Generative Adversarial Network based Image Super-Resolution to Enhance the shadow-less Image

19AD798 - DISSERTATION

Submitted by

CB.EN.P2AID19017 JISHNU P

in partial fulfilment of the requirements for the degree of

MASTER OF TECHNOLOGY

IN

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE – 641 112

JUNE 2021

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, COIMBATORE – 641 112

BONAFIDE CERTIFICATE

This is to certify that the thesis "Generative Adversarial Network based Image Super-Resolution to Enhance the shadow-less Image " submitted by Jishnu P **(CB.EN.P2AID19017)** in partial fulfilment of the requirements for the award of Degree of **Master of Technology in Artificial Intelligence and Data Science** is a bonafide record of the work carried out under my guidance and supervision at Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore.

Dr. Rajathilagam B.
Thesis Advisor
Associate Professor
Department of CSE

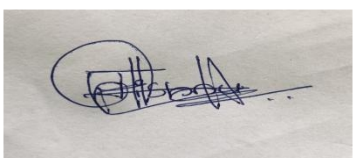
Dr. (Col) P. N. Kumar
Chairperson
Department of CSE

Evaluated on:

INTERNAL EXAMINER EXTERNAL EXAMINER

DECLARATION

I the undersigned solemnly declare that the thesis "Generative Adversarial Network based Image Super-Resolution to Enhance the shadow-less Image" is based on my own work carried out during the course of our study under the supervision of Dr. Rajathilagam B., Associate Professor, Computer Science & Engineering, and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgment has been made wherever the findings of others have been cited.



Signature of the Student

ABSTRACT

The presence of shadow is unavoidable while dealing with the outdoor images in a variety of computer vision applications. Shadows always trouble computer vision tasks like visual navigation, object detection and tracking. In order to unveil the information occluded by shadow, it is essential to remove the shadow. This is a two-step process which involves shadow detection and shadow removal. However, the field of shadow detection greatly improved compared to the shadow removal task, but the field of shadow removal has been difficult because it needs to be restored after removing the shadow.

While reconstructing region occluded by the shadow, the lack of quality in that area seems to be a challenging task. To address this issue, Image super-resolution is introduced. Single image super-resolution is a domain in image processing that aims at recovering a high-resolution image from a single low-resolution image. The need of high resolution image is essential for the computer vision applications for better performance in pattern recognition and analysis of images. In this paper, the low quality of the shadow-less image generated using the shadow removal GAN is improved using the Generative Adversarial network based Image Super-resolution model. By applying the image Super-resolution to the shadow-less image, we can improve the quality of the image as well as the overall performance of the shadow removal process. SRD (shadow removal dataset) , BSD 100(Berkeley Segmentation Dataset), DIV2K dataset are used as the datasets and the performance of the model evaluated using the metrics PSNR(Peak Signal to Noise ratio) and SSIM (Structural Similarity Index).

Keywords: Shadow Removal, Generative Adversarial Networks(GAN),Image Super-resolution, SSIM, PSNR

ACKNOWLEDGEMENT

I wish to record my deep sense of gratitude and profound thanks to my research supervisor Dr. Rajathilagam B., Associate Professor, Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, Coimbatore, for her keen interest, guidance, constant encouragement during all stages of my work, to bring this thesis into fruition.

I am extremely indebted to **Dr. (Col) P. N. Kumar**, Chairperson, Department of CSE, Amrita Vishwa Vidyapeetham, Coimbatore, and panel members for their valuable suggestions and support during the course of my research work.

I also thank the faculty members of the Department of CSE, Amrita Vishwa Vidyapeetham, Coimbatore, for their valuable support throughout the course of my research work.

JISHNU P

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Acronym	**Full Name**
GAN	Generative Adversarial Network
PSNR	Peak Signal to noise Ratio
SSIM	Structural Similarity Index
LR	Low Resolution
HR	High Resolution
SR	Super Resolution
SF	Shadow Free
ESF	Enhanced Shadow Free
MSE	Mean Square Error
BSD	Berkeley Segmentation Dataset)
SRD	Shadow Removal Dataset
MAE	Mean Absolute Error

CHAPTER 1

INTRODUCTION

In this chapter, the chosen problem is described in depth. In section 1.1 the problem definition of this work is explained, which contains a detailed introduction about shadow and the necessity for shadow removal, and an overview of the strategy which is followed to enhance the low resolution(LR) image and apply that module to the shadow-removal module. In section 1.2 the problem definition of this work is mentioned.

1.1 INTRODUCTION

Removing shadows from the images has been considered as a challenging task in the field of computer vision. The presence of opaque objects in the path of sunlight leads to the formation of shadows and depend on different factors such as the altitude of the sun and location of the object. For example, consider a bike and bus in traffic such that the bike is standing left side of the bus. There are chances for the shadow of the bus to cover the bike if the sunlight is from the right side of the bus. Shadow of different shapes distorts two different objects into a single object. This is called as the occlusion. This is a difficult situation in which we can't efficiently detect different objects. In this example, it will be difficult for us to distinguish between bike and bus. Probably the Bus and its shadow will merge together and form another shape which will be far different from the shape of a bus.

In traditional approaches, a common method to remove shadow consist of detecting shadows and using detected shadow masks as a clue for removing the shadows.

The field of shadow detection predicts the location of the shadowed region in the image and separates the shadowed and non-shadowed region of the original image in pixels. This has been considered a challenging task to classify shadows in an image because shadows have various properties. Depending on the degrees of occlusion by the object, the brightness of shadow varies such as umbra and penumbra. The dark part of the shadow is called the umbra, and the part of a shadow that's a little lighter is called the penumbra.

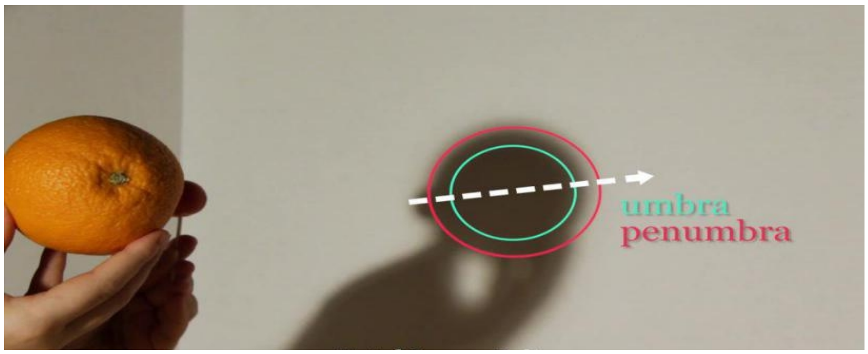


Fig 1 : Umbra and Penumbra regions
source : [https://www.youtube.com/watch?v=huVCQXYy4jc]

If there is an object with a black texture in the image, it can be misclassified as a shadow region.

Shadow removal is also a difficult task because we have to remove shadows and restore the information in that region according to the degree of occlusion.

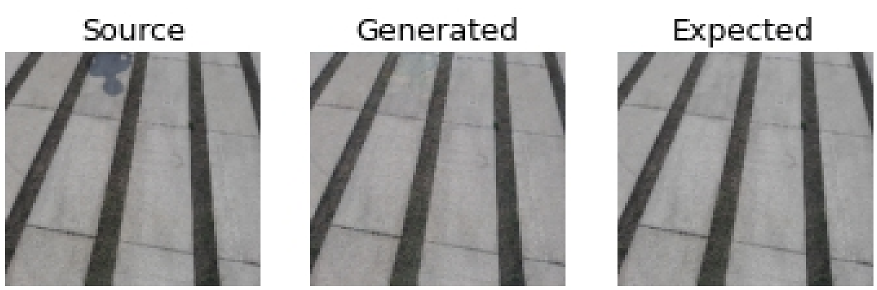


Fig 2: Example : Shadow removal (our model)

In Phase 1, the shadow removal module successfully developed using the Generated Adversarial Network. The model successfully removes shadow for the given image as illustrated in the figure 2. When the model applied on different dataset, we found that the information restored in the shadowed region need to be improved further and the scenario is depicted in the figure 3.

