Informatics Institute of Technology

In Collaboration With

University of Westminster, UK

**“TransformVector”**

**An automated raster to vector conversion platform for GIS (Geographical Information Systems)**

A dissertation by

Damodaran Dillon Thisaru Lakshman

Supervised By

Mr. Pumudu Fernando

Submitted in partial fulfilment of the requirements for the

BSc (Hons) Computer Science

Department of Computing

Declaration

Abstract

Acknowledgement

**Table of Contents**

[1. Chapter 1: Introduction 9](#_Toc39155569)

[1.1 Chapter Overview 9](#_Toc39155570)

[1.2 Project Overview 9](#_Toc39155571)

[1.2.1 Project Background 9](#_Toc39155572)

[1.2.2 Problem Domain 10](#_Toc39155573)

[1.2.3 Problem Justification 12](#_Toc39155574)

[1.2.4 Research Question 12](#_Toc39155575)

[1.2.5 Motivation 12](#_Toc39155576)

[1.3 Project Aim and Scope 12](#_Toc39155577)

[1.3.1 Project Aim 12](#_Toc39155578)

[1.3.2 Project Scope 13](#_Toc39155579)

[1.4 Objectives 13](#_Toc39155580)

[1.4.1 Research Objectives 13](#_Toc39155581)

[1.5 Overview of the Solution 14](#_Toc39155582)

[1.6 Resource Requirements 15](#_Toc39155583)

[1.7 Chapter Summary 16](#_Toc39155584)

[2. Literature Review 17](#_Toc39155585)

[2.1 Overview 17](#_Toc39155586)

[2.2 Problem Domain 17](#_Toc39155587)

[2.2.1 Geographic Information Systems 17](#_Toc39155588)

[2.2.2 Raster Image Data 18](#_Toc39155589)

[2.2.3 Vector Image Data 18](#_Toc39155590)

[2.2.4 Need for both Raster and Vector based data in GI systems 19](#_Toc39155591)

[2.2.5 Raster to Vector Conversion in GIS 21](#_Toc39155592)

[2.2.6 Image Processing 21](#_Toc39155593)

[2.2.7 Smoothening Images 22](#_Toc39155594)

[2.2.8 Image Colour Quantization using K-means technique 23](#_Toc39155595)

[2.3 Comparison of Similar Solutions 24](#_Toc39155596)

[2.3.1 Easy Trace 25](#_Toc39155597)

[2.3.2 FreeHand 25](#_Toc39155598)

[2.3.3 Illustrator 25](#_Toc39155599)

[2.3.4 Potrace 26](#_Toc39155600)

[2.3.5 Image Tracer 27](#_Toc39155601)

[2.4 Algorithmic Analysis 27](#_Toc39155602)

[2.4.1 General Raster to Vector Conversion Algorithms 27](#_Toc39155603)

[2.5 Image Tracer Raster to Vector Process Overview 28](#_Toc39155604)

[2.6 Justification of the Selected Approach 30](#_Toc39155605)

[2.7 Research Contribution/Research Gap 30](#_Toc39155606)

[2.8 Chapter Summary 31](#_Toc39155607)

[3. Chapter 3: Methodology 32](#_Toc39155608)

[3.1 Chapter Overview 32](#_Toc39155609)

[3.2 Research Methodology 32](#_Toc39155610)

[3.2.1 Research Paradigm 32](#_Toc39155611)

[3.2.2 Research Methodology 33](#_Toc39155612)

[3.3 Design Methodology 33](#_Toc39155613)

[3.4 Development Methodology 34](#_Toc39155614)

[3.5 Chapter Summary 35](#_Toc39155615)

[4. Chapter 4: Project Management 36](#_Toc39155616)

[4.1 Chapter Overview 36](#_Toc39155617)

[4.2 Project Management Methodology 36](#_Toc39155618)

[4.3 Potential Risks and Mitigation Plans 37](#_Toc39155619)

[4.4 Work Breakdown Structure 38](#_Toc39155620)

[4.5 Gantt chart 39](#_Toc39155621)

[4.6 Compliance with BCS Code of Conduct 40](#_Toc39155622)

[4.7 Social, Legal, Ethical and Professional Aspects 40](#_Toc39155623)

[4.8 Chapter Summary 41](#_Toc39155624)

[5. Chapter 5: System Requirements Specification 42](#_Toc39155625)

[5.1 Chapter Overview 42](#_Toc39155626)

[5.2 Stakeholder Analysis 42](#_Toc39155627)

[5.2.1 Onion Model 42](#_Toc39155628)

[5.2.2 Stakeholders and Roles 43](#_Toc39155629)

[5.3 Requirement Elicitation Process 44](#_Toc39155630)

[5.4 Use Case Diagram and Description 56](#_Toc39155631)

[5.4.1 Use Case Diagram 56](#_Toc39155632)

[5.4.2 Use Case Description 57](#_Toc39155633)

[5.5 Functional and Non Functional Requirements 64](#_Toc39155634)

[5.5.1 Functional Requirements 64](#_Toc39155635)

[5.5.2 Non-functional Requirements 65](#_Toc39155636)

[5.6 Chapter Summary 65](#_Toc39155637)

[6. Chapter 6: Design 66](#_Toc39155638)

[6.1 Chapter Overview 66](#_Toc39155639)

[6.2 Class Diagram 66](#_Toc39155640)

[6.3 Activity Diagram 67](#_Toc39155641)

[6.4 Sequence Diagram 68](#_Toc39155642)

[6.5 Chapter Summary 69](#_Toc39155643)

[7. Chapter 7: Implementation 70](#_Toc39155644)

[7.1 Chapter Overview 70](#_Toc39155645)

[7.2 Basic Overview of the System 70](#_Toc39155646)

[7.3 Selection of Technologies 71](#_Toc39155647)

[7.3.1 Selection of Programming Language 71](#_Toc39155648)

[7.3.2 Selection of UI Framework 71](#_Toc39155649)

[7.3.3 Selection of Libraries 72](#_Toc39155650)

[7.3.4 Selection of Tools 73](#_Toc39155651)

[7.4 Implementation of Components 74](#_Toc39155652)

[7.4.1 Parameter Trainer Component 74](#_Toc39155653)

[7.4.2 Image Classifier Component 80](#_Toc39155654)

[7.4.3 Backend API 82](#_Toc39155655)

[7.4.4 Frontend 89](#_Toc39155656)

[7.5 Screenshots of the System 90](#_Toc39155657)

[7.6 Chapter Summary 93](#_Toc39155658)

# Chapter 1: Introduction

## Chapter Overview

This chapter covers the overview of the project, it’s background, the justification of the problem being addressed and the research question relevant to this research. The motivation behind attempting this research is also explained. Furthermore, the Aim and the scope that will be attempted in this research will also be defined. The Research objectives that will be fulfilled and also personal objectives of conducting this research will be mentioned along with an overview of the solution. The requirements for such proposed solution will also be defined.

## Project Overview

### Project Background

**Images**Human beings can be considered predominantly visual creatures as we use our sense of vision when available to understand our surroundings. Due to this reason we have created images to capture moments of this ever constantly moving world. Images have since then evolved to not only capturing real world moments but to digital drawings and other representations of visual still media. An image can be a single picture which represents a certain object, location or scenery. When image files are attempted to be classified, two main classifications can be identified. Vector images and Raster images. There are different pros and cons of using either a Vector image or a Raster image.

**Raster Images**

A raster image is built up of colour pixels which are arranged to form the necessary result image.Raster images are mostly suited for linear art images because they can better represent subtle chromatic gradients due to the fact that each pixel can change its value independently of other pixels to form the necessary image. These types of images are also called continuous tone images.

A raster image while being faster to process and display as there is no methodical processing or such involved will have a larger file size as information is electronically stored on a pixel-by-pixel basis (Winnemoeller *et al.*, 2018), the size of the image is directly proportional to the resolution of the image

**Vector Images**Vector images on the other hand are created from points and lines and curves joining them. These are based off of mathematical formulas that create combinations of multiple true geometric primitives to create a final image (Seel-audom, Naiyapo and Chouvatut, 2017)..

Vector format based computer graphics tools have become very powerful tools allowing artists, designers etc. to mimic many artistic styles, exploit automated techniques, etc. and across different simulated physical media and digital media (SEVERENUK *et al.*, 2019). Similarly, in real world applications according to the factors that need to be considered, the image may be required in either Raster or in Vector format.

As stated above, as a vector file and raster file of the same image may have similar resulting image, when observed on a deeper level multiple differences can be identified between them.

### Problem Domain

Geographic Information System (GIS) is the processes of managing, manipulating, analyzing, updating and presenting metadata according to its geographic location, to be effectively used in different aspects of life (Al-Bayari, 2018). The increasing popularity GIS technology has increased the usage of spatial data. Making maps is relatively easy even for those who do not have much cartographic training (Wong and Wu, 1996).

According to the analysis needed to be performed on a certain image obtained, the requirement for a Raster image or a Vector image may vary. There are several pros and cons when you consider each type of image. There is an old GIS adage stating that “Raster is Faster but Vector is Corrector”.(Berry, 1995).

Vector GIS results in the geometrization of the geographical world, and generalizing and reducing its theory into theories about relations between points, lines, polygons and areas. Such objects which are in a GIS can be counted, moved about, stacked, rotated labelled, cut etc. and be handled like a variety of other everyday solid objects that bear no particular relationship to geography (Couclelis, 1992).

Vector maps use simple geometric components such as points, lines and polygons in adjacencies, unions and inclusions to describe spatial information and Raster maps are based on pixel matrices and are richer and realistic than vector maps (Lin and Guo, 2011).

There are several advantages of using a Vector data format. These can be stated as the output being more aesthetically pleasing and zoom able to very close detail as it is made up of points and line segments connecting them and not using fixed number of pixels which might look pixelated and less clear when zoomed into more than its resolution allows. It also provides higher geographical accuracy due the same reason as it being not of a fixed resolution. There are other reasons as why vector images are used in GIS such as data integrity, and allowing network analysis and proximity operations as they both use vector data structures. As well as there are advantages are there are disadvantages to using vector images as well. As these images are a result of mathematical calculations it is often very processing intensive. Vector data structures are also poor performing when displaying continuous data, and needs to be generalized in some manner to display, which can result in loss of some information.

While vector data structures in GIS over determines the geographic world by forcing it into a geometric objects generalizing them, the raster data structure feigns maximal ignorance on the nature of things in the world. Yet Raster data structures provide an implicit view of the geographical world with measurable values discretized into pixel arrays (Couclelis, 1992).

Raster data can store unique values per each pixel without any generalization being required. Therefore, is a good option when continuous data is required to be displayed. Even though continuous data can be very accurately represented in a Raster image, because of the resolution. Raster graphics display devices are capable of reproducing very complex images (Sloan and Tanimoto, 1979). It struggles when representing linear features and can cause pixilation if the resolution of the image obtained is low or when zoomed to obtain a closer look. Raster datasets can also be very large file as when the resolution is increased to get a more accurate image with high detail the file size increases proportionally with it.

From the statements above, it can be identified that both these formats are equally important when considering the use of imagery in GI Systems. Therefore, there becomes necessary a method of conversion between these two data types.

### Problem Justification

Automated conversion of engineering drawings and such similar content into Raster and vector data has been a very widely discussed topic. A critical step in this process can be considered as the conversion of these images into a vector format (Liu and Josep Lladós (Eds.), 2005). Many techniques for conversion of raster to vector have been proposed which has even led to development of commercial solutions to tackle this issue. The systems created all did provide quite acceptable results but each had their own drawbacks (Hilaire and Tombre, 2006). (Lacroix, 2009)

### Research Question

How is the problem of a platform that identifies the properties of a Raster image and converts it into a Vector file by using the best method of conversion using parameters which best fit the use case of the resulting vector image addressed currently in the research domain of GIS graphics processing?

### Motivation

After researching on the basic concept of converting Raster images into Vector images for graphic design purposes, I have come to find the importance of it but in a different domain which is in the field of GIS. Raster and Vector data structures are widely used in analysis in GIS and as both of these type of images are needed according to different situations. It has motivated me to create this automated Raster to Vector conversion tool.

## Project Aim and Scope

### Project Aim

To investigate design and implement a Raster to Vector conversion platform that selects the best method of conversion using image processing techniques.

### Project Scope

**In-Scope**

* Raster to Vector conversion tool is only developed geared towards GIS
* Training an image processing model to identify certain properties of images that affect conversion algorithms.
* Integrating Image processing for the identification special characteristics to identify the best conversion algorithm
* Considering of continuous tone images as well as line based images for the conversion process
* Set conversion method from either one of Accurate or Fast conversion

**Out-Scope**

* Conversion of Raster to Vector for other domains such as graphic design.
* OCR functionality out of image text is not considered, and will be represented in the converted images as graphical data and not textual data.

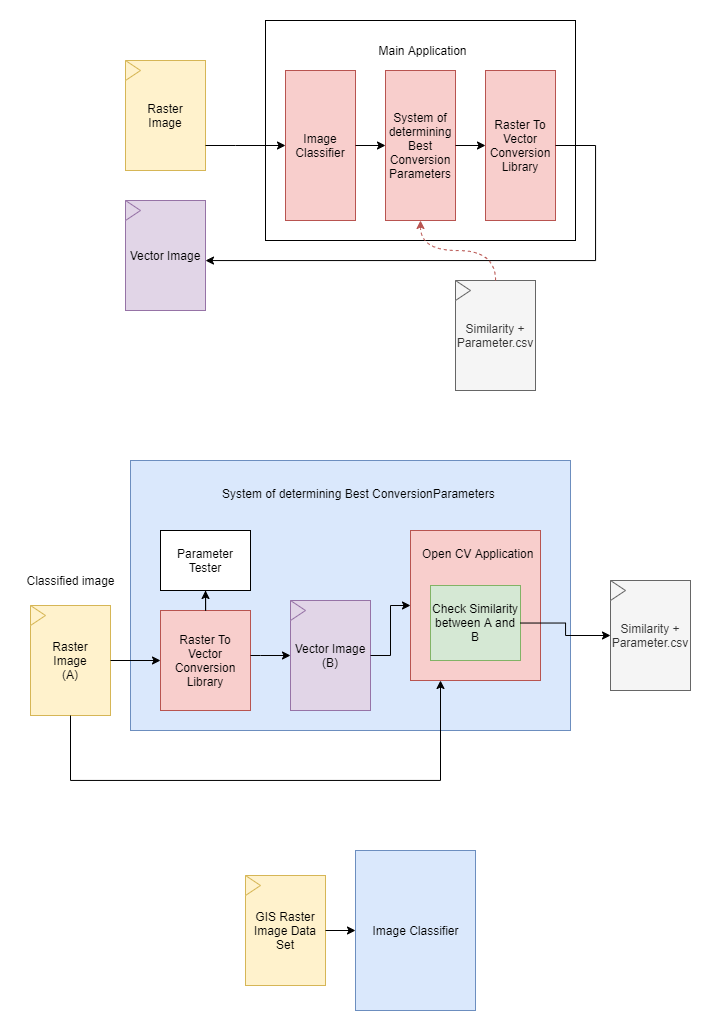
## Objectives

I have identified the following as the objectives to achieve to complete my Research successfully.

### Research Objectives

* To conduct a thorough literature review on existing solutions and platforms
* To design an image processing model to identify properties of a Raster image.
* To implement functionality to determine the most favourable algorithm for the conversion through the parameters identified from the image processing process.
* To evaluate the converted Vector image in terms of accuracy and speed of conversion.

