# Single Image Super-Resolution Using a Generative Adversarial Network

Mitali Meratwal
*Electrical Engineering Department*
*Indian Institute of Technology Bombay*
mitalimeratwal@gmail.com

Sai Saketika Chekuri
*Electrical Engineering Department*
*Indian Institute of Technology Bombay*
saketikachekuri@gmail.com

***Abstract*—For our project, we implemented a Generative Adversial Network (GAN) to realistically super resolve the low resolution images from our dataset. Though CNNs were used for super-resolution before, the SR-GAN framework gave substantially better results which motivated us to implement it ourselves. Also, for our project, we worked only on single image super-resolution (SISR). The train and validation datasets are sampled from the DIV2K dataset; Train dataset has 800 images, validation dataset has 100 images and test data has 100 images.**

## I. Introduction

The specific objective of this project was to take a low resolution image, and generate its respective high resolution estimate from a *single* image. GANs provide a robust framework for generating high resolution realistic-looking images. Also, to be able to distinguish between the generated image and the original image, we need a loss function which consists of an adversarial loss and a content loss. The adversarial loss using a discriminator network is to train to differentiate between the super-resolved images and original images. We used a super-resolution generative adversarial network (SR-GAN) with a VGG network combined with a discriminator with skip-connection and used Mean Squared Error (MSE) of the generated image and the original image, using pixel wise image differences, as the optimization target.

Another advantage of using MSE is that, minimizing MSE also maximizes the peak signal-to-noise ratio (PSNR), which is a common measure used to evaluate and compare SR algorithms. In our project, we used high upscaling factors, along with convolution layers to build our generator model. The drawback is, loss functions such as MSE result in the inaccuracy of details such as texture: minimizing MSE encourages finding pixel-wise averages of plausible solutions which are typically overly-smooth and thus have poor quality.

Instead, the network is designed such that it allows one to train a generative model G with the goal of fooling a discriminator D that is trained to distinguish super-resolved images from real images. With this approach our generator can learn to create solutions that are highly similar to real images and thus difficult to classify by D. This encourages realistic images as solutions which is in contrast to SR solutions obtained by minimizing pixel-wise error measurements, such as the MSE.

## II. Dataset

We have used the Div2K dataset for our project. The DIV2K dataset is divided into:

- Train data: 800 high definition high resolution images
- Validation data: 100 high definition high resolution images
- Test data: 100 diverse images

All the 1000 images are 2K resolution, that is they have 2K pixels on at least one of the axes (vertical or horizontal). All the images were processed using the same tools.
A few sample images from the dataset are added below:

Fig. 1. Example of a figure from the dataset.

Fig. 2. Example of a figure from the dataset.

## III. Analysis

The aim is to estimate a high-resolution, super-resolved image $I^{SR}$ from a low resolution input image $I^{LR}$. The low-resolution images are generated by downsampling the high-resolution images by a factor of 4, which comprise the dataset. We train a generator model that produces a high-resolution image for a given low-resolution input image.

### A. Method

We define a discriminator network $D_{\theta_G}$ where $\theta_G = \{W_{1:L}; b_{1:L}\}$ denoting the weights and biases of a L-layer deep network obtained by minimising the SR-specific loss fucntion $l_{SR}$. This discriminator network is trained alternatingly with the generator network $G_{\theta_G}$ to solve the adversarial min-max problem:

$$\min_{\theta_G} \max_{\theta_D} E_{I^{HR}\sim p_{train}(I^{HR})}[\log D_{\theta_D}(I^{HR})] + E_{I^{LR}\sim p_G(I^{LR})}[\log(1 - D_{\theta_D}(G_{\theta_G}(I^{LR})))] \quad (1)$$

This allows the generator model to fool the discriminator by generating data similar to the training set, while the discriminator tries not to be fooled by distinguishing between fake and real data.