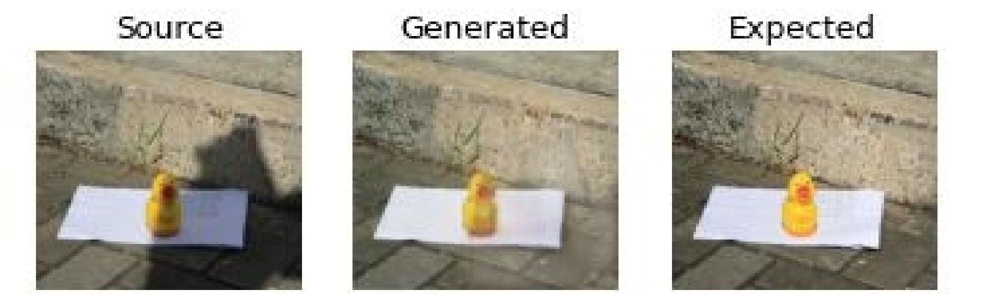


Fig 3: Sample output of GAN based shadow removal model on SRD dataset

In recent years, with the impact of the Internet boom and the rapid development of information technology, people's requirements for signal and information processing have gradually increased, and image processing is an important part of information processing. Additionally, image super-resolution (SR) technology is particularly important in image processing. The principle is to pass one or more low-resolution (LR) images to the final high-resolution (HR) image through information processing technology.

In this paper, we Introduce image Super-Resolution module in the shadow removal pipeline to improve the quality of shadow-less image before applying higher level computer vision tasks like Wild Animal detection, Road Accident detection.

1.2 PROBLEM DEFINITION

To develop an Image Super-Resolution model with Generative Adversarial Network(GAN) which can improve the quality of the shadow-less image and improve the performance of higher level computer vision tasks in surveillance systems.

CHAPTER 2

LITERATURE SURVEY

In this chapter, some existing shadow removal related works, and their limitations are described. Since our approach contains image-super resolution, some existing image-super resolution related works also described. In section 2.1 some of the literature work related to shadow removal, image super-resolution are discussed. In section 2.2 key findings from the survey were mentioned, which includes the limitations of the existing works and the approach to overcome these limitations. In section 2.3, the challenges are addressed. In section 2.4, the objectives of the proposed work are described.

2.1 SURVEY

Shadow detection and removal has been studied across a number of papers throughout the years. Shadow removal process extensively used for removing shadows from the images to make it suitable for higher level computer vision applications such as object detection, tracking. Shadow removal considered as the complex process compared to the shadow detection phase due to the difficulty while reconstructing the pixels in the detected shadow region. Jifeng Wang[1] introduced the ISTD dataset as part of the work titled “Stacked Conditional Generative Adversarial Networks for Jointly Learning Shadow Detection and Shadow Removal” and which considered as one of the benchmark dataset for the shadow removal process. From this view it is clear that the research in the shadow detection domain almost saturated and the focus is on the enhancements in the shadow removal phase. They proposed an architecture which contains two conditional GAN(cGAN) stacked together and performs shadow detection and removal tasks simultaneously. With this model RMSE value of 7.47 achieved on the ISTD dataset. Lack of considering the context information in the shadow removal phase was the drawback of this model.

In order to consider more details of the shadowed image like the illumination information, Ling Zhang[2] proposed a GAN based model which contains 4 generators and 3 discriminators in the work entitled “RIS-GAN: Explore Residual and Illumination with Generative Adversarial Networks for Shadow Removal”. This model achievedan RMSE value of 6.97 which is better than the model proposed by JifengWang[1]. The model trained on two different benchmark datasets for shadow removal which are SRD and ISTD datasets. RMSE value of 6.78 achieved for the the SRD dataset. Lower the RMSE value means better the quality of the generated shadow-less image.

The context information was still missing in these models. The model proposed by L. Qu[3] use the multi-context information for removing the shadow and reconstructing the shadowed region. Their work titled as “DeshadowNet: A Multi-context Embedding Deep Network for Shadow Removal ” uses only deep convolutional neural networks(DCNN) for shadow detection and shadow removal. Acquisition of multi-context information achieved by training three different networks named G-Net(Global Net), A-Net(Appearance net), S-Net(Semantic Net). Global context, appearance information and semantic context of the shadowed region captured by these networks. All these three networks together called as the DeshadowNet. The RMSE value scored on the SRD dataset is 6.64. Blurred effect was present in the shadowed region of the output images.

Generative Adversarial Network(GAN) base models produce high quality output images compared to the traditional model. Sidorov[4] proposed AngularGAN architecture to improve the color constancy on the images. Their work entitled “Conditional GANs for Multi-Illuminant Color Constancy: Revolution or Yet Another Approach?” used the GAN based model along with the angular information of the incident light on the object. The model trained on the synthetic dataset called GTAV dataset. Peak Signal to Noise Ratio(PSNR) used to evaluate the performance of the model. PSNR value for the GTAV dataset was 21dB. "Shadow Detection and Removal for Illumination Consistency on the Road" by Wang[5] showed promising results for shadow removal using non-linear SVM classifier. Proposed model was exclusively for traffic images and the model was not good for the large sized shadows in the scene. The model performs better for small sized shadows in the traffic images. Accuracy is used as the performance evaluation measure and achieved an accuracy of 83% on UCF dataset. They improved the accuracy of the previous work entitled "Shadow detection and removal for illumination consistency on the road"[9] by introducing adaptive variable scale regional compensation operator to remove the shadows.

Introduction of Generative Adversarial Networks[7] made drastic changes in the computer vision researches. Already implemented models migrated to GAN architectures as extension to their previous works to improve the results. Yun [6] Introduced GAN for shadow removal as an extension to the previous work entitled "Shadow Detection and Removal From Photo-Realistic Synthetic Urban Image Using Deep Learning" and improved the performance. However, traditional modeled methods have limited ability to remove shadows when irregular illumination or objects with various colors are present.

In order to address the color inconsistencies in the shadowed region, Xiaodong Cun[11] proposed a novel network structure called dual hierarchically aggregation network (DHAN) which contains a series of convolutions as the backbone without any down-sampling and hierarchically aggregate multi-context features for attention and prediction for successful shadow removal. The model proposed in the work entitled " Towards Ghost-free Shadow Removal via Dual Hierarchical Aggregation Network and Shadow Matting GAN" shows RMSE = 5.76 , PSNR = 34.48 dB for ISTD dataset and RMSE = 5.46, PSNR = 33.72 for SRD dataset respectively.

There are various ways of enhancing image quality. One of the most commonly used technique is interpolation. This is easy to use but this leads to distorted image or reduces the visual quality of image. Most common interpolation methods produce blurry images, i.e. bi-cubic interpolation. More sophisticated methods exploit internal similarities of a given image or, use datasets of low- resolution images and their high- resolution counterparts to effectively learn a mapping between them.

Image super-resolution using generative adversarial networks introduced by Christian Ledig[2]. The approach uses the generative adversarial network with perceptual loss which is a combination of the adversarial loss and the content loss. BSD100, set5, set14 benchmark datasets were used to evaluate the performance and the corresponding metrics are PSNR = 25.16 & SSIM = 0.6688. The deep learning model introduced by C.Dong[3] named SRCNN was the first deep learning method to outperform traditional ones. It is a Convolutional Neural Network consisting of only 3 convolution layers: patch extraction and representation, non- linear mapping and reconstruction.This method has improved metrics PSNR = 25.70 & SSIM = 0.7184, which is better than the earlier method.

To improve the performance of the model, J.Kim[4] introduced very deep convolutional neural network model for more accurate image super-resolution. This method uses the similar structure as the SRCNN with the depth of 20, which makes the network more deeper and produces more accurate results and PSNR = 27.29 & SSIM = 0.725. Enhanced Super-Resolution Generative Adversarial Networks by Wang X[5] introduced the Residual-in-Residual Dense Block (RRDB) without batch normalization as the basic network building unit in the generative adversarial network(GAN). The model has the performance metrics PSNR = 24.83 & RMSE = 15.15 . Another method using the Generative Adversarial Network for Image Super-Resolution[6] with Combined Texture Loss improved the performance of the super-resolution method with metrics PSNR = 27.67 & SSIM = 0.764 .

2.2 SUMMARY OF THE SURVEY AND FINDINGS

All existing methods to remove shadow are not ensuring the quality of the output image generated. The challenge of restoring the hidden details under the shadow is still open to research and further improvements are possible in this area. All the methods for image super-resolution are successfully enhance the Low Resolution images. Th evaluation metrics used such as PSNR, SSIM shows that the further improvements are needed in this area. There are no specific method available so far which always ensure the high quality shadow-less image for the given image.

