An Efficient Preprocessing Module For Incidental Scene Text Recognition

A Thesis submitted by

Anjan Giri

In the partial fulfillment of the requirements for the degree of

MASTER OF TECHNOLOGY

In

COMPUTER SCIENCE

Under the guidance of

Dr. UJJWAL BHATTACHARYA
Associate Professor (Equiv.)
Computer Vision and Pattern Recognition Unit

Indian Statistical Institute, Kolkata

September 5, 2020

Acknowledgement

I sincerely express my gratitude towards my supervisor, Dr.Ujjwal Bhattacharya for providing me with the necessary materials and for all his advice and support. I would also like to sincerely thank all the researchers working in the Handwriting Recognition Lab of the Computer Vision and Pattern Recognition Unit of my institute, Indian Statistical Institute, Kolkata.

Contents

Abstract

State-of-the-art scene text recognition systems perform satisfactorily on samples of benchmark datasets as long as the quality of the text in an image sample is not affected significantly by certain distortions such as blurring etc.
However, their performance may drop sharply whenever the input text appears well outside the focus of the image capturing device or it is suffered by motion blur etc.
In this study, we considered incidental scene texts which usually exhibit much more diversity, variability and complexity together with the common challenges of scene text recognition compared to their counterparts which are captured by properly positioning the camera and making possible adjustments of various image capturing parameters.
In this work, we introduce a trainable deep network that implements a super-resolution technique as the preprocessing module on low quality scene images to boost text recognition accuracy of the existing models.
There are various super resolution techniques for image available in the literature which mainly focus on reconstructing the detailed texture of image but fails to improve the quality of texts appearing in the image and thus the results of their recognition does not get improved.
Here, we propose a novel text-content aware super-resolution network to improve the quality of texts appearing in natural scene image leading to their more accurate recognition by automatic methods.
Simulation results of the proposed model on the ICDAR 2015 Incidental Scene Text dataset demonstrate its effectiveness as an efficient preprocessing model.
Code developed as a part of this dissertation is available at: https://github.com/AnjanGiri/Thesis.

Chapter 1

Introduction

Texts in natural scenes contain high level semantic information that is very useful in many text related vision based applications such as image retrieval, robot navigation, card recognition, industrial automation, intelligent inspection, Natural Language Processing etc. Despite the success of Optical Character Recognition (OCR), the robust scene text recognition and detection has been a research challenge for years. This is largely due to huge amount of variations in text appearance, the complicated image background, imaging artifacts etc. Also scene text images suffers from blurriness, perception distortions, orientation, curved texts, shape and low resolution. In recent years the advancement of the deep learning research and its success in many computer vision and Natural Language Processing tasks have pushed the boundary of scene text recognition.

The performance of many scene text recognizer are promising on focused image dataset, but the performance are not that promising on incidental scene texts. The problems of incidental scene texts are given in the next part. Super-resolution technique may have the answer for two particular problem of incidental scene texts: low-resolution image and blur image. So we have taken a Generating Adversarial Network (1) based super-resolution approach SRGAN (2). This showed a satisfactory result on natural images, but its performance on scene text is not promissing. In this work we tried to increase the performance of SRGAN by introducing a loss function which is very relevant for the texts. Our proposed loss function is similar with the loss function used by SRGAN. We have used the encoder of the pre-trained state of the art scene text recognizer ASTER (3) as the feature extractor similar to the SRGAN's pre-trained VGG-19 network. The main idea behind this loss function is, for an input image ASTER's encoder tries to

capture the information about the text contained in the image, which is later used to produce the output by the ASTER's decoder. The encoder of ASTER only concentrate on the texts in image not on the background of the image. Therefore if we use ASTER's encoder as the feature extractor and try to maximize the the similarity between the extracted features by giving the original high-resolution image and the generated image from the low-resolution image, then our generator will be forced to generate more clear and distinguishable texts from low-resolution image.
We have given a brief descriptions about why we chose scene text recognition and the problems of incidental scene texts.

1.0.1 Why Scene Text Recognition?

Scene text recognition (STR) refers to recognize a sequence of characters that appear in natural images. Optical character recognition (OCR) in scanned documents is well developed (4), but, STR remains challenging because of complex backgrounds, irregular fonts, different sizes, imperfect imaging conditions, diverse colors and multi-orientations.
It has very wide range of vision-based applications in various fields. Therefore, text recognition in natural scenes has drawn the attention of researchers and practitioners. Most of the recently proposed text recognizers have achieved impressive results on many benchmark dataset. However, their performances drop sharply when recognizing blurred text caused by low resolution, motion blur or camera shake. The main difficulty to recognize text in a blurred image is the lack of detailed information about the text contained in the image. This is the reason behind the poor recognition accuracy of many state of the art models on the incidental scene text dataset. In this work we tried to answer the problems of incidental scene text and boost the recognition accuracy of the ASTER model on the ICDAR15 benchmark.

