REMOVAL OF OCCLUDING FENCES FROM AN IMAGE

A PROJECT REPORT

Submitted by

DEEPTHI BHAT (1BM16CS003)
ADITI AWASTHI (1BM16CS008)
MEDHINI OAK (1BM16CS047)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Under the Guidance of

Dr. KAYARVIZHY N
Associate Professor, BMSCE

B. M. S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
MAY 2020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project entitled REMOVAL OF OCCLUDING FENCES FROM AN IMAGE is a bonafide work carried out by DEEPTHI BHAT (1BM16CS003), ADITI AWASTHI (1BM16CS008), and MEDHINI OAK (1BM16CS047) in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the academic year 2019-20. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Guide	Head of Department
Dr. Kayarvizhy N Associate Professor Dept of CSE, B.M.S. College of Engineering	**Dr. Umadevi V** Associate Professor and Head Of Department Dept of CSE, B.M.S. College of Engineering

Principal

Dr. B. V. Ravishankar

B.M.S. College of Engineering

External Viva

Name of the Examiners	Signature with Date
1.	
2.	

TABLE OF CONTENTS

ABSTRACT

Photographers are often hindered in their attempts at capturing pictures by unsuitable imaging conditions. One such condition is the presence of grills, fences or enclosures between the camera and the scene of interest. Due to growing security concerns for the safety of both private property and public places, the presence of such barriers cannot be avoided. Hence, there is a need for a tool which can remove the occlusion from the clicked image as if it were never there and replace it with content that blends with the background. In this project, we intend to detect and remove the occlusion from a given input image and then generate the content to be filled in the area occupied by the occlusion by using an image-processing algorithm. This system employs a defencing algorithm, followed by an exemplar-based image in-painting algorithm. The resulting image will be rid of the fence and will resemble an image taken without the presence of occlusions.

DECLARATION

We, hereby declare that the dissertation work entitled REMOVAL OF OCCLUDING FENCES FROM AN IMAGE is a bonafide work and has been carried out by us under the guidance of Dr. Kayarvizhy N, Associate Professor, Department of Computer Science and Engineering, B.M.S. College of Engineering, Bengaluru, in partial fulfillment of the requirements of the degree of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belgaum.

We further declare that, to the best of our knowledge and belief, this project has not been submitted either in part or in full to any other university for the award of any degree.

Candidate details:

Sl. No.	Student Name	USN	Student's Signature
1	DEEPTHI BHAT	1BM16CS003	
2	ADITI AWASTHI	1BM16CS008	
3	MEDHINI OAK	1BM16CS047	

Place: Bengaluru

Date:

This to certify that these candidates are students of Computer Science and Engineering Department of B.M.S. College of Engineering and have carried out the project work titled "**Removal of occluding fences from an image**" as final year (7th & 8th Semester) dissertation project. It is in partial fulfillment for completing the requirement for the award of B.E. degree by VTU. The works is original and duly certify the same.

Project Guide **Signature and Date**

Dr. Kayarvizhy N

Associate Professor,

Department of Computer Science and Engineering

B.M.S. College of Engineering

ACKNOWLEDGEMENTS

We have put a lot of effort into this project as a team. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We thank our dear Principal – Dr. B.V. Ravishankar, and the management of our college for providing the necessary infrastructure we needed to accomplish this project.

We would also like to take this opportunity to express our gratitude to the Head of Our Department – Dr. Umadevi V, for her kind and constant support.

We are highly indebted to Dr. Kayarvizhy N for her guidance and constant supervision as well as for providing necessary information regarding the project and for her unwavering support in completing the project.

We would like to express our gratitude towards our parents for their kind cooperation and encouragement which helped us in the completion of this project.

We would like to express our special gratitude and thanks to the freelancing photographers we consulted for giving us their attention and time.

We also extend our thanks and appreciation to our colleagues who assisted us in developing the project and people who have willingly helped us out with their abilities.

LIST OF TABLES

LIST OF FIGURES

Chapter 1
INTRODUCTION

Image processing is an area of study which deals with the manipulation of digital images through a computer in order to obtain an enhanced image or to extract some information of significance from it. It comes under the domain of signal processing. The input signal is essentially an image and the output yielded may be another image or certain noteworthy attributes associated with it.
In the present day, image processing is among the most rapidly growing technologies. It is a primary research area in the field of computer science and engineering.

Mathematically, an image can be viewed as a two-dimensional signal. It can be represented by the function f(x, y). The value of f(x, y) at any point gives the value of the pixel at that point of an image. This allows us the means to handle images in a more skillful way and transform them according to the requirement, instead of looking at them as just a collection of pixels which make sense only to the human eye and not to the computer.

1.1 OVERVIEW

Due to the inclusion of image capturing devices into portable devices like smart phones and digital cameras, the number of pictures generated has increased in recent times. Amateur photographers capture numerous pictures when they find an opportune moment, but they are often hampered by unwanted obstructions like fences, grills, enclosures or reflective surfaces. These structures distract the viewers from the actual focus of the picture, thereby ruining its visual appeal. Even if there are editing mechanisms available for photographers to remove obstructions from the picture, the removal of fence-like structures from the picture is particularly tedious as they generally cover the entire picture and it is difficult to remove them without losing some parts of the original picture. Instead of undergoing the harrowing process of manually editing such occlusions, a solution can be offered using image processing algorithms, where the fence is automatically detected and filled without or with minimal manual assistance.

1.2 MOTIVATION

Fences are one of the primary means of protection against intruders. They are used almost everywhere, from the protection of private property to public places like zoos, national heritage sites, scenic meadows and so on. While they are used as a layer of security, they also have drawbacks when it comes to photography since fences and enclosures cannot be removed from the frame by changing the angle of the camera or the plane of focus. While safety and security are the primary concerns, the presence of these obstructions ruins the aesthetic experience of the picture. This problem can be tackled by using image processing to remove the fence from the photograph after it has been clicked. By using accurate de-fencing and image inpainting techniques, the occlusions can be effectively removed and filled with content that fits the original image in a natural way, thereby removing the need to compromise with distorted photographs or performing manual editing on them.

1.3 OBJECTIVE

The focus of our project is to remove fence-like occlusions from the input image to generate a complete image without distortions. By using the frequency spectrum of the image, the fence in the image is segregated from the background. The Morphological close operation is used to convert the pixelated fence mask to a fence mask which is acceptable by the inpainting algorithm. The resultant de-fenced image will contain a mask of empty space that was previously occupied by the fence. It is then given as input to an exemplar-based image in-painting algorithm which uses the exemplar-based completion as the filling mechanism. This results in an occlusion-free image, in which the areas occupied by the fence are filled by the in-painting algorithm by using other parts of the image as a reference.

1.4 SCOPE

The algorithm will take the input as an image which requires occlusion removal. The image will undergo de-fencing and produce an image that will contain the area to be in-painted as a mask, which is the input to the in-painting algorithm. The resulting image will be an occlusion-free image.

1.5 EXISTING SYSTEMS

Defencing:

1. Video De-fencing tool
2. Texture Detection in Images to identify Fence-like Structures
3. Detection and Restoration of Image from Multi-Color Fence Occlusions

Inpainting:

1. Exemplar based image inpainting
2. Deep Convolutional Generative Adversarial Net (DCGAN) based inpainting

1.6 PROPOSED SYSTEM

The system essentially performs its task in 2 phases: first, it detects the fence and uses image processing techniques to produce an intermediate result without the fence masked and highlighted. In the next phase, this mask is filled with pixels to form a coherent image by extrapolating from the known ones (inpainting) by making use of an improved version of the foundational Criminisi algorithm.

