Vision Transformers for Image Restoration Problems

Team:

Ivan Gerasimov
Prateek Rajput
Vladimir Chernyy
Rustam Guseyn-zade
Hai Le

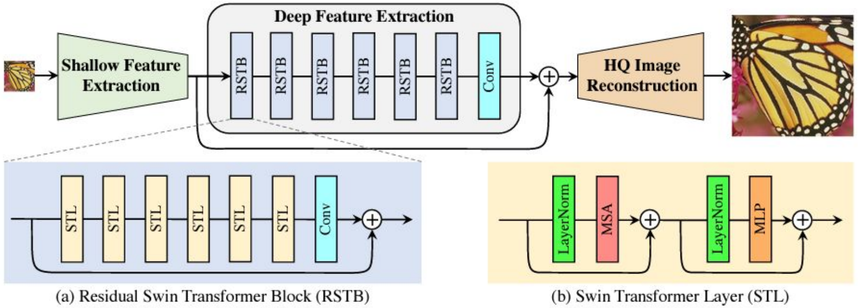
Problem Statement:

Replication: To train the SwinIR image restoration model with added gaussian noise and compare the results after denoising with the paper[1]. Also to add a projection layer in the model and see if the prior knowledge of known sigma affect the SSIM and PSNR measurement metrics

Research: To use the trained model on synthetically created blurred images and compare the results using the ISTA and FISTA optimization algorithms and comparing the similarity metrics with the results of the dataset used from the paper[2]

Papers 1 and 2 are posted on github: https://github.com/ctrlzet/imgrestore

Model and training parameters:



The architecture of the proposed SwinIR for image restoration

Model Training Parameters:

RSTB number: 6
STL number: 6
window size: 8
channel number: 180
attention head number: 6

Learning rate = 2e-5
Epochs = 400
batchsize: 6 (due to architecture constraints)

CNNs + Transformer blocks to extract local dependencies and long term dependencies in the image data respectively.

Gaussian noise: $y = x + \sigma \cdot n, n \sim \mathcal{N}(0, I)$

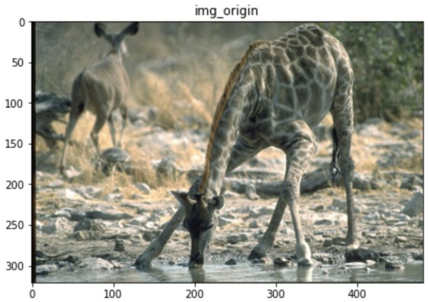
Noise used in training set: sigma [5,55]
Noise used in validation set: sigma{15,25,50}

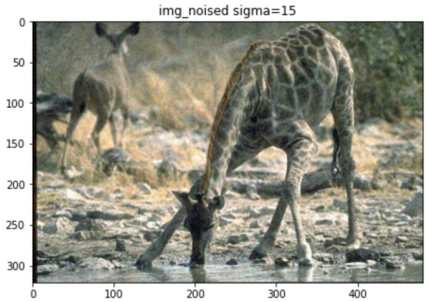
Projection Layer:

Projecting output of existing neural network to the following operator

$$\Pi_{\mathcal{C}}\left(\boldsymbol{v}\right) = \mathbf{y} + \varepsilon \frac{\boldsymbol{v} - \mathbf{y}}{\max\left(\left\|\boldsymbol{v} - \mathbf{y}\right\|_2, \varepsilon\right)}.$$