1.0.2 Problems with Incidental Scene Texts

Incidental scene text mean that texts appeared in natural images are captured without user's prior intentions or preferences. Therefore it has much more complexities and difficulties, such as blur, non-uniform illumination, low resolution and cluttered background. In the past few years, scene text detection and recognition have drawn much interest and concern from the computer vision community, and numerous inspiring ideas and effective approaches have been proposed. Though promising progresses have been made on several benchmarks for focused text, state of the art models perform poorly on incidental texts. Also, incidental scene text covers a wide range of applications linked to wearable cameras or identifying texts from moving objects (number plate of a car from surveillance camera) or where the capture is difficult to control.

Figure 1.1: Sample images from ICDAR15 dataset.

ICDAR 2015 data set contains lots of low resolution images causing low performance of each model on it. 1.1 shows some sample containing different kinds of irregularities from the ICDAR15 dataset

Next we have given a brief introduction of the Generative Adversarial Network and Super Resolution Generative Adversarial Network, which are related to our work.

1.1 Generative Adversarial Networks

Generative Adversarial Network (GAN)(1) is a framework for estimating the data distribution, in which we train two models:

- generative model G_{θ_G} , parametrized by θ_G
- discriminative model D_{θ_D} , parametrized by θ_D

via an adversarial process. Two network G_{θ_G} and D_{θ_D} compete with each other to optimize themselves. The generator G_{θ_G} tries to learn the data distribution and the discriminator D_{θ_D} learns to determine whether a sample is from the data distribution or the model distribution. The model D_{θ_D} estimates the probability that a sample came from the original training data or is generated by the generator and is trained to maximize the probability of assigning the correct label to both training examples and generated samples, i.e, the generator G_{θ_G} is trained to maximize the probability of the discriminator D_{θ_D} making a mistake for generated samples.

Let the generator's distribution be p_g. To learn the generator's distribution p_g over data x, a prior is defined on input noise variables $p_z(z)$, then represent a mapping to data space as $G_{\theta_G}(z;\ \theta_G)$. Also a second network $D_{\theta_D}(x;\ \theta_D)$ is defined, that outputs a single scalar, i.e, a classification model. D_{θ_D}(x) represents the probability that x came from the data rather than p_g. The generator G_{θ_G} is simultaneously trained with the discriminator to minimize $log(1D_{\theta_D}(G_{\theta_G}(z)))$. In other words, D_{θ_D} and G_{θ_G} play the following min-max game with the value function $V(G_{\theta_G}, D_{\theta_D})$:

$$\min_{\theta_G}\max_{\theta_D} V(G_{\theta_G}, D_{\theta_D}) = E_{x\sim p_{train}(x)}[logD_{\theta_D}(x)] + E_{z\sim p_z(z)}[log(1-D_{\theta_D}(G_{\theta_G}(z)))] \quad (1.1)$$

In theory it has shown in (1) G_{θ_G} and D_{θ_D} has enough capacity to recover the data generating distribution. The parameters θ_G and θ_D optimized alternately to restrict the overfitting. In practice, equation 1.1 may not provide sufficient gradient for G to learn well because early in learning, when G_{θ_G} has not learned anything, D_{θ_D} can reject samples with high confidence. In this case, $log(1D_{\theta_D}(G_{\theta_G}(z)))$ saturates. Therefore rather than training G_{θ_G} to minimize $log(1D_{\theta_D}(G_{\theta_G}(z)))$, we can train G to maximize $logD_{\theta_D}(G_{\theta_G}(z))$. The solutions are same for this two objective functions, but later one provides much stronger gradients early in learning. And this helps the network for faster convergence.

There exists unique solution for G_{θ_G} and D_{θ_D}, with G_{θ_G} recovering the training data distribution and D_{θ_D} equal to 0.5 everywhere. Since the network is differentiable, the entire network can be trained with the help of back-propagation. The disadvantages of GAN are discriminator D_{θ_D} must be synchronized well with the generator G_{θ_G} during the training.

1.2 Super Resolution Generative Adversarial Network

Super-resolution (SR) is a highly challenging task of estimating a high-resolution (HR) image from a low-resolution (LR) image. There are lots of different HR image estimations are possible from a given LR image, because of lots of missing information about the texture in the LR image. This makes the super-resolution tasks very hard and complex for high upscaling factors and this complexity increases exponentially with the upscaling factor.

Super Resolution Generative Adversarial Network (SRGAN) (2) is a special kind of GAN, is used for high resolution image estimation from a low resolution image. There exist some very simple super resolution techniques (Bi-Cubic etc), which are fast but the results are very smooth. Rather using these fixed methods the information from data is used to guide the learning. The MSE-loss is not good in capturing perceptually relevant differences, such as high texture detail since they are defined based on pixel-wise image differences. So, without using pixel wise MSE loss, the perceptual loss is used.