Chapter 2
LITERATURE SURVEY

1. Exemplar based image inpainting (Criminisi's Method):

The method proposed in [1] was one of the first methods to suggest the use of example patch information to fill missing patches in an image. It separates the image into target region; the region being filled in, and source region; the region with known pixel information. The target region is divided into patches of size 9 by 9 pixels. But the user should change this to a size slightly bigger than the biggest texture element or a Texel. The algorithm consists of two parts: deciding the filling order of the patches and filling the patches with appropriate colour and texture. To decide the filling order of the patches, a priority term is assigned to each patch. This priority is given by a "confidence term" and by a "data term". The patch P with the highest priority is selected to be filled in. To fill this selected patch with highest priority, a patch from the source region Q is selected that is lost similar to the given patch. This is done by calculating sum of squared distances between Q and the known pixels in P. After selecting appropriate Q, the pixel information from Q is copied onto P. The confidence terms of the patches are updated and this process is repeated until all the patches are filled. The proposed method outperformed every other existing method at that time and was able to remove large objects from images. Many improvements on the original algorithm have been proposed since such as using an image segmentation algorithm to segment the image based on topography before applying the algorithm used in [2], using a patch shifting scheme for cases where Criminisi's method might fail as used in [3], understanding the depth information to guide appropriate scale transformation as used in [4], using pixel inhomogeneity factor to drive the priority function as used in [5] and many more.

2. Deep Convolutional Generative Adversarial Net (DCGAN) based inpainting:

Content aware inpainting algorithms use the neighbouring pixel information to fill in target pixels. However, in some cases there is a need for the algorithm to have some intuitive knowledge as to what to fill as described in [6]. Supervised learning is used to solve that problem. DCGAN is essentially a convolutional neural network that is used to generate new content. It understands the semantics of the entire image and suggests new relevant content to complete the image. DCGANs consist of two parts, the discriminator and the generator. The discriminator comprises four convolution two dimensional layers activated by Rectified Linear Unit and lastly a fully connected layer. The generator comprises of a linear layer along with a reshape transformation so that the input is in appropriate dimensions followed by four convolution two dimensional transpose layers activated by Rectified Linear Units. The generator creates new content to fill into the target region whereas the discriminator distinguishes between what is real and fake. It is observed that using DCGAN produces superior inpainting although it takes longer time to run. However, it does not apply to every scenario since its applicability is limited by the training dataset. [7] uses Convolutional neural networks to inpaint parts of the human body that is obstructed by occlusions.

3. Video De-fencing:

The technique described in [8] involves automatic detection and removal of occlusions from video clips. This method takes advantage of the fact that consecutive frames aligned frame by frame has information of the pixels in the de-fenced video. Therefore, the fence-free video can be obtained by substituting the fence pixels to the pixel information in the frames with an unobstructed view. The algorithm can be divided into two parts: estimation of the term “Probability of Fence” (PoF) and pixel selection. The goal of PoF estimation is to find the confidence term of every fence pixel of every frame. This is done by calculating visual parallax by analyzing and inferring the visual flow and image appearance. After finding the affected pixels, pixel restoration is performed. This is done with an improved image alignment algorithm. This algorithm computes the optical flow by making use of a “robust temporal median filter” (R-TMF) which is able to give the correct pixel information even when the fence pixels dominate the pixel collection. This part of the algorithm determines its complexity since it is the most time consuming. The method proposed gave promising results on a dataset of actual world consumer videos with static scenes. However this method was unable to properly restore the video when the depth of the fence was the same as the background or with moving objects in the background. Methods described in [9], [10], [11] try to improve on this algorithm.

4. Texture Detection to identify fences:

The method described in [12] aims to automatically detect fences or fence-like objects present in the foreground of an image. An application of this is removal of occluding cages or wire mesh in pictures of animals in enclosures. The process is achieved in three stages: Frequency domain filtering, Multi-resolution Processing and SVM Classification. In frequency domain filtering, the image is considered as a two dimensional discrete time signal. The quasi periodic signal here is the fence which can be filtered using a bandpass filter in the frequency domain. For Multiresolution Processing, wavelet transformation is used. It uses a coarser to finer strategy where the fence masks at different levels of wavelet pyramid are combined. The detected fence mask, after multi-resolution processing, classifies a large number of pixels not picked up on the fence texture. Hence, some samples from the fence mask were picked and the features of those sample pixels were used to train an SVM classifier so that the fence texture can be segmented. The fence detection is complete after this stage. Exemplar based image inpainting technique is then used to fill the fence region in this approach. The proposed method works well for fence texture with different shapes, sizes, colours and orientations. Fence texture detection was successful not only for images having fence in the foreground but also for images having fence in the background.

5. A deep learning approach for video de-fencing:

The aim of the proposed method described in [13] is to formulate a sparsity based optimization framework to fill-in fence pixels in a video by using a convolutional neural network. The neural network consists of five layers, two convolutional and two pooling layers and finally a fully connected output layer. The output maps from the fourth layer are concatenated to form a vector while training and this is given to the next layer as input. The final output layer contains two neurons which correspond to each class. These are fully connected with the previous layer by weights. A linear sigmoid function is used to modulate the responses of the output layer. This output produces a resultant score for each class. Like in traditional video defencing, it is assumed that the areas of the frame that are covered by one frame might be revealed in the subsequent frames. This motion is estimated using the optical flow and wrapping matrix. Split Bregman Iterative framework is used to obtain the best estimate of the de-fenced image. The proposed algorithm is able to inpaint video-based frames with superior accuracy when compared to traditional video inpainting technique.

6. Detection and Restoration of Image from Multi-Color Fence Occlusions:

The method described in [28] aims to detect fences having different orientation, shapes, color and texture. The image is first converted from RGB to YCbCr. Histograms are computed for each channel of g(x, y) image. These histograms are analyzed to select threshold values based on which the segmentation of the fence is done. The segmented mask is amended by using morphological operations like elimination of false positives and insertion of false negatives. The fence mask obtained after an amendment is used for masking the original image. A hybrid inpainting algorithm is used to restore the occluded area. This algorithm is able to detect the fence in the image which may be of different orientation, texture, shape, multicolor and occluded. It is reliable with an average true positive rate of more than 95% and true negative rate of more than 97% of all the figures presented.

Chapter 3
REQUIREMENT ANALYSIS AND SPECIFICATION

3.1 FUNCTIONAL REQUIREMENTS

Given a picture with occluding elements, the system should effectively remove and inpaint the occlusions to produce a picture without distortions. (There should be enough coherence in the actual background to extrapolate and produce a complete image without occlusions. This depends on the discretion of the user.)

3.2 NON-FUNCTIONAL REQUIREMENTS

We aim at attaining the following attributes of quality in our project:

1. **Usability:** We want the project to be accessible to amateur photographers who do not have experience with complex editing tools. Hence the system needs to be user-friendly and easily comprehensible.

2. **Reliability:** Consistency in the system performance is very necessary and it should give accurate results for every input that falls within the scope of the project.

3. **Performance:** The system should perform as expected in an environment that meets the recommended specifications.

4. **Scalability:** The system design should be such that it can be extended to process batches of input images without reduction in performance

5. **Maintainability:** The code should be understandable and modularized so that making further improvements is not cumbersome.

3.3 HARDWARE REQUIREMENTS

Processor	Intel i5 / AMD A6 or greater
Graphics Card	NVIDIA GTX / AMD Radeon RX or greater
Memory (RAM)	Minimum 16GB
GPU	4GB Nvidia
System specification	x64-based processor, 64-bit Operating System

Table 3.3: Hardware Requirements

3.4 SOFTWARE REQUIREMENTS

Programming Language used:

- Python

Libraries Required:

- OpenCV
- Matplotlib
- NumPy

3.5 COST ESTIMATION

Our project does not incur any costs.

