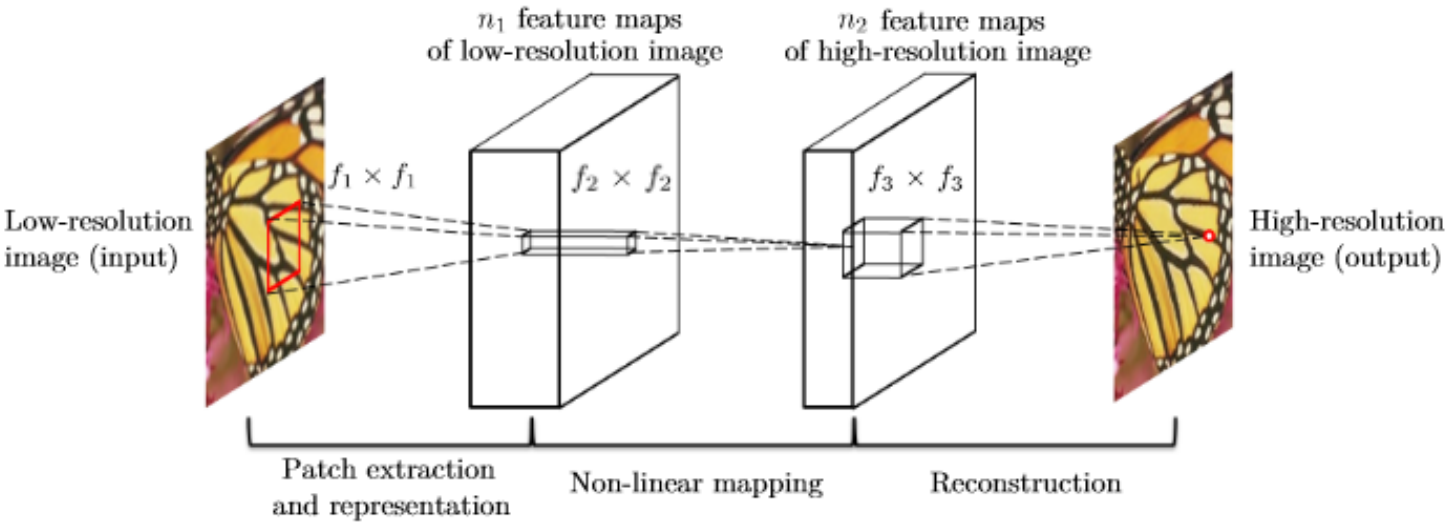
Super Resolution CNN

Christos Konstantas

Machine Learning Course

September 29, 2023

What is a super resolution CNN?



- **Patch Extraction:** Divide the image into patches.
- **Non-Linear Mapping:** Use convolutional layers for complex learning.
- **Reconstruction:** Combine mapped patches for high-resolution output.

Process Steps

- **Step 1:** Prepare data, convert to YCbCr, use Y channel for SRCNN input.
- **Step 2:** Convolve with $\boldsymbol{W}_1$ and apply ReLU.
- **Step 3:** Repeat Step 2 with $\boldsymbol{W}_2$ and ReLU.
- **Step 4:** Reconstruction without non-linearity using $\boldsymbol{W}_3$.
- **Additional Layers:** More layers can be added to SRCNN.
- **Loss Function:** Minimize MSE with optimizer.

Adam Optimizer (Simplified)

Algorithm Adam Optimizer (Simplified)

Initialize model parameters θ, step size α, β_1, β_2, and ϵ
Initialize first moment vector m and second moment vector v with zeros
Initialize time step $t \leftarrow 0$
while Not converged **do**
    Sample mini-batch data: inputs X and targets Y
    Compute gradients $\nabla_\theta \mathcal{L}(\theta; X, Y)$
    Increment time step $t \leftarrow t + 1$
    Update moments: $m \leftarrow \beta_1 \cdot m + (1 - \beta_1) \cdot \nabla_\theta \mathcal{L}(\theta; X, Y)$, $v \leftarrow \beta_2 \cdot v + (1 - \beta_2) \cdot (\nabla_\theta \mathcal{L}(\theta; X, Y))^2$
    Correct biases: $\hat{m} \leftarrow \frac{m}{1-\beta_1^t}$, $\hat{v} \leftarrow \frac{v}{1-\beta_2^t}$
    Compute parameter update: $\Delta\theta \leftarrow -\alpha \cdot \frac{\hat{m}}{\sqrt{\hat{v}}+\epsilon}$
    Update model parameters: $\theta \leftarrow \theta + \Delta\theta$
end while=0

Training the SRCNN

- Load HR and corresponding LR images, plus validation data.
- Use mini-batches and gradient accumulation.
- Perform forward-backward passes for loss optimization.
- Adapt learning rate during training.
- Evaluate on validation data; early stop if validation loss increases.
- Save the best model.