2.3 CHALLENGES

Some of the challenges to do this work are mentioned as following:

- The shadow detection phase has greatly improved , but the field of shadow removal has been difficult because it needs to be restored after removing the shadow.
- The shadow removed region seems to be poor in quality compared to the non-shadow region in the shadow-less image.

2.4 OBJECTIVES OF THE WORK

- Use Generative Adversarial Network for enhancing the quality of the given low resolution image.
- Use this image super-resolution module with the shadow removal module to enhance the quality of the shadow-less image. - “Enhanced Shadow Removal module”
- Evaluate the performance of the proposed model using the evaluation metrics and compare with existing models.

CHAPTER 3

PROPOSED WORK

In this chapter, a detailed description of the proposed work is given. In section 3.1 the abstract architecture and flow of the whole system are described, which can give the overall idea about the proposed work. In section 3.2 in-depth explanation of each module of the work is given as a part of the algorithm design and in section 3.3, dataset information and details about the platform used for the experiments are given.

3.1 ARCHITECTURE OF THE SYSTEM

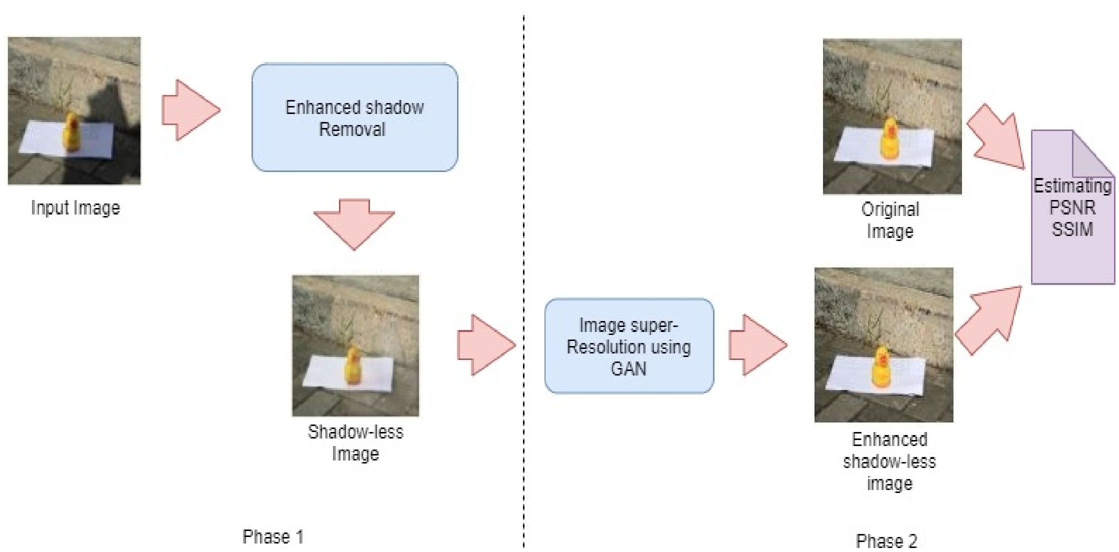


Fig. 4. Architecture Design of the system

In this section, the overview of the whole architecture design of the system is given.

The image super-resolution technique is very important in the shadow-removal process since it improves the resolution of the shadow-less image and make it suitable for the higher-level computer vision tasks.The basic flow diagram of the proposed approach illustrated in Fig4.. The shadowed input image is fed to the enhanced shadow removal module, which is a modified condition based generative adversarial network and generates the shadow-less image for the given shadowed image. In order to improve the quality of the shadow-less image, it is given as the input to the generative adversarial network based image super-resolution module and corresponding super-resolution image is generated. The evaluation metrics such as the PSNR and SSIM will be evaluated on the resulting image.

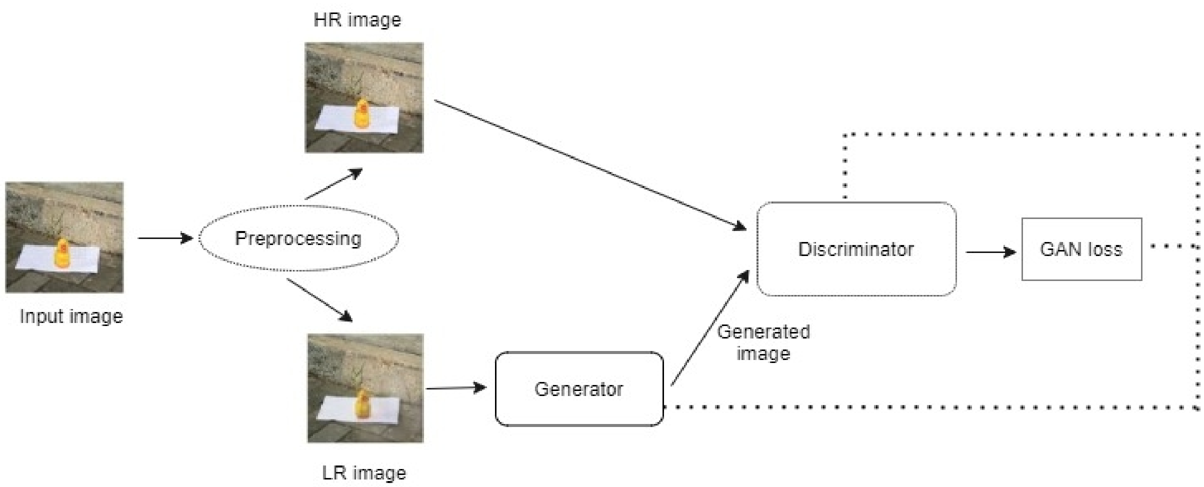


Fig. 5. Architecture Design of image super-resolution module

The basic architecture of the image super resolution module is illustrated in Fig5. The input image will be converted to HR(High Resolution) image and the LR(Low Resolution) image during the preprocessing stage using interpolation techniques such as bicubic interpolation. LR image will be given as the input to the generator and produces the SR(Super Resolution) Image called as the generated image. The discriminator will distinguish between the HR image and the SR image and ensure that the GAN generates the Super resolution image of the given Input image. The training of generator and Discriminator will be carried out using the GAN loss.

Here section 3.1.1 described the Architecture of the generator, and section 3.1.2 described the architecture of discriminator.

3.1.1 ARCHITECTURE OF GENERATOR

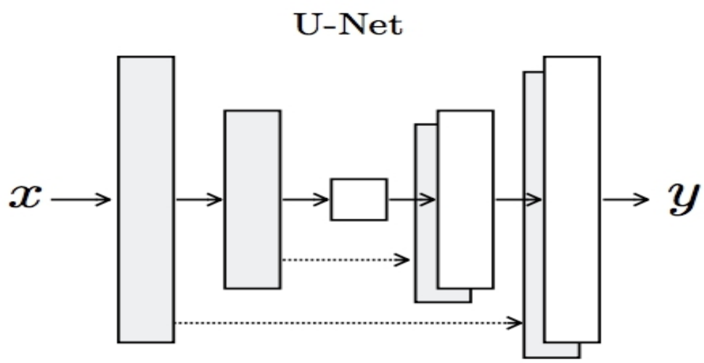


Fig. 6. Architecture of Generator

The Fig 6 Shows the overall abstract architecture of the Generator. The generator is an encoder-decoder model using a U-Net architecture[16]. The encoder and decoder of the generator contain convolutional, batch normalization, dropout, and activation layers. The encoder encodes the information in the given input image and the context information generated in the bottleneck is used for reconstructing the image using the decoder block. Skip connections are used between corresponding encoder and decoder layers for improving the quality of the image generated by the generator.

3.1.2 ARCHITECTURE OF DISCRIMINATOR

Architecture of the Discriminator is illustrated in Fig 7. The discriminator is a deep convolutional neural network that performs image classification. It takes both the source image (generated image) and the target image (shadow-less image) as input and predicts the likelihood of whether the shadow-less image is real or a fake translation of the generated image.