Chapter 4
DESIGN

4.1 HIGH-LEVEL DESIGN

4.1.1 SYSTEM ARCHITECTURE

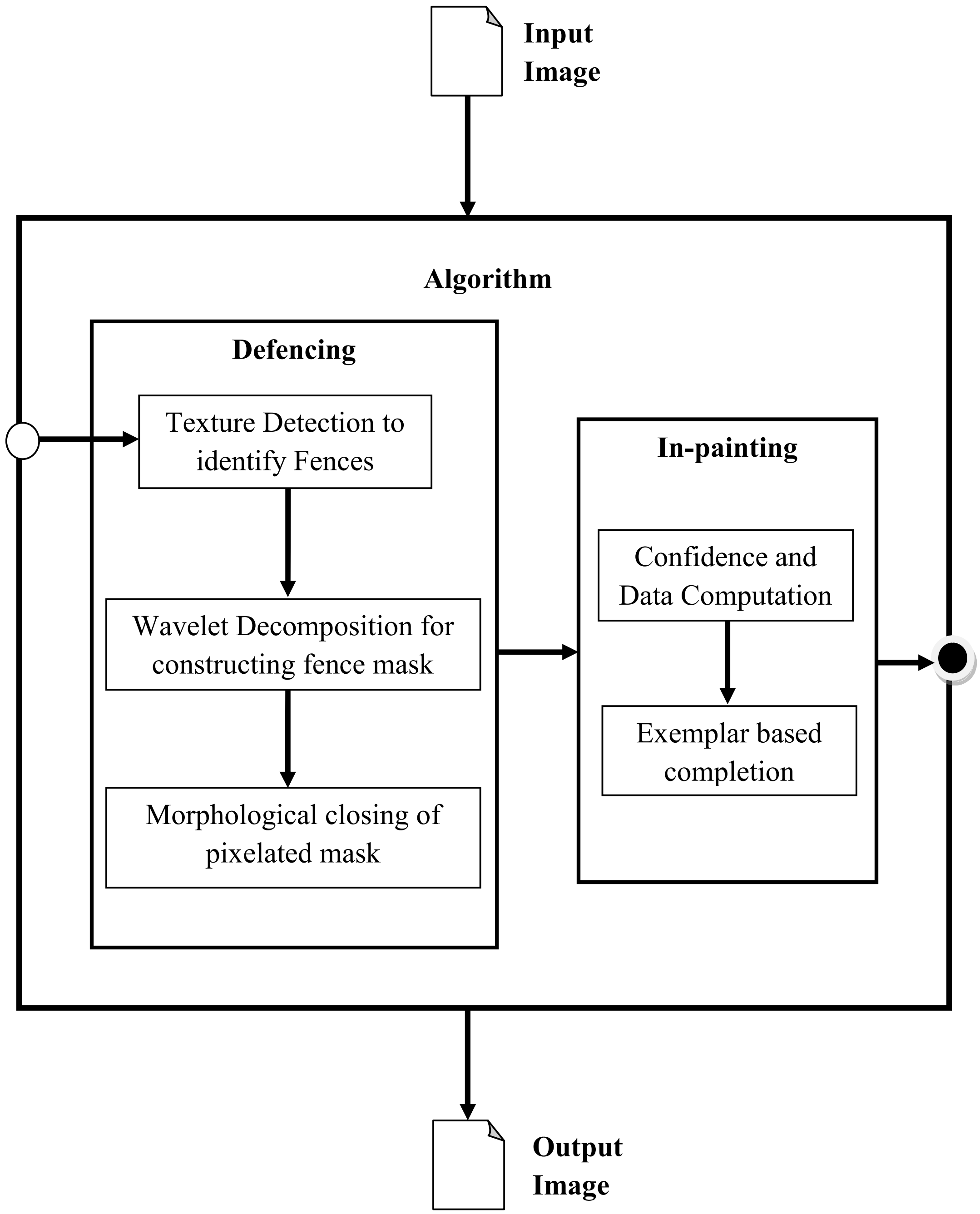


Figure 4.1.1: System Architecture

4.2 DETAILED DESIGN

4.2.1 SYSTEM ARCHITECTURE

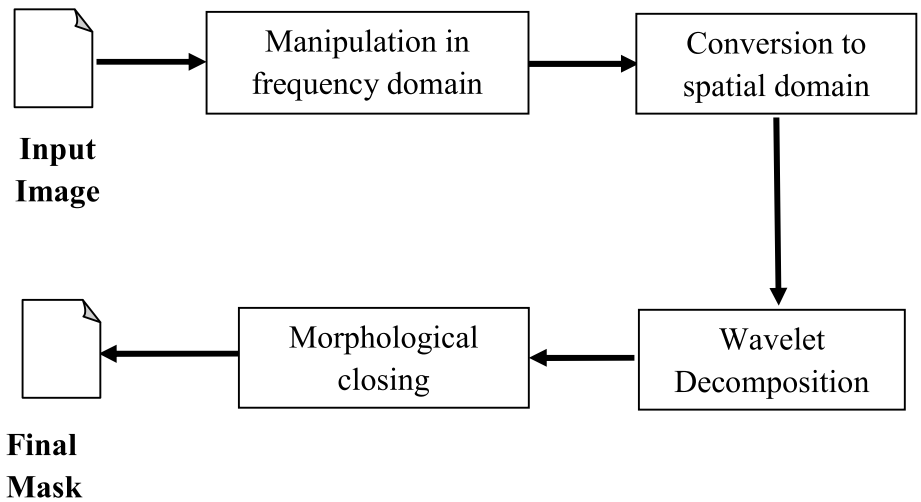


Figure 4.2.1.1: Detailed System Architecture - Defencing

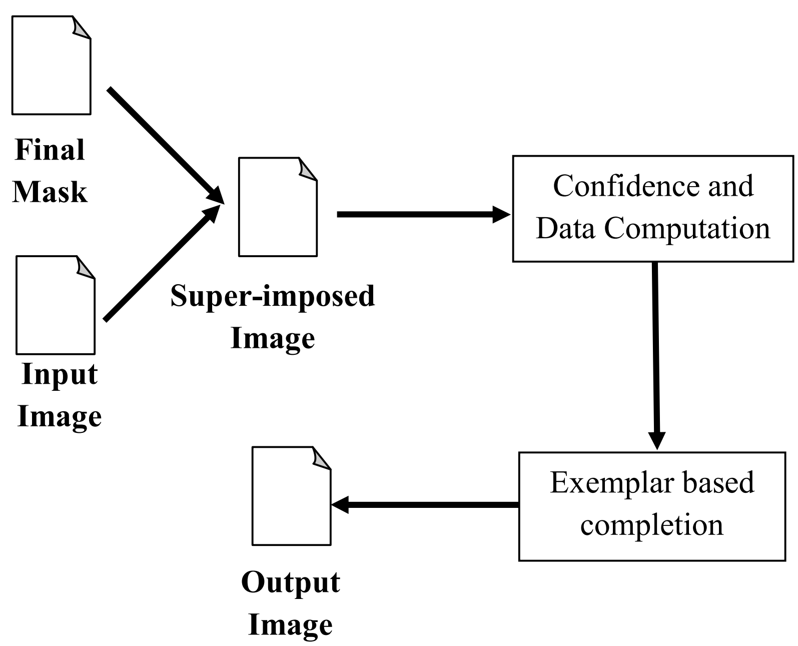


Figure 4.2.1.2: Detailed System Architecture - Inpainting

Chapter 5
IMPLEMENTATION

5.1 OVERVIEW OF TECHNOLOGIES USED

This project uses advanced image processing techniques which involve complex mathematics. To implement these techniques, we have made use of python's vast reserve of open source libraries. The most prominent libraries used are as follows:

- **OpenCV** (Open Source Computer Vision Library) is a widely used library for computer vision and image processing. Originally built with C++, it has grown to include Java, Python and MATLAB interfaces. It has more than 2500 optimized algorithms for diverse applications. We have employed OpenCV for most of the image processing functions.
- **Matplotlib** is a toolkit developed for python and NumPy to provide MATLAB-like interface for data visualization. It is a plotting library which includes functions to generate bar charts, scatter plots, histograms, power spectra etc. We have used this library to visualize the results of our algorithm.
- **PyWavelets** is a library that provides wavelet transformation functions for python. It provides a high-level interface for low level C functions for optimized performance. It includes functions for n-Dimension Forward and Inverse Discrete Wavelet Transform, Undecimated Wavelet Transform, Wavelet Packet decomposition and reconstruction, etc. We have used PyWavelets to perform wavelet decomposition which is an integral task for image defencing.
- **NumPy** is a package developed for python that provides functions and data structures that support scientific computing. Mostly used for its support for n-dimensional arrays and matrices. It additionally provides high-level mathematical functions to that process these data-structures for various applications. We use NumPy to store and perform operations on images in an optimized manner.
- **Skimage** or Scikit-image is a tool for image processing in python which works with NumPy arrays. Its core algorithms have been written in C for better performance. It provides functions which perform image segmentation, color space manipulations, morphology, feature detection, and more. We use this library to perform morphological operations for image defencing.

