# Super Resoultion Image Generation by Prompt and SRGAN

**Sudeep Ranjan Sahoo**
12141600
**Siddhi Agarwal**
12141570
**Konduri Naga Lakshmi Rekha**
12140930

## 1 Introduction

Generative Adversarial Networks (GANs) revolutionize computer vision, especially in SingleImage Super-Resolution (SR). SRGANs enhance low-res images, transcending edge sharpening to capture shape, color, and texture nuances. They find applications in surveillance, entertainment, and medical imaging. A pivotal aspect is the training dataset, shaping the network's ability to create precise, high-res images. SRGANs, thus, grasp and replicate object features, not just edges. This innovation holds immense potential across industries.

## 2 Initial Approach

Our approach aims to address this challenge by introducing a user-guided enhancement system. Users will provide a low-resolution image and specific prompts detailing the desired enhancements. This combined input will be fed into an SRGAN model with an attention mechanism. The attention model allows the network to focus its resources on the most relevant regions, optimizing the enhancement process.

We started with the implementation of the paper titled **"Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network."** Its implementation is done in the Pytorch library.

**Network Architecture mentioned in the paper**:

1. **Generator Architecture:** The generator in the SRGAN implementation employs a specific architecture. It utilizes a kernel size of 9, 64 channels, and a stride of 1. Residual blocks play a significant role in enhancing the discriminator's performance. Additionally, two novel concepts, PRelu and PixelShuffler, are introduced in the network architecture. The generator's role is to upsample low-resolution images to high-resolution images.
2. **Discriminator Architecture:** An additional dropout layer is integrated into the architecture to prevent the discriminator from dominating the generator during training.

**Loss Functions**:

1. **Discriminator Loss:** The discriminator is trained to distinguish between real high-resolution images and generated images. It uses binary cross-entropy loss (BCELoss) to compute the adversarial loss.

2. **Generator Loss:** The generator loss includes three components:

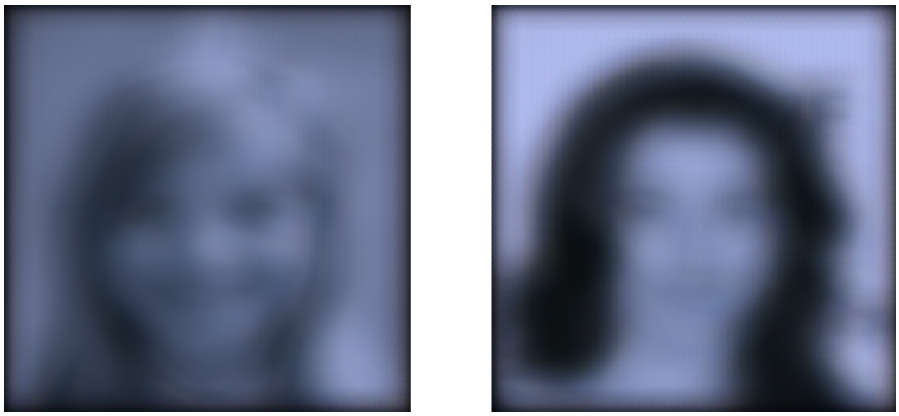
   (a) **Adversarial loss:** The generator tries to produce high-resolution images that the discriminator considers real. BCELoss is used.

   (b) **Content loss (VGG loss):** The generated and real high-resolution images are passed through a pre-trained VGG19 network to compute a content loss based on feature matching.

   (c) **Pixel-to-pixel mean square error:** This loss measures the pixel-wise difference between the generated and real images. These three losses are summed together to form the generator loss.

**Optimization:** The optimization strategy adopted in this implementation involves the use of the Adam optimizer with a learning rate of 0.0001 for training the SRGAN model.

**Our Implementation:** The dataset that we employed is the Celeba dataset. We converted the high-resolution images into low resolution by applying Gaussian blur and resizing into 64*64. We implemented the code as per the architecture defined in the paper and took inspiration when we were stuck. Unfortunately, the results are not satisfying. While training, one epoch showed to be 1-2 hours long. We started the training, and after 29 epochs, the images generated were less desired than expected. Moreover, we used Kaggle and the free resources it provides for our training, which made it difficult to speed up the process and conduct more epochs. The results can be seen in the image below.