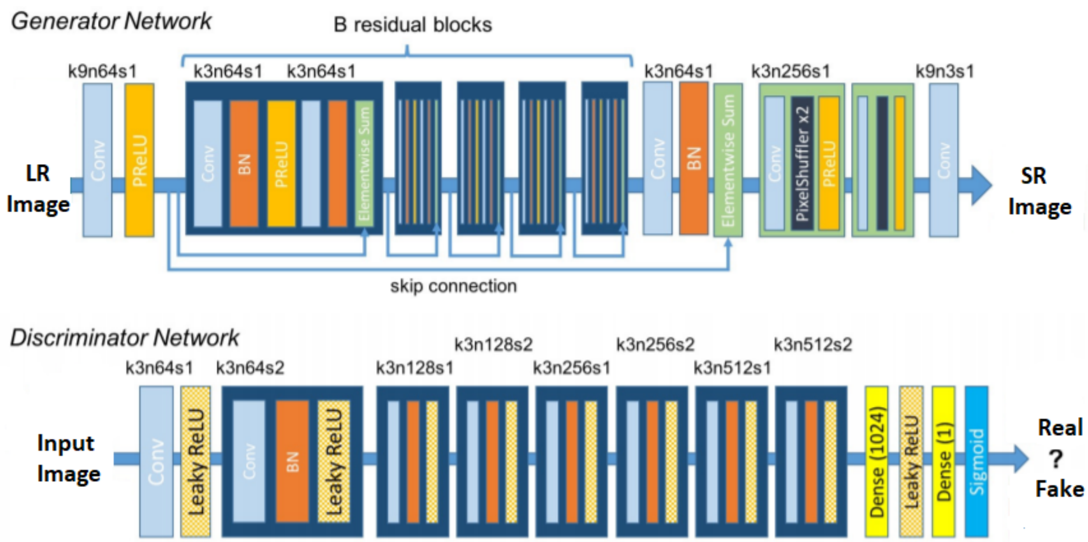

Figure 1.2: Architecture of Generator and Discriminator Network of SRGAN. For each convolutional layer k, n and s are indicating kernel size, number of feature maps and stride respectively. This Figure is taken from SRGAN (2)

Let I^{HR} be the high-resolution image and the corresponding low-resolution image be I^{LR} . The high-resolution images are only available during training. Therefore I^{LR} is obtained from the original image I^{HR} by down-sampling by a factor r , using a Gaussian filter. SRGAN consists of a generator, G_{θ_G} and a discriminator network, D_{θ_D}. The training data set contains low resolution image I_n^{LR} and the corresponding high resolution image I_n^{HR}, where n = 1,2, . . . , N. The parameters θ_G of the generator G_{θ_G} are optimized by the following equation:

$$\hat{\theta}_G = arg \min_n \frac{1}{N} \sum_{n=1}^{N} l^{SR}(G_{\theta_G}(I_n^{LR}), I_n^{HR}) \quad (1.2)$$

, where l^{SR} is the specifically designed perceptual loss that models distinct desirable characteristics of the recovered super-resolved image and is very critical for the good performance of the generator network. Following the Goodfellow et al. (1) a discriminator network D_{θ_D} is optimized in an alternating manner along with G_{θ_G} to solve the adversarial min-max problem:

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

It allows to train the generative network G_{θ_G} with the goal of fooling a discriminator network D_{θ_D}, that is trained to distinguish between SR images and real images. So our generator can learn to create highly similar solutions like real images which are very difficult to classify by D_{θ_D}.
To encourage to reconstruct perceptually superior images it uses a super-resolution specific loss function l^{SR} . l^{SR} is a weighted sum of the the content loss and the adversarial loss.

$$l^{SR} = l^{SR}_{VGG} + 10^{-3} l^{SR}_{GEN} \tag{1.3}$$

, where l^{SR}_{VGG} is the content loss and l^{SR}_{GEN} is the adversarial loss. This loss function helps the network to focus more on the generating content aware, clean images. Instead of using pixel wise mse loss, SRGAN proposed a VGG loss, defined based on the ReLU activation layers of the pre-trained VGG-19 network as described in (5). Here, VGG-19 is a pre-trained classification model on ImageNet dataset (6), which contains 1000 diffrent kinds of objects. The idea behind using the pre-trained VGG-19 as the feature extractor is that using the representation of the input image, that has more idea about the objects contained in the image. Let $\phi(i,j)$ be the feature map obtained by the j-th convolution (after activation) before the i-th maxpooling layer within the pre-trained VGG-19 network. The defined VGG loss is the euclidean distance between the feature representations of a generated image $G_{\theta_G}(I^{LR})$ and the original image I^{HR}, i.e, the VGG loss function is given by

$$l^{SR}_{VGG/i,j} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} [\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y}]^2 \tag{1.4}$$

Here, $W_{i,j}$ and $H_{i,j}$ are the dimensions of the feature maps within the VGG network. And the adversarial loss is defined by

$$l^{SR}_{GEN} = \sum_{x=1}^{N} -log(D_{\theta_D}(G_{\theta_G}(I^{LR}))) \tag{1.5}$$

1.3 Scene Text Recognition

Scene text recognition means recognizing the character sequence from a natural image. We have used a state of the art scene text recognizer ASTER (3) in our model as the feature extractor. This helps in calculation of the perceptual loss of our generator. We have have given a brief introduction of the ASTER model below.