5.2 IMPLEMENTATION DETAILS OF MODULES

I. Defencing Module

Image Defencing involves segregating the fence area of an image by using the mathematical properties of images and recognizing the quasi-periodic texture of a fence.
The algorithm takes the image to be defenced as input and produces a binary (black and white) fence mask as the output.

a) Manipulation in frequency domain

The input image is converted from the spatial domain (human understandable format) to the frequency or Fourier domain. This conversion is done by applying the Fourier Transform and helps us better visualize the rate at which the pixel values are changing in spatial domain.

For an image of size MxN pixels

2-D DFT

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)}$$

u = frequency in x direction, $u = 0, \ldots, M\text{-}1$
v = frequency in y direction, $v = 0, \ldots, N\text{-}1$

2-D IDFT

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi(ux/M + vy/N)}$$

$x = 0, \ldots, M\text{-}1$
$y = 0, \ldots, N\text{-}1$

Figure 5.2.1.1: Frequency Domain Transformation

In the frequency domain, the value and location are represented by sinusoidal relationships that depend upon the frequency of a pixel occurring within the image. The pixel location is represented by its x- and y-frequencies and its value is represented by the amplitude. This transform helps us determine which pixels contain more important information and whether repeating patterns occur.

An FFT Shift is then executed in order to enable ease of understanding when we look at the frequency domain image. It rearranges the Fourier transformed image by shifting the zero-frequency component to the center of the array.

b) Quasi-periodic texture filtering

Frequency component of an image can be divided into two major components:

- High frequency components correspond to edges in an image
- Low frequency components in an image correspond to smooth regions

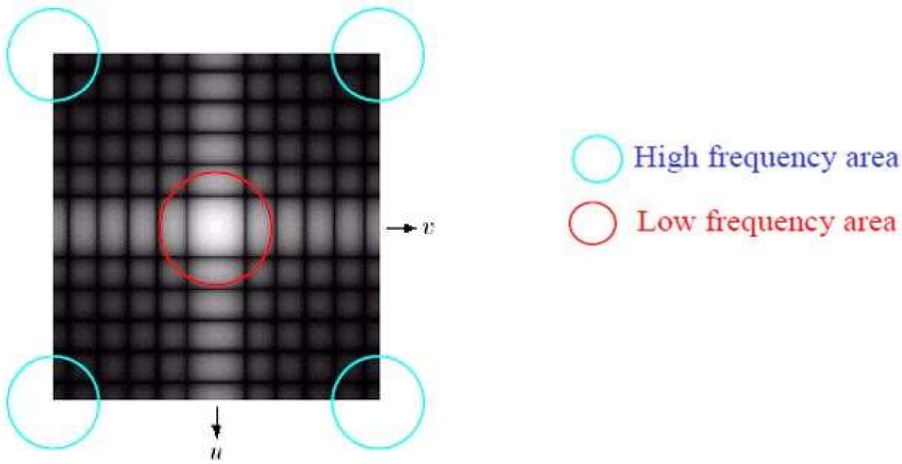


Figure 5.2.1.2: Quasi-periodic Texture Filtering

The fences which occur in the image can be categorized as edges which are to be segregated. This is done using a high pass filter, which allows only high frequencies to go through and blocks low frequencies. It can be used to sharpen an image and isolate edges.

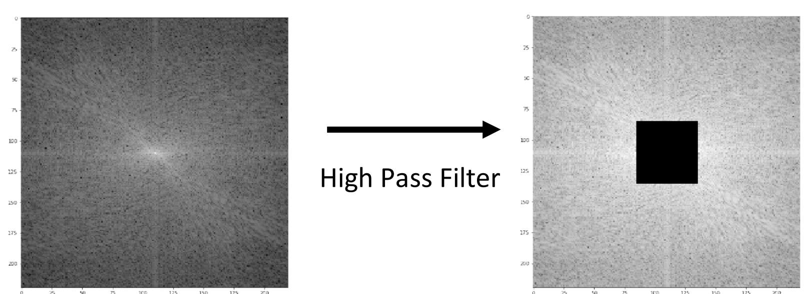


Figure 5.2.1.3: High Pass Filter

Applying a band pass filter helps reduce noise in the image. The actual changes, however, can be viewed after the image is inverse-transformed back to the spatial domain.

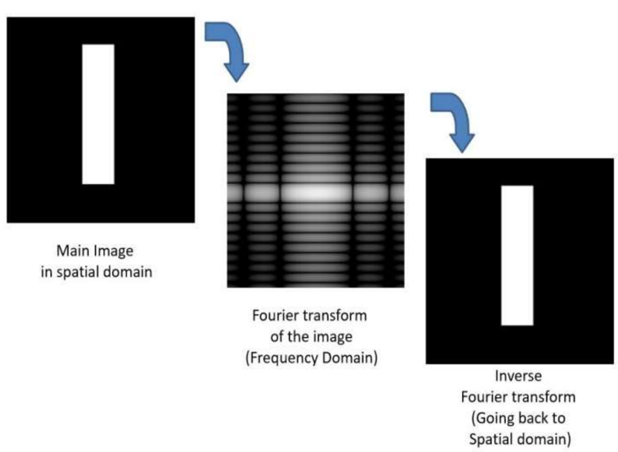


Figure 5.2.1.4: Applying Band-pass Filter

c) Wavelet Decomposition – Multiresolution Processing

A multi-resolution representation provides a simple hierarchical framework to analyze the signal at different resolution levels. Any image can be represented as a multi-resolution pyramid. At each level, the difference (residual) between the image at that level and the predicted image from the next level is stored. We can reconstruct the image by just adding up all the residuals.

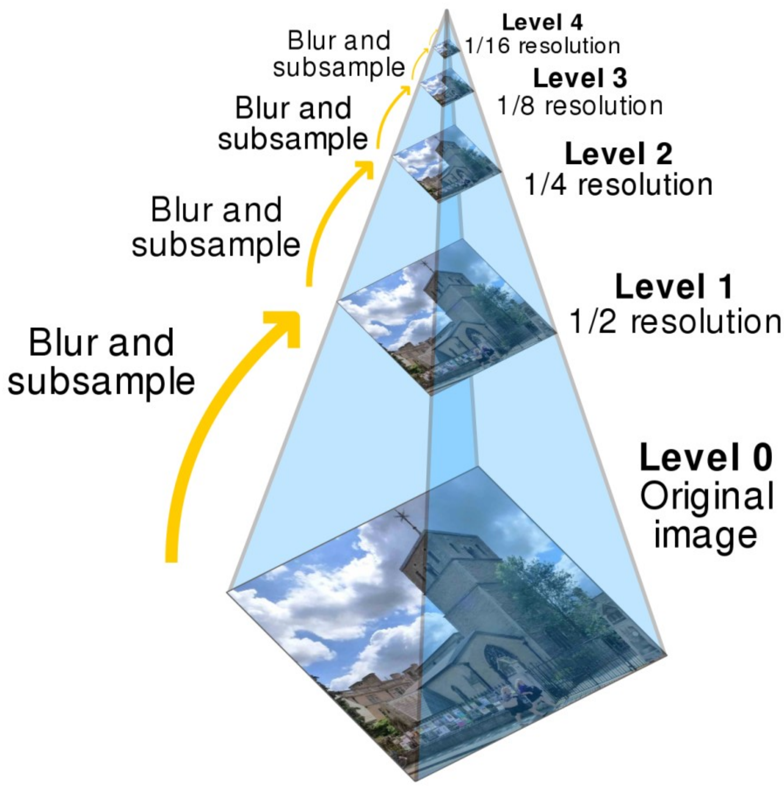


Figure 5.2.1.5: Wavelet Decomposition

Wavelets are a more general way to represent and analyze multi-resolution images.
They represent the scale of features in an image, as well as their position. Wavelet decomposition of an image produces diagonal (*D*), vertical (*V*) and horizontal (*H*) components, which are employed for feature extraction.
Orthogonality represents a unique and complete representation of a signal. Each stage of the algorithm generates wavelets with sequentially finer representation of signal content.