## Overview of the Solution



As shown in the figure above the main system is made up of three components. The image classifier, system of determining best conversion parameters and the raster to vector conversion library.

When taking a closer look at each system; the system of determining best conversion parameters is a machine learning based system that finds the best conversion parameters by interpolating data from a set of previously identified conversion parameters for each image classification and the match rate of the image to its particular classification determined by the Image classifier.

The initial best fit parameters are identified by converting a single image into multiple vector images each time with different parameters within a certain range of each parameter and comparing its visual similarity with the original using an image processing library. This similarity is then recorded in a csv file for the system of determining the best conversion parameter to use.

A data set of several hundred labelled GIS image types are used to train the image classifier.

The raster to vector conversion library is used to convert the images from its raster to vector format in each step that its necessary to do so.

These three main components make up the main system and together convert any given GIS based image which can be classified into an accurate vector image.

## Resource Requirements

\*Note these requirements can be subjected to change

#### Software Requirements

* **Windows 10 (64-bit version):** To accommodate and run the software
* **Java, Python:** For the conversion algorithms to function
* **NodeJS:** For backend related scripting
* **MSWord:** For documentation Requirements
* **GraalVM:** For cross platform application support

#### Hardware Requirements

* Core i7 Processor
  + High processing power required for algorithms to be executed
* Minimum 4GB RAM
  + For application to run smoothly and not run out of system memory
* Disk Space: Up to 10GB
  + For storing of application and images and temporary files created while converting and running algorithms

## Chapter Summary

The project overview was discussed which further explained the Project background, Problem domain, Justification and the Research question and the motivation behind this research. The project Aim and the scope of the project was also discussed. The research objectives of this project and the overview of the solution proposed was also discussed. Finally, an estimation of the software and hardware requirements of the system was discussed.

# Literature Review

## Overview

This chapter will provide a critical analysis on the domain of the problem based on literature. This chapter will analyze the current methods of raster to vector conversion why they are needed, the different variations of raster to vector conversion system. This chapter will also review similar work for the process of conversion of Raster to Vector.

## Problem Domain

### Geographic Information Systems

Geographic Information Systems (GIS) are computer based tool for process of manipulating, updating, managing, analyzing and presenting data of geographic locations and cartographic data which can be thereafter used in different fields of study or be applied in daily applications which require such data. (Al-Bayari, 2018) (Chrisman, 1999).

Due to proliferation of GI science and the usage of spatial data, map making has been simplified even for someone with little to no experience in cartographic training (Wong and Wu, 1996).

During the earlier stages of GI Science, the definition of the it simply putting maps into computers. The process of implementing this idea however was a tedious task and to complete satisfactorily was a major undertaking. Recently however the limitation of spatial data in 2 dimensional vertical view of the surface of the globe is not as sought after and scientist demand for 3 dimensional views and the ability to simulate various geographic processes though GIS data (Gold, 2006).

This GIS information can be stored and processed in image formats after being generated. There are two main types of image structures that can be used when storing this information. This can be classified as Raster data and vector data (Wade *et al.*, 2003).

### Raster Image Data

Raster images, which are also known as bitmap images are categorized under digital images as being formed of tiny rectangular pixels which are arranged in a grid formation that combines together to represent an image. This format of image can support a wide range of colours and depict subtle gradients, it allows for a very accurate visual representation of continuous tone images such as shaded drawings, photographs and other highly detailed and complex imagery.

Raster graphics initially originated in television technology with images constructed much like the pictures on a television screen. Raster graphics are made of small uniform sized pixels which are arranged in a two dimensional grid which is made up of vertical and horizontal columns. A single pixel contains information of a single or multiple bits depending on the degree of information in the image. For example, black and white images contain only one bit per pixel in the image this can either be a true black bit or a true white bit, an image with shading or colour commonly contains 24 bits of information per pixel, this allow more than 16 million possible states of colour value for the pixel. Images with 24 bits of information per pixel are known as ‘truecolor’ images with 24-bit colour and can realistically depict colour images. The amount of detail stored into a single bit is represented by the colour depth and the number of pixels that form together to form the image can be represented as its resolution and affects how much detail is depicted in an image (Britannica, 2014).

### Vector Image Data

Vector graphics are created using a sequence of commands or mathematical formulas that connect edges and nodes in a 2 or 3 dimensional space to render an image. In the field of physics, a vector is the representation of a quantity that contains both a direction and a quantity or value. For example, instead of storing the data as an array of pixels as when a raster images or bitmaps, a vector file creates its complex images by mathematically aligning and stacking multiple polygons created by connecting a series of point with lines segments. This results in the creation of typically smaller files with extremely high fidelity, lossless compress and scalability of any kind without the distortion or loss of fidelity of it (He, Xu and Zheng, 2009).

### Need for both Raster and Vector based data in GI systems

According to the type of processing and analysis to be performed on a certain data set obtained the requirement for the format of the data type, either Raster or Vector may be different. Each data type has its advantages and disadvantages, strengths and weaknesses and to further solidify the need for both data types an old GUS adage states that “Raster is Faster but Vector is Corrector” (Berry, 1995).

A well-known logical consequence of the difference of the data structures vector and raster is that as while vector data can record position to and degree of accuracy, raster data have a built in level of positional accuracy. Therefore we can also classify raster positional data as integer and vector positional data as real(Holroyd and Bell, 1992).

GIS data in vector format is produced by geometricizing the real geographical world, generalizing it and reducing it into theories about points, lines polygons and areas. Objects thus created using these theories can be counted translated, stacked, rotated, labeled, cut and etc. and can be handled like every day geometrical shapes that are not directly related to geographical data in any manner (Couclelis, 1992).

While vector maps use simple geometric structures as points, lines, polygons and relationships between them such as adjacencies, unions and inclusions to depict the geographical data, Raster based maps are stacked matrices of pixel based data which can be layered in a sandwich like structure and are more rich and realistic than vector data (Lin and Guo, 2011).

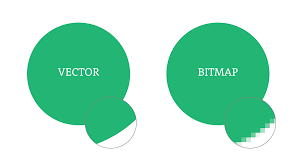


Figure 2.2‑1 Visual comparison between vector and bitmap image

Advantages of using a vector data format can be stated as the output produced being more aesthetically pleasing, highly zoom able as it is made up points and line segments contented to each other to form multiple polygons and not a pixel array which when zoomed into more than its set resolution allowed may look pixelated. There are a number of other advantages of using a vector data structure rather than a raster data structure, and these can be listed down as follows.

1. Data integrity
2. Allowing network analysis
3. Proximity operations as they both use vector data structures

While there is strength in using vector data structures to represent and process GIS data, there are weakness of it as well. As vector data is produced by the result of geometrical mathematical calculations, this data produced can be very processor intensive when for example depicting continuous data regarding a point if stored in vector format is much more complex than a single pixel storing information which makes up a raster image. Therefore, due to the feasibility of the idea of storing data as a series of points made up of geometrical data being not practical these images must be generalized in some manner to ease storing and processing when needed to be displayed. This result in the loss of some portions of information.

Even though GIS data that is stored as vector data structures over determines the geographic world by generalizing them and forcing them into geometric objects, Raster data structures provides an implicit view of the world with measurable values discretized into pixel arrays but simulates ignorance on the nature of an actual object in a physical space (Couclelis, 1992).

A simple advantage of using raster data over vector is the ability to store continuous tone images as pixel array which stores unique values of data per pixel without performing any generalization which may lead to data loss in some manner. But while continuous data can be stored very accurately using raster images, as the resolution of a raster graphic is set and cannot be varied, it may limit the fidelity of final output if the process requires the image to be enlarged. Raster graphics display devices are capable of reproducing very complex images (Sloan and Tanimoto, 1979). Raster images also have the weakness of not being able to produce true linear features as pixels used to form as arranged horizontally and vertically, are limited when reproducing line data at any other angle. Raster data sets also have very big file sizes as the higher the resolution the more pixel data it contains. Therefore, the file size is directly proportional to the resolution of the image, and by resulting the file size of the image to be directly proportional to the visual fidelity of the image.

### Raster to Vector Conversion in GIS

Spatial data obtained as aerial imageries, for example is very continuous in nature and must be represented as raster data. These types of maps are unable to be displayed by vector maps (Lin and Guo, 2011).

With development of high speed computing and its usage in a wide range of fields of study leading to improve process times and increasing the ability of highly computational tasks to be performed on smaller and more affordable system. Vector data can be used as a much more accurate data structure as the images even with complex processing requirements can be easily displayed and rendered faster than before (Winnemoeller *et al.*, 2018). Vector images are also known for its flexibility and compact nature etc.(Lin *et al.*, 2015). By going through the above use cases for each raster and vector images, we can come to the conclusion that both of these data structure is widely used in the Geographical Information Sciences. Therefore, there arises a need for conversion between these data structures. Even though many commercial solutions and researches have been conducted in achieving a satisfactory result, they each have their drawbacks and hence arises a need for a common solution to this problem.

### Image Processing

Digital Image Processing is the process of manipulation of images using computers. Image processing use cases have been increased exponentially during recent times. Its application ranges from entertainment, passing by geological processing and remote sensing. Multimedia systems which play a huge role in modern society also heavily rely on image processing.

When studying the various disciplines of image processing we can identify that it is a vast topic with many techniques that are applied specific to the type and qualities of an image. An image can be regarded as a function f (x, y) of two continuous variables x and y. For images to be processed digitally the image has to be sampled and transformed into a matrix of number. And quantization is then required as a computer represents number using finite precision. Image processing then can be identified as the manipulation of those finite precision numbers.

Image processing can be mainly categorized into (Eduardo A.B. da Silva, 2005).

* **image enhancement**
* **image restoration**
* **image analysis**
* **image compression**
* **color quantization**

### ****Smoothening Images****

Smoothening images can also be known as blurring of images. This can be used to de-noise Images if they contain a lot of noise that might affect the quality of the raster to vector conversion process resulting in a final output with unnecessary polygons that were a result of noise on the image. As all images will not have noise in them. This should be selectively applied to images if only necessary.

The basic process of smoothing operations is done by applying a filter to the image. The most commonly used filter is the linear filter in which the output pixel value (i.e. *g(i,j)*) is a calculated as a weighted sum of input pixel values (i.e. *f(i+k,j+l)*) and *h(k,l)* which is called the kernel, which is the coefficients of the filter.

g(i,j) = \sum_{k,l} f(i+k, j+l) h(k,l)

The commonly used filter types are mentioned below.

#### Normalized Box Filter.

The output pixels are calculated to be the mean of its kernel neighbors.

K = \dfrac{1}{K_{width} \cdot K_{height}} \begin{bmatrix}
    1 & 1 & 1 & ... & 1 \\
    1 & 1 & 1 & ... & 1 \\
    . & . & . & ... & 1 \\
    . & . & . & ... & 1 \\
    1 & 1 & 1 & ... & 1
   \end{bmatrix}

#### Gaussian Filter

Convolving of each point of the input images array with a Gaussian kernel and adding them to produce the output array as a sum.

A 2D Gausian can be represented as follows

G_{0}(x, y) = A  e^{ \dfrac{ -(x - \mu_{x})^{2} }{ 2\sigma^{2}_{x} } +  \dfrac{ -(y - \mu_{y})^{2} }{ 2\sigma^{2}_{y} } }

* Mean (the peak)
* Standard Deviation
* The values of the two axis



#### Median Filter

Replaces each pixel in the input images pixel array with the median values of its neighboring pixels. The neighboring pixels are of a square neighborhood around the pixel.

### Image Colour Quantization using K-means technique

Initially a fixed number of clusters and initial cluster centers in the color space are identified and chosen. The reason for this is to change the position of cluster centers so long as the sum of distances between all points of clusters and their cluster centers will be minimal.

During these modifications all points are allocated to closest cluster centers using a predefined metric. Typically used metrics are,

* Euclidean distance
* City Block metric.

After each allocation a new positions of cluster centers are computed as arithmetical means of cluster points. The algorithm usually stops if the difference between new and old positions of cluster centers is too small. K-means converges to a locally optimal solution.



Therefore, in this project, K-means clustering has been identified as the preferred colour quantization method to be used.

## ****Comparison of Similar Solutions****

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name of Software | Supported Image Formats | Max Zoom | Adjust Image | OCR | Scripting Support | Ability to adjust conv. params |
| Easy Trace | ArcINFO, ArcView, AutoCAD, Credo, MapInfo, MicroStation | 12800x | Yes | No | No | No |
| Free Hand | PNG, JPEG, TIFF, SVG, EPS | 64x | No | No | Yes | No |
| Illustrator | PNG, JPEG, TIFF, SVG, EPS | 640x | No | No | Yes | No |
| Potrace | PBM, PGM, PPM, or BMP (Bitmap only) | N/A | No | No | No | Yes |
| Scan2Cad | PDF (Scanned files) | 64K | No | Yes | Yes | No |
| Image Tracer | PNG, JPEG, TIFF, SVG, BMP | N/A | No | No | Yes | Yes |

Table 2.3‑1 Comparison of Similar Solutions

### Easy Trace

The easy Trace pro software package is a convenient and powerful tool for vector map generation with editing capabilities. The program also is widely used in GI systems and supports most of the import and export formats such as.

* ArcINFO
* ArcView
* AutoCAD
* Credo
* MapInfo
* MicroStation

This software application is only functional on a windows operating system. It also contains raster to vector conversion tools and is create by the Easy Trace group (Russia). Old versions are free versions but all newer updated version are paid versions of this software. The drawback of this software comes when we consider that it does not support scripting or the fact that its conversion parameters cannot be modified by the user.

### FreeHand

Adobe Freehand (a.k.a Macromedia Freehand) is a commercial computer application for creating vector graphic content. Created by the Altsys Coporation in 1988 and licensed to Aldus corporation this software released version 1 through4 in 1994, and was discontinued since 2003. This application also allowed a user to convert a placed raster image into a vector. This application is now discontinued.

### Illustrator

Adobe illustrator is an industry grade application for multimedia, online and print graphics creation. It provides tools to create technical illustrations or graphic related content for print publishing and also allows web related graphic content creation. It is a commercial tool. This software also allows a functionality known as image trace which allows the conversion of images from raster to vector. As this Raster to vector conversion is only a small part covered in this digital graphics creation software and it’s lack of support for scripting and parametrization of conversion options we have moved to other solutions that can be used in the project.

### Potrace

Portace(TM) is a bitmap tracing tool which allow the transforming of bitmaps into raster based scalable images. The input type of the bitmap can be of any of the following formats

* PBM
* PGM
* PPM
* BMP

And the output will be into a vector format file. A usecase of this software can be create SVG or PDF files from scanned raster images. The resulting vector image then can be scaled to any resolution without any distortion in the image quality.

Protrace uses the output formats

* SVG
* PDF
* EPS
* PostScript
* DXF
* GeoJson
* PGM

It is a software package that is currently in development and further image type support will be added in future times. Portrace does not preprocess the image before it is converted to a vector file format and this has to be performed beforehand by a different application. Though this application is a great software when it comes to doing what it is designed to do. The support for only BITMAP and grayscale line drawing output, really puts it in a bad spot when being considered to be used as a tool in this project.