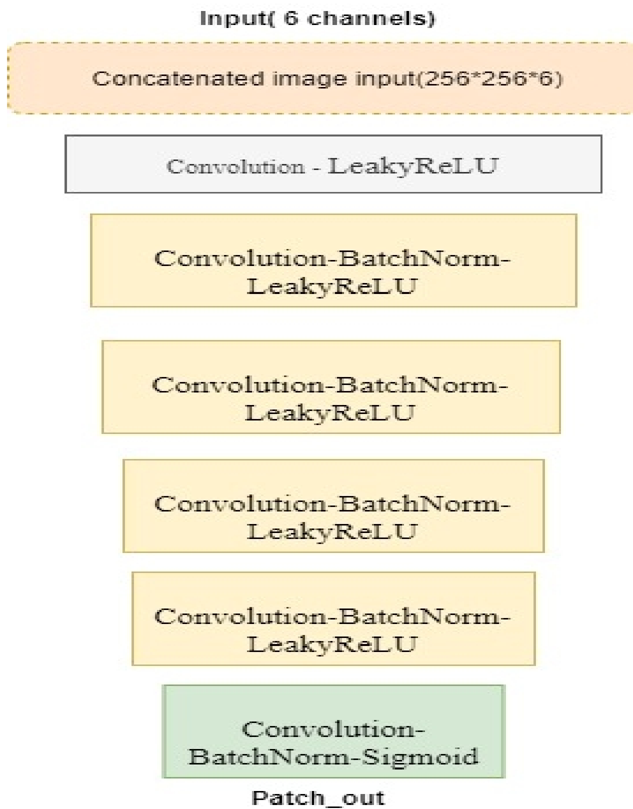


Fig. 7. Architecture of Discriminator

The discriminator model is trained in the same way as a traditional GAN model. Adversarial training is used for training the discriminator model.

3.2 ALGORITHM DESIGN

In this section, the algorithm design of image super-resolution architecture is discussed in detail. In section 3.2.1, proposed algorithm for image super-resolution using Generative adversarial network(GAN) is given along with its algorithm.

3.2.1 IMAGE SUPER-RESOLUTION ALGORITHM

The Adversarial training is used for training the discriminator model. The training of generator is too slow compared to the discriminator. This will lead to the issues in the GAN such as Vanishing gradient, Mode collapse, Non - convergence. The obvious solution is to balance generator & discriminator training to avoid over- fitting.

```
Algorithm 1: Enhancing Low Resolution image using GAN

Input : The LR and HR image
Output : Super-resolution(SR) image
d_epochs - discriminator training epochs
bs - batch size

for number of training epochs do
  for d_epochs do
        * sample minibatch of bs samples from LR images
        * sample minibatch of bs samples from HR images
        * Update the discriminator using adversarial loss


     end for
        * sample minibatch of bs samples from LR images
        * Update generator using combined loss
          Combined loss = λ1 * reconstruction loss + λ2 * adversarial loss
     end for
```

Fig. 8. Algorithm for Image Super-resolution module

Algorithm for the image super-resolution module is illustrated in Fig 8. Low Resolution(LR) and High Resolution (HR) images will be given as the input and the

Super Resolution image is the expected output. The training epochs of the discriminator is given as 'd_epochs'. The batch size is given as 'bs'. The discriminator will be trained first using the LR and HR images of the specified batch size and updated using the adversarial loss. After the training of the discriminator, generator training will start. The generator will be updated based on the combined loss.

Combined Loss(1) introduced to balance the training of Generator and Discriminator. Giving more importance to the reconstruction loss(2) rather than the adversarial loss(3) during the Generator training will reduce the effect of the fast training of discriminator on the Generator.

$$\textbf{Combined loss} = \boldsymbol{\lambda 1} * \textbf{ reconstruction loss} + \boldsymbol{\lambda 2} * \textbf{ adversarial loss} \qquad (1)$$

Where,

$\lambda 1, \lambda 2 = 100, 1$

Loss function between the generated fake image and the shadow-less image is called as the reconstruction loss.

$$\textbf{Lreconstruction} = \sum_{i=1}^{n} |Ytrue - Ypredicted| \qquad (2)$$

Where,

Ypred = The predicted value for the ith pixel.

Ytrue = The observed(actual) value for the ith pixel

n = Total number of pixels.

Adversarial loss (Binary Cross Entropy loss)

$$L_{adversarial}(G,D) = E_{xy}[\log D(x,y)] + E_{xy}[\log(1-D(x,G(x,y)))] \quad (3)$$

Where,

G - Generator , D - Discriminator

x - LR image , y - HR image

D(x,y) is the discriminator's estimate of the probability that real data instance x is real with respect to y.

G(x,y) is the generator’s output.

3.3 DATASETS FOR THE STUDY AND PLATFORM

Here in section 3.3.1, all the datasets [34,35] are discussed which were used for the experiments. In section 3.2.2 we mentioned the architectural information of the development platform of this work.

3.3.1 DATASET INFORMATION

BSD100 is the dataset used in this project for training the image-super resolution module. It is a benchmark datasets used for the single image super resolution.

ISTD and SRD datasets were used to evaluate the performance of the shadow removal module.

3.3.2 PLATFORM AND SPECIFICATIONS

All experiments are performed on the on-premise system, which contains an Intel Core i7 10th generation CPU with 2.60GHz clock speed. It also contains a 4 GB Nvidia GeForce GTX 1650 GPU and 8 GB of system memory.

CHAPTER 4

RESULTS AND INFERENCES

In this chapter, the evaluation metrics, setting the best parameter values to get good results in the experiments, and final generated results are briefly discussed. In section 4.1 all the evaluation metrics which are used for this experiment with their detailed description are described along with the reason behind choosing that evaluation matrics. In section 4.2 all the parameter values used for the experiment are mentioned and In section 4.3, a detailed description is given about the all obtained results and inferences.

4.1 METRICS FOR EVALUATION

To measure the performance of the model, the selection of the proper evaluation metrics plays a major role. To check the efficiency of the proposed approach, Peak Signal-to-noise Ratio(PSNR) and Structural Similarity Index(SSIM) are calculated.

- **Peak Signal-to-noise Ratio(PSNR):**

  This ratio is used as a quality measurement between the original and a compressed image.Higher the PSNR, the better the quality of the compressed or reconstructed image.The metric is commonly used to quantify reconstruction quality for images and video subject to lossy compression.

$$\mathbf{PSNR(dB) = 10*log(\frac{255^2}{MSE})}$$

Where,

MSE - Mean Squared Error

- **Structural Similarity Index(SSIM):**

The SSIM is used as a metric to measure the perceptual similarity between two given images. This metric is more aligned with human visual system. Instead of the pixel-to-pixel comparison, group of pixels are considered from both the images and the resultant value ranges from -1(very different) to +1(very similar).

$$\textbf{SSIM(x,y)} = \frac{(2\mu x\mu y + c1)\,(2\sigma\, xy + c2)}{(\mu x^2 + \mu y^2 + c1)\,(\sigma\, x^2 + \sigma\, y^2 + c2)}$$

Where,

Standard Deviation - σ

Mean - **μ**

4.2 PARAMETERS SETTINGS

The experiments were performed for this work based on the following parameters:

- **Activation:** ReLU
- **Loss function:** Combined loss (1)
- **Optimizer:** Adam
- **Learning Rate:** 0.1
- **Maximum Epochs:** 1000
- **CPU:** Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz
- **GPU:** 4 GB Nvidia GeForce GTX 1650
- **RAM:** 8 GB

4.3 RESULTS AND DISCUSSION

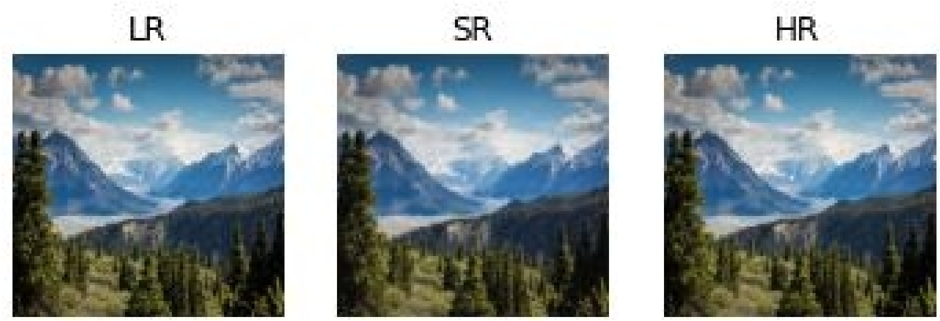


Fig. 9. Sample output of the Image super-resolution module

As discussed in Section 3.2.1, a Low Resolution image and corresponding High resolution image are given to the image super-resolution module and generate the corresponding Super Resolution image. The sample output is depicted in Fig 9. The PSNR is 69dB which is a good value. SSIM metric for the output image given in Fig 9 is 0.77, which states that the HR image and the SR images have high similarity according to human visual system perspective.

