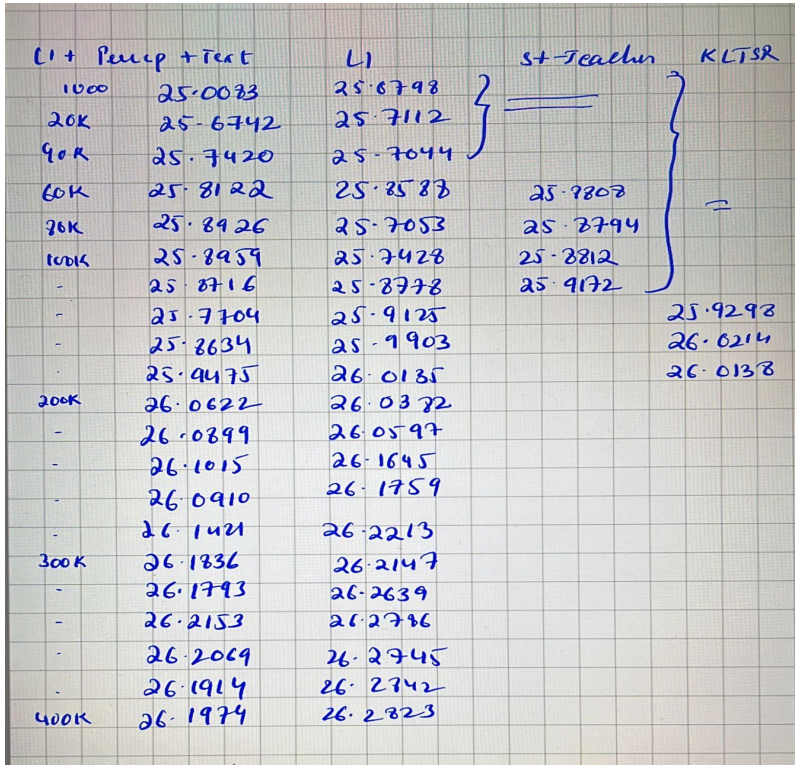
Incorporation

- Using Two Teachers in place of One, and one student of course.
- Comparing performance of teachers for every patch, and correspondingly using their respective Loss Functions for back-propagation.

Downside of Student Teacher Model

- The two teachers will also require processing time on gpu.
- One way is to perform the forward propagation for the two teachers and the student in parallel, but that would optimally require 3 GPUs running in parallel.
- All three running on the same GPU result in repetitive context switching with respect to the models and this slows down the training speed a lot.
- On an 8-GB GPU, the expected training time reached till **53 days**, for 4,00,000 iterations. We performed the training till 1L iterations, after which we switched to KLT-SRQA.

Results and Observations of Student-Teacher Model



CI + Pmcp + Text		LI	St-Teacher	KLTSR
1000	25.0033	25.6798		
20K	25.6742	25.7112		
40K	25.7420	25.7044		
60K	25.8122	25.8587	25.9803	
80K	25.8926	25.7053	25.8794	
100K	25.8959	25.7428	25.8812	
-	25.8716	25.8778	25.9172	
-	25.7704	25.9125		25.9298
-	25.8634	25.9903		26.0214
	25.9475	26.0185		26.0138
200K	26.0622	26.0382		
-	26.0899	26.0597		
-	26.1015	26.1645		
-	26.0910	26.1759		
-	26.1421	26.2213		
300K	26.1836	26.2147		
-	26.1793	26.2639		
-	26.2153	26.2786		
-	26.2069	26.2745		
-	26.1914	26.2742		
400K	26.1974	26.2823		

Switch to Quality Assessment based Losses

- Necessity for a Monolithic architecture.
- Processing Time.
- Incorporating new variety of losses.
    - MAN-IQA Loss
    - KLT-StereoSR-QA Loss [KLTSRQA]

MANIQA Loss

MANIQA - Multi-dimension Attention Network for No-Reference Image Quality Assessment

No-Reference Image Quality Assessment (NR-IQA) aims to assess the perceptual quality of images in accordance with human subjective perception without any reference.