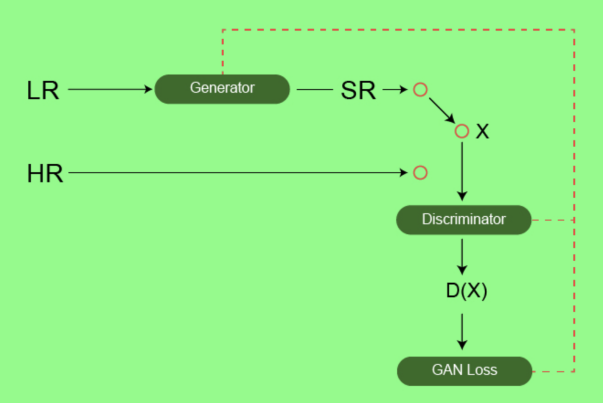


Fig. 3.

### B. Architecture

*1) Generator Architecture:*
The generator architecture contains residual network instead of deep convolutional networks because residual networks are easy to train and allows us to make the models deep along with faster convergence. The networ comprises of 16 residual blocks, originated by ResNet. Each residual block has two convolutional layers, 3X3 kernels, 64 feature maps followed by batch normalisation layers and ParametricReLU as the activation function. The resolution of the input image is increased by 2 PixelShuffler layers (feature map upscaling). During the training, a high-resolution image is downsampled to a low-resolution image. The generator architecture than tries to upsample the image from low resolution to super-resolution. After then the image is passed into the discriminator, the discriminator and tries to distinguish between a super-resolution and high-resolution image and generate the adversarial loss which is then backpropagated into the generator architecture.

*2) Discriminator Architecture:*
The task of the discriminator is to discriminate between real high-resolution images and generated super-resolution images. The discriminator architecture used is similar to DC-GAN architecture with LeakyReLU as activation. The network contains 8 convolutional layers with of 3×3 filter kernels, increasing by a factor of 2 from 64 to 512 kernels. Strided convolutions are used to reduce the image resolution each time the number of features is doubled. The resulting 512 feature maps are followed by two dense layers and a LeakyReLU applied between and a final sigmoid activation function to obtain a probability for sample classification.

### C. Losses

The perceptual loss ($l^{SR}$) is weighted sum of two loss components: content loss( $l_X^{SR}$) and adversarial loss ($l_{Gen}^{SR}$) .

$$l^{SR} = l_X^{SR} + 10^{-3} l_{Gen}^{SR} \quad (2)$$

*1) Content loss:*
The pixel-wise MSE loss for the SR-ResNet architecture.

$$l_{MSE}^{SR} = \frac{1}{r^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} (I_{x,y}^{H} R - G_{\theta_G}(I^{LR})_{x,y})^2 \quad (3)$$

However, MSE loss is not able to deal with high frequency content in the image that resulted in producing overly smooth images. Therefore, VGG based content loss is used. This loss is based on the ReLU activation layers of the pretrained VGG19 by using feature extraction from the $9^{th}$ layer or $3^{rd}$ convolutional layer from the $3^{rd}$ block.

$$l_{VGG/i.j}^{SR} = \frac{1}{W_{i,j} H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2 \quad (4)$$

*2) Adversarial loss:*
The adversarial loss is the loss function that forces the generator to generate images which are more similar to the true high-resolution images by exploiting the discriminator whose job is to differentiate between high-resolution and super-resolution images. If the generator is succesfull in fooling the discriminator, then the adversarial loss is less. Therefore, adversarial loss is defined as:

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -log D_{\theta_D}(G_{\theta_G}(I^{LR})) \quad (5)$$