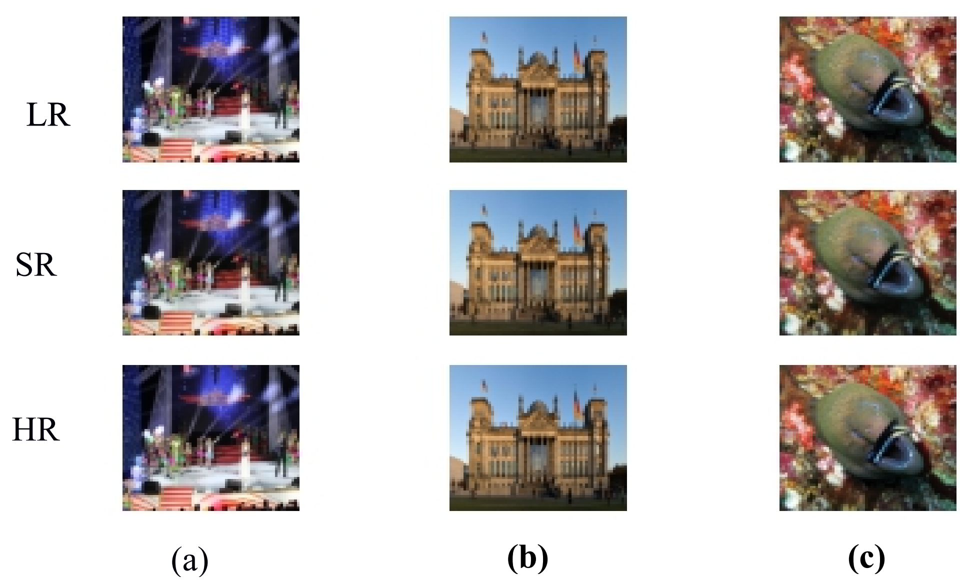


Fig. 10. Sample output of the Image super-resolution module

The evaluation metrics such as SSIM, PSNR values corresponding to the Fig 10 is given in the Table 1.

Table 1 : PSNR & SSIM metrics for the sample output in Fig 10

	Image (a)	Image (b)	Image (c)
PSNR(dB)	73.56	73.01	71.15
SSIM	0.80	0.79	0.78

As per the metric values given in Table 1, It’s clear that our model successfully enhance the given low resolution image. The detailed comparison of the image-resolution module with the standard algorithms on the BSD100 dataset is given in Table 2.

Table 2: Performance Analysis of the image super-resolution module

Model	PSNR(dB)	SSIM
SRGAN[7]	25.16	0.668
ESRGAN[8]	27.66	…..
GAN combining Texture loss[6] (2020)	27.99	0.778
Our Model	72.67	0.798

From Table 2, Its evident that our image super-resolution module shows more efficiency in converting the Low resolution image to the Corresponding Super

Resolution image. The introduction of the combined loss(1) and selection of $\lambda 1=100$ made more importance to the reconstruction loss(2) while training the Generator. This lead to the reduction of the pixel-to-pixel comparison values.

After the successful training of the image enhancement module, resultant model will be added to the shadow removal pipeline to create the "Enhanced Shadow removal module".

Enhanced shadow removal module output ↑ .

Image super resolution module output ↓

Fig. 11. Sample output of the Enhanced Shadow removal module

Sample output of the Enhanced Shadow Removal Module is illustrated in Fig 11.The shadowed input image will be given to the module. The module consist of the shadow removal module and the image super-resolution module connected in sequence. Shadow Free(SF) image will be generated by the shadow-removal module and the image will then fed to the image super-resolution module. The Image super resolution module will produce Enhanced Shadow free(ESF) image which is of higher quality in terms of contrast, luminance, Structures. The PSNR = 70.55dB and SSIM =0.732 for the output shown in Fig 11.

Fig. 12. Sample output of the Enhanced Shadow removal module

The sample output of the Enhanced shadow removal module on the SRD dataset is shown in Fig 12. The shadow_free image is the output of the shadow_removal module for the given shadowed input. Enhanced_output is the final output of the Enhanced shadow-removal module. Overall quality of the shadow_free image improved further by introducing the image super-resolution module.

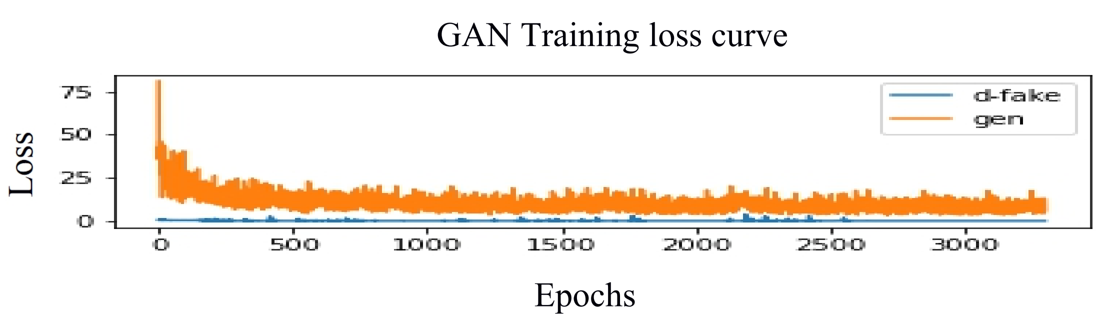


Fig. 13. Generator and Discriminator loss

The generator and discriminator loss is illustrated in the Fig 13. The discriminator loss is low compared to the generator and the generator loss reduces over time and it indicates that the generated started generating super resolution images which are almost similar to the HR images given to the Discriminator as the ground truth image.

Table 3: Performance Analysis of the Enhanced Shadow Removal module.

Model	RMSE	PSNR	SSIM
ST-GAN[7]	7.47	30.66	…...
RIS-GAN[8]	6.67	31.64	…...
Our model	1.997	53.38	0.722

Performance analysis of the enhanced shadow removal module is given in the Table 3. Our model performs well compared to the standard algorithms. SSIM introduced as the new evaluation metric and shows good results.

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENTS

In this chapter, the proposed work is concluded and given a detailed explanation about the work summary and contributions towards this work. In section 5.1, the summary of the whole project work is given, In section 5.2 the major contributions towards this work are mentioned and in section 5.3 “Where to Apply this model? “ and the final process ready Enhanced shadow-Removal module are described. In section 5.4, the future directions of this work are given.

5.1 SUMMARY OF THE WORK

In this work, an Enhanced shadow-removal method were introduced and the pipeline consists of the shadow-removal module and the image super-resolution module. Generative adversarial network with combined loss improved the quality of the image and the PSNR of the image super-resolution module was 72.67dB. This clearly shows the improvement of our model compared to the standard algorithms whose highest PSNR value achieved was 27.99dB.

The Enhanced shadow Removal module successfully removes shadows from the given image and ensure the higher quality image at the end of the pipeline. For the Enhanced Shadow-Removal module, the SSIM metric evaluated was 0.722 and it shows that the model produce more similar images like the ground truth images given to the model.

5.2 CONTRIBUTIONS IN THIS WORK

Major technical contributions in this work are as follows:

- GAN based image super-resolution module with custom loss functions and generator, discriminator architecture which gives promising results compared to the standard algorithms
- Enhanced Shadow-Removal pipeline to remove shadow, which always guaranties high quality output image for the given input image.

- Introduction of the combined loss which improved the overall performance of the GAN and helped to generate higher quality images as well as extensive shadow-removal.

5.3 WHERE TO APPLY THIS MODEL?

Enhanced shadow removal can be used in the preprocessing phase of the higher level computer vision applications as depicted in the Fig 14.