We leverage these components to filter out the fence-like components of the image more effectively. At lower resolutions, it becomes easier to demarcate the fence pixels from the non-fence pixels, which is achieved by thresholding the pixel value of the directional components of the grayscale image we have.

d) Morphological closing

Morphology is a broad set of image processing operations that process images based on shapes. In a morphological operation, each pixel in the image is adjusted based on the value of other pixels in its neighborhood. By choosing the size and shape of the neighborhood, we can construct a morphological operation that is sensitive to specific shapes in the input image.

It needs two inputs, one is the original image and the second one is called structuring element or kernel – which gives the shape of the closing element to be used.

Figure 5.2.1.6: Morphological Closing

We use the Morphological close operation to convert the pixelated fence mask to a fence mask which is acceptable by the inpainting algorithm.

Closing is similar in some ways to dilation but it is less destructive of the original boundary shape. As with other morphological operators, the exact operation is determined by a structuring element. The effect of the operator is to preserve *background* regions that have a similar shape to this structuring element, or that can completely contain the structuring element, while eliminating all other regions of background pixels.

Closing is Dilation followed by Erosion. It is useful in closing small holes inside the foreground objects, or small black points on the object.

II. Inpainting Module

Image Inpainting is a technique of filling in missing or unwanted areas of an image by using the knowledge of the existing or wanted pixels in the image. Exemplar-based algorithms are widely used as an inpainting technique, where the main idea is to find suitable patches in the source region that can be used to fill the missing regions. This is done in an iteration that involves two major steps, choosing the missing region to fill and selecting an exemplar region to fill it with.

The algorithm takes the image to be inpainted and the corresponding mask, with the mask pixels marked as white and the background pixels marked as black, as input.

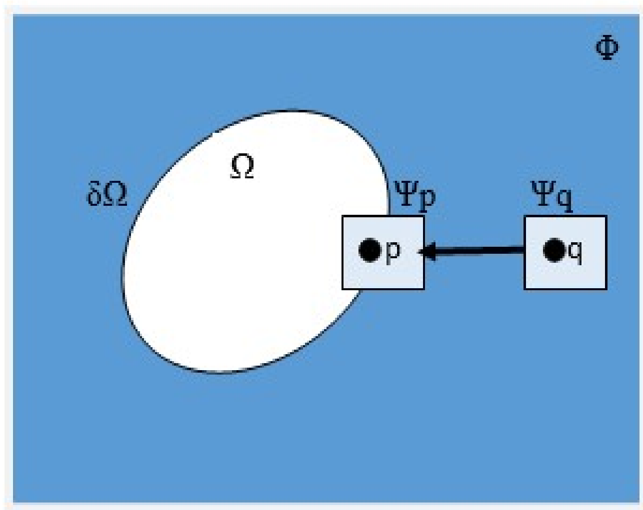


Figure 5.2.2.1: Inpainting using Exemplar based Algorithm

Consider the image to be inpainted as Φ and the working mask as Ω. The border of the working mask is known as the fill front, $\delta\Omega$.

a) Priority Computation

The priority of each pixel in the mask region decides the filling order. It is calculated as the product of the *data* and *confidence* values of the patch surrounding the pixel. For each pixel *p* of the image, the confidence and data terms are calculated as follows:

Initialization:

Confidence (p) =0, if pixel *p* belongs to the mask

Confidence (p) =1, if pixel *p* does not belong to the mask

Data [] = [0]

Calculation:

$$Confidence\ (p) = \frac{\sum Confidence(q)}{Area(\Psi p)} \quad , \forall\ q \in (\Psi p \cap (\Omega)')$$

*Data (p) = unit normal (p) * max_gradient (p)*

The maximum gradient is a term used to denote the relative difference that a pixel has with respect to its surroundings. For every pixel that belongs to the fill front, the gradient is calculated for a patch of a pre-determined size that surrounds it (preferably a size of 9X9). The pixel is assigned a gradient value that is equivalent to the maximum gradient that is exhibited by the patch around it.

The unit normal term is obtained by performing a convolve operation on the working mask using the Sobel Edge operator which removes the noise and provides a smoothing effect on the mask by clearly highlighting the edge of the mask.

The Confidence and Data terms hence calculated are then used to compute the priorities of all the pixels along the fill front to determine the filling order.

*Priority (p) = Confidence (p) * Data (p),* $\forall\ q \in \delta\Omega$

b) Exemplar based completion

Once priorities of the patch are computed, and the maximum priority patch is chosen as the patch to be filled in the current iteration. The exemplar portion to fill it is selected based on the minimum sum of squared differences between the target patch Ψp and all the patches that are a part of the background region. The patch with the minimum difference is chosen as the exemplar patch and is used the fill the missing part of the target patch. At the end of the iteration, the fill front, confidence terms and data terms of the image are updated. This cycle continues till the image is complete and the fence mask has been entirely covered.

