Practical deep learning workshop 4 - Image super-resolution exercise

Introduction

The goal of this task is to gain practical experience in building and training fully convolutional networks for image super-resolution. Using the PascalVOC 2007 dataset, we developed network architectures in different forms in order to study and understand their impact on image resolution. The goal was to create a network that can accept a low-resolution 72x72 image and upscale it to a higher resolution. To do this, we explored and studied this world and decided to use a loss function called PSNR.

PSNR (Peak Signal-to-Noise Ratio) is a common measure of image quality, mainly used to compare original images to reconstructed/compressed images..

Dataset Preparation

- **Dataset Source:** The PascalVOC 2007 dataset was downloaded and extracted to serve as the base for our image super-resolution tasks.
- **Image Resizing:**
    - **X:** 72x72x3
    - **y_mid:** 144x144x3
    - **y_large:** 288x288x3

To expedite development, we initially worked on a smaller sample of the dataset and validated our pipeline before scaling up.

- **Data Splitting:** The dataset was split into training and validation sets:
    - **Training:** 4011 images
    - **Validation:** 1000 images
- **Visualization:** Sample images from each size were displayed to ensure proper preprocessing

Step 2

Model Architectures and Training

Initial Model: SuperResNet

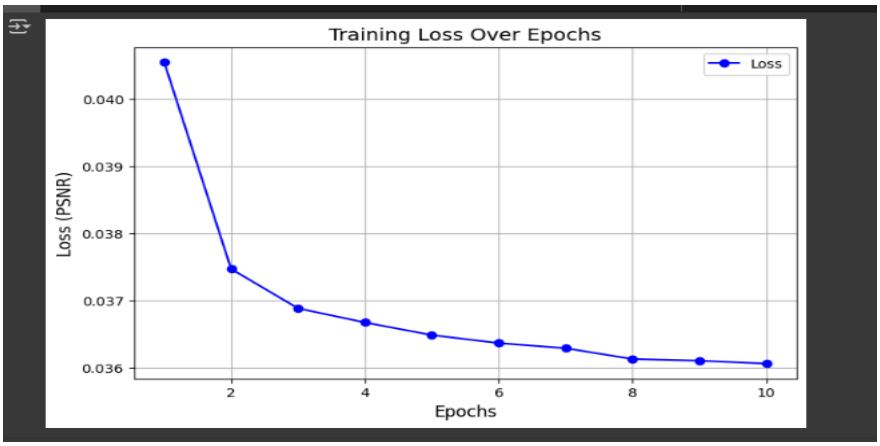
- **Architecture:** The initial model, named *SuperResNet*, is a fully convolutional network designed for basic image super-resolution. It includes the following components:
    - **Conv2D Layer 1:** 64 filters, kernel size 3x3, padding 1
    - **Conv2D Layer 2:** 64 filters, kernel size 3x3, padding 1

- **Upsample Layer:** Bilinear upsampling with a scale factor of 2
- **Conv2D Layer 3:** 3 filters, kernel size 1x1 for output
- **Activations:** ReLU activations after the first two convolutional layers and a sigmoid activation after the final layer

Training Setup:

- **Device:** CPU (fallback in absence of GPU)
- **Loss Function:** PSNRLoss
- **Optimizer:** Adam with a learning rate of 0.001
- **Epochs:** 10
- **Results:**

- **Loss over Epochs:**
  - Epoch 1: Loss = 0.0405
  - Epoch 2: Loss = 0.0375
  - Epoch 3: Loss = 0.0369
  - Epoch 4: Loss = 0.0367
  - Epoch 5: Loss = 0.0365
  - Epoch 6: Loss = 0.0364
  - Epoch 7: Loss = 0.0363
  - Epoch 8: Loss = 0.0361
  - Epoch 9: Loss = 0.0361
  - Epoch 10: Loss = 0.0361



The model showed steady improvement in loss reduction over the epochs, indicating effective learning during the training process.

Step 3

Dual-Output Model: SuperResNetMulti

- **Architecture:** The second model, *SuperResNetMulti*, builds on the initial architecture by generating outputs at two different resolutions: 144x144 (mid) and 288x288 (large). The architecture components include:
    - **Conv2D Layer 1:** 64 filters, kernel size 3x3, padding 1
    - **Conv2D Layer 2:** 64 filters, kernel size 3x3, padding 1
    - **Upsample Layer (Mid):** Bilinear upsampling to scale input by a factor of 2
    - **Conv2D Layer (Mid):** 3 filters, kernel size 1x1 for mid-resolution output
    - **Upsample Layer (Large):** Additional bilinear upsampling applied to the mid-resolution output
    - **Conv2D Layer (Large):** 3 filters, kernel size 1x1 for large-resolution output
    - **Activations:** ReLU activations after convolutional layers and sigmoid activations for both outputs