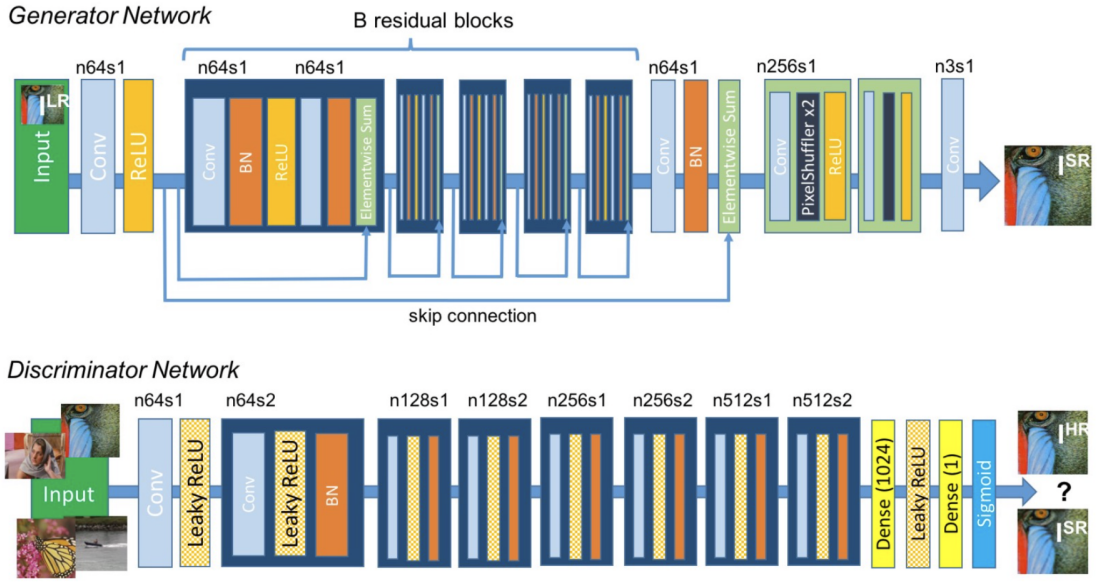


Fig. 4.

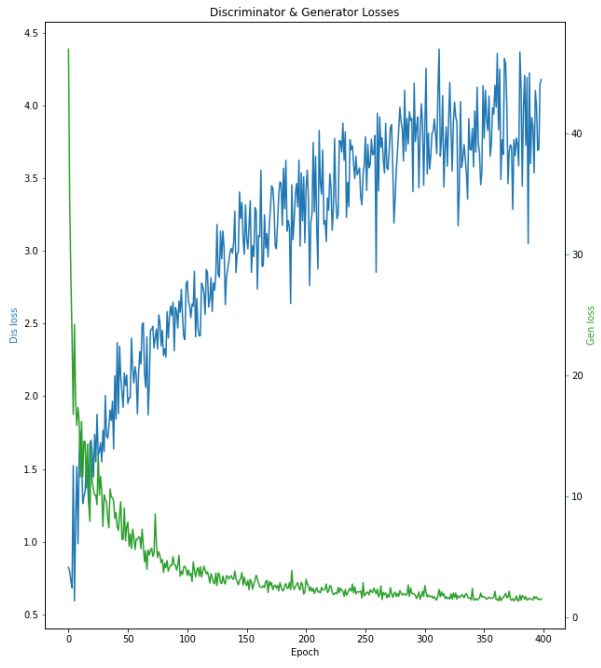


Fig. 5. Discriminator and Generator loss

## IV. Results

Losses after training over 400 epochs are illustrated in Fig5. The decreasing generator loss implies that the generator is improving in its task to fool the discriminator and the discriminator loss increasing suggests its performance to distinguish the between the fake and real images is decreasing.

## V. Discussion

The focus has been on perceptual quality of the super-resolved images rather than computational efficiency. The official paper discusses deeper networks (B¿16) increasing the performance of **SRResNet** at a cost of longer training and testing times. Also SRGAN variants of much deeper network become difficult to train. Image super-resolution has its application in medical imaging, satellite imaging, surveillance among many others. Real time video super-resolution or increasing the resolution of old videos for content recovery is more sought after research area.


## Acknowledgment

We would like to thank our TA, Drumil Trivedi, for greatly helping us out with the doubts we had while working on this project.


## References


[1] C. Ledig et al, *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*, 2016.