### Image Tracer

Image Tracer is a simple Raster to vector conversion tool. It is an open source software and has an active github that fixes and improves its code base. It allows the ability to modify its code as necessary for use and also provides the ability for its conversion parameters to be adjusted by the user for each conversion to get the best output. It supports the following Input formats.

* PNG
* JPEG
* TIFF
* SVG
* BMP

This application can be built and deployed as a Java application and can be executed using wrappers with other programming languages and also through the command line. As its output image quality is also of a high quality and as it also supports many features required on this project. Image Tracer is identified most suitable to be used for Raster to Vector Convesion in this project.

## Algorithmic Analysis

### General Raster to Vector Conversion Algorithms

The task of converting a raster based data structure in to line data or into vector based data structures can be divided into three basic operations. These are as follows

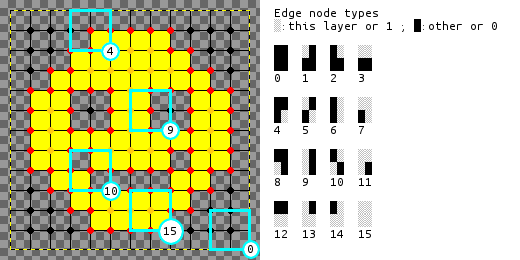
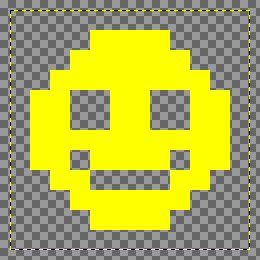
* Skeletonization/Line Thinning
  + This is process of reducing or thinning the thickness of the lines found in the image to a unit thickness relative to the given resolution
* Line Extraction/Vectorization
  + The process of identifying a series of data entities or coordinates that are used o form a single line segment in the input document
* Topology Reconstruction
  + The process of determining the adjacency relationship among the lines identified during line extraction.

The individual line segments are joined into whole lines if desired and can also be combined as polygons for continuous representation. Two other additional post processing tasks that can applied onto the basic raster to vector conversion process can be stated as follows.

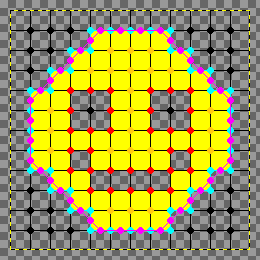
* Line smoothening
* Gap Removal

## Image Tracer Raster to Vector Process Overview

Initially colour quantization for the image is carried out. This colour quantization is done using the indexed color method. This method is a commonly used color quantization technique to manage colors in a digital image to reduce its size. This helps in further computational tasks and reduces the number of colours therefore making polygonization of continuous colours easy moving forward. After colour quantization is complete layer separation and edge detection of the image is carried out. The layering function creates an array for every colour and calculates edge nodes based on the data obtained. These are at the center of every 4 dots as shown in the illustration below.

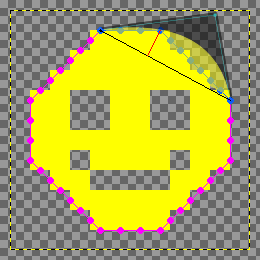


After this step, a path scan is function finds and chains the edge nodes of the image. Once complete, the internode function interpolates the coordinates of every edge node path and finally the interpolated paths are traced to obtain the basic paths of the image.



A function is then carried out to draw a connection between two lines to fit a straight line on the start - and endpoint of the sequence. If the distance error between the calculated points and actual sequence points is greater than the threshold, the point with the greatest error is selected.

Then a quadratic spline through the error point determined in the previous step is drawn and fitted appropriately.



Finally, the coordinated are rendered into SVG paths.



## Justification of the Selected Approach

As the literature review has been conducted, the various similar solutions available on a commercial level and on a research level are identified not address identifying the qualities of the raster image before conducting the process of conversion of the image from raster to vector. Hence this project will be further looking into the usage of image processing to identify various image processing techniques for the pre-processing of the image before converting it from raster to vector and also identifying the most suitable parameters for the algorithm to be executed on for an accurate conversion.

## Research Contribution/Research Gap

Raster to Vector Image conversion systems found in the market and that are developed on a research-level mostly operate on fixed parameters on their conversion process and algorithms. And when application does have these parameters for access of the user to change to streamline an image output to be of a better quality and more suitable to the type of image which is to be converted. The user may not have enough knowledge on the factors that each of these parameters and their values might do the result output image obtained after conversion. Due to this, there arises a need for an application to analyze the image, classify it and identify the best parameters required for a better quality and accuracy to be obtained in the resulting vector image that is produced from the conversion process.

Images are also sent through a pre-processing stage which the user can customize as they see fit for image quality and other options such as de-noising and colours to be used in the resulting image through quantization.

The contribution of this research includes analyzing an image and identifying the best fit conversion parameters dynamically to obtain a higher structural accuracy between the input and output images. This research will also contribute to GIS researches in converting raster images of common types of classifications of GI data that can be found in use among GI researchers accurately to a vector format to aid in their research and studies. This research will also

## Chapter Summary

As outlined in this chapter, an image processing solution to identifying the best conversion method and parameter identification when raster to vector conversion is considered is discussed.

This is to combat the fact of there not being a standard when identifying the best method of conversion from raster to vector. A number of raster to vector conversion tools have been reviewed and researched on for this purpose. And a conclusion was reached upon after carefully analyzing their functionality and algorithmic features.

# Chapter 3: Methodology

## Chapter Overview

In this chapter the methodology behind the research approach which has been selected for this research will be discussed. The research paradigm and the research methodology selected will be elaborated and justified. The design methodology and the development methodology of the project will also be discussed.

## Research Methodology

This research is based on the conversion of the process of Raster to Vector in classified GIS imagery with the use of best-fit conversion parameters. The theory of the research is based on a hypothesis that has been established as follows. "Images that have similar characteristics and can be classified as such using an image classifier can have a similar set of best-fit conversion parameters for Raster to Vector conversion to obtain optimal results of conversion."

### Research Paradigm

When considering the different approaches that can be considered when approaching this research topic. It can be identified that there are three clear approaches that have been established by researchers. These are the approaches of 1) Positivism 2) Interpretivism 3) Critical theory. This is essential because as consumers of research, we have to be able to look deeper into claims made by researchers who adhere to different research paradigms (Reference here).

As a hypothesis is established and the testing to obtain empirical data to form a relation between the phenomena is carried out. The research paradigm of this research can be concluded to be of a **positivism approach**.

### Research Methodology

When considering research methodologies, there are two main types of methods that researches generally can be categorized under. 1) Deductive approach 2) Inductive approach. The deductive approach starts with specifying the objective. The inductive starts through an observation and arriving to a conclusions through the observation. (Reference here).

As the hypothesis that is being established at the beginning of this section is clear, and the work of this research is to prove its validity. We can consider this research to be of the **deductive approach**.

## Design Methodology

Two main design methodologies can be considered to be used to design the proposed system. These can be identified as Object-Oriented Analysis and Design (OOAD) and Structured Systems Analysis and Design Method (SSADM). Each of these methods have their pros and cons according to what kind of system is being designed. In this section we will identify factors when considering each design methodology to pick the most suitable approach for designing the proposed system.

In the Object-Oriented analysis and design approach initially, requirements are identified and a software specification and documentation is developed in terms of an object model. These are objects that integrate both data and function ns and are modeled after real-world objects. After these objects are identified it is then mapped into classes and constraints and relationships are identified. This methodology gives access to certain OOD principles such as classes and objects, encapsulation, polymorphism and interfaces/abstract classes. OOD can be considered a good design methodology to manage complexity in applications and to enable the reusability of components.

In the structured system analysis and design methodology graphical tools are used in a systematic approach to refine objectives out of well-defined user requirements. In the case of these requirements not being clearly described initially, it can lead to problems in the process of the solution created. This approach also does not accommodate dynamic user requirements that might be subjected to change through the development life cycle of the solution.

When considering the above factors as the requirements of the proposed solution may be subject to change through the life cycle of this research. The **OOAD** methodology is better suited for the analysis and design methodology of this project.

## Development Methodology

When starting a project that has as purpose the software development, it is very important to use a methodology that increases its success rate. The success rate of a Project can be improved by using a methodology that fits the characteristics of the project. Multiple software development methodologies that can be employed when determining the development life cycle of a software application. These can be identified as follows. Waterfall, Agile, Feature-Driven, Iterative, Spiral, Prototyping and RAD (Rapid Application Development). The following features that might be considered as factors that affect which mentioned methodology is most suitable for this project can be identified due to their repeated occurrence during this project.

* The initial requirements are identified and recorded in the Project initiation document.
* These requirements are subjected to change over the development of the project due to feedback and development constraints
* Development is done by breaking down the project into components
* Development will be done component by component.
* Each component will have the ability to be tested independently.

The Agile development methodology can be considered an iterative and incremental process that focuses on rapid delivery of working prototypes that finally create the working product in incremental builds. When considering the above stated factors, it is evident that an **Agile** Software development methodology is better suited for the development methodology of this project.

## Chapter Summary

As to conclude this chapter, the projects Research methodology and Research paradigm has been selected as the deductive approach and the positivism approach respectively. The Design methodology of the project is also selected as to the OOAD (Object Oriented Analysis and Design) methodology. Further the chapter also mentions selecting Agile Development Methodology as the suitable Software development methodology of this project as the software as the initial requirements identified may be subjected to change during the course of the project and a dynamic development methodology is thus required to adapt and overcome the challenges that follow along with such a project lifecycle.

# Chapter 4: Project Management

## Chapter Overview

This chapter will discuss the Project Management Methodology of the project. Various project management mythologies will be evaluated and a suitable methodology will be selected as to fit the characteristics of this project. The potential risks and the mitigation plan in a case where these occur will also be discussed. Further the work breakdown structure and the project time line as a Gantt chart will be described. Finally, the Social, Legal, Ethical and Professional Aspect along with the chapter summary will be included in this chapter.

## Project Management Methodology

There are many unique project management methodologies that have been tested and approved by many industry professionals. Some of these can be stated as Agile, Scrum, Kanban, PRINCE2 etc. An agile development methodology can be simply defined as iteratively trying to deliver whatever works at given equal stages of time. Scrum can be defined as using a small cross functional team to deliver result fast. Among these project management methodologies, we take a closer look at the PRINCE2 Project management framework.

PRINCE 2 is a project management framework that is scalable and can be adaptable to any type of project with ease. PRINCE 2 focuses on dividing the work process of a project into manageable and controllable stages. PRINCE 2 is a widely used standard among many professionals in the industry and is a flexible project management methodology that can accommodate with changing requirements of a project. PRINCE 2 is based on seven principles. These can be stated as follows, ***Continued Business Justification, learn from Experience, Defined Roles and Responsibilities, manage by Stages, manage by Exception, Focus on Products, Tailor to Suit Project Environment.***

When considering these principles, and as we can see that the project being considered in this research does have tendency for requirement alteration as progress is being made and there is much to learn and discover through the life cycle of the research. PRINCE 2 can be considered a suitable project management framework to be employed for the management methodology of this project.

## Potential Risks and Mitigation Plans

The following tables define potential risks and its respective mitigation plan.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk 01** | Lack of Domain Knowledge | | | | |
| **Risk Level** | High | **Phase** | All | **Frequency** | Medium |
| **Mitigation** | * Perform an in-depth review and analysis of currently available literature regarding domain and relevant topics | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk 02** | Inability to meet set deadlines | | | | |
| **Risk Level** | High | **Phase** | All | **Frequency** | High |
| **Mitigation** | * Meet with mentor and keep track of progress made on a regular basis. * Keep personal deadlines for each task by breaking it down further within main timeline | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk 03** | Changes in Project Requirements | | | | |
| **Risk Level** | High | **Phase** | All | **Frequency** | Medium |
| **Mitigation** | * Maintain priorities and make changes accordingly so as not to affect project timeline | | | | |

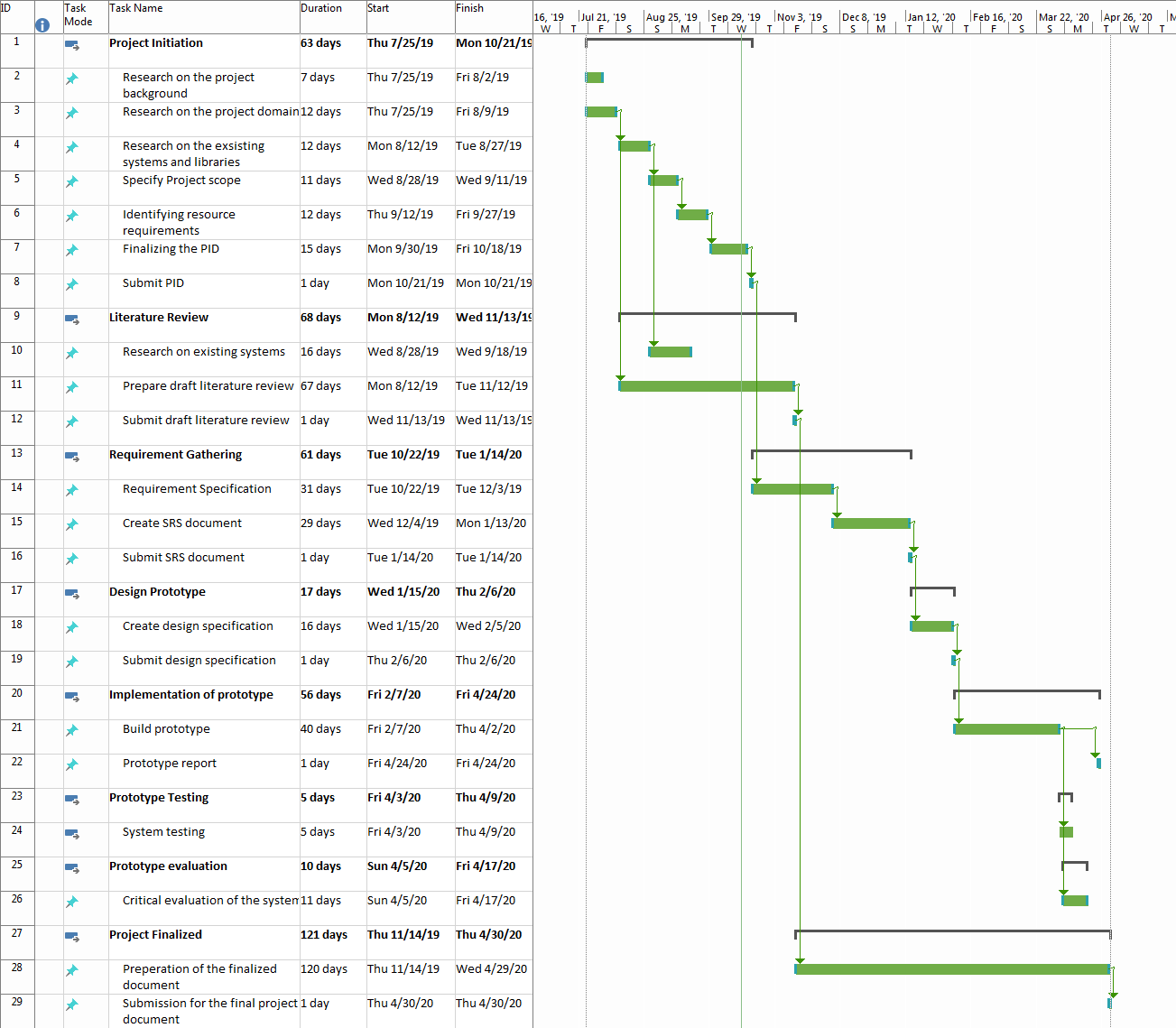
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk 04** | Technical limitations | | | | |
| **Risk Level** | High | **Phase** | Implementation | **Frequency** | Medium |
| **Mitigation** | * Alter method of development to find work around said limitation | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Risk 04** | Progress loss due to loss of data | | | | |
| **Risk Level** | High | **Phase** | All | **Frequency** | Low |
| **Mitigation** | * Regular data back-ups to cloud storage * Maintain Github repository for code management * Have local backup of project if drastic change is being made in case roll-back is necessary. | | | | |