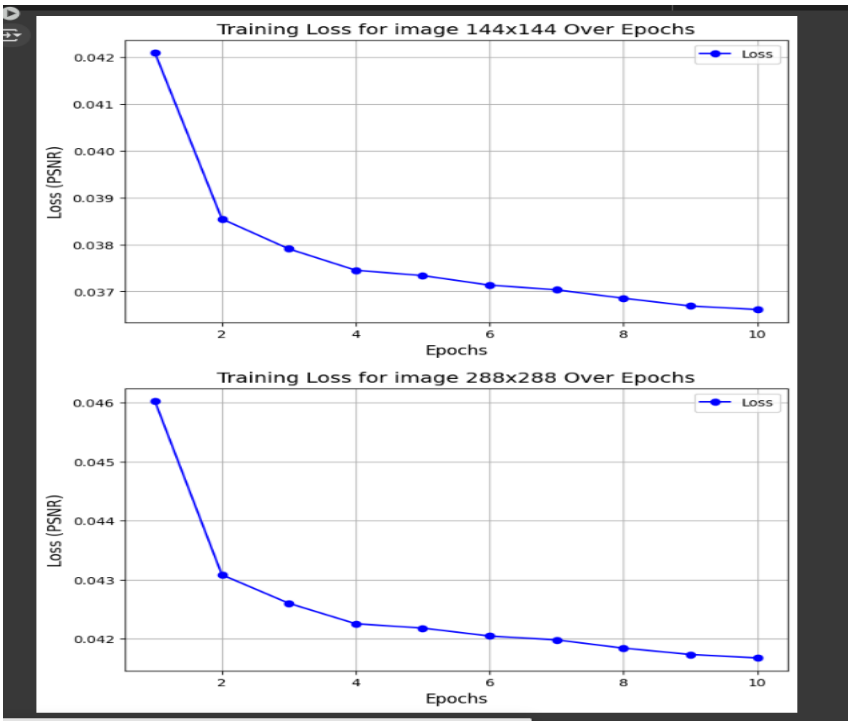
Training Setup:

- **Loss Function:** PSNRLoss
- **Optimizer:** Adam with a learning rate of 0.001
- **Epochs:**10
- **Results:**

Loss over Epochs:

- **Epoch [1/10], Mid Loss: 0.0421, Large Loss: 0.0460**
- **Epoch [2/10], Mid Loss: 0.0385, Large Loss: 0.0431**
- **Epoch [3/10], Mid Loss: 0.0379, Large Loss: 0.0426**
- **Epoch [4/10], Mid Loss: 0.0375, Large Loss: 0.0423**
- **Epoch [5/10], Mid Loss: 0.0373, Large Loss: 0.0422**
- **Epoch [6/10], Mid Loss: 0.0371, Large Loss: 0.0420**
- **Epoch [7/10], Mid Loss: 0.0370, Large Loss: 0.0420**

- **Epoch [8/10], Mid Loss: 0.0369, Large Loss: 0.0418**
- **Epoch [9/10], Mid Loss: 0.0367, Large Loss: 0.0417**
- **Epoch [10/10], Mid Loss: 0.0366, Large Loss: 0.0417**



The dual-output model demonstrated consistent loss reduction across both mid and large resolution outputs, suggesting balanced performance improvements in super-resolution tasks.

Step 4

Residual Blocks Model: SuperResNetResidualBlock

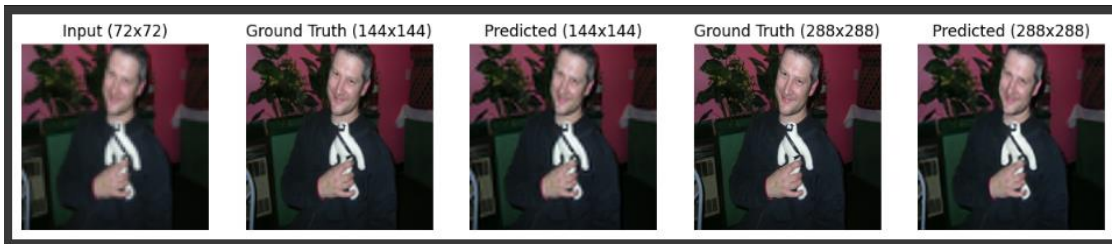
- **Architecture:** The third model, *SuperResNetResidualBlock*, incorporates residual blocks to improve feature learning and mitigate vanishing gradient issues. Residual connections help the model learn more effectively by allowing gradients to flow directly through skip connections. The architecture includes:
    - **Conv2D Layer 1:** 32 filters, kernel size 3x3, padding 1
    - **Residual Block 1 & 2:** Each containing two Conv2D layers with 32 filters, kernel size 3x3, and ReLU activations
    - **Upsample Layer 1:** Bilinear upsampling with a scale factor of 2 for mid-resolution output
    - **Conv2D Layer (Mid):** 3 filters, kernel size 1x1 for 144x144 output
    - **Residual Block 3:** Applied after mid-resolution upsampling