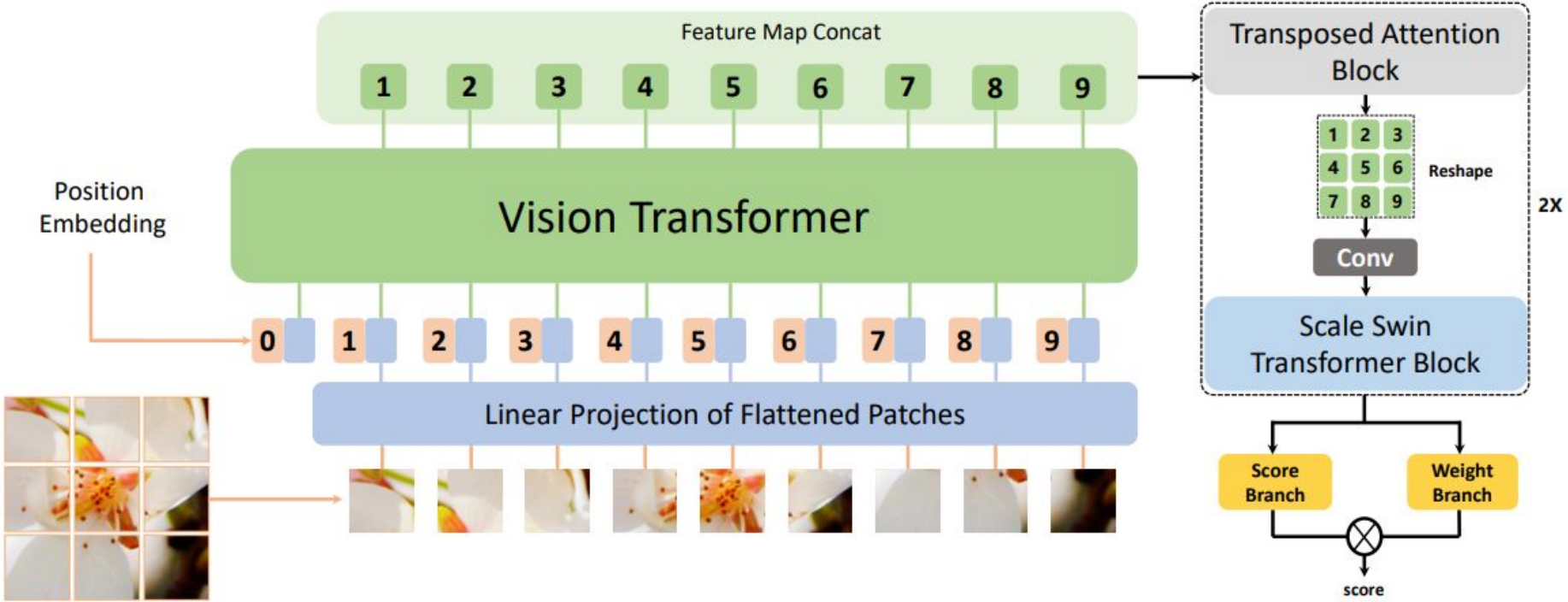
Steps in finding the NR-IQA:

1. Extract features via ViT
2. Strengthen global and local interactions via the Transposed Attention Block (TAB) and the Scale Swin Transformer Block (SSTB)
3. A dual branch structure for patch-weighted quality prediction is applied to predict the final score depending on the weight of each patch's score.

Why MANIQA?

- NR-IQA provided a new kind of measure which could be used a loss that needs to be minimized between the LR and HR images.
- Proven to work in both case - natural as well as GAN based distortions - in images.
- Captures both high level and low level features in the spatial dimension of the image but also uses the information present between channel to channel.