## Work Breakdown Structure

|  |  |  |  |
| --- | --- | --- | --- |
| **Task Name** | **Duration** | **Start** | **Finish** |
| 1. Project Initiation | 63 days | 7/25/19 | 10/21/19 |
| Research on the Project Background | 7 days | 7/25/19 | 8/2/19 |
| Research on the Project Domain | 12 days | 7/25/19 | 8/9/19 |
| Research on the existing systems and libraries | 12 days | 8/12/19 | 8/27/19 |
| Specify Project Scope | 11 days | 8/28/19 | 9/11/19 |
| Identify Resource Requirements | 12 days | 9/12/19 | 9/27/19 |
| Finalizing PID | 15 days | 9/30/19 | 10/18/19 |
| Submit PID | 1 day | 10/21/19 | 10/21/19 |
| 1. Literature Review | 68 days | 8/12/19 | 11/13/19 |
| Research on Existing Systems | 16 days | 8/28/19 | 9/18/19 |
| Prepare draft literature review | 67 days | 8/12/19 | 11/12/19 |
| Submit draft literature review | 1 day | 11/11/19 | 11/11/19 |
| 1. Requirement Gathering | 61 days | 10/22/19 | 1/14/20 |
| Requirement Specification | 31 days | 10/22/19 | 12/3/19 |
| Create SRS document | 29 days | 12/4/19 | 1/13/20 |
| Submit SRS document | 1 day | 1/14/20 | 1/14/20 |
| 1. Design Prototype | 17 days | 1/15/20 | 2/6/20 |
| Create Design Specification | 16 days | 1/15/20 | 2/5/20 |
| Submit Design Specification | 1 day | 2/6/20 | 2/6/20 |
| 1. Implementation of Prototype | 56 days | 2/7/20 | 4/24/20 |
| Build Prototype | 40 days | 2/7/20 | 4/2/20 |
| Prototype Report | 1 day | 4/24/20 | 4/24/20 |
| 1. Prototype Testing | 5 days | 4/3/20 | 4/9/20 |
| System Testing | 5 days | 4/3/20 | 4/9/20 |
| 1. Prototype Evaluation | 10 days | 4/5/20 | 4/17/20 |
| Critical evaluation of System | 11 days | 4/5/20 | 4/17/20 |
| 1. Project Finalization | 121 days | 11/14/19 | 4/30/20 |
| Preparation of the finalized Document | 120 days | 11/14/19 | 4/29/20 |
| Submission of the final Project Document | 1 day | 4/30/20 | 4/30/20 |

## Gantt chart



## Compliance with BCS Code of Conduct

The BCS code of conduct is a professional set of standards to be followed to ensure ethical computing practices being followed in the United Kingdom. It is directed to all member of the British Computer Society. The principal duties of the BCS Code of conduct is as follows.

* The Public Interest
* Duty to the Profession
* Duty to Relevant Authority
* Professional Competence and Integrity

While implementing and conducting research on the project the author has followed the BCS code of conduct to ensure that Legal, Ethical and Social issues within the IT industry have been followed and carried out in a professional manner. The author also confirms that there have been no breaches of conduct of the BCS code of conduct during the implementation and research of this project.

## Social, Legal, Ethical and Professional Aspects

|  |  |
| --- | --- |
| **Social** | **Ethical** |
| * The platform will be available to anyone who wishes to use the software under a Standard End User License Agreement (EULA) * Any individual willing to use this research for further development or use it as it is, will be allowed to do so. | * Creating inaccurate result images - * In terms of an ethical standpoint, for data which might define county boundaries for example if produced inaccurately can cause a lot of issues if used for high influential statements. |
| **Legal** | **Professional** |
| * Computer Miss use Act Violation * Data obtained for training of the system has been obtained through publicly available image training data sets with fair use. | * BCS Code of conduct adhered to and not violated.   + Having due regard for rights of Third Parties   + Promote equal access to benefits to all sectors |

## Chapter Summary

In conclusion to this chapter we have identified the Project Management Methodology to be used in this project to be PRINCE2 after identifying the characteristics of the project and determining its suitability. The Potential risks that may occur through the lifespan of the project have also been identified and mitigation plans for each scenario has been discussed to avoid to factors that might lead to short coming in the project development. A work breakdown structure and Gantt chart for the time of the project have been illustrated. Further to conclude the compliance to the BCS code of conduct and Social, Legal Ethical and Professional aspect analysis of this project is discussed.

# Chapter 5: System Requirements Specification

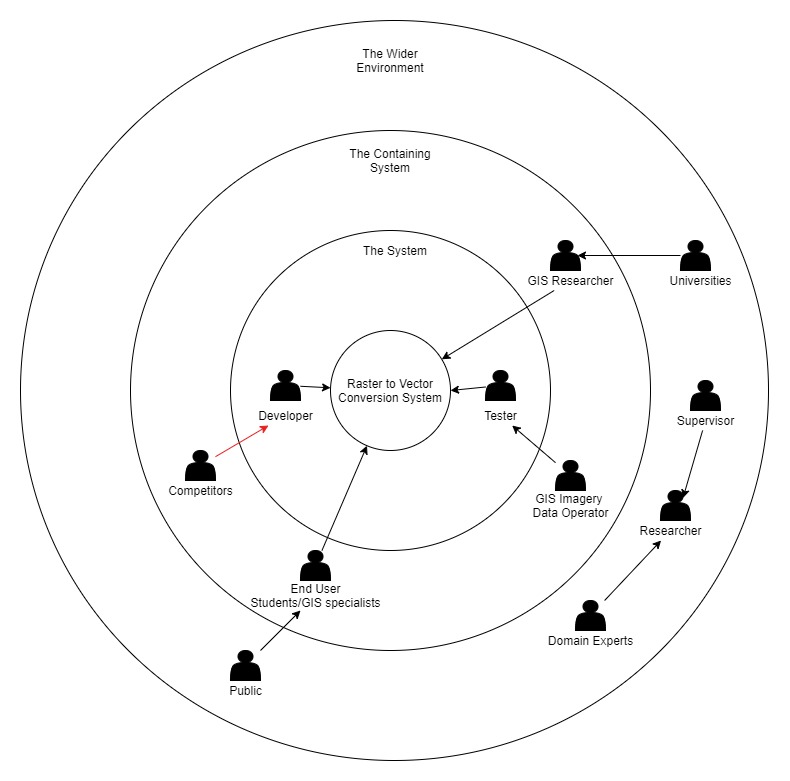
## Chapter Overview

The goal of this chapter is obtain information on the requirements of the Raster to Vector Conversion system. Initially a stake holder analysis is carried out and then several requirement gathering techniques are used to identify functional and non-functional requirements that should be considered when implementing the system. Finally, user case descriptions are created to further strengthen the functional requirements identified and how they impact the operational flow of the system once implemented along with the chapter summary.

## Stakeholder Analysis

The stakeholder analysis will be visually represented using an onion model diagram in this chapter and be further described using the roles and benefits table for each stakeholder.

### Onion Model



### Stakeholders and Roles

The following table describes the stakeholders and their roles and benefits to the system.

|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Roles** | **Benefits** |
| Developer | Develop System | Develops the platform with less cost |
| Tester | Test System | Test and report the accuracy of the system |
| Domain Experts | Expert on the field of study (GIS) | Provide expertise on domain related matters to make system results accurate and to evaluate them |
| Universities | Functional Beneficiary | Is allowed to convert valuable geographical data from raster formats into vector formats and other educational uses |
| GIS Researches | Is allowed to convert valuable geographical data from raster formats into vector formats |
| End User |
| Public |
| GIS Imagery Data Operator | Gather Training and Testing Data | Make system more accurate if more data sets are found for training and testing |
| Supervisor | Assist in documentation and process of building system | Improve quality of system and its documentation by giving appropriate feedback |
| Researcher | Conduct Literature reviews and research to help implement system and identify requirements | Creates valuable documentation for ease of implementation of system |
| Competitor | Negative Stakeholder | Build similar solutions with better features |

## Requirement Elicitation Process

There are multiple techniques that can be used in order to validate and verify the requirements gathered. These can be states as Questionnaires, Observations, Literature Reviews etc. This section will briefly discuss the strengths and weaknesses of each such method and justify the method(s) of approach selected for this project.

|  |  |
| --- | --- |
| **Method 1: Questionnaire** | |
| Questionnaires are carried out to gain knowledge of the developers who have a similar experience in the industry. This form is sent out to the target audience to get their feedback regarding the research. | |
| **Advantages** | * Larger research audience * Easy to analyses data obtained from questionnaire quantitatively |
| **Disadvantages** | * Feedback could contain facts that are untrue |
| **Execution:** A questionnaire is circulated among the target audience of the project which mostly comprises of developers working in the same field along with GIS Researchers who work on a daily basis with the cartographic and geographical image data that might require conversion between raster and vector. The questions were as follows.   * To identify the percentage of users familiar with raster to vector conversion software . * To identify the level of the domain where these conversion methods are used. * To analyze if satisfactory result was obtained when using existing systems. * To understand the user reaction towards the system proposed. | |

The information gathered from the questionnaire will be further discussed as this section continues.

**Operationalization on Questionnaire**

|  |  |
| --- | --- |
| **Objective** | **Question No.** |
| * To identify the percentage of users familiar with raster to vector conversion software | **1** |
| * To identify the level of the domain where these conversion methods are used | **2, 3, 5, 6** |
| * To analyze if satisfactory result was obtained when using existing systems | **4, 7** |
| * To understand the user reaction towards the system proposed | **8, 9** |

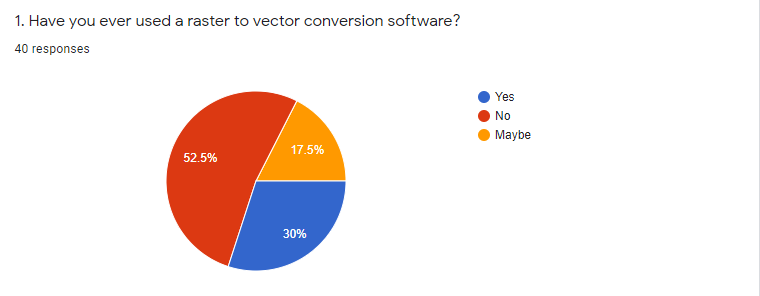
This questionnaire was sent out to individuals working in the IT and design industry who may or may not have experience with raster and vector graphics and platform that allows conversion between these two formats. This survey was helpful in determining if:

* Users were satisfied with the current industry options available for conversion.
* To gauge their reaction towards the proposed solution

**Question 1**

**Aim:** Identify the percentage of users familiar with raster to vector conversion software.

**Observation:**



The purpose of this question was to determine the percentage of the users that have used raster to vector conversion software previously. It seemed that 30% of the users had used a raster to vector conversion software and 52% of the user had no used any tool that would resemble raster to vector conversion.

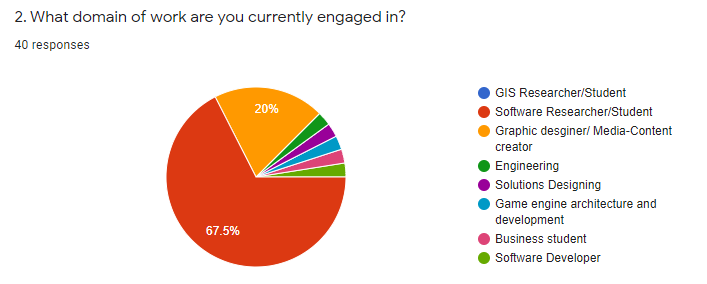
**Conclusion:**

The use of raster to vector conversion even in the technical and design domain is only just below a perfect third of the users. This shows that the requirement for the raster to vector conversion platform even though is niche amongst the user is necessary for some users.

**Question 2**

**Aim:** Identify domain of the users who have used raster to vector conversion software

**Observation:**



The purpose of this question was to determine the domain specification of the user base of the questionnaire so a better understanding as to who it needs to cater to can be identified.

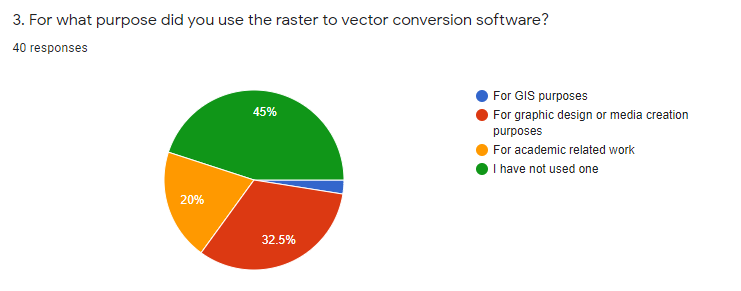
**Conclusion:**

It can be observed that a large base of the users that provided feedback for the questionnaire are Software Researchers/Students. While the second demographic being Graphic designers/media content creators. There is also a portion of GIS Researchers and Students. Therefore, we can conclude that this questionnaire has reached the target demographic of users that are most likely to use a Raster to Vector conversion tool.

**Question 3**

**Aim:** Determine purpose of using a raster to vector conversion software among users

**Observation:**



The purposes of this question was to identify for what purposes the target demographic as mentioned above of this questionnaire use Raster to Vector conversion platforms.

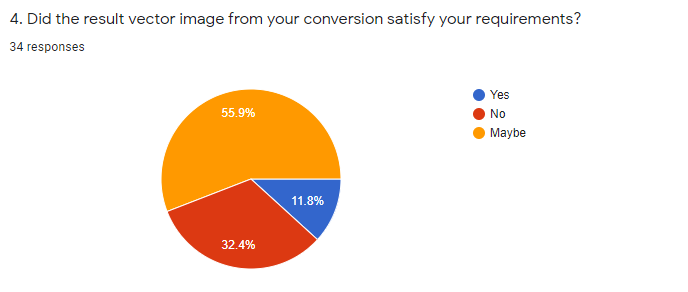
**Conclusion:**

It can be concluded that a majority (32.5%) of the users that use raster to vector conversion tools use it for graphic design and media creation purposes. It can also be observed that 20% of the users use these tools for academic purposes and Some users also use these platforms and tools for GIS Specific purposes.

**Question 4**

**Aim:** Identify if currently available solutions provide a satisfactory result

**Observation:**



The purpose of this question was to determine if the solutions currently available to users of raster to vector conversion software satisfy their needs of providing a satisfactory result of conversion between these two formats.