- **Upsample Layer 2:** Further bilinear upsampling for large-resolution output
- **Conv2D Layer (Large):** 3 filters, kernel size 1x1 for 288x288 output

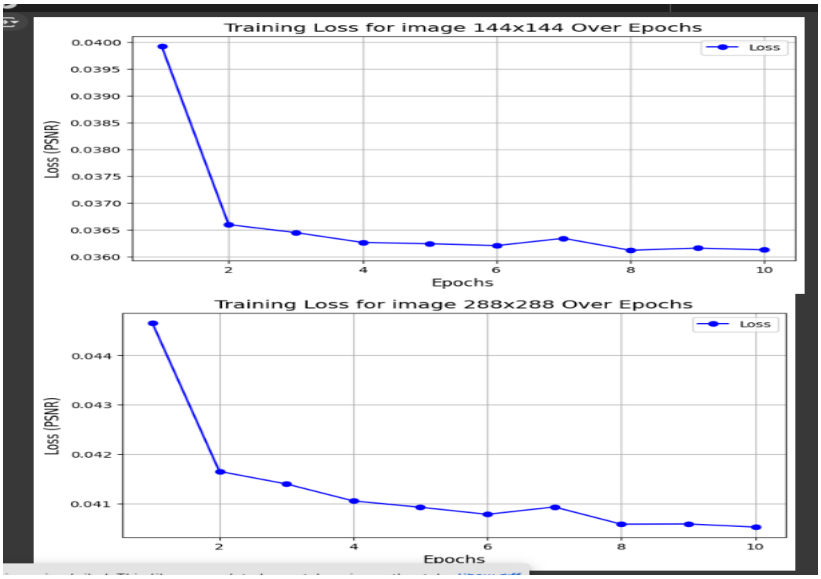
Training Setup:

- **Loss Function:** PSNRLoss
- **Optimizer:** Adam with a learning rate of 0.001
- **Epochs:** 10

Results:



- **Loss over Epochs:**
  - Epoch [1/10], Mid Loss: 0.0399, Large Loss: 0.0447
  - Epoch [2/10], Mid Loss: 0.0366, Large Loss: 0.0416
  - Epoch [3/10], Mid Loss: 0.0364, Large Loss: 0.0414
  - Epoch [4/10], Mid Loss: 0.0363, Large Loss: 0.0411
  - Epoch [5/10], Mid Loss: 0.0362, Large Loss: 0.0409
  - Epoch [6/10], Mid Loss: 0.0362, Large Loss: 0.0408
  - Epoch [7/10], Mid Loss: 0.0363, Large Loss: 0.0409
  - Epoch [8/10], Mid Loss: 0.0361, Large Loss: 0.0406
  - Epoch [9/10], Mid Loss: 0.0362, Large Loss: 0.0406
  - Epoch [10/10], Mid Loss: 0.0361, Large Loss: 0.0405



The introduction of residual blocks resulted in more stable training and improved performance, particularly noticeable in the large-resolution output.