MANIQA Architecture



MANIQA Explanation

1. From the ViT, we use 4 from the total 12 layers to extract features from different semantic degrees, and concatenate them.
2. Next, the feature map is sent to the TAB (Transposed Attention Block), which applies SA across channel rather than the spatial dimension to compute cross-covariance across channels to generate an attention map encoding the global context implicitly.
3. Advantages of TAB:
    a. TAB rearranges channels' (the 4 from ViT) weight in terms of their importance to perceptual quality score.
    b. The attention map generated by the TAB encodes the global context implicitly.

MANIQA Explanation - Contd...

4. The output from TAB, is passed onto the Scale Swin Transformer Block, which consists of 2 Swin Transformer Layers (STL) and a convolutional layer.

$$F_{out} = \alpha \cdot H_{CONV}(H_{STL}(\tilde{F}_{i,2})) + \tilde{F}_{i,0}, \qquad (4)$$

where $H_{CONV}(\cdot)$ is the Swin Transformer layer. α denotes the scale factor of the output of STL. This design

5. The advantages of the SSTB are:

a. The convolutional layer with spatially invariant filters can enhance the translational equivariance
b. The scale factor α stabilizes the training through residual connection.

MANIQA Explanation - Contd...

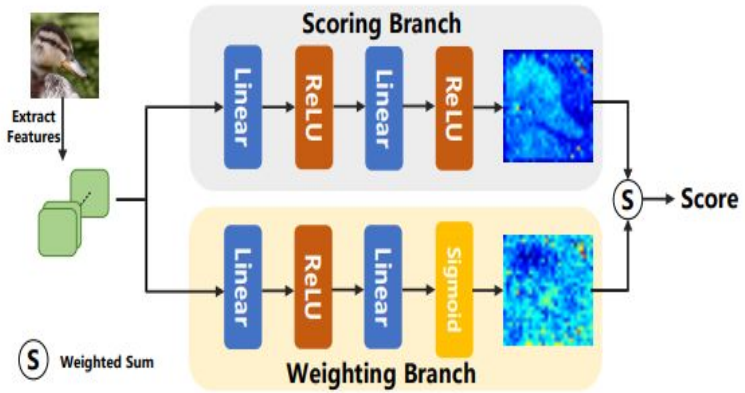


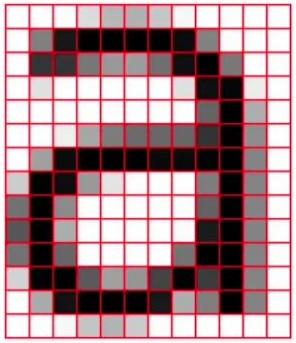
Figure 5. Dual branch structure for patch-weighted quality prediction.

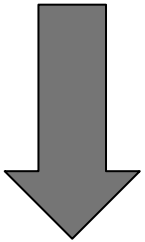
6. Lastly the output from the SSTB part, goes into the Patch-weighted Quality Prediction part.

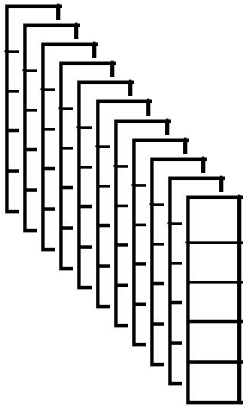
7. The final patch score of the distorted image is generated by multiplication of each patch's score and weight, then the final score of the whole image is generated by the summation of final patch scores.

KLT-SRQA

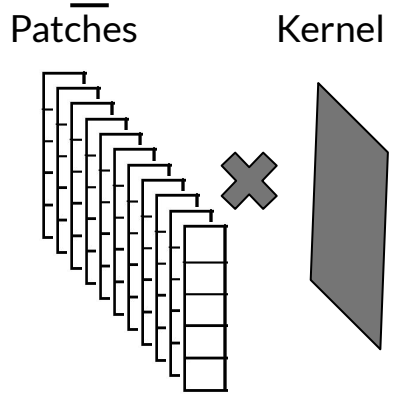
- Karhunen-Loéve Transform (KLT)
- We extract features from the output image, that we call KLT Features.
- In the reference research paper:
    - KLT Features mapped to Human Subjective Study Scores
    - Using a regression model.
- Instead of mapping KLT Features to a subjective score, we use the features from predicted image, and ground truth image to calculate a loss function and use it for our training.