Hence, we started to research more, gain awareness of other state-of-the-art models, and read the following papers.

## 3 Next Step

Since we were not able to implement the model using SRGAN due to scarce computer resources we decided to shift to another method that can be both efficient and fast.

So we chose three papers which were about the implementation of the text-based diffusion model and various other techniques.

Below are the literature reviews of the three paper:

### 3.1 Siddhi-"Imagic: Text-Based Semantic Image Editing with Real Images":

This research paper titled presents a novel approach for text-based semantic image editing using real images. The method, known as Imagic, is unique in its ability to perform complex non-rigid edits on a single high-resolution image, all while preserving the original image's structure and composition.

1. **Introduction:** The paper introduces the problem of semantic image editing, which involves making meaningful edits to real images based on natural language text prompts. Existing methods have limitations, such as being restricted to specific types of edits or requiring multiple input images of the same object. Imagic overcomes these limitations and can handle a wide range of complex edits with just one input image and a target text prompt.
2. **Related Work:** The paper discusses related work in the field of image editing, including techniques based on GANs, diffusion models, and text-to-image generation models. It highlights the advancements in using diffusion models for image manipulation and points out the unique features of Imagic.
3. **Imagic: Diffusion-Based Real Image Editing:** This section provides an overview of the Imagic framework, which involves three main stages:

   (a) **Text Embedding Optimization:** The target text is encoded into a text embedding, and an optimization process is used to find an embedding that closely matches the input image.

   (b) **Model Fine-Tuning:** The pre-trained generative diffusion model is fine-tuned to better reconstruct the input image while preserving the optimized text embedding.

   (c) **Text Embedding Interpolation:** A linear interpolation is performed between the optimized embedding and the target text embedding to obtain the final edited image.

The paper showcases the generality, versatility, and quality of Imagic, emphasizing its capability to perform complex non-rigid edits on real images with just a single input image and a text prompt. It also introduces a novel benchmark called TEdBench for evaluating text-based image editing methods.

### 3.2 Rekha-"Prompt-to-Prompt Image Editing with Cross Attention Control":

1. **Introduction:** This paper introduces an intuitive textual editing method called Prompt-to-Prompt, which allows semantically editing images in pre-trained text-conditioned diffusion models. The method uses

internal cross-attention maps, high-dimensional tensors that bind pixels and tokens extracted from the prompt text, to control the generated image. The method allows for editing tasks that are challenging otherwise and does not require model training, fine-tuning, extra data, or optimization. The method also allows for global editing, style changes, and amplifying or attenuating semantic effects. The method can be applied to real images using an existing inversion process.

2. **Related Work:** Image editing is a crucial task in computer graphics, involving modifying input images using auxiliary inputs like labels, scribbles, masks, or reference images. Text-driven image manipulation methods like GANs and CLIP have made progress, but struggle with large and diverse datasets. Crowson et al. use VQ-GAN, while Diffusion models achieve high-quality generation. Bar-Tal et al. propose a text-based localized editing technique without masks, but lacks complex structures.

3. **Method:**

   This paper introduces a method for semantically editing images using pre-trained text-conditioned diffusion models. The approach consists of two key steps.

   (a) **Cross-Attention in Text-Conditioned Diffusion Models**: The paper uses the Imagen text-guided synthesis model to create cross-attention in text-conditioned diffusion models. It uses noise prediction during image generation and uses cross-attention layers to create attention maps for each textual token. Multi-head attention is used to enhance expressiveness, and the model includes two types of attention layers: **cross-attention and hybrid attention.**

   (b) **Controlling the Cross-Attention:** The paper discusses methods to control cross-attention in image generation, which involves injecting attention maps from the original prompt into a new one to preserve its structure. Editing capabilities include limiting injection to specific steps or controlling diffusion steps for different tokens. Alignment functions preserve common details when adding new phrases, and users can adjust the influence of specific tokens by scaling their attention maps.