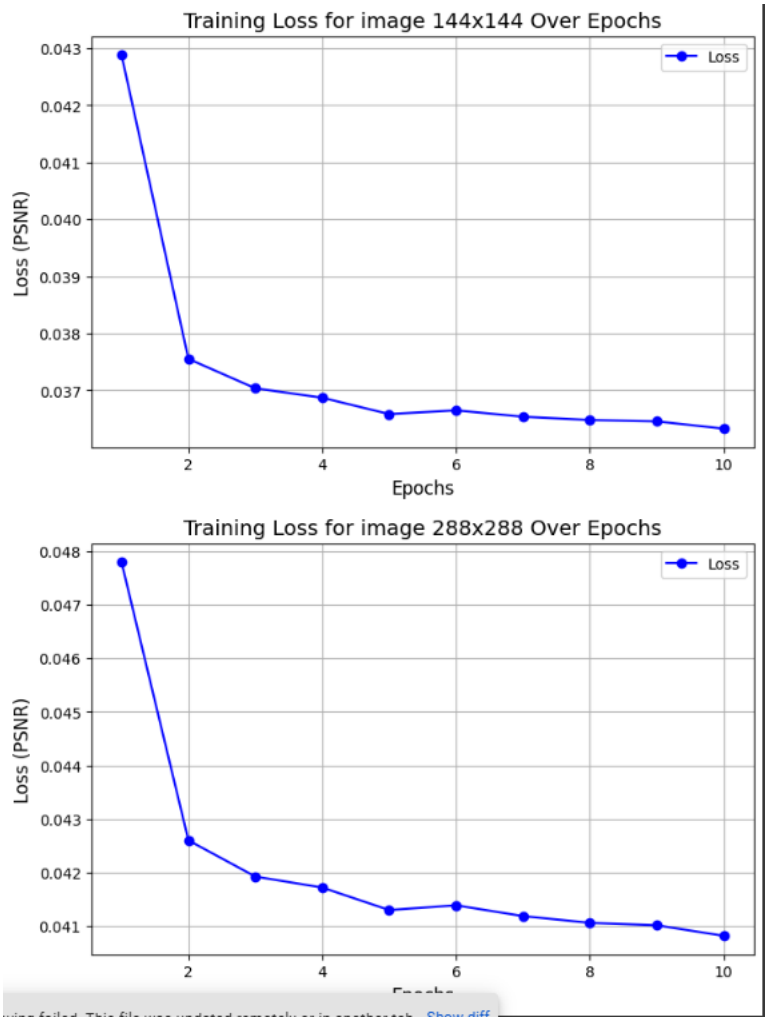
Step 5

Dilated Convolutions Model: SuperResNetDilated

- **Architecture:** The fourth model, *SuperResNetDilated*, introduces dilated (atrous) convolutions to enhance the receptive field without increasing the number of parameters significantly. This approach allows the model to capture multi-scale contextual information effectively. The architecture consists of:
    - **Initial Conv2D Layer:** 32 filters, kernel size 3x3, padding 1
    - **DilatedConvBlock 1 & 2:** Each containing three Conv2D layers with dilation rates of 1, 2, and 4, followed by concatenation and a final convolution
    - **Upsample Layer 1:** Bilinear upsampling to achieve mid-resolution (144x144)
    - **Conv2D Layer (Mid):** 3 filters, kernel size 1x1 for mid-resolution output
    - **DilatedConvBlock 3:** Applied after the first upsampling to further refine features
    - **Upsample Layer 2:** Additional bilinear upsampling to achieve large-resolution (288x288)
    - **Conv2D Layer (Large):** 3 filters, kernel size 1x1 for large-resolution output
    -
- **Training Setup:**
    - **Loss Function:** PSNRLoss
    - **Optimizer:** Adam with a learning rate of 0.001
    - **Epochs:** 10
- **Results:**

    - **Loss over Epochs:**
        - Epoch [1/10], Mid Loss: 0.0429, Large Loss: 0.0478
        - Epoch [2/10], Mid Loss: 0.0376, Large Loss: 0.0426
        - Epoch [3/10], Mid Loss: 0.0370, Large Loss: 0.0419
        - Epoch [4/10], Mid Loss: 0.0369, Large Loss: 0.0417
        - Epoch [5/10], Mid Loss: 0.0366, Large Loss: 0.0413
        - Epoch [6/10], Mid Loss: 0.0366, Large Loss: 0.0414

- Epoch [7/10], Mid Loss: 0.0365, Large Loss: 0.0412
- Epoch [8/10], Mid Loss: 0.0365, Large Loss: 0.0411
- Epoch [9/10], Mid Loss: 0.0365, Large Loss: 0.0410
- Epoch [10/10], Mid Loss: 0.0363, Large Loss: 0.0408



The use of dilated convolutions enabled the model to capture multi-scale features effectively, leading to steady improvements in both mid and large-resolution outputs.

Step 6

Pretrained Feature Extractor Model: SuperResNetWithVGG

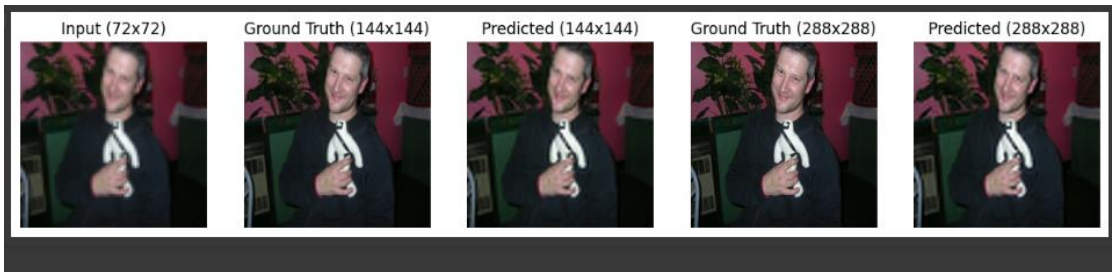
- **Architecture:** The fifth model, *SuperResNetWithVGG*, incorporates a pretrained VGG16 network as a feature extractor. This enhances the super-resolution process by leveraging rich, high-level features from a network trained on a large image dataset. The architecture includes:

- **Initial Conv2D Layers:** Two convolutional layers with 64 filters, kernel size 3x3, and padding 1
- **Feature Extractor:** VGG16's first four convolutional layers used as a fixed feature extractor (parameters are frozen)
- **Merging Layer:** Concatenation of VGG features with the main network output, followed by a Conv2D layer with 64 filters to merge them
- **Upsample Layer 1:** Bilinear upsampling to achieve mid-resolution (144x144)
- **Conv2D Layer (Mid):** 3 filters, kernel size 1x1 for mid-resolution output
- **Upsample Layer 2:** Further bilinear upsampling to achieve large-resolution (288x288)
- **Conv2D Layer (Large):** 3 filters, kernel size 1x1 for large-resolution output

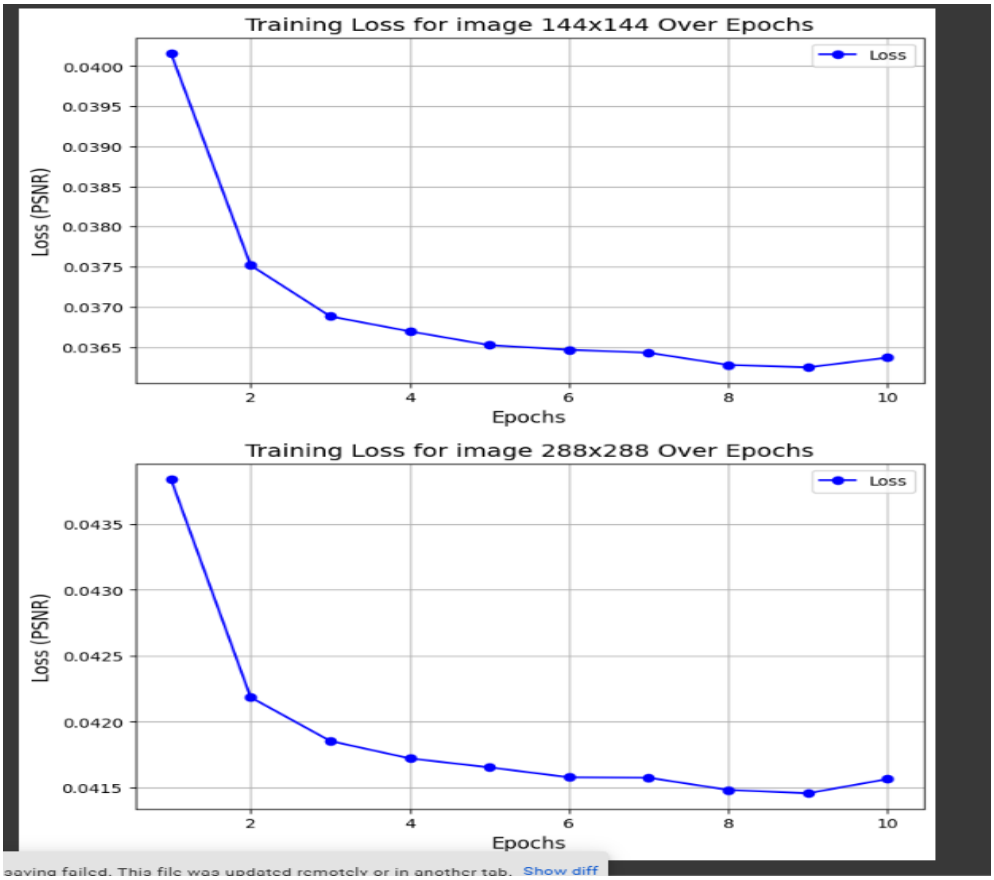
Training Setup:

- **Loss Function:** PSNRLoss
- **Optimizer:** Adam with a learning rate of 0.001
- **Epochs:** 10

Results:



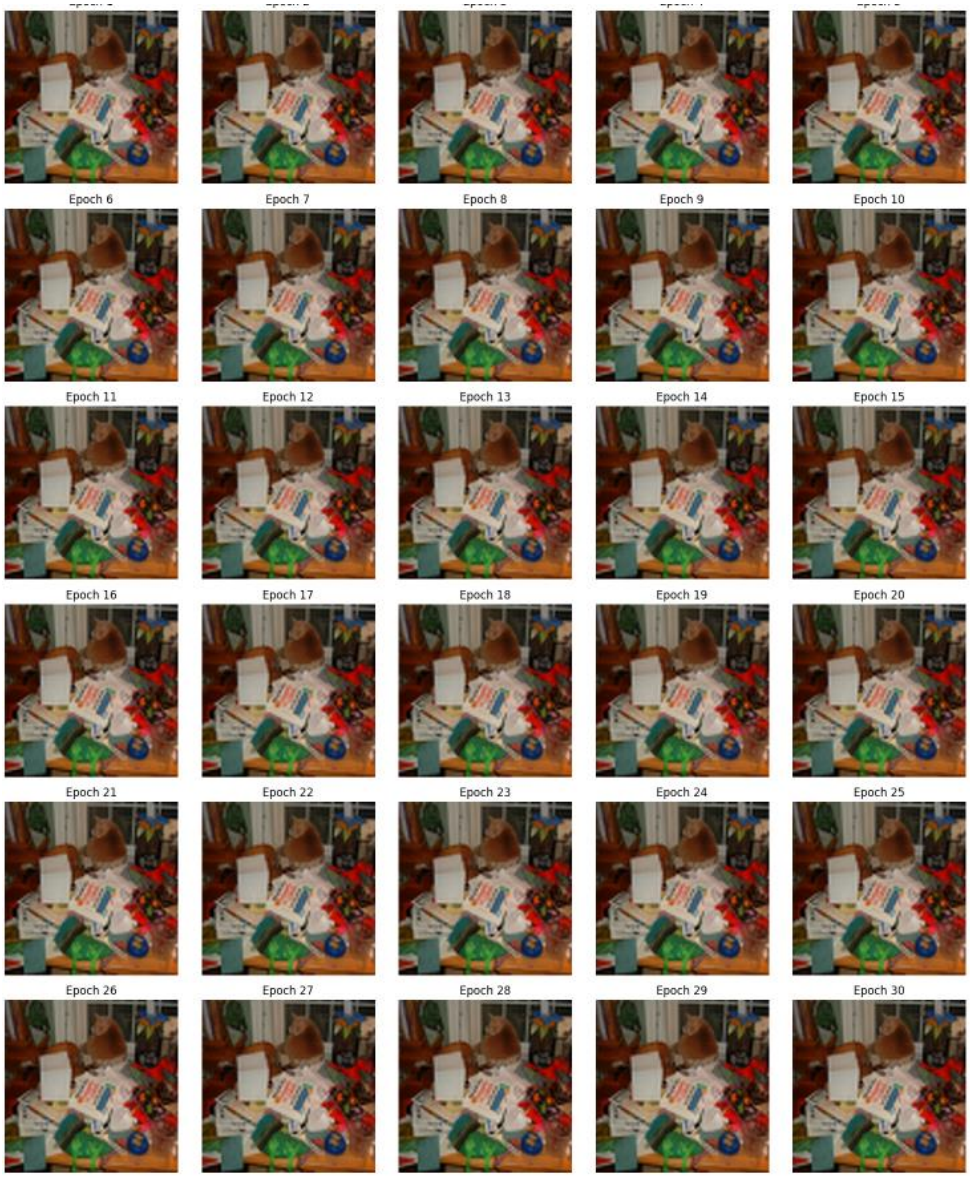
- **Loss over Epochs:**
  - Epoch [1/10], Mid Loss: 0.0402, Large Loss: 0.0438
  - Epoch [2/10], Mid Loss: 0.0375, Large Loss: 0.0422
  - Epoch [3/10], Mid Loss: 0.0369, Large Loss: 0.0419
  - Epoch [4/10], Mid Loss: 0.0367, Large Loss: 0.0417
  - Epoch [5/10], Mid Loss: 0.0365, Large Loss: 0.0417
  - Epoch [6/10], Mid Loss: 0.0365, Large Loss: 0.0416
  - Epoch [7/10], Mid Loss: 0.0364, Large Loss: 0.0416
  - Epoch [8/10], Mid Loss: 0.0363, Large Loss: 0.0415
  - Epoch [9/10], Mid Loss: 0.0362, Large Loss: 0.0415
  - Epoch [10/10], Mid Loss: 0.0364, Large Loss: 0.0416