[2] AvivSham, SRGAN-Keras-Implementation, GitHub Repository, https://github.com/AvivSham/SRGAN-Keras-Implementation

[3] deepak112, Keras-SRGAN, GitHub Repository, https://github.com/deepak112/Keras-SRGAN

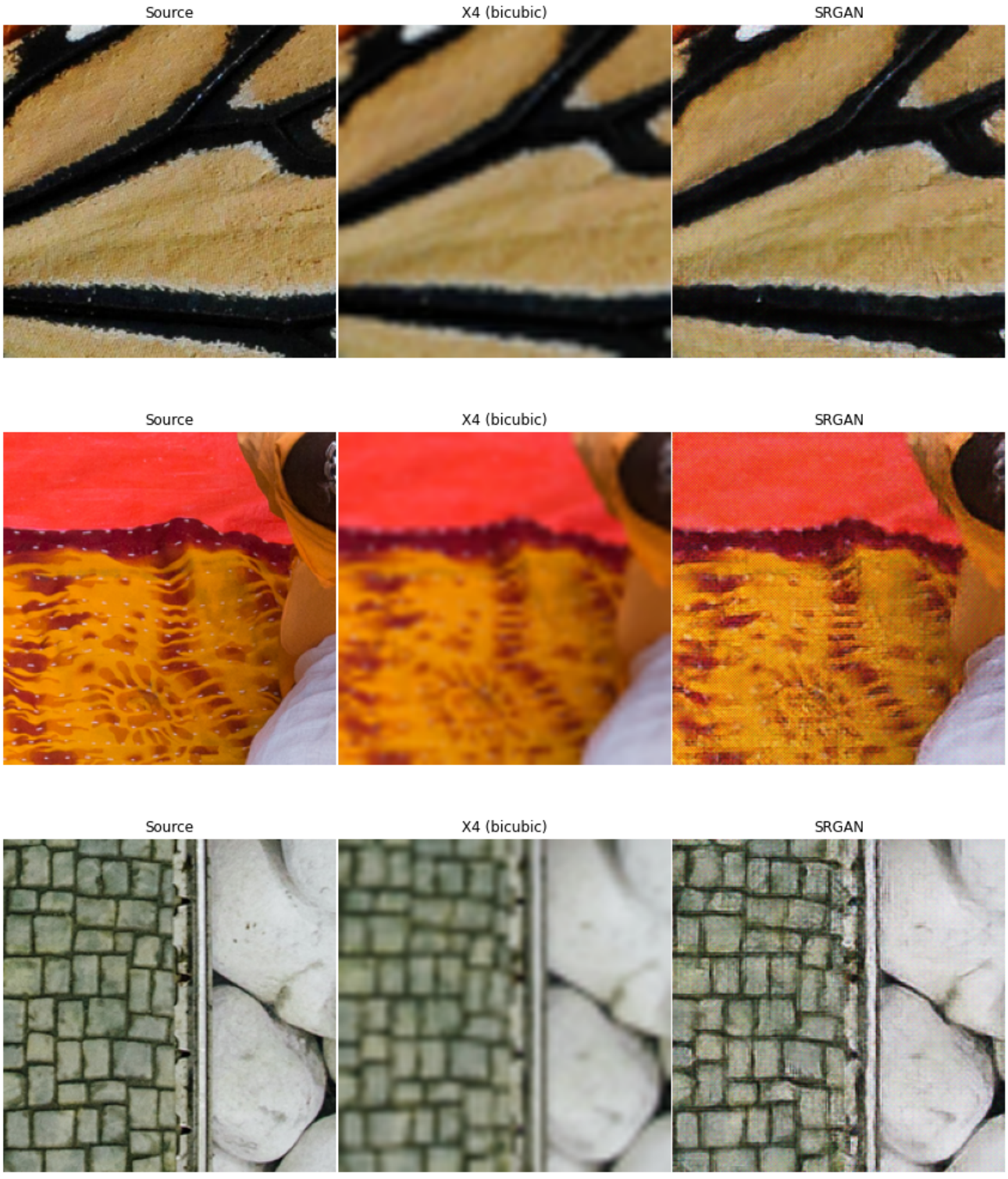
Source
X4 (bicubic)
SRGAN
Source
X4 (bicubic)
SRGAN
Source
X4 (bicubic)
SRGAN

Fig. 6.