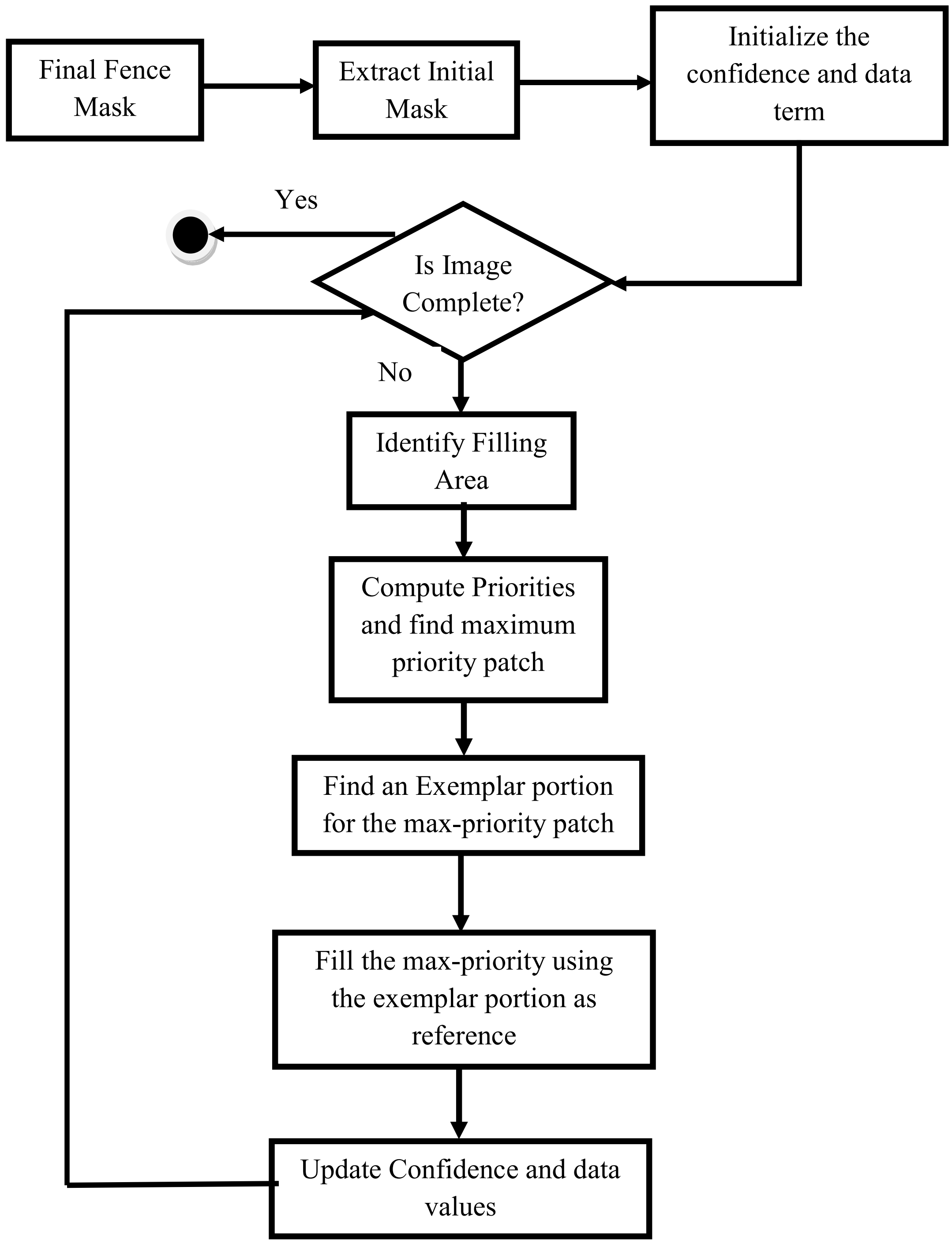


Figure 5.2.2.2: Exemplar Based Inpainting Algorithm

Pseudo Code:

Defencing:

1. Perform a Fast Fourier Transform on the image containing the fence
2. Shift the origin of the frequency representation
3. Apply high pass filter after selecting an appropriate threshold
4. Apply band pass filter with appropriate threshold
5. Perform inverse Fourier Transform on the image
6. Decompose the image into 'H', 'V' and 'D' components
7. Apply a threshold and perform on these components
8. Choose an appropriate thickness and perform morphological close operation to get a coherent fence mask

Inpainting:

Initialize the Confidence and Data terms of all the pixels.

Repeat until done

1. Identify the fill front. If fill front does not exist, exit.
2. Compute priorities *P (p)*, where *p* is the pixel belonging to the fill front.

$$P(p) = Confidence(p) * Data(p)$$

3. Find the pixel *p* with maximum priority and derive the patch Ψp around it.
4. Find the exemplar patch Ψq which has the minimum difference with target patch Ψp.
5. Copy image data from Ψq to Ψp.
6. Update the Confidence and Data of the newly filled patch Ψp.

5.3 DIFFICULTIES ENCOUNTERED AND STRATEGIES USED TO TACKLE

This project has been a demanding but exciting learning experience. Following are some of the main difficulties we encountered during the process:

1. **Unfamiliarity with the chosen technical domain and use of complex mathematical concepts:**

Image processing was an entirely novel field for all members. After selecting the topic, we were briefly overwhelmed by the tsunami of information that came along with this field and the complex high-level mathematics which accompanied it. However, we exposed ourselves to this knowledge in a systematic way with our guide helping us every step of the way.

2. **Choice of platform and algorithm:**

During implementation, we were faced with a very critical question – which language and which platform should be chosen. At first, we chose MATLAB as a suitable platform as it is a favourite of image processing experts. However, we later decided to go with Python, so as to enable efficient development which was at the correct level of abstraction.
Doing a comprehensive literature survey and our guide's help enabled us to form our own creative ideas about how to modify the algorithm and make the existing systems better.

3. **Maintaining simple and understandable code despite complexity of calculations:**

Given the complex intricacies of the problem at hand, it was a difficult task to maintain the code at a level which could be easily understood by all developers. We made continuous efforts to keep the code modularized and inserted regular comments to make sure we understand each other's work.

4. **Navigating roadblocks in implementation**

Translation of mathematical concepts into effective code proved to be a time-consuming and arduous task.

5. **High number of variable factors like dependency on illumination level**

This was a major problem in the initial stage of development and has been duly tackled after research. The algorithm was tweaked accordingly to accommodate such situations.

Chapter 6
TESTING, EXPERIMENTAL ANALYSIS AND RESULTS

6.1 UNIT TESTING

I. Defencing Module

Input: Image from which mask is to be identified

Output: Binary (black and white) fence mask corresponding to the input image

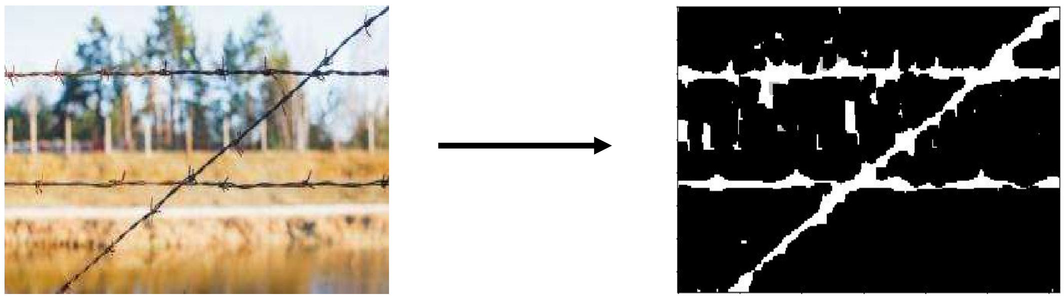


Figure 6.1.1: Result of Defencing

Result: Works as expected for all resolutions and sizes of the image given as input

II. Inpainting Module

Input: Image from which fence has to be removed and a binary (black and white) fence mask corresponding to the input image (made manually using marking tool)

Output: Inpainted image (should not contain the fence element)

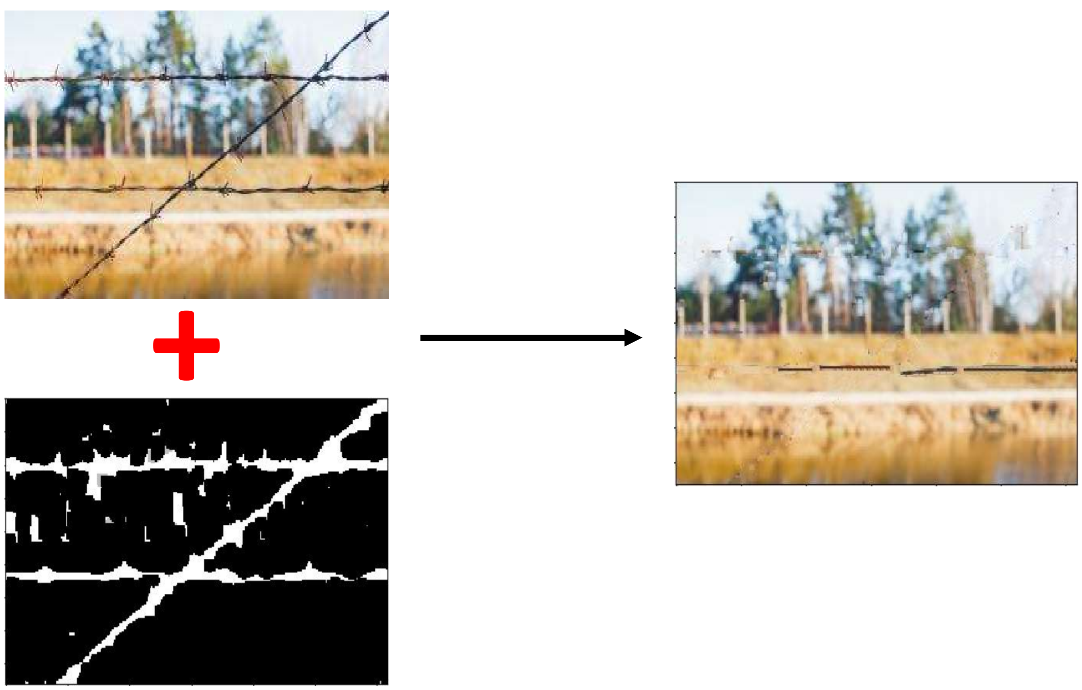


Figure 6.1.2: Result of Inpainting

Result: Provides satisfactory result when a good quality mask is provided

6.2 INTEGRATION TESTING

The main points of integration in this project were as follows:

1. Integrating the defencing and inpainting modules

The main point of notice here was that the dimensions and resolution of the input image and the mask should be exactly the same. This was handled from the design stage itself to ensure no further problems. After successful establishment of the flow, the project worked as expected.