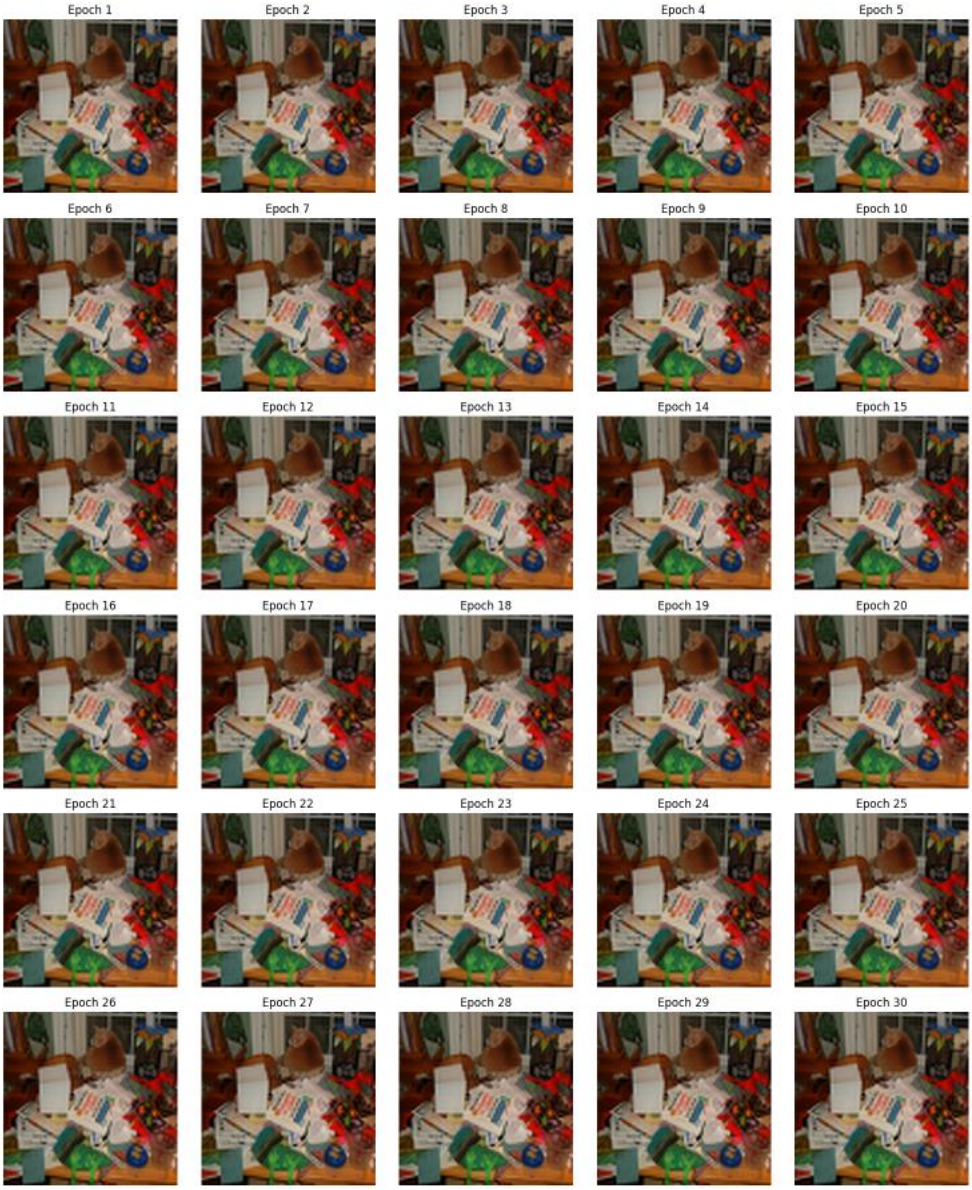
- Incorporating VGG16 as a feature extractor significantly improved model performance, particularly in capturing detailed textures and structural information in both mid and large-resolution outputs.

Step 7:

144x144



288x288:



Step 8:

The SuperResNetWithCLIP model integrates a pretrained CLIP model as a feature extractor, replacing the traditional use of VGG16. This approach leverages CLIP's multi-modal feature extraction capabilities to enhance the super-resolution process by incorporating semantic-rich embeddings from a model trained on large-scale image-text datasets.

Initial Conv2D Layers:

Two convolutional layers with 64 filters, kernel size 3x3, and padding 1

These layers extract low-level spatial features from the input image.

Feature Extractor: CLIP Model

Uses CLIP's ViT-B/32 as a feature extractor, processing the image into 512-dimensional embeddings.

The extracted features are expanded to match the spatial resolution of the convolutional layers.

Parameters of CLIP are frozen to leverage its pretrained knowledge.

Merging Layer:

CLIP features (512 channels) are concatenated with the main convolutional features (64 channels).

This results in a 576-channel tensor, which is processed through a Conv2D layer (64 filters, kernel 3x3, padding 1) to merge the information.

Upsample Layer 1:

Bilinear upsampling is applied to generate a mid-resolution output (144x144).

Conv2D Layer (Mid-Resolution Output):

A 1x1 Conv2D layer with 3 filters to generate the final mid-resolution image.