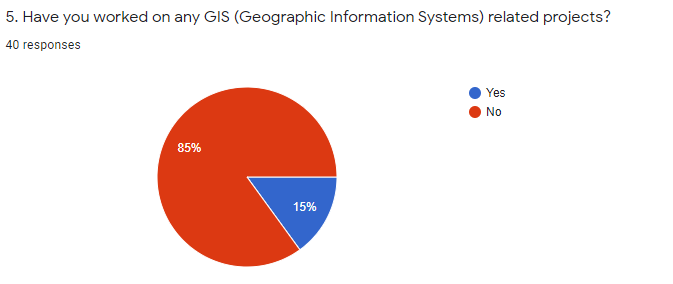
**Conclusion:**

A majority (55.9%) of the users were not sure if they were satisfied with their result. But we can also see that almost a third (32.4%) of the users were clearly not satisfied with the result that the conversion software produced while only a 11.8% of the users were satisfied with their results. We can conclude that there can be room for improvement in conversion methods and therefore is a good impact on the research being carried out.

**Question 5**

**Aim:** Identify percentage of users who have worked in GIS related project or activities

**Observation:**



The purpose of this question was to determine the percentage of the users that provided feedback on this questionnaire who had engaged or worked in any GIS related activities.

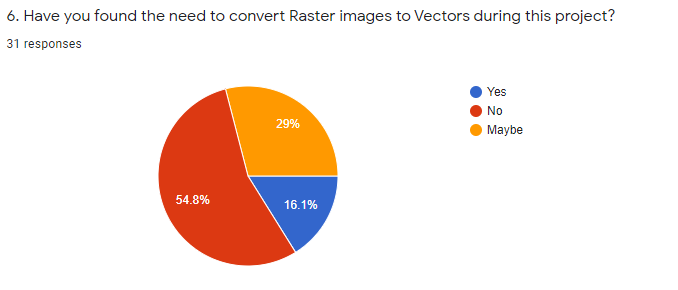
**Conclusion:**

We can see that only 15% of the users who responded to this questionnaire have worked in any GIS related project or activities.

**Question 6**

**Aim:** Identifying the need for raster to vector conversion during GIS projects

**Observation:**

The purpose of this question is to identify if raster to vector conversion is necessary when conducting GIS related projects.

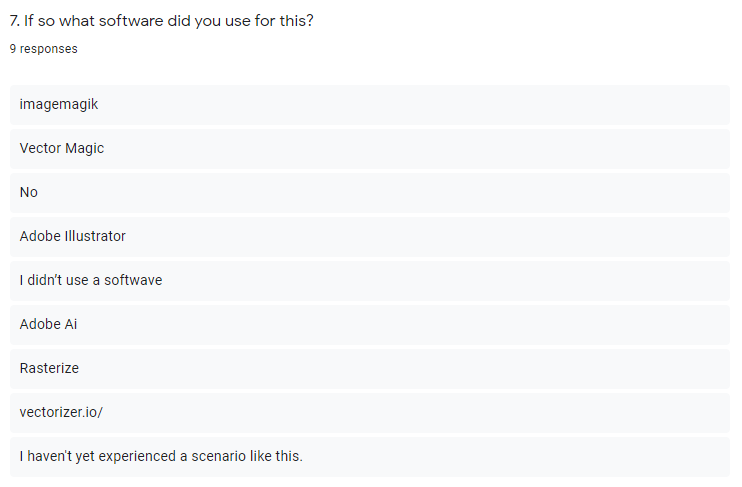
**Conclusion:**

We can see that though the majority of users did not require a Raster to vector conversion software there were 16.1% results that said it was required during their GIS tasks. In conclusion we can understand that even though raster to vector conversion is not a process that is always required in GIS related tasks, there are instances that require tools that carry out this function.

**Question 7**

**Aim:** Identify what commercial soft wares are used by the target demographic of this research questionnaire.

**Observation:**



The purpose of this question was to get a better understanding on the currently used solutions for raster to vector conversion by the users who provided feedback to this questionnaire.

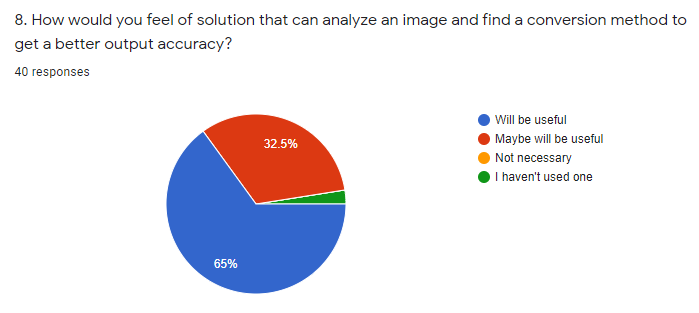
**Conclusion:**

It is clear that there are several soft wares that can be identified through the feedback obtained that perform raster to vector conversion as one of its functionalities or development entirely to do so. Almost all of these software has been analyzed and covered under the similar solutions analysis in the Literature review section. And can be concluded that their strengths and weakness have been taken into consideration when implementing this project.

**Question 8**

**Aim:** To get feedback on reaction to the research idea of this project.

**Observation:**



The purpose of this question is to get the reaction of the users who provided feedback through this questionnaire to the solution proposed in this project.

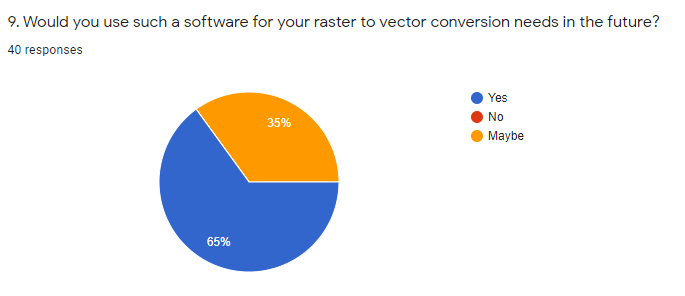
**Conclusion:**

When observing the feedback obtained to this question it can be identified that majority of the feedback (65%) which is more than half the feedbacks obtained are in favor of such a software being developed and assume that it will be useful to them. 32.5% of the feedback also state that may find it useful while no feedback says that this solution will be not necessary contributing very well to the direction of this research and its progress.

**Question 9**

**Aim:** Understand the user reaction towards the system proposed

**Observation:**



The purpose of this question is to understand if the users who provided feedback to this questionnaire would use the proposed solution in this research to perform their raster to vector conversion tasks, when necessary.

**Conclusion:**

When analyzing the observation of the results obtained through this questionnaire, 65% pf the feedback obtained are in favor of using such an implemented solution for their future raster to vector conversion needs and 35% of the feedback of the users are willing to try it as an option in their conversion necessities. Therefore, it is concluded that this greatly supports the research idea of this project and affects this project in a greatly favorable manner.

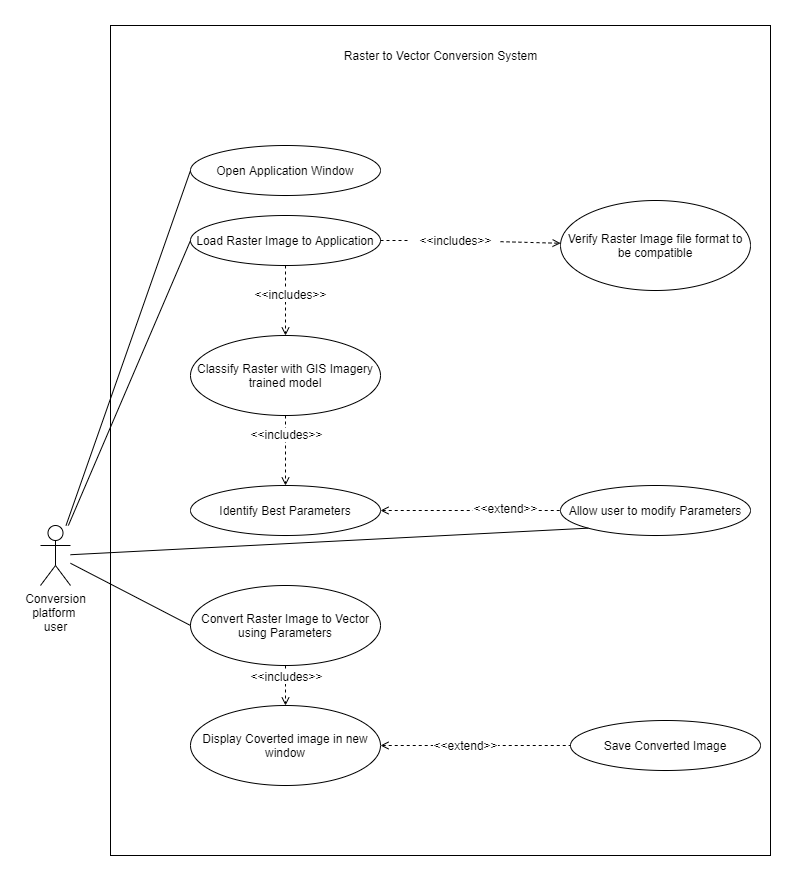
|  |  |
| --- | --- |
| **Method 2: Observations** | |
| Observation of existing solutions proprietary and open source available in this domain, and to a find a unique gap that can be addressed and solved in this research. | |
| **Advantages** | * Identify short comings of current solutions |
| **Disadvantages** | * Observation varies according to observing individual |
| **Execution:** The author is reviewing existing systems and their functionalities, to identify their strength and weaknesses to address them in this research | |

|  |  |
| --- | --- |
| **Method 3: Literature Review** | |
| A widely used method in the research community when it is necessary to gain knowledge on a domain or various techniques and technologies currently being used is by conducting a Literature review on existing material. Research repositories such as IEEE, Science Direct and Google Scholar can be used for this purpose. Hence by studying this material a research gap can be identified in current system to be addressed. | |
| **Advantages** | * Identify short comings of current solutions |
| **Disadvantages** | * Observation varies according to observing individual |
| **Execution:** A literature review has been conducted using the reference of research papers and documents found from the research libraries mentioned. The literature found has been categorized as Image Processing, GI systems and Raster to Vector Conversion related topics. This section is addressed in the second chapter of this thesis document. | |

## Use Case Diagram and Description

### Use Case Diagram

The following is a diagram that represents the use cases of the system visually.



### Use Case Description

The following table will further elaborate the use cases shown in the above diagram.

|  |  |
| --- | --- |
| **Use Case ID** | **UC-1** |
| **Use Case Name** | **Open Application Window** |
| Priority | High |
| Actors | Conversion Platform User |
| Description | Open the application and initialize the system |
| Pre-condition | Application package must be downloaded and extracted onto machine running a windows OS |
| Extending Use Cases | none |
| Including Use Cases | none |
| Triggering Event | Click on Application icon from windows file explorer |
| Main Flow | 1. Open Application 2. Perform Initialization functions to prepare application |
| Alternative Flow | none |
| Exceptional Flow | none |

|  |  |
| --- | --- |
| **Use Case ID** | **UC-2** |
| **Use Case Name** | **Load Raster Image to Application** |
| Priority | High |
| Actors | Conversion Platform User |
| Description | Load a valid raster type image file onto application for processing and conversion |
| Pre-condition | Application must be opened and initialization functions should have been executed |
| Extending Use Cases | none |
| Including Use Cases | * Verify Raster Image file format to be compatible * Classify Raster with GIS imagery trained model |
| Triggering Event | Click on open file from application window |
| Main Flow | 1. Open file explorer window to browse and locate file 2. Once file is selected load file into application as an array or matrices for processing and classification |
| Alternative Flow | none |
| Exceptional Flow | none |

|  |  |
| --- | --- |
| **Use Case ID** | **UC-3** |
| **Use Case Name** | **Verify Raster Image file format to be compatible** |
| Priority | High |
| Actors |  |
| Description | Verify if the file format of the loaded file is one of the compatible file formats accepted by the system |
| Pre-condition | Application must be opened and initialization functions should have been executed and a file must be selected for the conversion process |
| Extending Use Cases | none |
| Including Use Cases | none |
| Triggering Event | Click on open file from file browser window |
| Main Flow | 1. Load file onto the application 2. Check if the file belongs to one of the valid types 3. Display file loaded successfully message |
| Alternative Flow | 3. If File is not one of the valid types show  failed to load file message |
| Exceptional Flow | none |

|  |  |
| --- | --- |
| **Use Case ID** | **UC-4** |
| **Use Case Name** | **Classify Raster with GIS imagery trained model** |
| Priority | High |
| Actors | Conversion Platform User |
| Description | Classify image with trained classification model to aid automated identification of best fit conversion parameters for that particular loaded image |
| Pre-condition | Image must be loaded onto the application and be of a valid file format type |
| Extending Use Cases | none |
| Including Use Cases | Identify Best Parameters |
| Triggering Event | Click on Analyze button on application |
| Main Flow | 1. Run classification model and classify image 2. Return classification accuracy data to find best match of GIS image classification |
| Alternative Flow | none |
| Exceptional Flow | none |

|  |  |
| --- | --- |
| **Use Case ID** | **UC-5** |
| **Use Case Name** | **Identify Best Parameters** |
| Priority | High |
| Actors | Conversion Platform User |
| Description | Use classification data to identify best fit conversion parameters for the image that was loaded |
| Pre-condition | Image classification must be performed on the loaded image |
| Extending Use Cases | Allow user to modify parameters |
| Including Use Cases | None |
| Triggering Event | System triggered after classification of image |
| Main Flow | 1. Use parameter and classification data to identify best parameters for analyzed image 2. Return parameters for the conversion 3. Display parameters and allow editing to the user |
| Alternative Flow | none |
| Exceptional Flow | none |

|  |  |
| --- | --- |
| **Use Case ID** | **UC-6** |
| **Use Case Name** | **Convert Raster image to vector using parameters** |
| Priority | High |
| Actors | Conversion Platform User |
| Description | Convert Image using raster to vector conversion library using the parameters defined |
| Pre-condition | Conversion parameters for the particular image must be set |
| Extending Use Cases | none |
| Including Use Cases | Display converted image in new window |
| Triggering Event | Click on convert button on application |
| Main Flow | 1. Initialize conversion package 2. Convert image using parameters set 3. Display success message for conversion |
| Alternative Flow | none |
| Exceptional Flow | 3. If image conversion fails, due to not  enough memory, catch exception and  handle crashes that might occur |

|  |  |
| --- | --- |
| **Use Case ID** | **UC-7** |
| **Use Case Name** | **Allow User to Modify Parameters** |
| Priority | High |
| Actors | Conversion Platform User |
| Description | Allow the user to adjust parameters obtained for best conversion within a certain range at the users own discretion |
| Pre-condition | Conversion parameters for the particular image must be set |
| Extending Use Cases | none |
| Including Use Cases | none |
| Triggering Event | Edit parameter button clicked |
| Main Flow | 1. Display editable fields for user to change parameters found before conversion 2. Edit parameters 3. Save new edited parameters |

|  |  |
| --- | --- |
| **Use Case ID** | **UC-8** |
| **Use Case Name** | **Display Converted Image in new Window** |
| Priority | High |
| Actors | Conversion Platform User |
| Description | Show preview of converted image to user |
| Pre-condition | Image conversion must be executed |
| Extending Use Cases | Save Converted Image |
| Including Use Cases | none |
| Triggering Event | System triggered after image conversion |
| Main Flow | 1. Open new preview window 2. Show converted image in window |
| Alternative Flow | none |
| Exceptional Flow | none |

|  |  |
| --- | --- |
| **Use Case ID** | **UC-8** |
| **Use Case Name** | **Display Converted Image in new Window** |
| Priority | High |
| Actors | Conversion Platform User |
| Description | Save converted image |
| Pre-condition | Preview window of converted image must be open |
| Extending Use Cases | none |
| Including Use Cases | none |
| Triggering Event | System triggered after image conversion |
| Main Flow | 1. Open file explorer window 2. Select save location 3. Press save button |