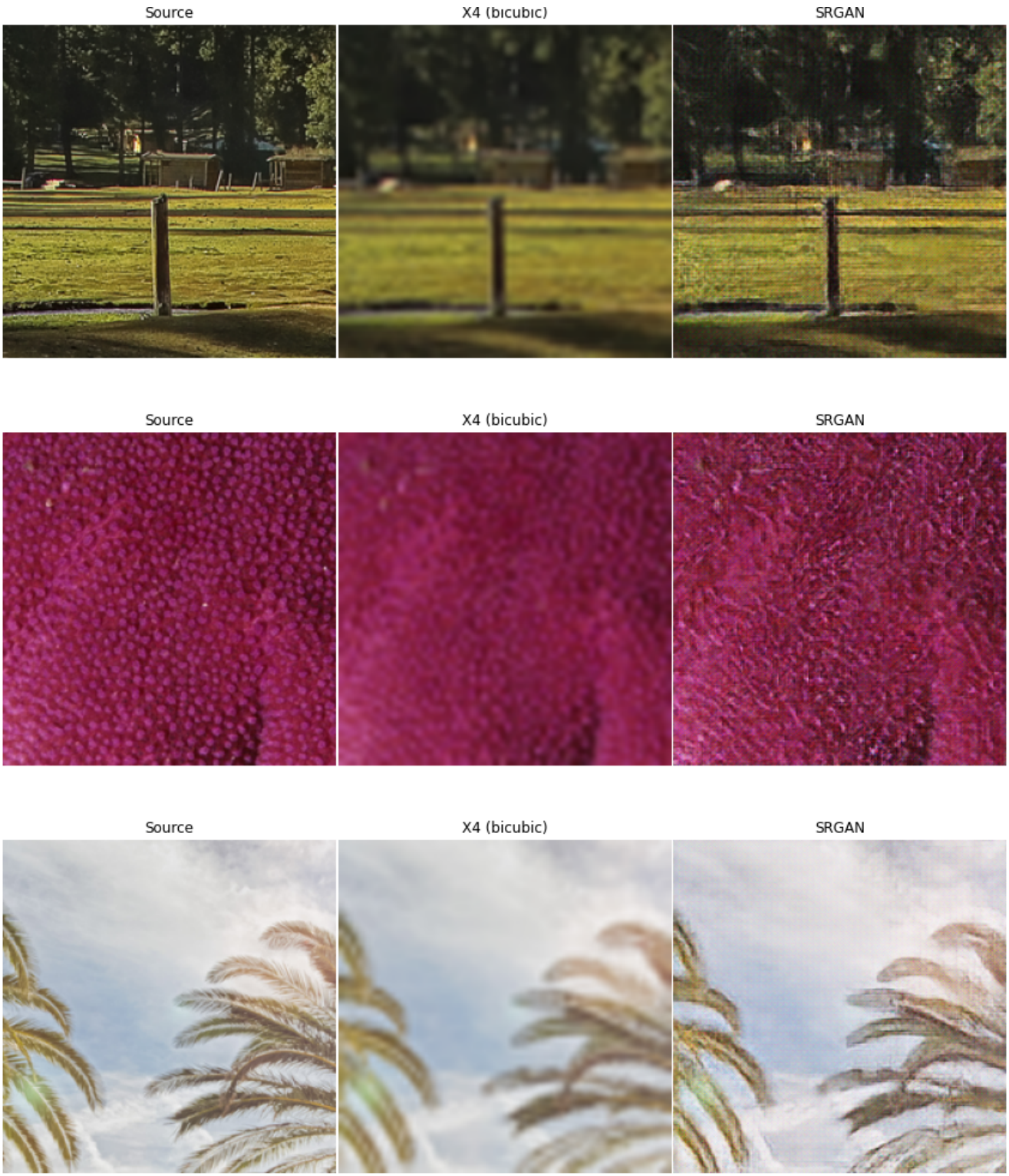
Source
X4 (bicubic)
SRGAN
Source
X4 (bicubic)
SRGAN
Source
X4 (bicubic)
SRGAN

Fig. 7.