Upsample Layer 2:

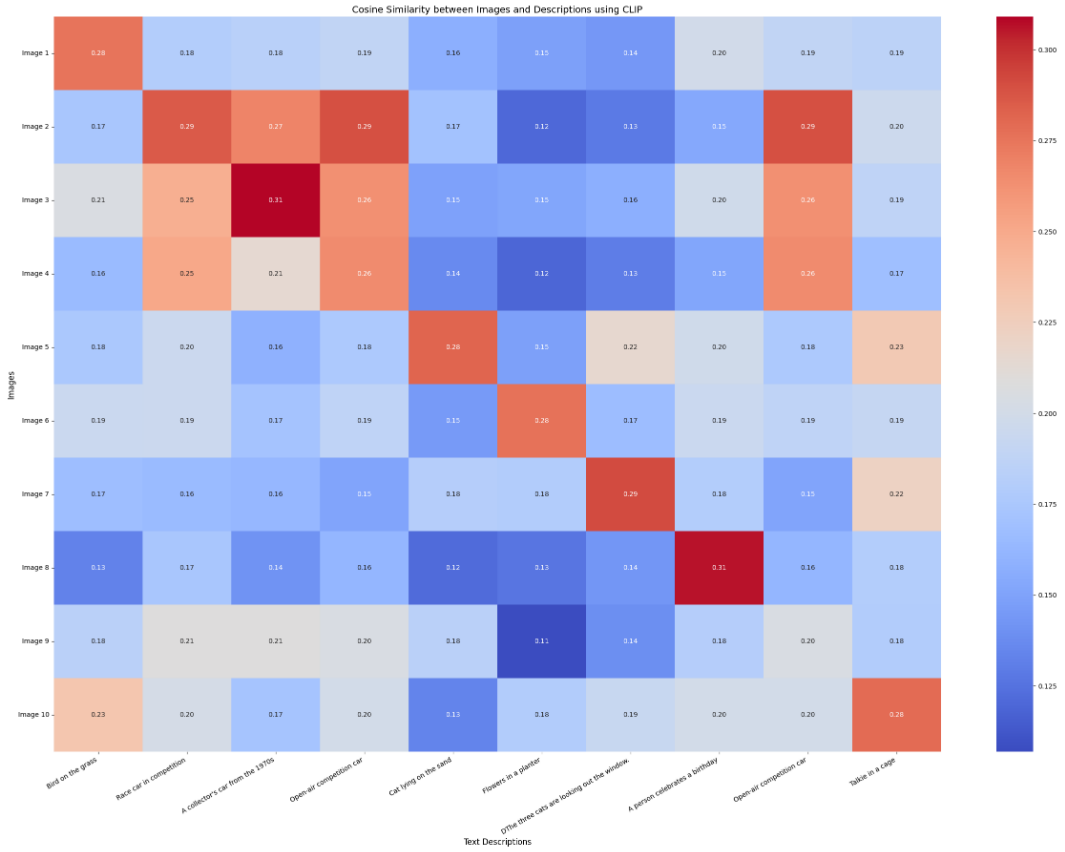
Another bilinear upsampling step increases the resolution to large-resolution (288x288).

Conv2D Layer (Large-Resolution Output):

A 1x1 Conv2D layer with 3 filters to generate the final large-resolution image.

The image white descriptions:

The cosine_matrix:


Cosine Similarity between Images and Descriptions using CLIP
Images
Image 1
Image 2
Image 3
Image 4
Image 5
Image 6
Image 7
Image 8
Image 9
Image 10
0.28 0.18 0.18 0.19 0.16 0.15 0.14 0.20 0.19 0.19
0.17 0.29 0.27 0.29 0.17 0.12 0.13 0.15 0.29 0.20
0.21 0.25 0.31 0.26 0.15 0.15 0.16 0.20 0.26 0.19
0.16 0.25 0.21 0.26 0.14 0.12 0.13 0.15 0.26 0.17
0.18 0.20 0.16 0.18 0.28 0.15 0.22 0.20 0.18 0.23
0.19 0.19 0.17 0.19 0.15 0.26 0.17 0.19 0.19 0.19
0.17 0.16 0.16 0.15 0.18 0.18 0.29 0.18 0.15 0.22
0.13 0.17 0.14 0.16 0.12 0.13 0.14 0.31 0.16 0.18
0.18 0.21 0.21 0.20 0.18 0.11 0.14 0.18 0.20 0.18
0.23 0.20 0.17 0.20 0.13 0.18 0.19 0.20 0.20 0.28
Bird on the grass
Race car in competition
A collector's car from the 1970s
Open-air competition car
Cat lying on the sand
Flowers in a planter
Three three cats are looking out the window.
A person celebrates a birthday
Open-air competition car
Tabby in a cage
Text Descriptions
0.300
0.275
0.250
0.225
0.200
0.175
0.150
0.125