1.3.1 ASTER

Attentional Scene Text Recognizer with Flexible Rectification (ASTER) (3) is a very popular state of the art model for scene text recognition and text detection work. This model comprises of two parts:

- Text rectification network
- Text recognition network

ASTER model uses a text rectification network to tackle irregular text problems of text appearing in scene texts. The rectification network takes an image I as input and outputs a rectified image, I^R by rectifying the text in it. Then this rectified image I^R is given as input to the recognition network to get the output text. We have given brief introduction of these two networks.

Text Rectification Network

The text rectification network transforms the input image into a new image by rectifying the texts in it. There exist various 2D-transformations like affine and projection but they are not very good in rectifying text in natural images. So, ASTER uses a well known and very flexible image transformation Thin Plate Spline (7) (TPS) for text rectification. TPS can handle a variety of text irregularities and has a broad application in image transformation and image matching. Two common irregularities in natural image containing texts are perspective texts and curved texts. TPS is very good at tackle these problems, since it performs non-rigid deformation on images.
The rectification network of ASTER rectifies an input image with the TPS image transformation, T is based on the learnable Spatial Transformer Network (8) (STN). The text rectification network is combined of a text localization network, a grid generator and a sampler. All these are made differentiable so that the rectification network can be trained by the gradients back propagated through the recognition network. A TPS transformation can be determined by the two sets of control points, first on the input image and second on the output image, i.e, the rectified image. The size of this two sets of control points must be equal. The first set of control points are predicted by the localization network denoting the top and bottom boundary of the text in the input

image and the second set of control points are placed on the output image at fixed locations along the top and bottom image borders with equal spacings. So, when the control points on the input image are predicted along the upper and lower text edges, the resulting T outputs a rectified image with regular text. Then, T is calculated using the control points C and $C^{'}$where C has the same number of fixed equi-spaced points on the rectified image. The computation of the TPS transformation is given in details in the section-3.1.2 of (2). Using this TPS transformation, T the sample grid is generated, i.e., for the given pixel co-ordinates in $I^{'}_{R}$ it computes the corresponding pixel co-ordinates in the I. At last, a sampler generates the rectified image by interpolating the neighbour pixels of the image $I^{'}_{R}$. The sampler made differential so that the Back propagation algorithm can be used for the learning of the rectification network.

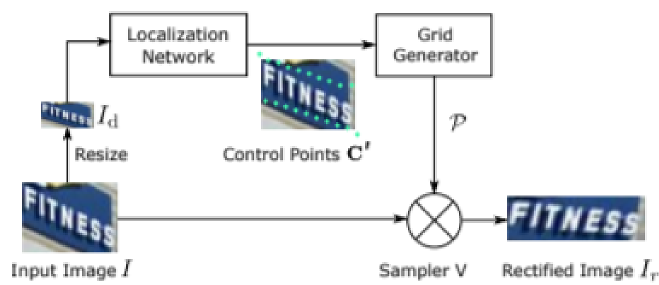

Figure 1.3: Rectification Network structure of ASTER. This Figure is taken from ASTER (3)

Text Recognition Network

The recognition network predicts a sequence of characters from the rectified image, I^R in an attentional sequence-to-sequence manner by taking the rectified image as input. To achieve more accurate recognition, the recognition network effectively encloses the language modeling, character detection and character recognition into a single model. This network comprises of two parts: encoder and a decoder. The whole recognition network is trained on the image using the ground truth annotations.

Encoder : The encoder is trained to extract the rich and important discriminative features from the image. Generally, the characters of a text are arranged in a line and are therefore can be represented by a feature sequence that describing local image regions arranged from left to right (or likewise, right to left).

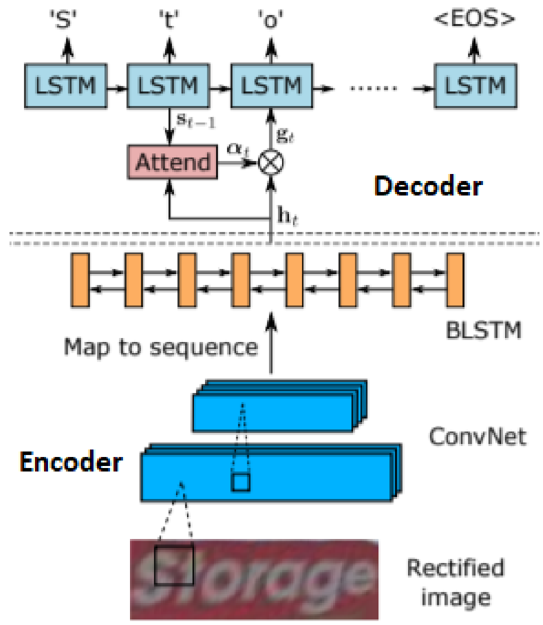

Figure 1.4: Structure of ASTER's text recognition network. This Figure is taken from ASTER (3)

As shown in Figure 1.4 the encoder first extracts a feature map from the input image. The convolutional layers with residual connections are designed so that it can extract strong image features. But this features are strongly effected by their receptive fields, i.e., the image region they capture. To enrich the context of the feature and enable to capture long range of character dependencies, a multi-layer bi-directional LSTM (9) (BLSTM) network is used on the top of the convolutional network in the encoder.