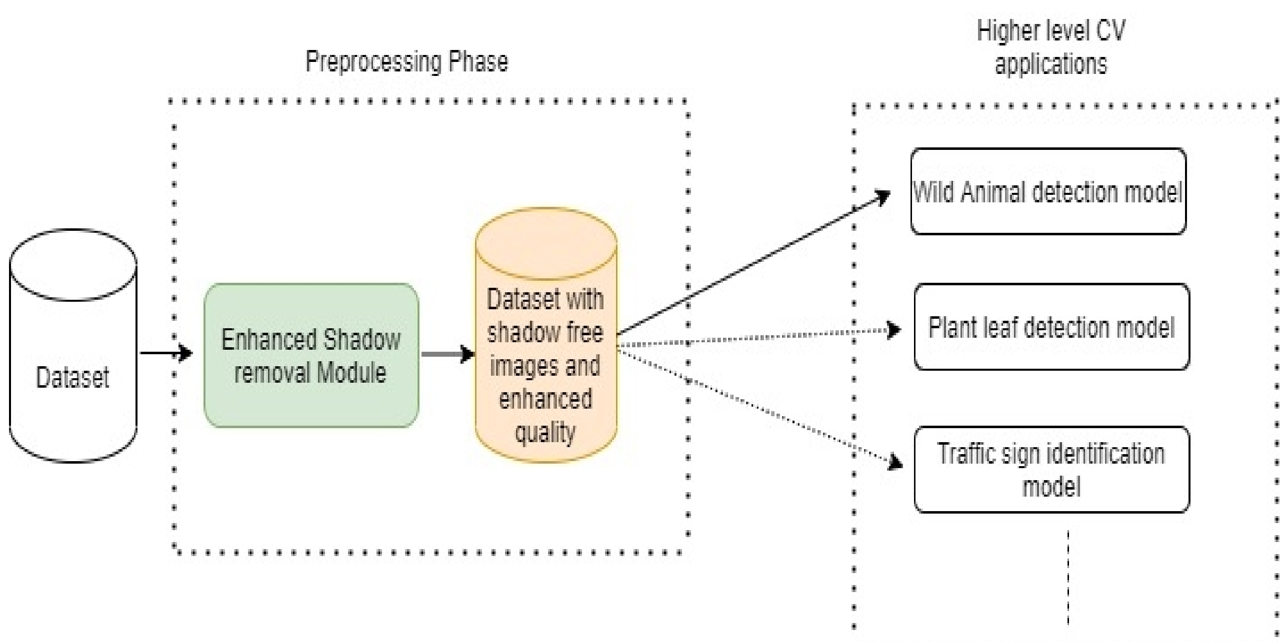


Fig. 14. Application of Enhanced Shadow-Removal module

During the preprocessing phase, the dataset can be passed through the enhanced shadow removal module and the corresponding dataset with shadow-free images and enhanced quality will be generated. Then, this dataset can be used in further applications like Wild Animal detection model, plant leaf detection model , etc.

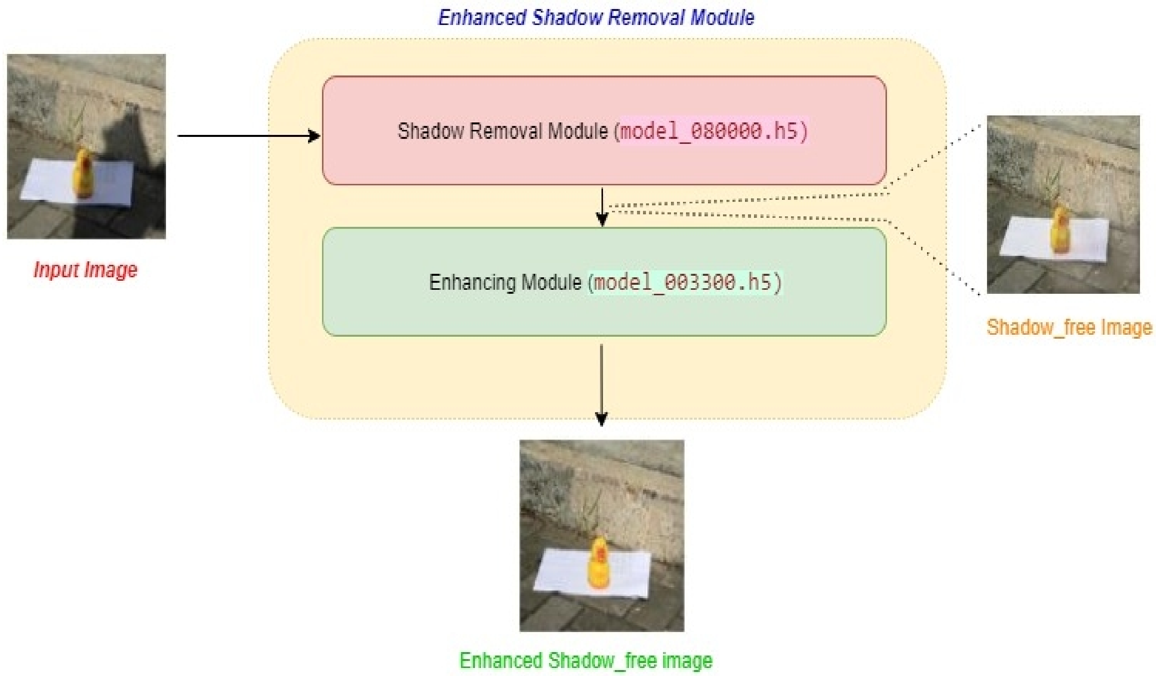


Fig. 15. Process ready Enhanced Shadow-Removal module

The process ready Enhanced Shadow-removal module illustrated in Fig 15. The module contains the saved models of Image-super resolution(model_003300.h5) and Shadow-removal(model_080000.h5) modules after the training. The input image will be given to the Enhanced shadow-removal module. It will generate corresponding shadow-less image first. Then the image will pass through the image super-resolution module and gives the final output.

5.4 FUTURE ENHANCEMENTS

The investigated model successfully removes shadow and ensure that the resultant image is of higher quality. The Enhanced shadow removal module performs well and removes shadows as well as improves the image quality. Our final module for shadow removal not only removes the shadow but also guaranties high quality shadow free images for further high level Computer Vision Applications.

Genetic algorithm can be introduced for tuning the hyper parameters as a future enhancement. The model can be trained on large dataset and for more epochs to improve the performance according to the availability of computation resources.

The process ready Enhanced shadow-removal module produces the shadow-free images for the given shadowed input within 43secs. This time is huge in the real-time applications. The light-weight models can be introduced as the future enhancement to this problem.

CHAPTER 6

REFERENCES

1. [1]Jifeng Wang, Xiang Li2, Jian Yang, “Stacked Conditional Generative Adversarial Networks for Jointly Learning Shadow Detection and Shadow Removal”, CVPR 2017,https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00192

2. [2]Ling Zhang,1 Chengjiang Long,2∗ Xiaolong Zhang,1 Chunxia Xiao3, “RIS-GAN: Explore Residual and Illumination with Generative Adversarial Networks for Shadow Removal”, AAAI 2020,https://doi.org/10.1609/aaai.v34i07.6979

3. [3]L. Qu, J. Tian, S. He, Y. Tang and R. W. H. Lau, "DeshadowNet: A Multi-context Embedding Deep Network for Shadow Removal," IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017,https://ieeexplore.ieee.org/document/8099731

4. [4]Sidorov, Oleksii, “Conditional GANs for Multi-Illuminant Color Constancy: Revolution or Yet Another Approach?”,The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018,https://arxiv.org/abs/1811.06604

5. [5]Wang, H. Xu, Z. Zhou, L. Deng and M. Yang, "Shadow Detection and Removal for Illumination Consistency on the Road," in IEEE Transactions on Intelligent Vehicles, 2020, https://ieeexplore.ieee.org/document/9068460

6. [6]Yun, Heejin & Kang Jik, Kim & Chun, Jun-Chul. “Shadow Detection and Removal From Photo-Realistic Synthetic Urban Image Using Deep Learning”, Computers, Materials & Continua 2019, https://www.techscience.com/cmc/v62n1/38123

7. [7]Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, “Generative Adversarial Networks”, International Conference on Neural Information Processing Systems, 2014, https://arxiv.org/abs/1406.2661

8. [8]M. Mirza, S. Osindero, "Conditional Generative Adversarial Nets", In Arxiv 2014. http://arxiv.org/abs/1411.1784

9. [9]Wang, L. Deng, Z. Zhou, M. Yang and B. Wang, "Shadow detection and removal for illumination consistency on the road,"https://ieeexplore.ieee.org/document/8304275

10. [10]M. Wang, Y. C. Chung, C. L. Chang and S. W. Chen, "Shadow detection and removal for traffic images," IEEE International Conference on Networking, Sensing and Control, 2004, https://ieeexplore.ieee.org/document/1297516

11. [11]Xiaodong Cun, Chi-Man Pun, Cheng Shi, " Towards Ghost-free Shadow Removal via Dual Hierarchical Aggregation Network and Shadow Matting GAN" AAAI 2020, https://arxiv.org/abs/1911.08718

12. [12]Olaf Ronneberger, Philipp Fischer, Thomas Brox,"U-Net: Convolutional Networks for Biomedical Image Segmentation"MICCAI 2015, https://arxiv.org/abs/1505.04597

13. [13]Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network" , Computer Vision and Pattern Recognition , 2017,https://arxiv.org/abs/1609.04802v5

14. [14]J. Kim, J. K. Lee and K. M. Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1646-1654, doi: 10.1109/CVPR.2016.182.

