

Software Design Document

(SDD)

For

IMAGE SUPER-RESOLUTION FOR DEEP LEARNING

Name of Student(s):

1.AHMAD SIRAJ HASHMI(19C0B022)

2.ALWEERA KHAN(19COB470)

Name of the supervisor (s): Prof. Mohammad Sarosh Umar

Department of Computer Engineering

Aligarh Muslim University

Session 2019 – 23

**TABLE OF CONTENTS**

1. [Introduction 1-2](#_TOC_250014)
   1. [Purpose 1](#_TOC_250013)
   2. [Scope 1](#_TOC_250012)
   3. [Overview 1](#_TOC_250011)
   4. [Definitions and Acronyms 1](#_TOC_250010)
   5. [Reference Material 2](#_TOC_250009)
2. [System Overview 2-3](#_TOC_250008)
3. [System Architecture 3-5](#_TOC_250007)
   1. [Architectural Design 3](#_TOC_250006)
   2. [Decomposition Description 4](#_TOC_250005)
   3. Design Rationale 5
4. [Data Design 6](#_TOC_250004)
   1. [Data Description 6](#_TOC_250003)

# Introduction

## Purpose

This software design document describes the architecture and system design of our final year project ‘Image Super Resolution using Deep Learning’. It provides an insight into the whole system and help in implementing the same. The main purposes of design document are listed below:

* Precise understanding of the requirements and constraints related with the programming language, and User Interface.
* It serves as ‘how to’guide for implementing the system.
* Provide a basic outline of the User Interface of the project.

## Scope

## High-quality images are something we all want without spending a lot of money on a costly system. Our network also struggles with image quality when training. The first step to improving image resolution is to understand what other solutions are available. There are two approaches to increasing the resolution: hardware-based and algorithmic-based approach. Hardware-based approach requires either decreasing the size of pixel or increasing the sensor size. Decreasing the pixel size to less than a threshold value would result in less light on a pixel, which will increase shot noise. Increasing the sensor’s size leads to increase in system capacitance, which results in slow charge transfer rate.

## Overview

The document shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it contains all the information required by a programmer to write code. It gives the preliminary design in which the overall system architecture and data architecture is defined.

## Definitions and Acronyms

* + - * **HTML**: Hypertext Markup Language
      * **CSS**- Cascading Style Sheet
      * **HTTP**: Hypertext transfer protocol
      * **IP**: Internet Protocol
      * **SR:** Super Resolution
      * **HR:** High Resolution
      * **LR:** Low Resolution
      * **TCP**: Transmission control protocol
      * **ML**: Machine Learning
      * **Deep Learning Model**: Deep learning is a type of machine learning and artificial intelligence (AI) that imitates the way humans gain certain types of knowledge. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance.
      * **GAN:** A generative adversarial network (GAN) is a machine learning (ML) model in which two neural networks compete with each other to become more accurate in their prediction

## Reference Material

Software design Document <http://en.wikipedia.org/wiki/Software_design_document> Visited date: 09-11-2022

# System Overview

## In our final year project, we will be implementing the image super resolution model using Generative Adversarial Network as it is the one of the recent advances in the field of Image super resolution, even though Generative Adversarial Network is one of the finest models it still have some problems like the training process of Generative Adversarial Network is still difficult and unstable. Although there have been some studies on how to stabilize the GAN training how to ensure that the GANs integrated into SR models are trained correctly. So, we will try our best to remove these flows and achieve even better results. The technologies used are:

## Machine Learning techniques like Neural Networks, Generative Adversarial Network.

## Python libraries like NumPy, Pandas, Matplotlib, Seaborn for data analysis.

## Sckit learn library for using various ML Algorithms.

## HTML, CSS, JavaScript for frontend of Web Application (if we will integrate it in web application in future. The basic functionalities include:

## 

# System Architecture

## Architectural Design

## We process the HR(High Resolution) images to get down-sampled LR(Low Resolution) images. Now we have both HR and LR images for training data set.

## We pass LR images through Generator which up-samples and gives SR(Super Resolution) images.

## We use a discriminator to distinguish the HR images and back-propagate the GAN loss to train the discriminator and the generator.

## Decomposition Description

**Data Flow Diagrams:**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFD can also be used for the visualization of data processing (structured design).