## Functional and Non Functional Requirements

### Functional Requirements

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FR No.** | **Requirement** | **Inputs** | **Process** | **Outputs** | **Priority** | **Use case** |
| **1** | Convert Image from Raster to vector | Raster Image | Convert Raster to vector from identified parameters | Vector Image | Critical | UC-6 |
| **2** | Classify GIS image with trained model | Raster Image | Use image classification model trained to identify types of GIS imagery and classify image | Image  classification score | Critical | UC-4 |
| **3** | Identify conversion parameters | Classification score | Use classification score to find best fit conversion parameters | Best fit parameters | Critical | UC-5 |
| **4** | Review output vector image |  |  | Vector Image | Critical | Do tom. |
| **5** | Change auto selected parameters and redo conversion process |  | Convert Raster to vector from manual parameters |  | Critical | Do tom |
| **6** | Save converted file to local or cloud storage |  | Save file to storage |  | Important | Do tom |

Table 5‑1 Functional Requirements

### Non-functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Requirement title and description** | **Specification** | **Priority** |
| 1 | Give user proper feedback on conversion process as it can be a long and time consuming process depending on the raster image and conversion parameters | Usability | Desirable |
| 2 | Result image should be visually similar and accurate to input raster image | Accuracy | Important |
| 3 | Develop API to allow conversions using online platforms | Usability | Desirable |

Table 5‑2 Non Functional Requirements

## Chapter Summary

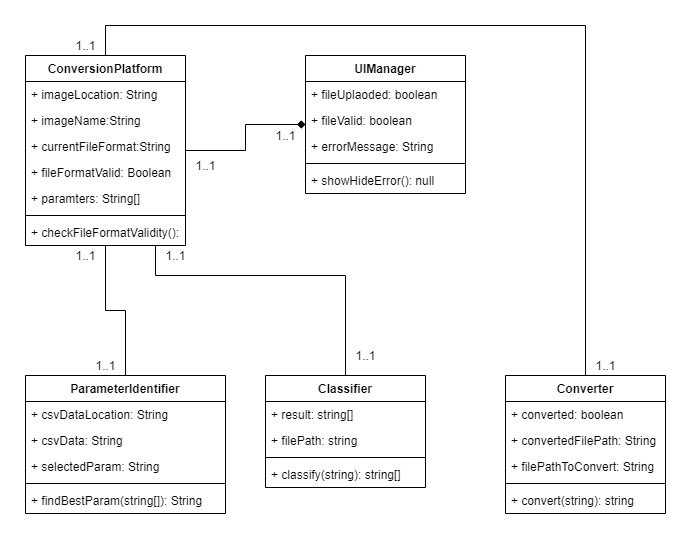
As chapter summary, first the stakeholders were identified and their roles were defined. After defining the roles, the requirement elicitation was carried out mainly with a questionnaire and a literature review. The outcomes of the questionnaire were discussed above with the statistics. Then the use case diagram of the system with the use case descriptions were discussed. After the use case diagram, the functional and non-functional requirements were identified.

# Chapter 6: Design

## Chapter Overview

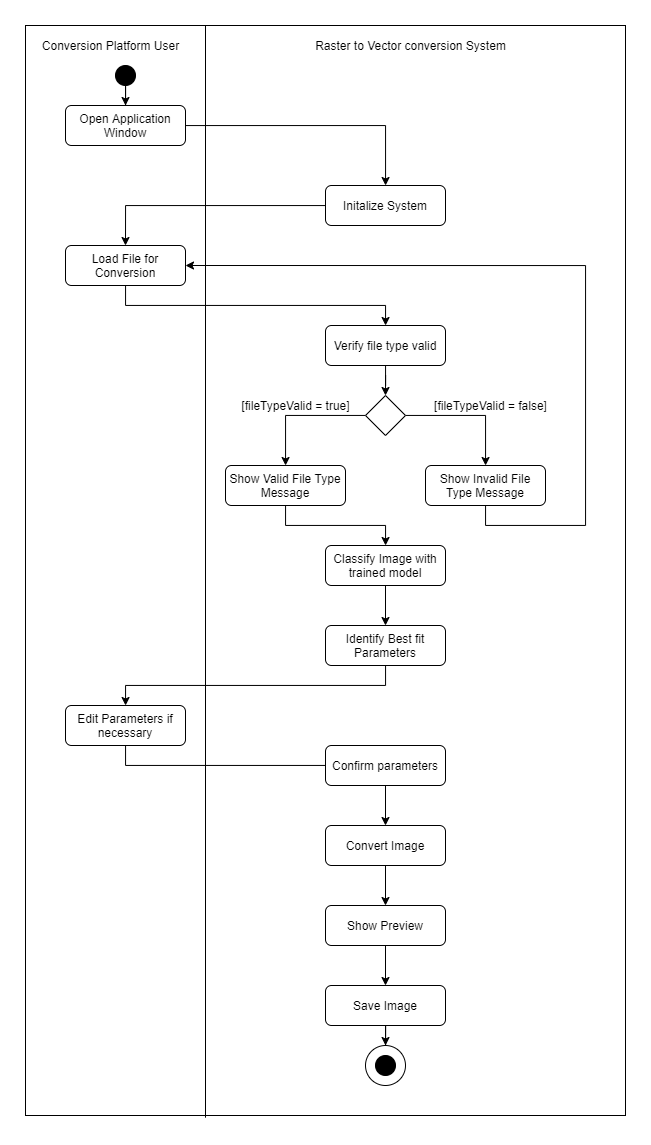
In this chapter, the class diagram, activity diagram and sequence diagram of the proposed solution will be elaborated and visually represented. Also an overview of all the diagrams will be elaborated at the end of this chapter.

## Class Diagram

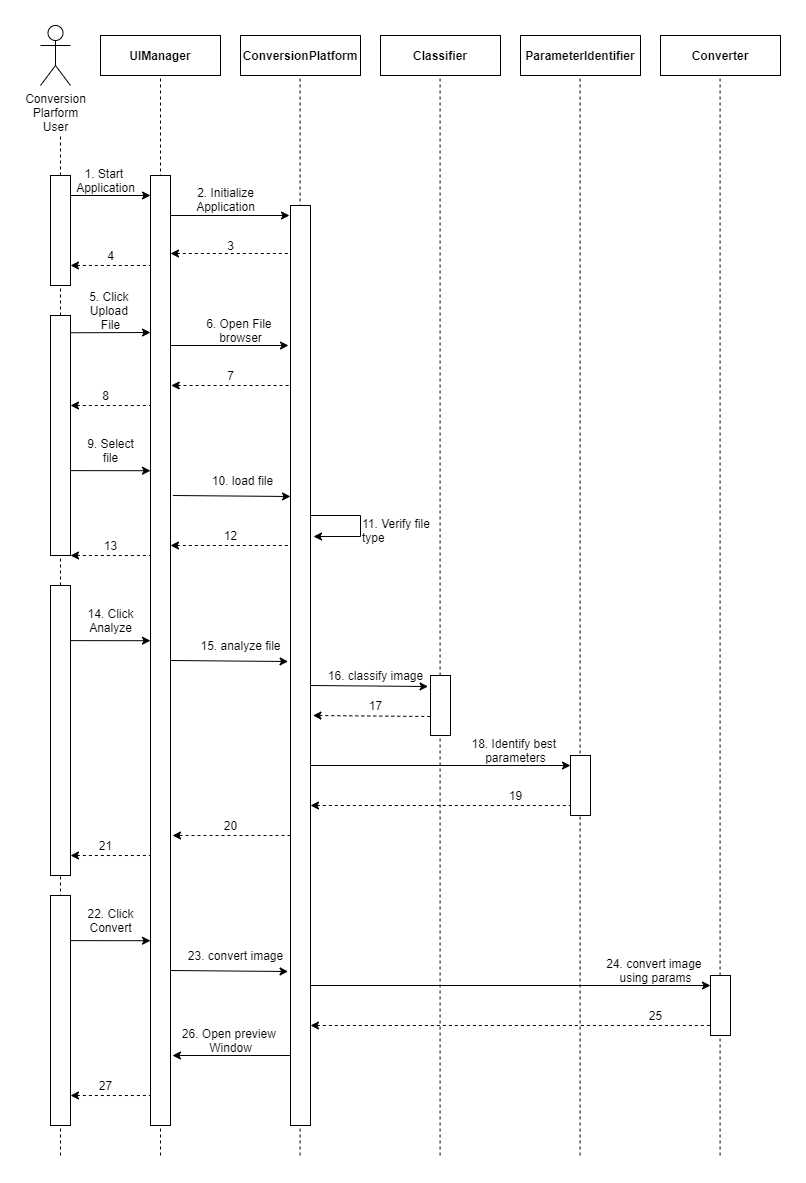
Class diagram below represents the attributes and methods of each class in the proposed raster to vector conversion platform. Each class’s relationship with each other is also shown and also aggregation/composition relationships. 

## Activity Diagram

The flow of activities in the Raster to Vector conversion platform is shown by the activity diagram shown below. This diagram is created according to the flows that were described in the use case diagrams and descriptions. This can be identified as the successful flow of activities for converting an image from raster to vector.



## Sequence Diagram



The diagram above represents how the user interacts with the raster to vector conversion platform. The user starts by opening up the application, which then initializes the conversion platform. Then the user can locate an image file that they wish to convert. Once the file is loaded. It will be verified if it is of the valid file types accepted by the system. If this is the case. The user will be allowed to analyze the file. The conversion platform then uses the image classification trained model to classify the image and get its accuracy as to which classification it belongs to. Using this information then the best fit parameters are identified. Then the user is allowed to click the convert button which then converts the image into a temporary location. The user is also showed a preview of the converted image. This can be considered the sequence followed for a successful image conversion.

## Chapter Summary

The Chapter depicted the design diagrams of the system. Initially the class diagram of the system along with the activity diagram for the main flow of activities in the system was described. A sequence diagram for the passing of messages between the components of the system and how they interact was also depicted.

# Chapter 7: Implementation

## Chapter Overview

This chapter elaborates on the technologies selected for the implementation of this project. This chapter also explains the reasons behind selecting them as the suitable technologies for this project. It also describes the libraries and tools used in this project. The chapter then depicts the High level architecture of the project according to the design specified in chapter 6 of this research document. Finally screenshots of the implemented system are given along with the chapter conclusion.

## Basic Overview of the System

The basic component of the system can be divided into three main sub sections. These are the Main application which gets a raster image as an input, performs all acts of preprocessing on it and identifies the best fit conversion parameters for an output with high structural accuracy. The Image classifier that identifies what type of GIS image the main application is dealing with at the moment and the System that determines the best parameters for raster to vector conversion for a certain classification of images according to the structural accuracy comparison between input and output image.

The core functionality of all three of these components are implemented using the programming language Python. There are several libraries which are being employed to carry out specific tasks in each of these components and some are recurring libraries such as the python machine learning library that handles structural image comparison and the basic image processing libraries that are being employed for reading images and conversion of them into matrices for further operation to be carried out on them.

The User Interface of the Raster to Vector converter is implemented using Electron JS. A UI frame work that employs the power of HTML, CSS and JS to create cross platform desktop applications top of Node.js and Chromium. This will be further elaborated as this chapter continues. Parameters calculated are stored for future access for processing in csv format and are accessed and manipulated using the core language of the application which is Python. The Backend of this application is implemented using python Flask API and will be further elaborated and justified as to its use in the sections below.

## Selection of Technologies

In the next sections the selection and justification for each of the technologies, libraries and tools used in this project will be discussed.

### Selection of Programming Language

As most of the image processing, pre-processing libraries and machine learning libraries are libraries that are implemented on the language **Python (Python v3.6)**, it has been chosen as the language of implementation for the core features of the applications backend. There were also other reasons contributing to the selection of this language as the preferred language to be used in this project. Those are as follows.

* It is multi-paradigm programming language Therefore; it allows different programming approaches. Hence allowing it to follow the OOP design approach.
* Backed up with good documentation and an active and helpful community that is passionate about it.
* Support for wrappers to enable other languages to be integrated to the application. Allowing flexibility in coding certain features.
* Ability to extend functionality with libraries that support various machine learning functions.

### Selection of UI Framework

The UI framework selected for implementing this project is Electron JS. Electron JS is a framework powered by Node.Js and Chormium that allows the implementation of cross platform desktop-suite applications using HTML, CSS and JS. The reasons for selecting electron JS as the preferred framework for implementing the UI component of this project are as follows.

* It is an efficient framework that takes care of tedious parts of developing a desktop application such as providing native menus and notification system, crash reporting system and tools for debugging etc.
* The author has prior experience and knowledge working on a similar framework. Therefore, allowing way adaptation to quickly implement and deploy the front end of the application.
* The framework allows for creative User interface to be created and to be developed in to a user friendly experience with ease.

### Selection of Libraries

#### Scikit-learn (Image quantization)

Scikit-learn is a python library that provides unsupervised machine learning algorithms. It is a library that is built upon other common python libraries such as matplotlib, NumPy and Pandas. Out of the many algorithms that Scikit learn provides, this project will be utilizing the K-Means Algorithm which is a Clustering algorithm which will be used to quantize the colours found in an image to ease in raster to vector conversion and execute other functionalities which ease processing overheads and therefore, reduce processing time of the system.

#### Scikit-Image (Structural Similarity Index)

Scikit-Image is a python Image processing library that is built on other common python libraries such as numPy and sciPy. We will be using the set of methods that are created for comparing structural similarity and obtaining and index between two images of the same dimensions for the purpose of this project.

#### Tensorflow (Keras – Image classification CNN)

Tensorflow is an open source end to end machine learning platform. It supports multiple programming languages including python which is the core programming language used in this project. Its API is also available in multiple versions; this project will be using version r2.1 (stable) to implement the image classification training model that will be deployed using Tensorflow. Tensorflow allows multiple levels of abstraction. Due to this, the project will be utilizing the high level Keras API provided by Tensorflow. The Keras API will be utilized to implement a simple Image processing model using the layers provided to create a Convoluted Neural Network (CNN) to classify GIS related image according to the training data sets obtained.

#### OpenCV

Open Computer Vision Library (OpenCV) is another Image processing library that supports the programming language python among other programming languages that is being used for image processing in this project.

#### Python Flask API Framework

For the Application Programming Interface that is used to expose the backend functionality to the frontend desktop application, FLASK a popular python based API will be used. The advantages of using Flask as the selected API for the project are as follows.

When compared to popular web frameworks such as Django, Rails, Play and Laravel etc. Flask can be considered a significantly minimal web framework. This allow the programmer to implement flask applications in any design structure. In this case OOP based. As OOAD is selected as the preferred Software Design methodology as specified in the third chapter of this document. Flask has been chosen the Framework to be used to implement the backend exposing API in this project.

### Selection of Tools

#### Selection of IDEs

IntelliJ Pycharm is a phenomenal IDE developed by Jetbrains s.r.o. This is a very popular IDE used for Python Software Development. PyCharm also enables the project to be created in its own Virtual environment for which python Conda will be used.