15. [15]C. Dong, C. C. Loy, K. He and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 2, pp. 295-307, 1 Feb. 2016, doi: 10.1109/TPAMI.2015.2439281.

16. [16]Wang X. et al. (2019), " ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks." In: Leal-Taixé L., Roth S. (eds)

Computer Vision – ECCV 2018 Workshops. ECCV 2018,Springer, Cham. https://doi.org/10.1007/978-3-030-11021-5_5

17. [17]Jiang Y, Li J. Generative Adversarial Network for Image Super-Resolution Combining Texture Loss. Applied Sciences. 2020; 10(5):1729. https://doi.org/10.3390/app10051729

CHAPTER 7

LIST OF PUBLICATIONS

1. Jariwala, R. C., & Nalluri, M. R. (2021). Orthonormal Bayesian Convolutional Neural Network for Detection of the Novel Coronavirus-19. In S. Mekhilef, M. Favorskaya, R. K. Pandey, & R. N. Shaw (Eds.), *Innovations in Electrical and Electronic Engineering* (pp. 819–836). Springer. https://doi.org/10.1007/978-981-16-0749-3_63

Abstract:

Novel coronavirus-19 (COVID-19) is almost affected all over the world, and daily thousands of peoples are getting infected and died due to this virus. All the existing methods to test coronavirus infections are taking a long time to give results and giving low sensitivity and specificity. This necessitates Artificial Intelligence (AI) in the detection of COVID-19 infected patients. Various deep learning and image processing techniques are available to recognize patterns from the X-ray and the Computed tomography (CT) image and giving higher testing accuracy, sensitivity, and specificity on a large dataset. In this research, we adapted the Bayesian inference approach and proposed a Bayesian convolutional neural network (BCNN) with an Orthogonal Normalization Technique (ONT) for automatic detection of the COVID-19, Pneumonia, and No-infection (Normal) from CT Images. To verify the strength of the model, the data is randomly split-ups into different training, validation, and testing set combinations. With the proposed model on (training%_validation%_testing%) 70_10_20 split-up, we achieved 99.91% testing accuracy with 99.91% F1 score and 100% Area under the Receiver Operating Characteristic Curve (AUC). We also achieved 100% sensitivity, 100% specificity on COVID-19 prediction, 100% sensitivity, 99.97 % specificity on Normal prediction, and 100% sensitivity, 99.91% specificity on the Pneumonia prediction. The higher accuracy, sensitivity, and specificity of this model would be extremely useful for the accurate and rapid screening of the COVID-19 and Pneumonia.

2. Jariwala, R. C., & Mathi, S. (2021). Quick Shift Clustering-Based Unsupervised Convolutional Neural Network To Detect The Severity Of Novel Coronavirus-19 (Under review)

Abstract:

The novel coronavirus-19 (2019-NCOV) disease has had a substantial influence on the worldwide global public healthcare sector and the situation has become more treacherous due to the new mutant variants of the virus. All of medical science's coronavirus detection procedures provide no information regarding the disease's severity. In this research, the quick shift clustering-based unsupervised convolutional neural network is investigated to detect the 2019-NCOV severity includes Severity-1 (Highly severe), Severity-2 (Moderate severe), and Severity-3 (Low severe) from the Computed Tomography scan images. The proposed approach averagely outperforms the compact watershed algorithm by 2.53% of the Ground Glass Opacification patterns. The model is also giving the Intersection over Union greater than 0.86 and Dice coefficient score greater than 0.92 for all testing images from the MedSeg 2019-NCOV segmentation dataset. The proposed approach is quickly giving accurate results so it is useful to prioritize the patients for the medical treatment in case of overflooding of the 2019-NCOV cases due to the new mutant variants of the virus.

CHAPTER 8

APPENDIX

- **Code for Lung extraction approach (MATLAB)**

```
%----------------------Read Image and preprocess it----------------------

folder = 'D:\Academics\Amrita\Final_Year_Project\Phase-
2\severity_detection\images';
baseFileName = 'NCP_17_1166_0052.png';
% Get the full filename, with path prepended.
fullFileName = fullfile(folder, baseFileName);

%converting image to grayscale for simplification
grayImage = imread(fullFileName);
imwrite(grayImage, 'D:\Academics\Amrita\Final_Year_Project\Phase-
2\severity_detection\temp.png');
% Get the dimensions of the image.
% numberOfColorBands should be = 1.
[rows, columns, numberOfColorBands] = size(grayImage);
disp(numberOfColorBands)
if numberOfColorBands > 1
    % It's not really gray scale like we expected - it's color.
    % Convert it to gray scale by taking only the green channel.
    grayImage = grayImage(:, :, 2); % Take green channel.
end

%---------------------------Logic Starts -------------------------------

% Threshold (binarize) the image.
thresholdValue = 255;
binaryImage = grayImage < thresholdValue; % Do the thresholding.
%imshow(binaryImage)
imwrite(binaryImage, 'D:\Academics\Amrita\Final_Year_Project\Phase-
2\severity_detection\binary_image.png');

% Get rid of stuff touching the border
```

```
binaryImage = imclearborder(binaryImage);
%imshow(binaryImage)
imwrite(binaryImage, 'D:\Academics\Amrita\Final_Year_Project\Phase-
2\severity_detection\after_clear_border.png');

% Extract only the two largest blobs(group of connected pixels in a binary image)
binaryImage = bwareafilt(binaryImage, 2);
%imshow(binaryImage)
imwrite(binaryImage, 'D:\Academics\Amrita\Final_Year_Project\Phase-
2\severity_detection\after_extrected_two_largest_blob.png');

% Fill holes in the blobs to make them solid.
% This will produce a gray scale image in the lungs and black everywhere else.
binaryImage = imfill(binaryImage, 'holes');
%imshow(binaryImage)
imwrite(binaryImage, 'D:\Academics\Amrita\Final_Year_Project\Phase-
2\severity_detection\fill_holes.png');



% Adding the mask of newly generated binaryImage on the original gray Image
maskedImage = grayImage; % Initialize (original image)
maskedImage(~binaryImage) = 0;
imwrite(binaryImage, 'D:\Academics\Amrita\Final_Year_Project\Phase-
2\severity_detection\adding_Mask.png');

imwrite(maskedImage, 'D:\Academics\Amrita\Final_Year_Project\Phase-
2\severity_detection\masked_image.png');
```