   In summary, the paper uses cross-attention in text-conditioned diffusion models to semantically edit images, enabling fine-grained control over image generation and enabling various editing tasks without additional model training or data.

This paper discusses the challenge of extending text-driven image synthesis models to text-driven image editing. Text-driven synthesis models are capable of generating diverse images based on textual prompts, making them appealing to humans for describing their intentions. However, editing images using these models is difficult because even small changes to the text prompt can lead to drastically different results. Existing methods often require users to provide a spatial mask to localize edits, which can ignore the original image structure.

### 3.3 Sudeep-"Composing Text and Image for Image Retrieval - An Empirical Odyssey "

1. **Introduction:** This paper focuses on the task of image retrieval, where the input query is an image plus text describing desired modifications to the image. The authors propose a new method called Text Image Residual Gating (TIRG) for combining image and text features. They compare TIRG with several existing

methods and show that it outperforms them on three benchmark datasets. The authors also demonstrate the effectiveness of TIRG for image classification with compositionally novel labels.

2. **Method:**

   As explained in the introduction, our goal is to learn an embedding space for the text+image query and for target images, such that matching (query, image) pairs are close

   It is a three-step process:

   (a) First, we encode the query (or reference) image x using a ResNet-17 CNN to get a 2d spatial feature vector fimg(x) = φx  RW ×H×C , where W is the width, H is the height, and C = 512 is the number of feature channels

   (b) Next we encode the query text t using a standard LSTM. We define ftext(t) = φt  Rd to be the hidden state at the final time step whose size d is 512.

   (c) Finally, we combine the two features to compute φxt = fcombine(φx, φt). This is done using a method called as We propose to combine image and text features using the following approach which we call Text Image Residual Gating (or TIRG for short).

3. **Result:** The results of the study show that the proposed Text Image Residual Gating (TIRG) method outperforms existing approaches on three benchmark datasets: Fashion-200k, MIT-States, and a new synthetic dataset called CSS. TIRG achieves significant improvement in image retrieval performance compared to previous methods, particularly on the Fashion-200k dataset. It also achieves state-of-the-art results on image classification with compositionally novel labels on the MIT-States dataset. The ablation studies conducted on TIRG demonstrate the efficacy of its feature modification mechanism. Overall, the results indicate that TIRG provides improved performance for image retrieval and image classification tasks.

# 4 Future work:

Since we were not able to implement the model using SRGAN due to scarce computer resources we decided to shift to another method which can be both efficient and fast too.

So we are going to implement the paper **Composing Text and Image for Image Retrieval - An Empirical Odyssey "** This paper in which,we study the task of image retrieval, where the input query is specified in the form of an image plus some text that describes desired modifications to the input image.We then in future will try to decode the composite image to generate the image which we require.
These are the steps we are going to take:

1. Do a thorough study of the paper and work on the blueprint.
2. Data Pre-processing : data collection and preporcessing suitable for the implementation of the TIRG.
3. Training with the basic model, validation, and completion of the data pipeline

4. Adding a decoder over the composite image to generate a new image using faster GAN Models like ESRGAN or SwiftGAN.

## 5 Contribution

1. **Sudeep** Literature Review of the "Composing Text and Image for Image Retrieval - An Empirical Odyssey".Starting imolementtion of the codebase of the paper.Worked on evaluation of various other models and helped in finalising the future steps.
2. **Siddhi**: Literature Review of the paper Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network and implementation of the architecture mentioned in paper. Also, helped in deciding future steps.
3. **Rekha** Literature Review of the "Prompt-to-Prompt Image Editing with Cross Attention Control".Extraction of the database.Also, helped in deciding future steps.

## 6 Concluding Remarks

In conclusion, while training deep neural networks like SRGAN, there is a high demand of substantial memory, and running out of GPU memory is a common issue. High-resolution images, large batch sizes, or complex model architectures resulted to memory constraints. Insufficient or low-quality training data hindered the model's ability to learn to perform super-resolution effectively.Training SRGAN is time-consuming due to the high computational load. Lengthy training times can delay model development and experimentation. So, we finally chose to implement the TIRG approach.

Link to the github for the existing codebase.