Process Synopsis

- Use SRCNN for image enhancement.
- Convert RGB to YCbCr, focus on Y channel.
- Combine reconstructed Y with bicubic interpolated Cb and Cr.
- Convert YCbCr back to RGB.
- Start with single-image processing, then evaluate on datasets.
- Focus on enhancement rather than upscaling; use a transpose convolution for upscaling if needed.

Y Cb Cr color space (1)

- Separates luminance (Y) from chrominance (Cb and Cr).
- Efficient for image and video compression.
- Conversion from RGB to YCbCr using equations.
- Train models to enhance Y (luminance) while keeping Cb and Cr unchanged.
- Reasons: Human visual perception, reduced complexity.
- Reconstruct RGB from YCbCr using specific equations.

Y Cb Cr color space (2)

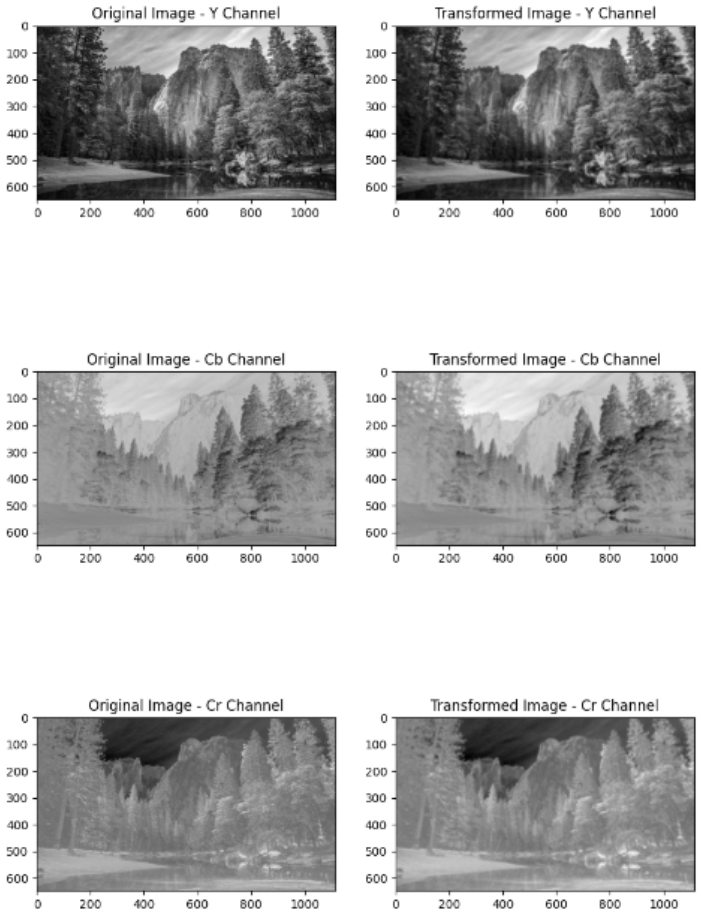


Figure: Y Cb Cr color channels in an image.

PSNR (Peak Signal to Noise Ratio)

- PSNR, or Peak Signal-to-Noise Ratio, is a metric for image quality comparison.
- It measures image quality relative to the original image.
- PSNR is calculated using Mean Squared Error (MSE).
- The formula for PSNR is:

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

- Where:
  - PSNR represents the Peak Signal-to-Noise Ratio.
  - MAX is the maximum possible pixel value (e.g., 255 for an 8-bit image).
- A higher PSNR value indicates better image quality and less distortion.

MSE (Mean Squared Error)

- MSE, or Mean Square Error, is a widely used metric in image processing.
- It measures the average squared difference between values in the original and processed images.
- MSE is calculated using the following formula:

$$\text{MSE} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(I(i,j) - K(i,j) \right)^2$$

- Where:
  - MSE represents the Mean Square Error.
  - $I(i,j)$ is the pixel value of the original image at position (i,j).
  - $K(i,j)$ is the pixel value of the processed or reconstructed image at position (i,j).
  - m and n are the dimensions of the image.
- A lower MSE indicates a smaller difference between the original and processed images, suggesting better image quality.