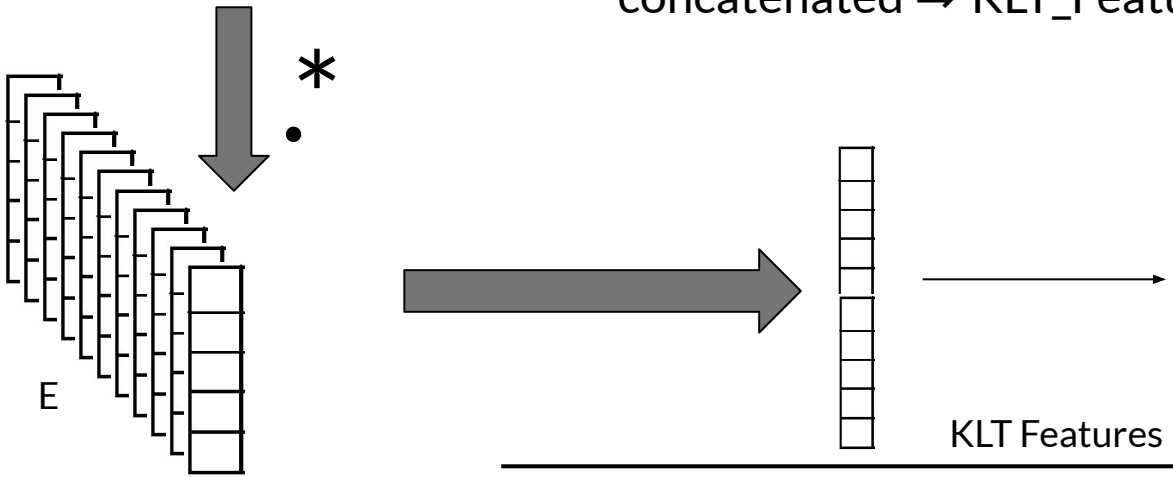




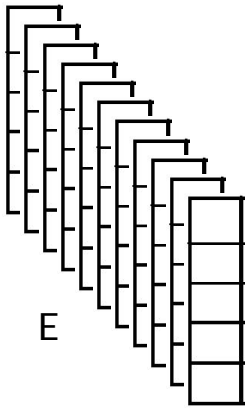
- Image $X \in R^{MxN}$; M = 120 & N=360 in our case
  - Color Space Change and MSCN (mean-subtracted-contrast-normalized)
- Extracting a set of non-overlapping $\sqrt{K}$ X $\sqrt{K}$ patches, where K = 64. Hence we obtain 675 Patches in total
- Patches: $X \in R^{675X\sqrt{K}X\sqrt{K}}$. Where K = 64
- Vectorizing the patches: $X \in R^{675X64}$



- Patches @ KLT_Kernel
  - @: matrix multiplication
  - KLT_Kernel: default Kernel taken from paper: $\in R^{1 X 64 X 64}$
  - Patches: $\in R^{N X 64 X 675}$
  - KLT_Kernel^{T} @ Patches $\Rightarrow$ KLT $\in R^{N X 64 X 675}$
- Energy and AGGD Parameters Computed and concatenated $\Rightarrow$ KLT_Features $\in R^{N X 777}$



- Pred_features = KLT_Features from predicted image
- Target_features = KLT_Features from target image
- Final Loss value = MSE(pred_features, target_features)



Energy Compute

- We fit the values in the E matrix to an exponential function: f(x, a, b, c) = a * exp(b*x) + c; and obtain parameters a, b, c…
  - Method used: Least Squares fit
  - X = range(0, 64), Ydata = values in E matrix
- Energy is given by f(x, a, b, c) once we obtain the optimal parameters a, b, c

AGGD Parameters:

- Asymmetric Generalized Gaussian Distribution.
- We fit the E matrix to an aggd.
- We avoid a usual gaussian distribution because that is symmetric and less generic.