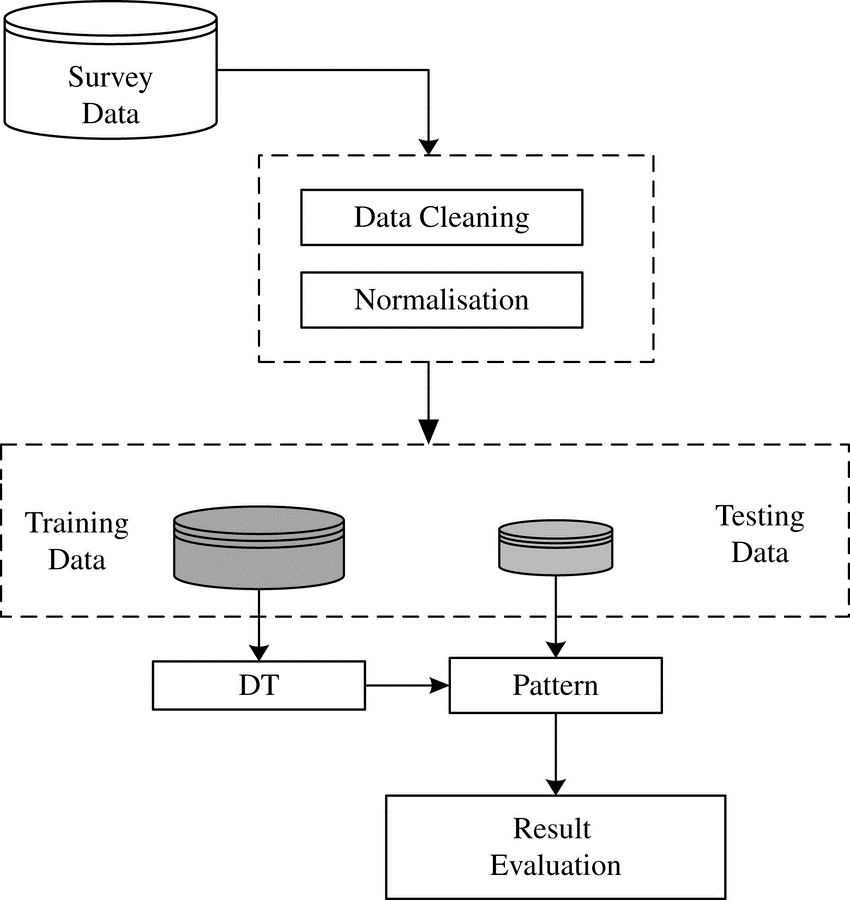


Fig: 4.1 Context Diagram of TRS

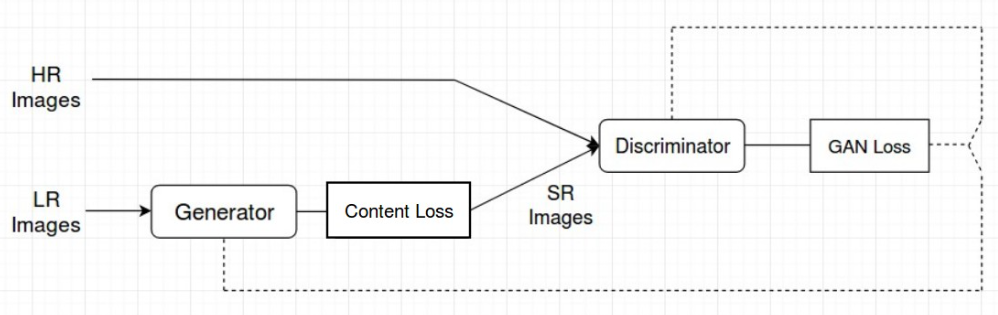


Fig: 4.2 Data Flow Diagram of Image Super Resolution

## Design Rational:

## It is an efficient design. It divides our program into various logical and independent functions. Thus, it ensures smooth execution of a particular operation without any hindrance and intervention. It also provides modularity.

# Data Design

## Data Description:

Once the model and the loss functions are determined, the dataset to be used in the training process must be selected. In the case of GAN training, it is specially important to dispose of a large image dataset, as the quality of the results obtained after the training session will be highly dependent on the amount and variety of cases the model learns during the process.