where $\varepsilon = e^{\alpha}\sigma\sqrt{N_t - 1}$

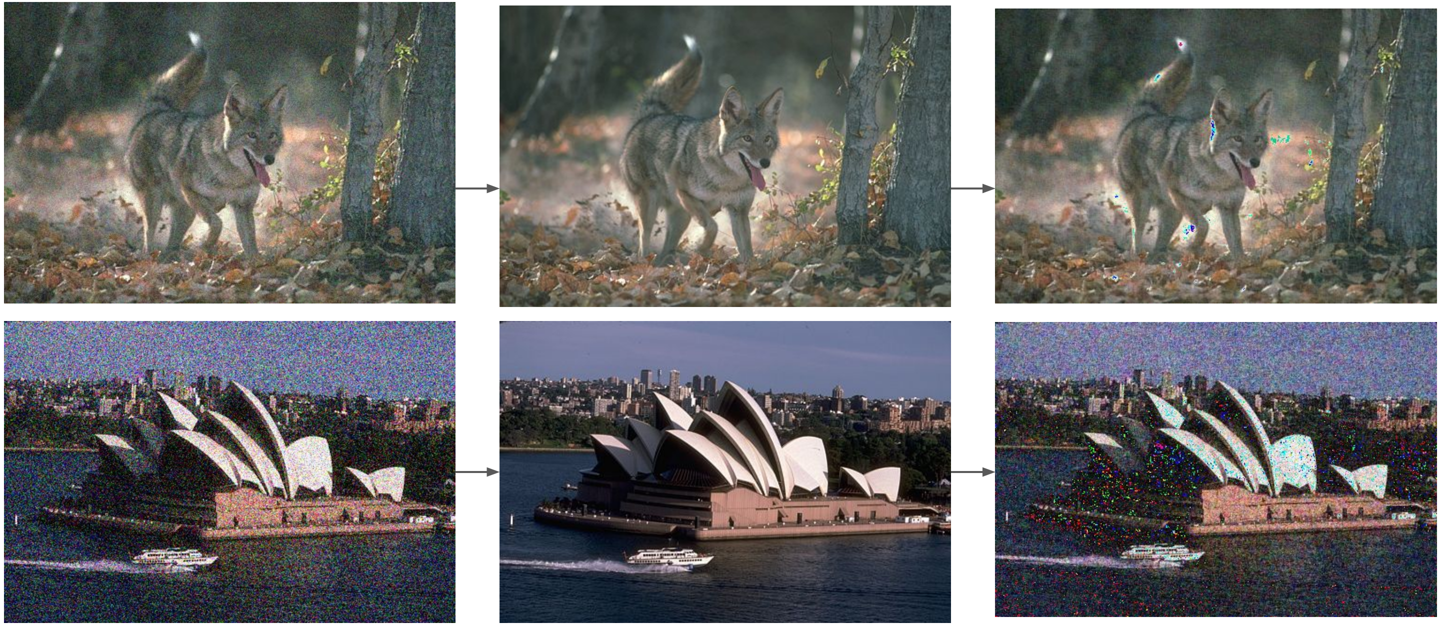




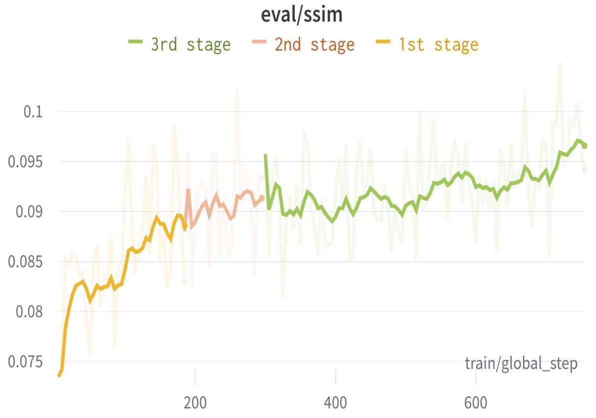
Introduce the projection layer as last module of neural network and train an alpha parameter

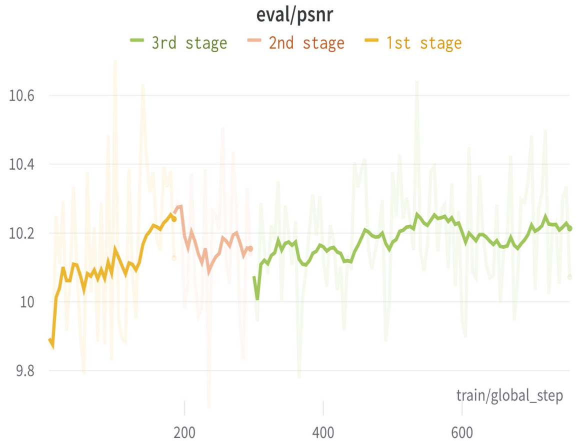
Sample images from the trained model:

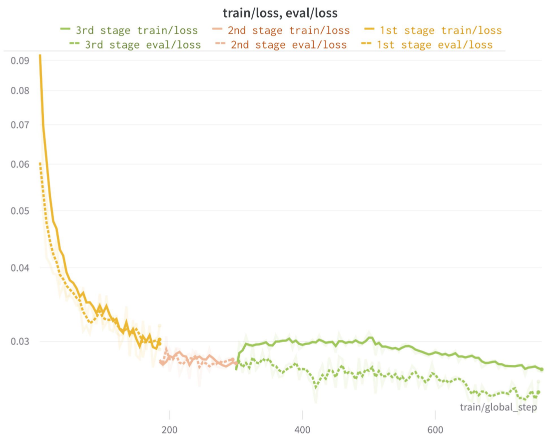
Input + noise | True value | Reconstructed result