2. Integrating the minimalistic UI with the rest of the project

This was a trivial task which did not change the behaviour of any module.

6.3 EVALUATION METRICS

Assessing the performance of any algorithm in image processing is difficult because it depends on several factors:

1. the algorithm itself
2. the nature of images used
3. the algorithm parameters used
4. the method used for evaluation

Evaluation with a set of “easy” images may produce a higher accuracy than the use of more difficult images containing complex regions. This relates to the diversity and complexity of the selected images and the significance of the images.

Typical performance indicators include:

- Accuracy
- Robustness
- Sensitivity
- Adaptability
- Reliability
- Efficiency

6.4 EXPERIMENTAL DATASET

Pictures of different sizes and resolutions were used in our dataset. We also made sure that the fenced area in the picture was of different kinds – regular, irregular, broken, etc.

Some of the pictures from our dataset are given below:

Figure 6.4.1:
Medium-high resolution
Broken fence

Figure 6.4.2:
High resolution
Regular fence

Figure 6.4.3:
Low resolution
Regular fence

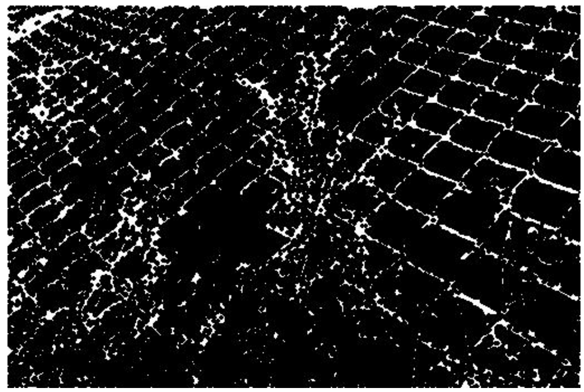
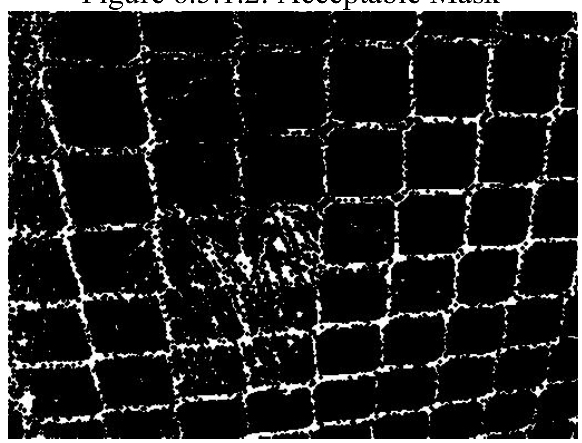
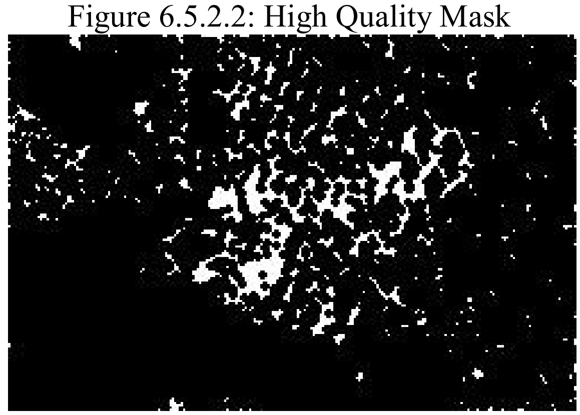
Figure 6.4.4:
Medium-low resolution
Irregular fence

6.5 PERFORMANCE ANALYSIS

As discussed before, we have analyzed the performance on the basis of the following factors:

- the algorithm parameters used
- the nature of images used

With different images given as input, we get different kinds of results, depending on the resolution, type of fence, orientation, clarity, foreground and background features, etc.:

Original image	**Fence Mask**
	
Figure 6.5.1.1: Background fence	Figure 6.5.1.2: Acceptable Mask
	
Figure 6.5.2.1: Foreground fence	Figure 6.5.2.2: High Quality Mask
	
Figure 6.5.3.1: Oblique Orientation	Figure 6.5.3.2: Low Quality Mask

Chapter 7
CONCLUSION AND FUTURE ENHANCEMENTS

5.1 CONCLUSION

In this project, we have demonstrated how to remove occluding elements from images using advanced image processing techniques which involve complex mathematics. It is remarkable to see the impact of pure math-based computer applications in such varied real world problems. This project tackles a task which is quite cumbersome for photographers when done manually. We hope this tool is incorporated into the mainstream set of tools that are used professionally and that image processing experts contribute to make it even more effective.

5.2 FUTURE ENHANCEMENTS

Currently, the algorithm gives accurate results for high resolution images. This work can be improved to provide better accuracy for low resolution images as well. There is also a scope for improvement in the efficiency of the inpainting algorithm. Reduction in the number of variable factors involved in producing the fence mask will also increase the efficiency of the algorithm. Removing the dependency on controlled illumination level is also a suggested improvement. This work can be extended to include random structures and reflective occlusion (distortions created due to reflective surfaces present in the frame). It can also be enhanced to perform de-fencing and in-painting for videos.

REFERENCES

[1] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpaint-ing," IEEE Trans. Image Process., vol. 13, no. 9, pp. 1200–1212, 2004.

[2] Huang Ying, Li Kai a*, Yang Ming "An Improved Image Inpainting Algorithm based on Image Segmentation" International Congress of Information and Communication Technology 2017 a Institute of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China 400065.

[3] S. Tae-o-sot and A. Nishihara, "Exemplar-based image inpainting with patch shifting scheme," 2011 17th International Conference on Digital Signal Processing (DSP), Corfu, 2011, pp. 1-5.

[4] Xiao M, Li G, Xie L, Peng L, Chen Q (2018) Exemplar-based image completion using image depth information. PLOS ONE 13(9): e0200404.

[5] Q. Fan, H. Liu, Z. Fu and X. Li, "Exemplar-based image inpainting based on pixel inhomogeneity factor," 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Kuala Lumpur, 2017, pp. 1164-1168.

[6] Chenduo Huang & Koki Yoshida, (2017). Evaluations of Image Completion Algorithms: Exemplar-Based Inpainting vs. Deep Convolutional GAN. Stanford University.

[7] Xian Wu, Rui-Long Li, Fang-Lue Zhang, Jian-Cheng Liu, Jue Wang, Ariel Shamir, Shi-Min Hu, (2018) "Deep Portrait Image Completion and Extrapolation".

[8] Y. Mu, W. Liu, and S. Yan, "Video de-fencing," IEEE Trans. Circts. Sys. Vid. Tech., vol. 24, no. 7, pp. 1111–1121, 2014.

[9] Jonna, Sankaraganesh & Satapathy, Sukla& Sahay, Rajiv. (2016). Stereo image de-fencing using smartphones.