- The Covariance matrix of $X \in R^{675X64}$ is defined as follows:

$$\begin{aligned}\mathbf{C} &= \mathbb{E}[(\mathbf{x}_s - \bar{\mathbf{x}})(\mathbf{x}_s - \bar{\mathbf{x}})^{\mathrm{T}}] \\ &= \frac{1}{S-1}\sum_{s=1}^{S}(\mathbf{x}_s - \bar{\mathbf{x}})(\mathbf{x}_s - \bar{\mathbf{x}})^{\mathrm{T}}\end{aligned}$$

- Where: $\bar{\mathbf{x}} = \frac{1}{S}\sum_{s=1}^{S}\mathbf{x}_s$ denotes the mean vector obtained by averaging each column of X and $\mathbf{C} \in R^{K X K}$.

Issues Faced While Implementing KLT-SRQA Loss.

- There was no initial code base available in python.
- Implemented the initial version in python, using various libraries like scipy, numpy, and pytorch.
  - Too much inference time because of CPU computations.
  - Decided to move computations to GPU, replacing all numpy functions and arrays with torch functions and tensors.
- Version Two was able to run on GPU, with little inference time, but overall the function was non-differentiable, so not able to perform back-propagation
  - Fix: Had to remove all scipy-library functions and code them from scratch using torch tensors, keeping in mind to maintain the tensors on same device while computations, avoiding non-differentiable operations, and keeping the Computation Graph of the loss function in form of a directed acyclic graph.

Key Takeaways

- In depth knowledge about Super Resolution in general - problems, requirements, what to do and what not to do etc.
- Use and comparison of various loss functions and their overall effect on outcomes
- How to submit and write a report for a formal publication

Future Improvements

- Incorporate new loss function(s) in the Image2Image Model.
- Increase the size of the model and train for a longer time
- Use a different architecture on top of NAFNET (instead of UNET)
  - CycleGAN
  - Pix2Pix
  - ESRGAN (Enhanced Super Resolution GAN)
  - Deep Image Prior
  - DeblurGAN
  - Neural Style Transfer
  - DeepDream

Challenges Faced

- Difficulty in super-resolution of natural scenes (Trees/Vegetation/Plants/Faces)
- Super-resolution of high level features. For example a complex network of tangled wires won't be super-resolved with much accuracy.
- Lack of adequate computing resources for this task
  - HPC
  - 3 X VMs
  - UG-PG Lab PCs
  - Underwater Lab PC

Our Current Publication - CVPR 2023

Publication - Link

Co-Authors - Pulkit Mahajan, Shreyas Chatterjee, Vinit Jakhetiya, Badri Subudhi, Sunil Jaiswal

References

1. NAFSSR: https://arxiv.org/abs/2204.08714
2. NAFNet: https://arxiv.org/pdf/2204.04676.pdf
3. UNet: https://arxiv.org/pdf/1505.04597.pdf
4. MANIQA: https://arxiv.org/pdf/2204.08958.pdf
5. SISRQA: https://ieeexplore.ieee.org/document/9727079
6. Knowledge Distillation: https://arxiv.org/pdf/1503.02531.pdf
7. ReCon: https://martina.lindorfer.in/files/papers/recon_mobisys16.pdf
8. AntShield: https://arxiv.org/pdf/1803.01261.pdf
9. Microsoft MEB Model: https://www.microsoft.com/en-us/research/blog/make-every-feature-binary-a-135b-parameter-sparse-neural-network-for-massively-improved-search-relevance/
10. Shreyas BTP 1 PPT - https://docs.google.com/presentation/d/1dq3PQUYps_oMP-hXNgYzp2gVopyOskV7IKQg8LRwisw/edit?usp=sharing
11. Pulkit BTP 1 PPT - https://docs.google.com/presentation/d/1-VhyvHMzsSjchfWdd6PI-wS7DIuixT89t6mI0-3a2e0/edit#slide=id.g19368f11819_2_57

Thank You