Decoder : Here decoder is a attention based sequence-to-sequence model that translates the feature sequence from encoder into a character sequence of arbitrary lengths. Due to the simplicity and its ability to capture output dependencies this type of model are very popular. Decoder generates the characters of the output text one by one and the accuracy is used as loss function.

Chapter 2

Related Work

2.1 Super Resolution

Image super-resolution (SR) is an important class of image processing techniques in computer vision and image processing. This problem is inherently ill-posed since there are always multiple HR images corresponding to a single LR image. There are various approaches to recover HR images from LR images. The most simple SR techniques interpolations such as the nearest neighbour, Bi-Linear, Bi-Cubic, etc. Where nearest neighbour interpolation selects a value from nearest neighbour pixel value for each position. Bi-Linear and Bi-Cubic interpolations perform linear and cubic interpolations on both the axis respectively. There is another single image super-resolution (SISR) approach Lanczos (10), which is a Fourier method to filter digital data. These methods are simple and fast but oversimplify the SISR problem by smoothing the textures in images. More powerful approaches aim to establish a complex mapping between low- and high-resolution image information and usually rely on training data. Also, a sequence of undersampled and degraded image sequence can be used to obtain a super-resolved image to take advantage of the additional Spatio-temporal data available in the image sequence (11) and (12). An edge-directed interpolation method have been presented in (13) for digitally interpolating images to higher resolution.

With the rapid development of deep learning techniques in recent years, deep learning-based SR models often achieve very good performance. Various Deep Learning based SR method was proposed in recent years. In general, the family of SR algorithms using deep learning techniques differ from each other in the following major aspects: different types of network architectures, different types of loss functions, different types of learning principles and strategies, etc.

Although Convolutional Neural Network (CNN) based SR algorithms (SRCNN) (14), (15) have shown excellent performance, Generative Adversarial Networks (GAN) based SR method (2) produce perceptually rich super-resolution images, SRGAN (2) shown superior results than the SRCNN capturing perceptually rich informations in the super-resolved images.

2.2 Text Recognition

Existing works on Scene Text Recognition (STR) can be roughly divided into traditional and deep learning based methods. Most traditional STR work follow a bottom-up pipeline that first detects and recognizes individual character and then links up the recognized characters into words or text lines by language models. For example, (16) uses a fully connected network for character recognition and (17) uses CNNs to recognize unconstrained character. These bottom-up methods need to localize each individual characters, which is costly both for location labeling and training. Besides, these methods also prone to errors such as overlaps between adjacent characters.

Deep learning methods have dominated STR in recent years. (18) start to take STR as a word classification problem by CNN model which is constrained to the pre-defined vocabulary. Later, various sequence-to-sequence models (3), (19), which is believed to embed the language model in the decoding layer, are applied for STR. Instead of using VGG (5) in the feature extraction layer, ResNet (20) became the tendency to use it as feature extractor. (21) developed a focusing attention mechanism to improve the performance of general attention-based encoder decoder framework. To improve the recognition accuracy further the rectification was used in STR models based on the spatial transform network (STN). ASTER (3), ESIR (19) adopts the thin-plate spline (TPS) transformation based on STN for scene text distortion correction. ASTER uses TPS based on STN to rectify the warped image, the points on two sides are predicted without any constraint, two independent decoders was exploited. ESIR extends the rectification of ASTER to an iterative way. It takes the rectified image iteratively multiple times and uses the final rectified image as the input to the text recognition network. The author of (22) used a finer grid rectification method to rectify the distorted images. It uses TPS transformation but with different control points. Instead of predicting a set of boundary control points, it predicts a set of control points of the rectangular grid and then uses it to compute the TPS transformation parameters.

Chapter 3

Our Method

In this section, we present our proposed method in detail. Super-resolution can be a very good option to tackle the problems in incidental scene texts like low-resolution, blurred texts. Although SRGAN is a popular and promising generating adversarial networks used for the estimation of the super-resolution images from the low-resolution natural images but isn't good for the texts. The high-resolution images generated by the SRGAN lack fine details of text desired for text recognition and preserve the detailed texture of the natural image and enhance the text area as well as the background. Also, super-resolution has some other drawbacks. Super resolving a scene text image is much more challenging compared to a natural image, because of arbitrary poses and illumination of the texts in images. Therefore, to generate clear, sharp and identifiable text images for the recognition we need a text-specific content-aware super-resolution network.