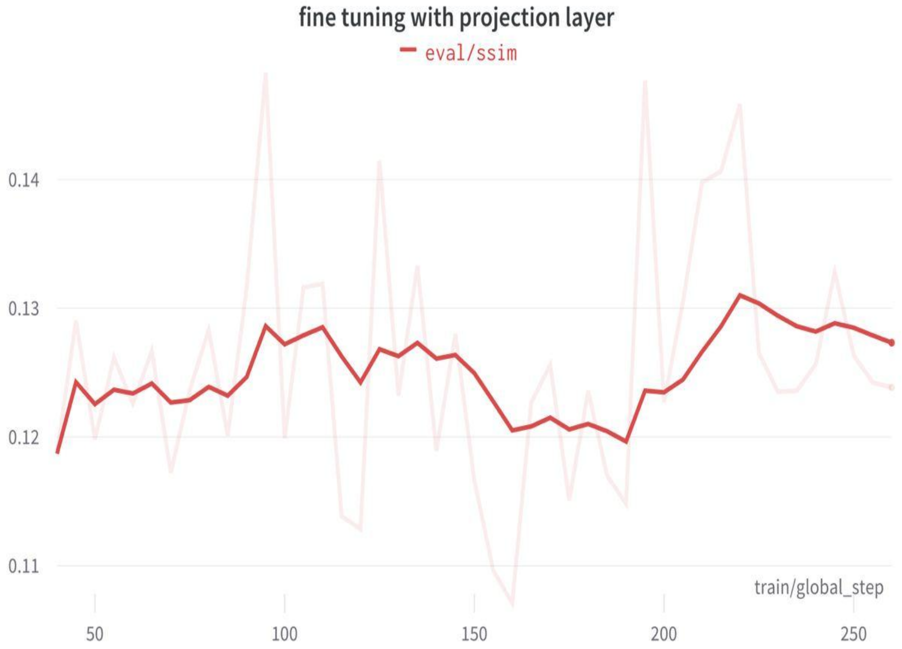
SSIM/PSNR for the Validation set without projection layer

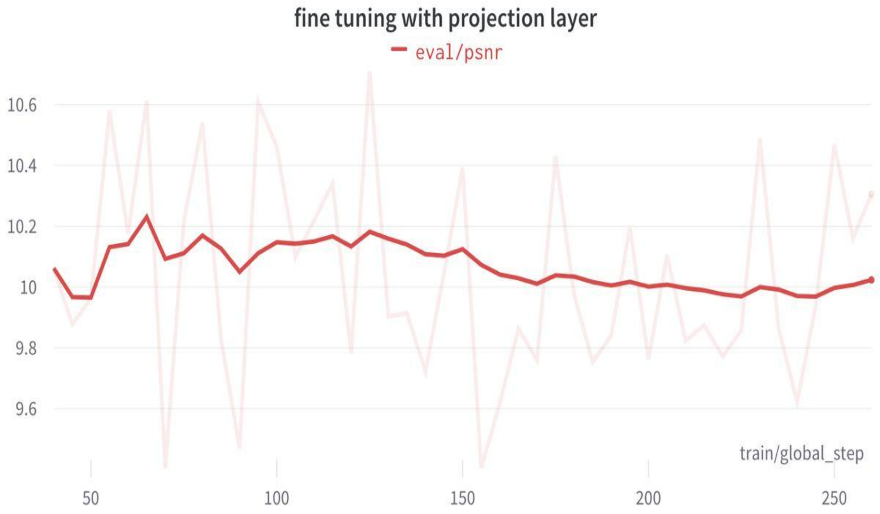






SSIM/PSNR for the Validation set with projection layer





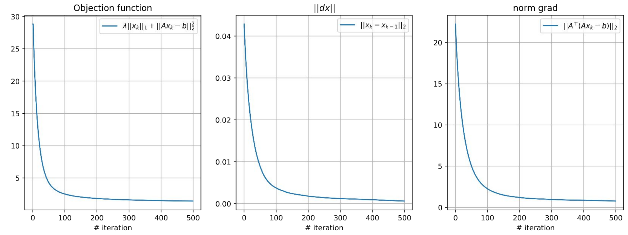
Research Part: understand algorithms and then implement them in the model

ISTA

ISTA with constant stepsize
Input: $L := L(f)$ - A Lipschitz constant of ∇f.
Step 0. Take $\mathbf{x}_0 \in \mathbb{R}^n$.
Step k. $(k \geq 1)$ Compute

$$\mathbf{x}_k = p_L(\mathbf{x}_{k-1}). \tag{3.1}$$

,where $p_L(\mathbf{y}) = \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ g(\mathbf{x}) + \frac{L}{2} \left\| \mathbf{x} - \left(\mathbf{y} - \frac{1}{L} \nabla f(\mathbf{y}) \right) \right\|^2 \right\}$



FISTA

FISTA with constant stepsize
Input: $L = L(f)$ - A Lipschitz constant of ∇f.
Step 0. Take $\mathbf{y}_1 = \mathbf{x}_0 \in \mathbb{R}^n$, $t_1 = 1$.
Step k. $(k \geq 1)$ Compute

$$\mathbf{x}_k = p_L(\mathbf{y}_k), \tag{4.1}$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \tag{4.2}$$

$$\mathbf{y}_{k+1} = \mathbf{x}_k + \left(\frac{t_k - 1}{t_{k+1}} \right) (\mathbf{x}_k - \mathbf{x}_{k-1}). \tag{4.3}$$