[10] Jonna, Sankaraganesh&Nakka, Krishna & Sahay, Rajiv. (2016). Towards an Automated Image De-fencing Algorithm Using Sparsity.

[11] Xue, Tianfan& Rubinstein, Michael & Liu, Ce & Freeman, William. (2015). A Computational Approach for Obstruction-Free Photography. ACM Transactions on Graphics. 34. 79:1-79:11. 10.1145/2766940.

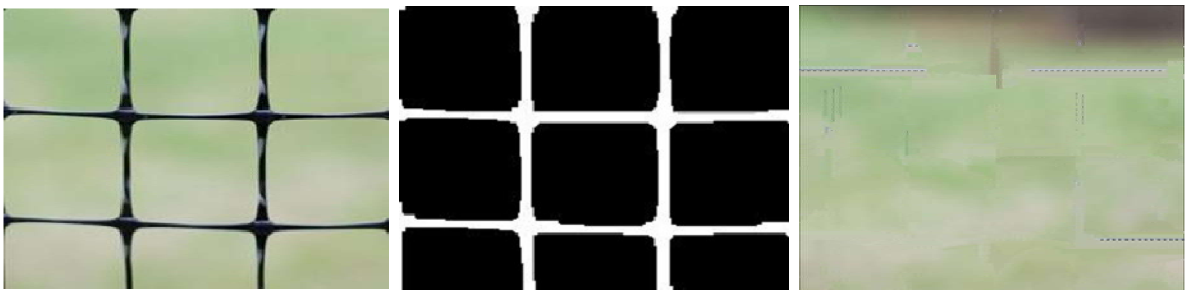
[12] Hettiarachchi, Randima& Peters, James & Bruce, Niel. (2014). Fence-like Quasi-periodic Texture Detection in Images. Theory and Applications of Mathematics & Computer Science.

[13] S. Jonna, K. K. Nakka and R. R. Sahay, "My camera can see through fences: A deep learning approach for image de-fencing," 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), Kuala Lumpur, 2015, pp. 261-265.

[14] Varalakshmamma, M. &Venkateswarlu, T. (2019). Detection and Restoration of Image from Multi-Color Fence Occlusions. Pattern Recognition and Image Analysis. 29. 546-558. 10.1134/S1054661819030209.

[15] Patel, Pritika& Prajapati, Ankit & Mishra, Shailendra. (2012). Review of Different Inpainting Algorithms. International Journal of Computer Applications. 59. 30-34. 10.5120/9650-4411.
[16] Vreja, Raluca & Brad, Remus. (2014). Image Inpainting Methods Evaluation and Improvement. The Scientific World Journal. 2014. 937845. 10.1155/2014/937845.
[17] Marcelo Bertalmio, Guillermo Sapiro, Vicent Caselles, Coloma Balleste (1999) "Image Inpainting", University of Minnesota.
[18] L. Liu and H. Li, "Image Completion and Its Application in Cultural Relic Preservation," 2009 2nd International Congress on Image and Signal Processing, Tianjin, 2009, pp. 1-5.
[19] Zarif, Sameh& Faye, Ibrahima& Awang Rambli, Dayang. (2014). Image Completion: Survey and Comparative Study. International Journal of Pattern Recognition and Artificial Intelligence. 29. 1554001. 10.1142/S0218001415540014.
[20] V.V. Voronin, V.I. Marchuk, E.A. Semenishchev, S. Makov, R. Creutzburg: Digital Inpainting with Applications to Forensic Image and Video Processing, IS&T International Symposium on Electronic Imaging — Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications, San Francisco, CA, USA, 14-18 Feb. 2016.
[21] Jia-Bin Huang, Sing Bing Kang, Narendra Ahuja, Johannes Kopf (2014): Image Completion using Planar Structure Guidance" - ACM Transactions on Graphics (Proceedings of SIGGRAPH 2014) vol 33 num 4.
[22] Q. Zhang, Y. Yuan and X. Lu, "Image de-fencing with hyperspectral camera," 2016 International Conference on Computer, Information and Telecommunication Systems (CITS), Kunming, 2016, pp. 1-5.
[23] S. Jonna, V. S. Voleti, R. R. Sahay and M. S. Kankanhalli, "A multimodal approach for image de-fencing and depth inpainting," 2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR), Kolkata, 2015, pp. 1-6.
[24] C. S. Negi, K. Mandal, R. R. Sahay and M. S. Kankanhalli, "Super-resolution de-fencing: Simultaneous fence removal and high-resolution image recovery using videos," 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), Chengdu, 2014, pp. 1-6.
[25] Farid, Muhammad Shahid & Mahmood, Arif Grangetto, Marco. (2016). Image de-fencing framework with hybrid inpainting algorithm. Signal Image and Video Processing. 10. 1193–1201. 10.1007/s11760-016-0876-7.
[26] LIU, Y., BELKINA, T., HAYS, J., AND LUBLINERMAN, R. 2008.
Image de-fencing. In CVPR.
[27] Li, Yinlong& Wang, Yu & Piao, Yan. (2017). Extraction of thin occlusions from digital images. 1025540. 10.1117/12.2268111.
[28] LUO, Meng-xiao& XU, Wei-sheng & YU, You-ling. (2019). Image De-fencing Based on Binary Morphology. DEStech Transactions on Computer Science and Engineering. 10.12783/dtcse/icaic2019/29429.
[29] Park, Minwoo& Brocklehurst, Kyle & Collins, Robert & Liu, Yanxi. (2010). Image De-fencing Revisited. 6495. 422-434. 10.1007/978-3-642-19282-1_34.

APPENDIX A
Result Snapshots



a) Original image b) Fence Mask c) Inpainted Image

Result Set I

a) Original image b) Fence Mask c) Inpainted Image

Result Set II

APPENDIX B
Plagiarism Report

REMOVAL OF OCCLUDING FENCES FROM AN IMAGE

ORIGINALITY REPORT

16%	9%	7%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Liberty Union High School District Student Paper	3%
2	www.uav.ro Internet Source	2%
3	doaj.org Internet Source	1%
4	jainpoojith.blogspot.com Internet Source	1%
5	www.harrisgeospatial.com Internet Source	1%
6	www.mathworks.com Internet Source	1%
7	Jonna, Sankaraganesh, Krishna K. Nakka, and Rajiv R. Sahay. "My camera can see through fences: A deep learning approach for image de-fencing", 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015. Publication	1%

APPENDIX C

Attainment of POs and PSOs

B.M.S. College of Engineering
Department of Computer Science and Engineering

Batch no.: B33 Date:

Project Title: **Removal of occluding fences from an image**

Programme Outcomes	**Level (1/2/3)**	**Justification (if addressed)**
PO1	2	Applying the knowledge of mathematics and image processing to develop the project to address the current challenges in the domain of image completion.
PO2	2	Analyzing the problem statement, performing research in the project domain and quoting the same in the literature survey.
PO3	2	Designing the solution for the problem statement with ethical and social considerations in mind.
PO4	2	Referring many publications of similar research base and analysis of their results.
PO5	3	Utilizing the latest tools and libraries of image processing to come up with accurate and improved solutions.
PO6	3	Developing solutions in adherence to ethical and moral standards defined in the professional engineering practice.
PO7	2	Producing sustainable results which cause no harm to the user or the environment.
PO8	3	Adhering to the ethics and norms of the engineering practice.
PO9	3	Distributing work among the team members in an equal and efficient manner and conducting periodic review for knowledge sharing
PO10	3	Communicating our research to the society via well-written reports, self-explanatory presentations and thorough documentation
PO11	3	Quoting cost and resource requirements at the beginning of the project and following the same throughout the duration of the project
PO12	3	Recognizing scope for future enhancements to the project and documenting the same for future generations
PSO1	2	Using standard software development practices at every step of the development process
PSO2	3	Producing a user-friendly system that can be used with minimum domain knowledge
PSO3	2	Optimizing codes to the best of our abilities and increasing the efficiency of the design throughout the development process.