We have used a SRGAN with different loss functions and want our generator G_{θ_G} to be content-aware, i.e., generates a high-resolution image from low-resolution image focusing on the text region, not the back-ground. In this way, our generator can generate clear, sharp and identifiable text images. A traditional approach of training SRGAN on scene text is, training the generator separately with the guidance of an perceptual loss. But this approach fails for scene texts, since it super-resolve everything (the detailed texture of natural image) and doesn't focus more on the text part, because it is lacking a good perceptual loss function specially for scene texts. So we have introduced a special perceptual loss for scene texts to capture more text information in a image. Our perceptual loss contains a content loss similar to the VGG-19 content loss in the SRGAN (2), which computes the similarity of two feature maps of super resolved image and the original image. We have taken the encoder of the pre-trained ASTER as the feature extractor. ASTER is a state of the art recognition model consisting encoder and decoder. The encoder encodes the input image and then the decoder takes this encoding as its input and generates the sequence of characters in text. The general idea for this is, for an input image ASTER is learning the information about the texts contained in the image and its encoder capaturing the features that are related to texts in the image. If we try to maximize the similarity of the features extracted by the encoder of ASTER for the super-resolved image and original image, then the generator will be forced to generate images with more clean text. Let the encoder of ASTER be ϕ_{enc} . Then our text specific content loss for the generator is:

$$l^{SR}_{ENC} = \frac{1}{WH} \sum_{x=1}^{W} \sum_{y=1}^{H} (\phi_{enc}(I^{HR})_{x,y} - \phi_{enc}(G_{\theta_G}(I^{LR}))_{x,y})^2, \tag{3.1}$$

where W and H describe the dimensions of the of the feature maps within the Encoder network of ASTER.

Our loss function is the weighted sum of the content loss and the adversarial loss. It is enough powerful and capable to force the to generate more natural and clear text images. The content loss focus on the contents, i.e., the texts in the image and the adversarial loss force the generator to generate images similar to the natural images.

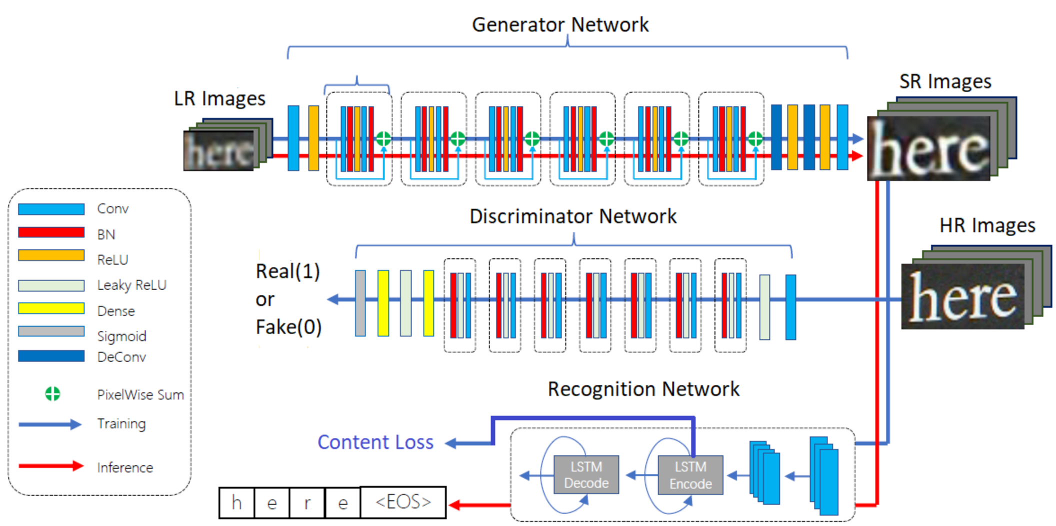

Figure 3.1: The network architecture of our model. At the inference time we give the low resolution image to the generator and then the super-resolved image is given to the pre-trained ASTER to get the output.

Chapter 4

Data Set

The proposed model is trained on selected images from SynthText dataset and tested on ICDAR 2015 Incidental Text without any finetuning. The details of these datasets are given below.

4.1 SynthText

SynthText is the synthetic text dataset proposed in (23). The dataset contains 9 million images generated from a set of 90k common English words. Words are rendered onto natural images with random transformations and effects. Every image in SynthText is annotated with a ground truth word and targeted for text detection. We used the cropped images as training data, using the ground truth word bounding boxes.

4.2 ICDAR 2015 Incidental Text (ICDAR15)

ICDAR 2015 Incidental Scene Text is the Challenge 4 of the ICDAR 2015 Robust Reading Competition (24). This challenge features incidental text images, which are taken by a pair of Google Glasses without careful positioning and focusing. The dataset includes 1500 natural images in total. This is different from the previous ICDAR competitions, in which the texts are well-positioned and focused. The images from ICDAR 2015 are taken in motion and without focusing, so the texts are usually skewed or blurred. Therefore the dataset contains a lot of irregular text. Training and testing images are obtained by cropping the words using the ground truth word bounding boxes.