SSIM (Structural Similarity Index)

- SSIM, or Structural Similarity Index, is a metric that evaluates structural similarity between two images, including luminance, contrast, and structure.
- It is computed using the formula:

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

- Where:
  - $\text{SSIM}(x, y)$ is the SSIM between images x and y.
  - μ_x and μ_y are the means of images x and y.
  - σ_x and σ_y are the standard deviations of images x and y.
  - σ_{xy} is the covariance between images x and y.
  - c_1 and c_2 are constants used for stability in the division.
- The SSIM value ranges from -1 to 1, where 1 indicates perfect similarity. In the project, SSIM is normalized and ranges from 0 to 1.
- Higher SSIM values imply better image quality and similarity.

Single image enhancement (no generalization)

Figure: Scale Factor = 4

Figure: Scale Factor = 3

Set14 Fast Training Results

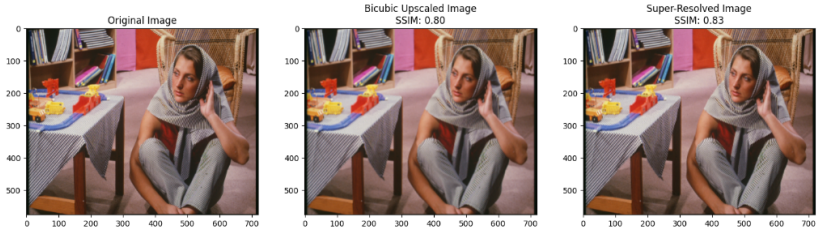


Figure: SSIM on barbara.png of Set 5

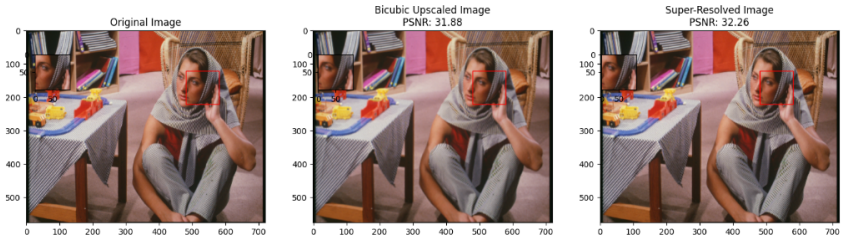


Figure: PSNR on barbara.png of Set 5

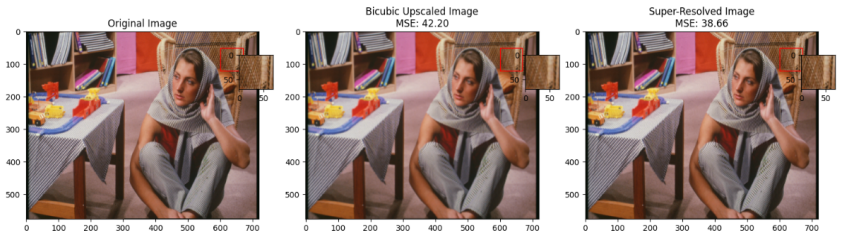


Figure: MSE on barbara.png of Set 5

DIV2K training results

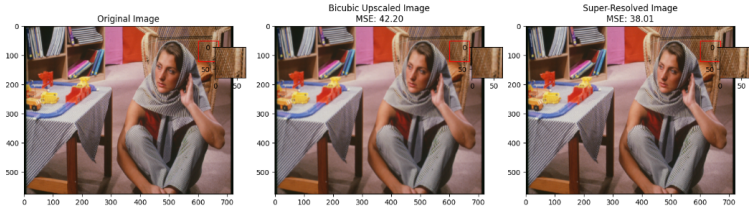


Figure: Better image quality for DIV2K training on barbara.png

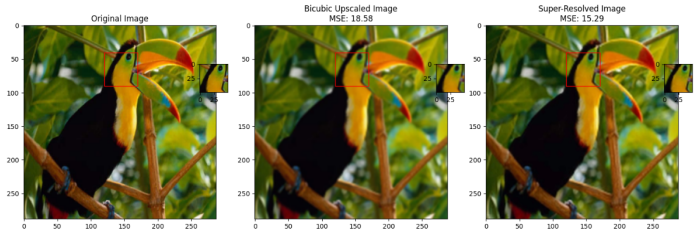


Figure: bird.png from set 5

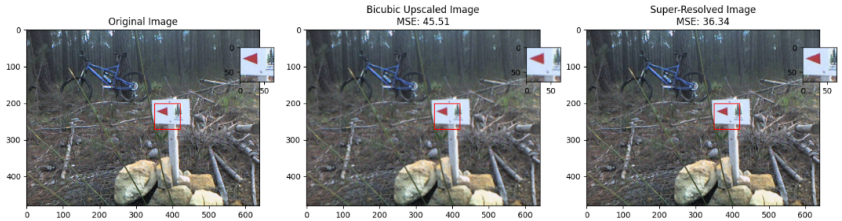


Figure: Nature image from an amateur photographer