* Existing experience of the author.
* Better intellisense for Python Development provided by the
* PEP8 checks for better code quality and alerts when violated with fix suggestions

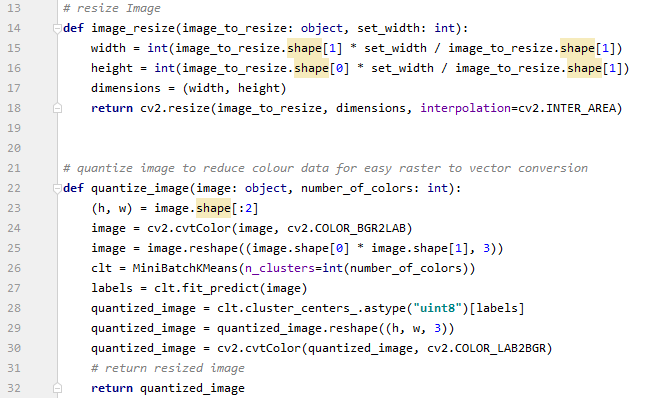
For Frontend Development Visual Studio Code will be used, as it supports code suggestion and formatting support for the mainly used three languages used in the front end framework which are HTML, JS, CSS.

## Implementation of Components

### Parameter Trainer Component

This component handles the pre training of the system to determine the best conversion parameters for each classification of GIS image determined by the image classification model. It contains the methods for various image pre-processing functions that are used throughout the system as well.

#### Image Resizing



The image pre-processing functions that ready the image for the training and ssim functions are a vital part of the whole system. Every time an image is to be imported and converted using open cv. After it is imported it has to be resized to certain dimension for operation such as quantization and structural analysis to be run on them. As some images are too large. They may cause the system to run out of memory. Hence for operations such as image classification model training and Structural similarity index to be calculated on the image data sets it should be scaled down to a constant size. This also helps with training of the image classification model as images with similar dimensions train better with less inconsistencies.

This function gets an image and a width and scales the image height, maintaining its aspect ratio to the newly defined width and returns it.

#### Image Colour Quantization

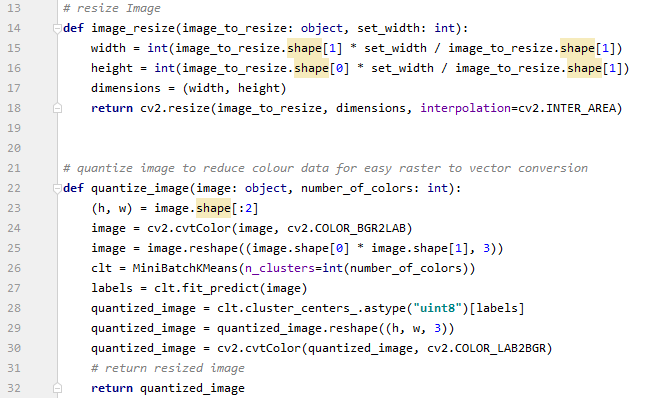


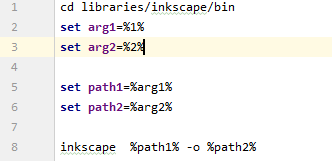
Image quantization is also an important pre-processing functionality that is being employed to reduce the computational power required and time of running of the system when dealing with high quality images with high number of colours that also might make the conversion software run out of memory due there being really high polygon count when raster to vector conversion is to be done. Colour quantization in method is done by using the clustering algorithm found in k-means and passing the number of clusters to be used when quantization of the image. The image should be converted to the LAB colour format before being used to in the clustering algorithm the image after quantization is converted back into the BGR format the default colour space configuration for Open CV. The quantized image is then returned.

#### Converting SVG into PNG



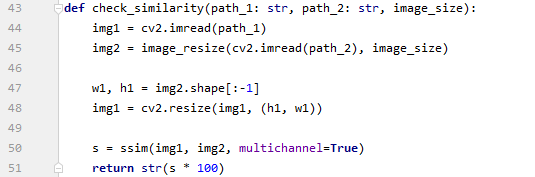
As there is no python library to convert SVG images into PNG format, a library known as inkscape is being used to do this process. The requirement for an SVG image to be converted to PNG comes when the result conversion has to be compared to the original image to identify its Structural similarity as that is the index used to measure the quality of the output in this project. This code calls a sub process call and runs the svg2png batch file which accepts 2 arguments which are the path of the image to be converted and the output path where the converted image will be saved to. Sub process is an inbuilt module in python that allows the spawning of new processes and obtain their return codes.

#### SVG2PNG.bat



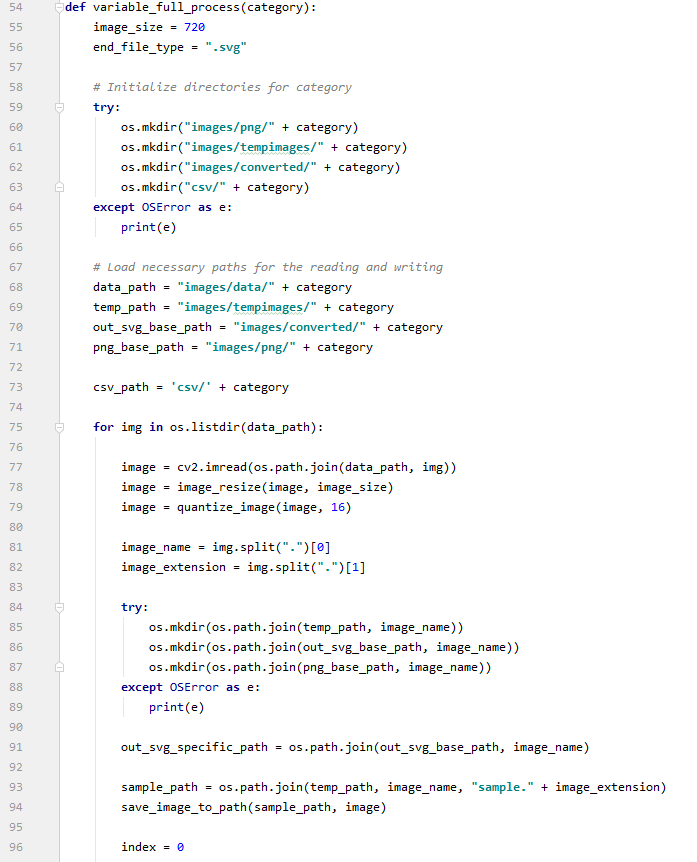
Inkscape is an open source Vector graphics editor. It also has a command line version of its tool and also supports functionalities such as conversion vector data formats to PNG or jpeg formats. This batch file takes the power of the inksacpe command line tool and converts the image that it obtains as the first augment and saves it in the location provided in the second argument. As mentioned above this batch file is invoked though python using sub processes.

#### Image Structural Similarity Index Calculator



Structural similarity index measure (SSIM) calculates the mean squared error (MSE). Structural similarity takes texture of an image to account when comparing the original and to be compared to image. Two images are passed in as parameters where the later image is resized to be the same size as the original image, then using Scikit-images SSIM algorithm the similarity between these two images are calculated and returned as a percentage.

#### Variable Trainer Process





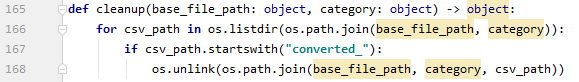
The full process of identifying best fit parameters of each classification of image as classified by the trained image classification model starts by creating and initializing the directories required by the function to store and read images to and from. Once all directories are initialized and the paths to the data set directories are all validated A for loop loops through all the data for the necessary category (images). The image is then resized and quantized using the methods described above. The processed image is then stored in a temporary location which gets overwritten every time the for loop moves into a new iteration. A csv writer then is opened and a csv of the name of the image is created to store the parameters, and similarity of each conversion. Then the code enters a series of nested for loops that iterate each through a single parameters possible values. The temporary image is then converted using these parameters each time. Converted to PNG using the method described above and stored and then the converted PNG and the temp image is compared to identify similarity between them using the method described above. The values are then stored in the csv file and after all the parameter values are iterated through the initial for loops moves on to the next image. After all images in the directory have been iterated through the function is ended.

#### Create Best Fit Parameters for category



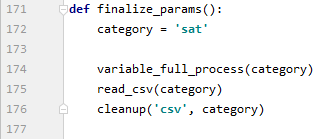
The parameter and ssim csv files are read by a for loop where each is arranged in descending order of their ssim values and the highest ssim value of each parameter file is taken and noted down in a csv file. This csv file then contains the best parameters that were used to get the highest ssim value in all of the images that were used as the data set. In the previous method where the parameter iteration and ssim is carried out. This csv is then saved.

#### Cleanup



Deletes all the temporary csv files created in the process of sorting and creating the final category best fit parameter csv file.

#### Full Training Process Executor

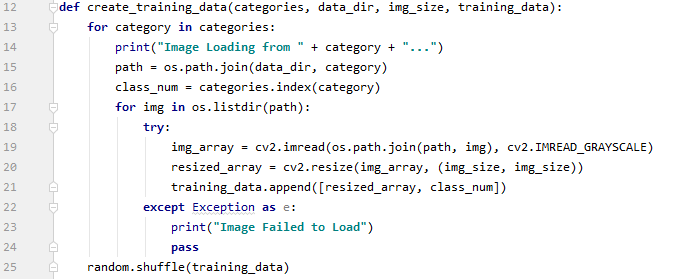


This method runs all of the methods described above in their proper sequence. This process is fully auto mated and the sample space for the best fir parameters can be increased simply by increasing the dataset of images per category. And as this is an automated process. There is no monitoring required by the user of the application to create the best fit parameter files.

### Image Classifier Component

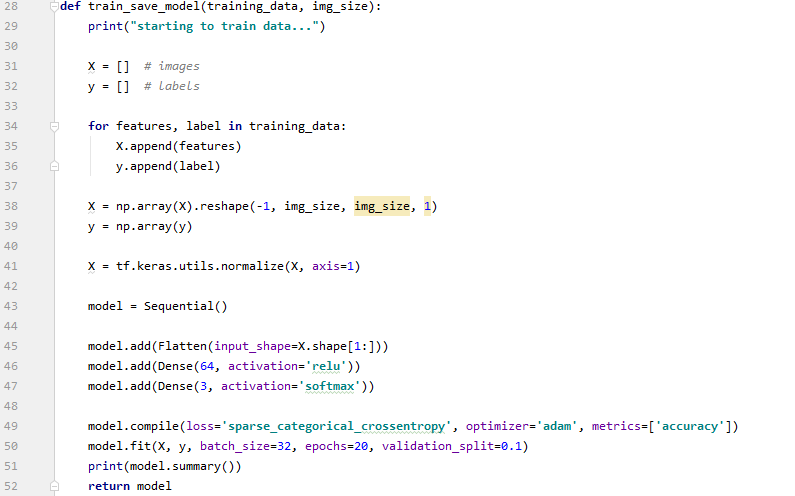
This component handles the classification of various types of GIS data into three main categories. Satellite Imagery, Land classification Images and Scanned maps. This component has three main subcomponents. One that prepares the data and creates the training data set, the Model training component and the component that creates the neural network model with its layers and other parameters.

#### Create and Prepare Training Data



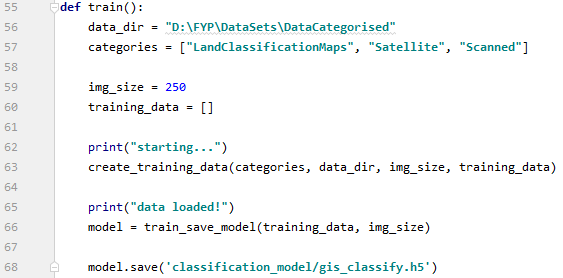
The images from a directory are loaded onto the application and stored in an numpy image array. The images here are loaded in grayscale (2 channels) and are indexed with the category, which is their folder name in which they reside in the data directory. The data is then shuffled to get better test result when training the system as it gets trained to each category randomly without being trained one after the other. This can help with the performance of the trained models accuracy.

#### Training Image Classification Model



Once the prepared training data and image sized that is being used is passed as arguments into this method. It creates two arrays, one with the features (the image) X and the other with their corresponding label y. The X array of images is converted to a numpy array and reshaped using the image size passed as an argument. The values in X are then normalized to obtain better accuracy when training. The model is then created using the **layers** library that is provided by Keras by Tensorflow. In this model, a flatten layers then two dense layers with 64 and 3 nodes respectively are added sequentially the activation function for the first dense layer is **relu** which is a linear activation function that ramps up for any value that is not negative. The second dense layer has a **softmax** activation function and as we have only three categories it has 3 nodes. This model is then compiled using the *optimized* **adam**, *loss function* **sparse categorical crossentropy,** and to obtain metrics of accuracy. The model is then fitted with the data set. Which trains it to classify images between those three categories. And finally returns this model.

#### Training and Saving Model

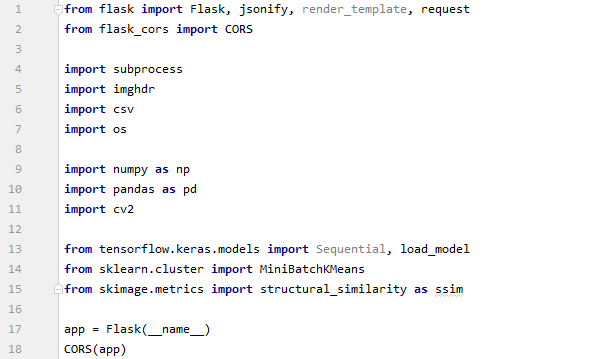


This function calls the above two described methods of this component by passing in the necessary parameters for each function such as the data directory for the data sets, and categories array which is an array of the folder name of the categorized data set which will be used for labelling the images. The image size is also then defined. After the data set is created and the model is trained the model is saved locally to a file of file type h5. It is a Hierarchical Data Format that contain multi-dimensional array of scientific data.

### Backend API

This component exposes the backend functionalities to the front end application through an application programming interface. This API uses the python Flash API framework.

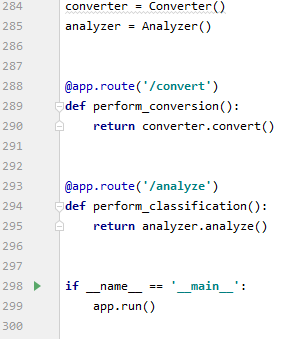
#### Flask API Initialization





This segment of code initializes the Flask application, enables CORS for cross origin access as both these backend and frontend to communicate without the browser blocking it. And starts the application.

#### Flask API method signatures



These two methods are the methods of the API that are exposed through the routes defined and can be called through an http request from the front end. The methods which are being executed here will be further explained below.

#### Analyzer Class



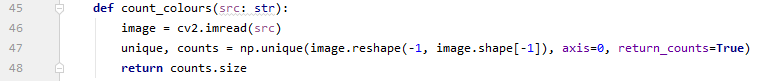
This is a python class that contains the method necessary for the image to be analysed and also for various pre-processing functions to be carried out before analysis, and finally identify the best fit parameters for the input image uploaded by the user. In this section we will not discuss on the methods that are used for quantization, resize and reshaping as it has been discussed precious and the same concept is implemented in this class as well. Instead the unique functions of the analyzer class will be described.