Chapter 5

Training Details

We have trained our generator and discriminator model from scratch on the one NVIDIA Tesla P6 GPU using a random sample of 100K images from a subset of SynthText data set. We filtered training images by image size greater than (32, 128) from SynthText dataset. And tested our results on the ICDAR15 test images without any fine-tuning of the parameters on training dataset of ICDAR15, i.e., the training and testing data comes from two different datasets.
All the experiments are performed with scale factor, r=2 and r=4. This results in 4x times and 16 times reduction in the pixels reducing the image quality and information. The low-resolution images in the training data set are obtained by downsampling the original HR images (BRG, C=3), using Bi-Cubic kernel. Although we can apply our generator to an arbitrary size image, we resize all the LR image to 16 x 50 and all HR image to 32 x 100 for each mini-batch with batch size 256. We initialized all the weights of the generator parameters by the normal distribution with mean 0 and standard deviation 0.2. And all weights of the discriminator and all the biases are initialized by zeros. We adopt the Adam](25) optimizer for the optimization of the trainable parameters of the network with betas as 0.01 and 0.9. We fixed the parameters of ASTER and only updating the parameters of the generator and discriminator throughout the training. We trained our integrated model for 1000 epochs with learning rate 10^{-4} for the first 500 epochs and 10^{-5} for the next 500 epochs.
Throughout the entire training, all the weights of the ASTER remain fixed and our network only optimizes the parameters of the generator and discriminator.
We were unable to train our model on large dataset due to the system constraint. Also, this restricts us to train the model for the big number of iterations, since the training

was slow. Therefore we trained our model on a small collection of data and we think that reflects on the performance of our model.

Chapter 6

Experiments and Results

We evaluate the accuracy of ASTER using our proposed generator as a preprocessor on ICDAR 2015 Incidental Text (ICDAR15) test dataset and compare its performance with the original ASTER model, which is a state-of-the-art scene text recognition model. We also have shown a comparison of the quality of the super-resolved image of the incidental scene texts and the original images in Figure 6.1. The results are given in the following Table 6.1. The accuracy of the pre-trained ASTER model (available at https://github.com/ayumiymk/aster.pytorch), by the author (3), is little less than the claimed result on the ICDAR15 dataset. We couldn't train our model for a sufficient number of epochs due to system unavailability. So we have given our model's accuracy with the increasing epoch numbers along with the claimed accuracy of ASTER in the following Table 6.1.

Table 6.1: Results of our simulation

Models	Iteration	Downsample Factor (r)	Accuracy on ICDAR15
ASTER	50K	-	76.1
Our proposed model	500	2	68.14
Our proposed model	600	2	69.37
Our proposed model	700	2	69.21
Our proposed model	1200	2	71.41

From the above Table 6.1 it is clear that the accuracy of our model integrated with the ASTER is increasing and may have improved with more training.
Also accuracy is not a very good metric to measure the performance of the scene text

recognition models, because the penalty is same for both failing to recognize only one character of the ground truth text and failing to recognize the entire text. The metric should be able to capture the character recognition accuracy with the same order as in ground truth text.

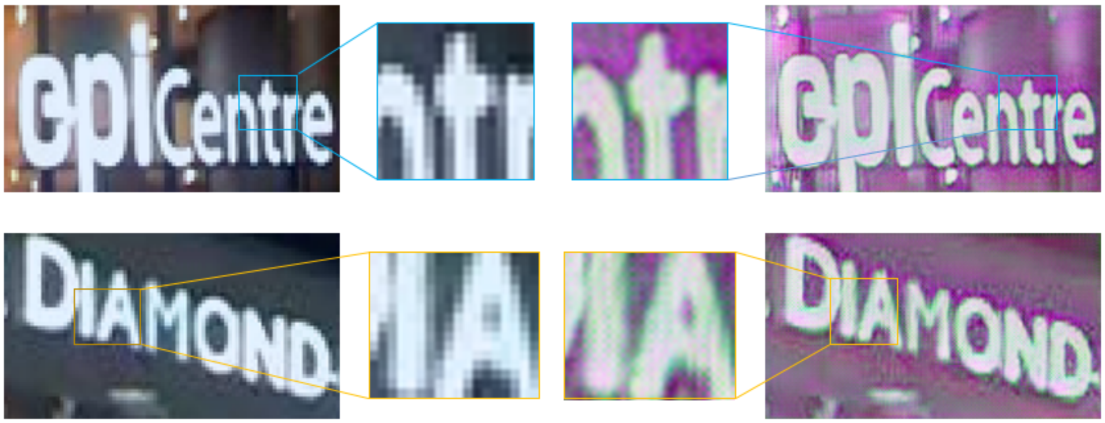

Figure 6.1: Result of our model on two randomly chosen images from ICDAR15 dataset. From two images it is clear that our model can improve the texts in the image by removing blurriness from it.