- **Code for Unsupervised segmentation and Severity detection (Python)**

```
import cv2
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torchvision import datasets, transforms
```

```
from torch.autograd import Variable
import sys
import numpy as np
from skimage import segmentation
import torch.nn.init

input_image = "masked_image.png"
image = cv2.imread(input_image)

use_cuda = torch.cuda.is_available()

nChannel = 100
maxIter = 1000
minLabels = 3
lr = 0.07

# CNN
class MyNet(nn.Module):
    def __init__(self,input_dim):
        super(MyNet, self).__init__()
        self.conv1 = nn.Conv2d(input_dim, nChannel, kernel_size=3, stride=1,
padding=1 )
        self.bn1 = nn.BatchNorm2d(nChannel)
        self.conv2 = nn.Conv2d(nChannel, nChannel, kernel_size=3, stride=1,
padding=1 )
        self.bn2 = nn.BatchNorm2d(nChannel)
        self.conv3 = nn.Conv2d(nChannel, nChannel, kernel_size=3, stride=1,
padding=1 )
        self.bn3 = nn.BatchNorm2d(nChannel)
        self.conv4 = nn.Conv2d(nChannel, nChannel, kernel_size=3, stride=1,
padding=1 )
        self.bn4 = nn.BatchNorm2d(nChannel)
        self.conv5 = nn.Conv2d(nChannel, nChannel, kernel_size=3, stride=1,
```

```
padding=1 )
        self.bn5 = nn.BatchNorm2d(nChannel)
        self.conv6 = nn.Conv2d(nChannel, nChannel, kernel_size=1, stride=1,
padding=0 )
        self.bn6 = nn.BatchNorm2d(nChannel)

def forward(self, x):
        x = self.conv1(x)
        x = F.relu( x )
        x = self.bn1(x)

        x = self.conv2(x)
        x = F.relu( x )
        x = self.bn2(x)
        x = self.conv3(x)
        x = F.relu( x )
        x = self.bn3(x)

        x = self.conv4(x)
        x = F.relu( x )
        x = self.bn4(x)
        x = self.conv5(x)
        x = F.relu( x )
        x = self.bn5(x)

        x = self.conv6(x)
        x = self.bn6(x)
        return x

im = image
data = torch.from_numpy( np.array([im.transpose( (2, 0, 1) ).astype('float32')/255.]) )
if use_cuda:
```

```
    data = data.cuda() #[1,3,512,512]

model = MyNet( data.size(1) )
if use_cuda:
    model.cuda()
model.train()

labels = segmentation.quickshift(im, kernel_size=3, max_dist=6, ratio=0.5)
labels = labels.reshape(im.shape[0]*im.shape[1])
u_labels = np.unique(labels)
l_inds = []
for i in range(len(u_labels)):
    l_inds.append( np.where( labels == u_labels[ i ] )[ 0 ] )

loss_fn = torch.nn.CrossEntropyLoss()
optimizer = optim.SGD(model.parameters(), lr=lr, momentum=0.9)
label_colours = np.random.randint(255,size=(100,3))

# Train
for batch_idx in range(maxIter):
    optimizer.zero_grad()
    output = model( data )[ 0 ]
    output = output.permute( 1, 2, 0 ).contiguous().view( -1, nChannel ) #shape:
[262144, 100]

    ignore, target = torch.max( output, 1 )
    im_target = target.data.cpu().numpy()
    nLabels = len(np.unique(im_target))

    for i in range(len(l_inds)):
        labels_per_sp = im_target[ l_inds[ i ] ]
        u_labels_per_sp = np.unique( labels_per_sp )
```

```
        hist = np.zeros( len(u_labels_per_sp) )
        for j in range(len(hist)):
            hist[ j ] = len( np.where( labels_per_sp == u_labels_per_sp[ j ] )[ 0 ] )
            im_target[ l_inds[ i ] ] = u_labels_per_sp[ np.argmax( hist ) ]
    target = torch.from_numpy( im_target )
    if use_cuda:
        target = target.cuda()
    loss = loss_fn(output, target) #loss_fn([262144, 100],[262144])

    loss.backward()
    optimizer.step()

    print (batch_idx, '/', maxIter, ':', nLabels, loss.item())

    if nLabels <= minLabels:
        print ("nLabels", nLabels, "reached minLabels", minLabels, ".")
        break

# Save output
output = model( data )[ 0 ]
output = output.permute( 1, 2, 0 ).contiguous().view( -1, nChannel )
ignore, target = torch.max( output, 1 )
im_target = target.data.cpu().numpy()
im_target_rgb = np.array([label_colours[ c % 100 ] for c in im_target])
im_target_rgb = im_target_rgb.reshape( im.shape ).astype( np.uint8 )
cv2.imwrite( "output.png", im_target_rgb )

#Severity detection
import webcolors
from PIL import Image
im = Image.open('output.png')
from collections import defaultdict
```

```
by_color = defaultdict(int)
for pixel in im.getdata():
    by_color[pixel] += 1
by_color

total_pixels = 0
all_pixels=[]
for k,v in by_color.items():
    all_pixels.append(v)
    total_pixels+=v
pixels_covered_by_lungs = total_pixels-max(all_pixels)
percentages =[]
all_pixels.remove(max(all_pixels))
for i in all_pixels:
    percentages.append(i/pixels_covered_by_lungs)

Lungs_covered_COVID19 = min(percentages)*100
print("Lungs coverd with COVID19:",Lungs_covered_COVID19,'%')

if(Lungs_covered_COVID19 >= 10):
    print("severity-1 (Highly severe)")
elif(Lungs_covered_COVID19 >=5 and Lungs_covered_COVID19 <10):
    print("severity-2 (Moderate severe)")
else:
    print("severity-3 (Low severe)")
```

- **Validation on MedSeg dataset (Python)**

```
prefix = 'D:/Academics/Amrita/Final_Year_Project/Phase-
2/severity_detection/Validation datasets/Kaggle-medseg/'
images_radiopedia = np.load(os.path.join(prefix,
'images_radiopedia.npy')).astype(np.float32)
masks_radiopedia = np.load(os.path.join(prefix,
```

```
'masks_radiopedia.npy')).astype(np.int8)
images_medseg = np.load(os.path.join(prefix,
'images_medseg.npy')).astype(np.float32)
masks_medseg = np.load(os.path.join(prefix, 'masks_medseg.npy')).astype(np.int8)

test_images_medseg = np.load(os.path.join(prefix,
'test_images_medseg.npy')).astype(np.float32)



import uuid
def visualize(image_batch, mask_batch=None, pred_batch=None, num_samples=8,
hot_encode=True):
    num_classes = mask_batch.shape[-1] if mask_batch is not None else 0
    fix, ax = plt.subplots(num_classes - 2, num_samples, figsize=(num_samples * 2,
(num_classes + 1) * 2))

    for i in range(num_samples):
        u = uuid.uuid4()
        ax_image = ax[0, i] if num_classes > 0 else ax[i]
        if hot_encode:
            ax_image.imshow(image_batch[i,:,:,0], cmap='Greys')

np.save('./New_validation_data/Original_images/'+str(u)+'.npy',image_batch[i,:,:,0])
        else:
            ax_image.imshow(image_batch[i,:,:])

np.save('./New_validation_data/Original_images/'+str(u)+'.npy',image_batch[i,:,:])
        ax_image.set_xticks([])
        ax_image.set_yticks([])

        if mask_batch is not None:
            for j in range(2,num_classes-1):
                if pred_batch is None:
                    mask_to_show = mask_batch[i,:,:,j]

np.save('./New_validation_data/Ground_Truth/'+str(u)+'.npy',mask_to_show)
                ax[j- 1, i].imshow(mask_to_show, vmin=0, vmax=1)
                ax[j -1, i].set_xticks([])
                ax[j -1, i].set_yticks([])

    plt.tight_layout()
    plt.show()
```

```
prediction= cv2.imread("./Extracted_validataion_data/Results/image1/output.png")
target = cv2.imread("./Extracted_validataion_data/Results/image1/ground.png")

Target_norm = cv2.normalize(target, None, alpha=0, beta=2.2,
norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_32F)
prediction_norm = cv2.normalize(prediction, None, alpha=0, beta=2.2,
norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_32F)

#IoU

def IoU(y_true, y_pred):
    y_pred_f = K.flatten(y_pred)
    y_true_f = K.flatten(y_true)
    intersection = K.sum (y_true_f * y_pred_f)
    union = K. sum (y_true_f) + K.sum(y_pred_f) - intersection
    return intersection/union
IoU = IoU(Target_norm,prediction_norm)
print("IoU score==>",np.array(IoU).tolist()*100,' %')

#Dice coefficient

def dice_coef(y_true, y_pred, smooth=1):
    y_true_f = K.flatten(y_true)
    y_pred_f = K.flatten(y_pred)
    intersection = K.sum (y_true_f * y_pred_f)
    return (2. * intersection + smooth) / (K.sum(y_true_f) + K.sum(y_pred_f) + smooth)
dice = dice_coef(Target_norm,prediction_norm)
print("Dice coefficient score==>",np.array(dice).tolist()*100,' %')
```