To show the effectiveness of our generator model for text detection as a preprocessor we tested our model on randomly chosen images from IIIT5k-Words (IIIT5k) (26) dataset. This is more challenging, since the image data may contain more complex and irrelevant background with small portion of text. We have given a result of the original image and the super-resolution image in Figure 6.2. From Figure 6.2 it is clear that our model has capability to enhance the texts in a complex image.

Figure 6.2: Original image (left) vs generated image (right) by our model. It is clear from the image, that the quality of texts in the image has improved. The texts in the generated image are more identifiable than the original image (the texts 'ONE WAY' and 'CHINESE' are easily recognizable). This shows that our model can detect the text and enhance it from an image with a complex background.

To show the robustness of our model we have randomly chosen an image from Google search containing the texts of Hindi, English and Tamil scripts and tested our model on that image. The result is shown in Figure 6.3. The super resolved image shows that our model don't memorize the data and captures more information of texts. Although our model has been trained on English script, the results on the other scripts are also very good and can be easily generalized on different type of scripts.

Figure 6.3: Original image (left) vs generated image (right) for an input image containing different scripts than English (Hindi, English and Tamil). Although, the model isn't trained on scene texts containing Hindi and Tamil scripts still it can enhance texts from these scripts and remove the blurriness from an input image.

Chapter 7

Future work

We want to explore a few more ideas in the near future and extend our work. The ideas we want to explore are:

- We can use the generator as a preprocessor for scene text detection problem.
- We have used pre-trained ASTER as the feature extractor which is a state of the art recognition model. But there are many other state-of-the-art models that can be used as a feature extractor. Also, this may lead us towards a more efficient solution.
- We want to use the recognition accuracy as the adversarial loss for our generating model. This could lead us to a more efficient solution for scene text preprocessor.
- Without using accuracy as a metric to measure the performance of any text recognition model we can search for metrics that measure the correctly identifying characters with the same order as in ground truth text and use that as a loss function of scene text recognition.
- We have trained our generator by taking 2x downsampled images. If we take this downsample factor, $r=4$ then this may have boosted the performance of our model, since our model would learn to restore images from images with fewer contents.
- Without using a pre-trained model as the recognizer we also can end to end learn the recognizer model.

Chapter 8

Conclusions

This work addresses the problem of small images with a content-aware super-resolution technique. Despite the limited access to the computational systems and small number of a training epochs, the results of our model are quite satisfying. From Figure 6.2 it is clear that the generated text image preserving more text information than the background. Also, training for a large number of iterations may increase the results.

Chapter 9

Bibliography

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[2] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[3] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "ASTER: An attentional scene text recognizer with flexible rectification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2035–2048, 2019.

[4] G. Nagy, "Twenty years of document image analysis in pami," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 38–62, 2000.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ArXiv Preprint arXiv:1409.1556*, 2014.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of the IEEE conference on computer vision and pattern recognition.* IEEE, 2009, pp. 248–255.

[7] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 567–585, 1989.

[8] M. Jaderberg, K. Simonyan, and A. Zisserman, "k. kavukcuoglu,"," *Spatial transformer networks," in Advances in Neural Information Processing Systems*, vol. 28, pp. 2017–2025, 2015.

[9] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2008.

[10] C. E. Duchon, "Lanczos filtering in one and two dimensions," *Journal of applied meteorology*, vol. 18, no. 8, pp. 1016–1022, 1979.

[11] S. Borman and R. L. Stevenson, "Super-resolution from image sequences-a review," in *1998 Midwest symposium on circuits and systems (Cat. No. 98CB36268)*. IEEE, 1998, pp. 374–378.

[12] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE transactions on image processing*, vol. 13, no. 10, pp. 1327–1344, 2004.

[13] J. Allebach and P. Wong, "Edge-directed interpolation," *Proceedings of 3rd IEEE International Conference on Image Processing*, vol. 3, pp. 707–710 vol.3, 1996.

[14] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision*. Springer, 2014, pp. 184–199.

[15] ——, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[16] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "Photoocr: Reading text in uncontrolled conditions," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 785–792.

[17] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep structured output learning for unconstrained text recognition," *ArXiv Preprint arXiv:1412.5903*, 2014.

[18] ——, "Reading text in the wild with convolutional neural networks," *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1–20, 2016.

[19] F. Zhan and S. Lu, "Esir: End-to-end scene text recognition via iterative image rectification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2059–2068.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[21] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou, "Aon: Towards arbitrarily-oriented text recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5571–5579.

[22] G. Wang, "Scene text recognition with finer grid rectification," *ArXiv Preprint arXiv:2001.09389*, 2020.

[23] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2315–2324.

[24] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu *et al.*, "ICDAR 2015 competition on robust reading," in *Proceedings of 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 1156–1160.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ArXiv Preprint arXiv:1412.6980*, 2014.

[26] S. Lu, B. M. Chen, and C. C. Ko, "Perspective rectification of document images using fuzzy set and morphological operations," *Image and Vision Computing*, vol. 23, no. 5, pp. 541–553, 2005.