#### Classify Image (Analyzer Class)



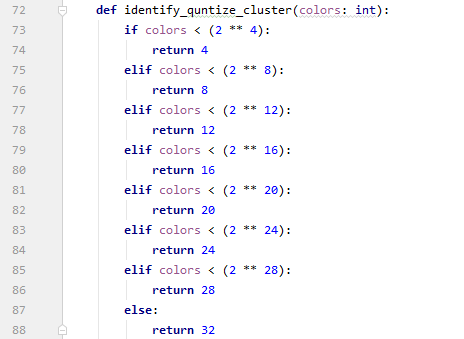
This method uses the h5 file of the model during the classification model training process and classifies an image that is passed into it. And return the type of the image and the category which is assigned by nested if statements that reads the prediction result and decides the classification of the image.

#### Colour Counter (Analyzer Class)



This method calculates the number of unique colours in the image by converting it into a numpy array and counting the number of unique values in each pixel. It is a rough value of the number of colours in the image but can be helpful to understand how to quantize the image without losing a lot of quality.

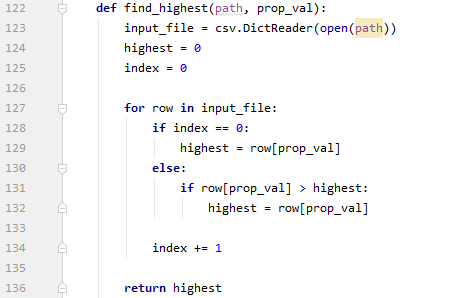
#### Determine Colour Clusters (Analyzer Class)



It is a simple nested If loop to determine the number of colour clusters required for quantization. It calculates this value using the colour counter value that is returned.

#### Get Best Parameters (Analyzer Class)





This method is responsible for determining the best fit parameters range for the category of the image that has been uploaded by the user. This category is a parameter passed into the method and is the classification category determined when the image is classified using the model. The method then reads the best parameter csv regarding that category and determines the highest and lowest values for each of the parameters and determines a range for the best fit parameters to exist in and returns this as an array of tuples.

#### Determining Best parameters for user image (Analyzer Class)

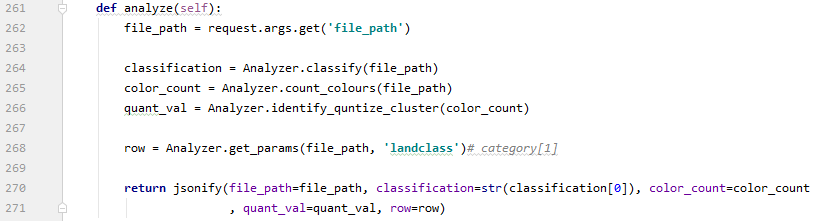






This method is the main function of the Analyzer class which determines the best fit parameters for the image selected by the user. This method too gets the category as a parameter for its operations. It selects the csv file of best parameters for its category. And then after obtaining the best parameter range from the method previous defined it runs conversion on them similar to the parameter trainer component and finally returns the set of parameters with highest Structural similarity index.

#### Analyze (Analyzer Class)



This method combines all the methods defined above and creates the final JSON that will be passed to the front end with all of the parameters and some additional data that will be shown to the user before conversion.

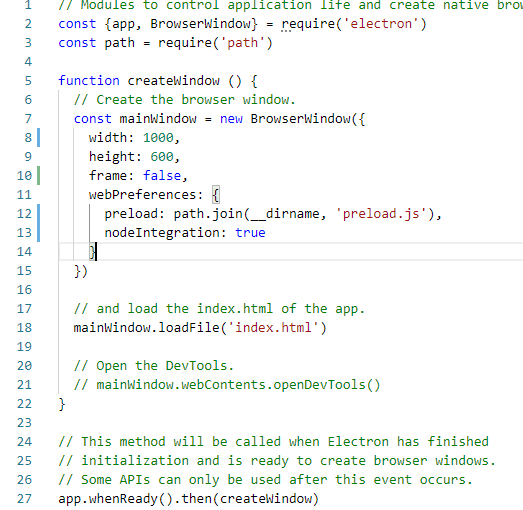
#### Converter Class



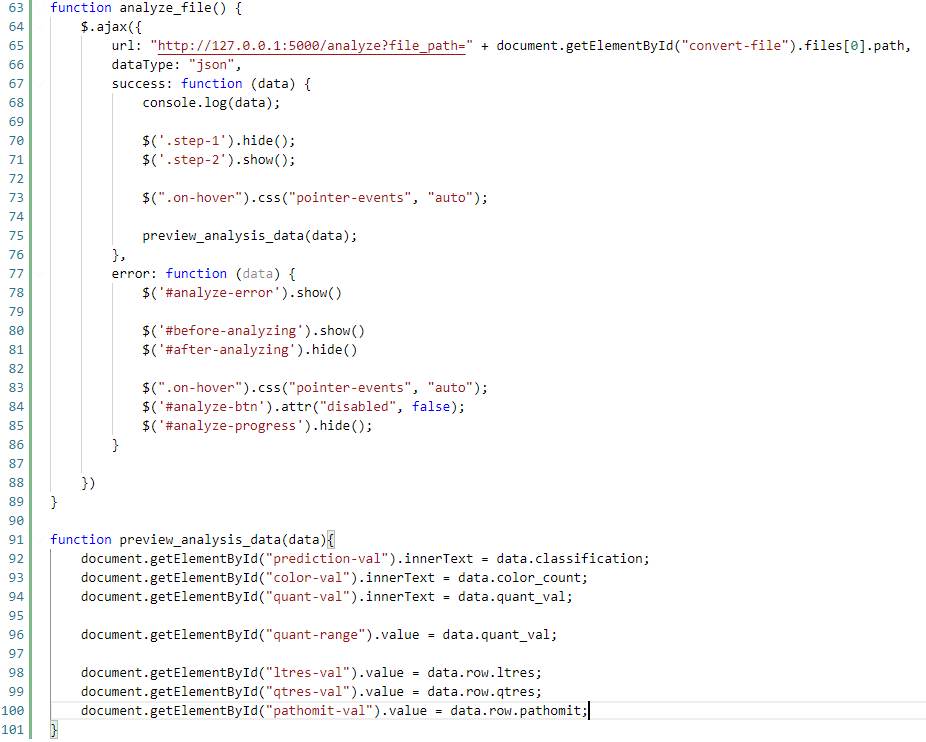
This component is the class that handles the conversion of the image from all the parameters that have been identified using the analyser phase of the applications flow. It obtains the parameters from the URL params of the request and converts the image using the imagetrace library and output the image to the specified location. Once the operation is complete returns a JSON with the success flag to update the UI.

### Frontend

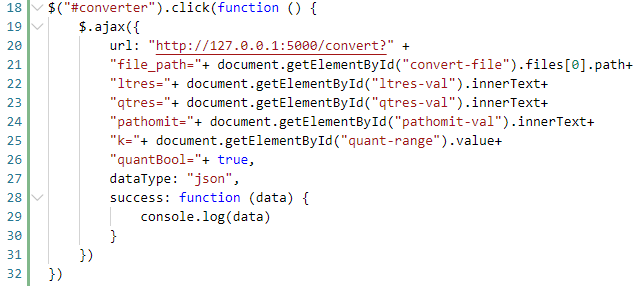
#### Initialization



Electron JS application initialization configuration. The configuration defined here will be used when creating the window for the frontend application which will be initialized.



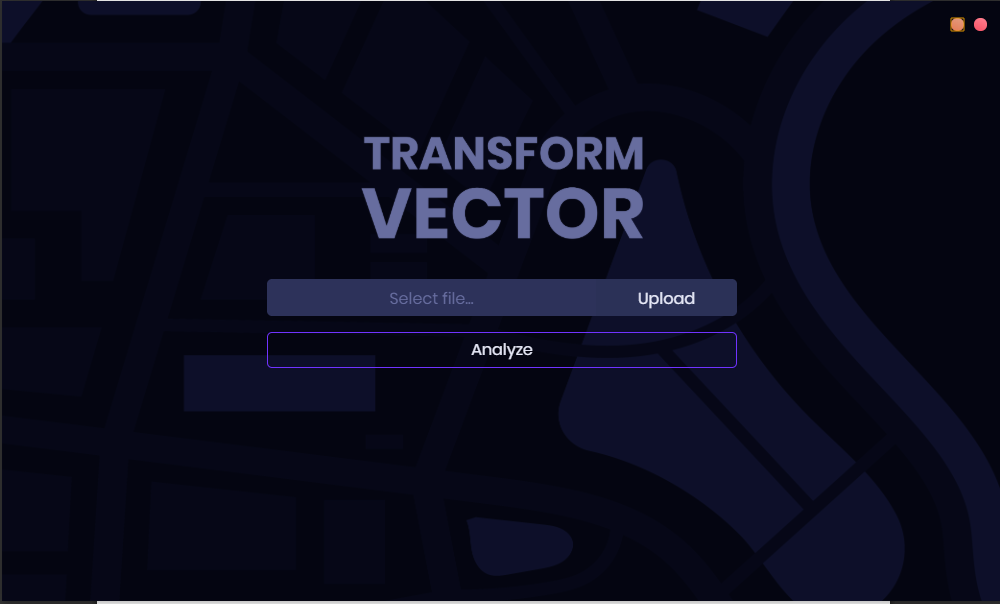
This function sends a request to the backend Flask API and gets the parameters for the file defined in the file path URL parameter. And after a successful response updates the UI with these values preparing the application for the conversion process.

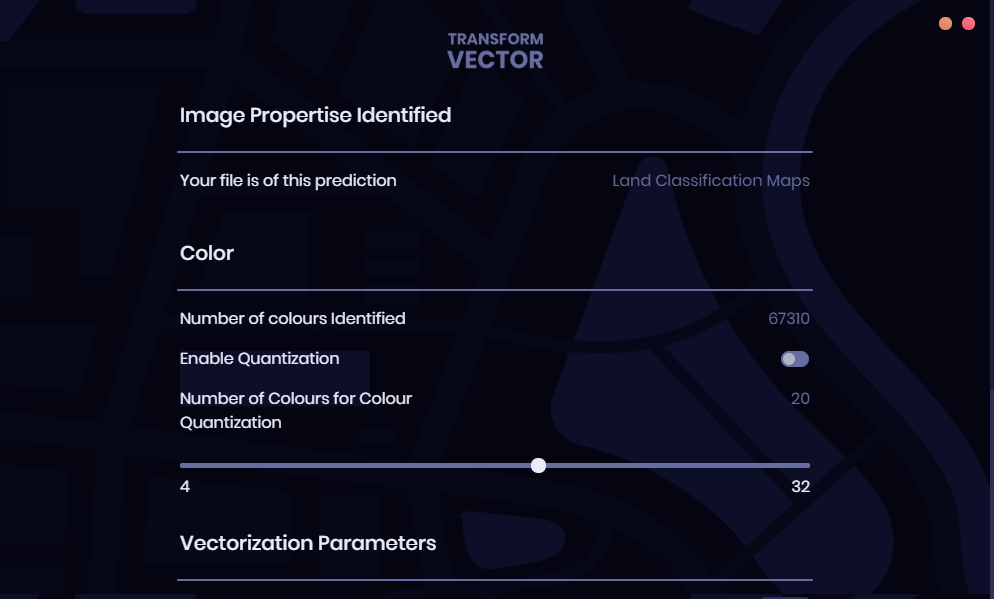


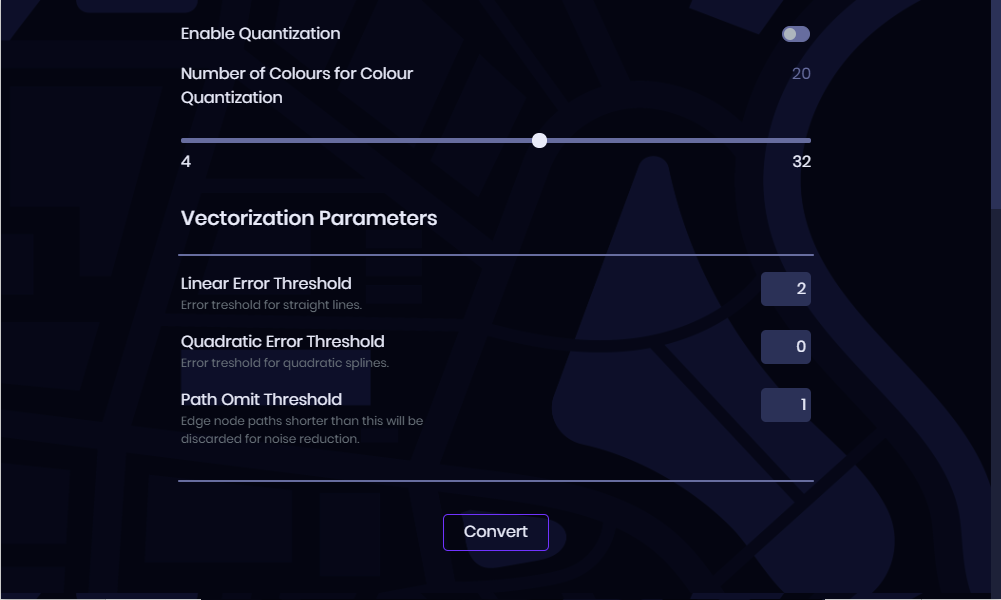
And after converts the image from raster to vector using the parameters found. When triggered by the user.

## Screenshots of the System

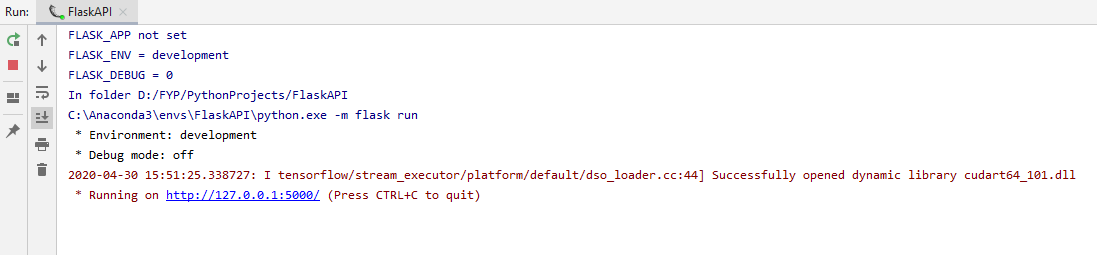
The following screenshots depict the UI developed for the raster to vector conversion application. This is custom UI created using HTML,JS and CSS on the Frontend framework Electron JS



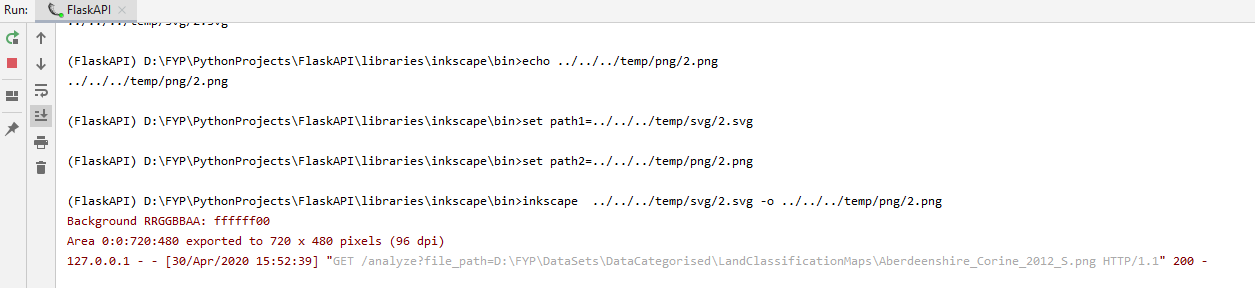




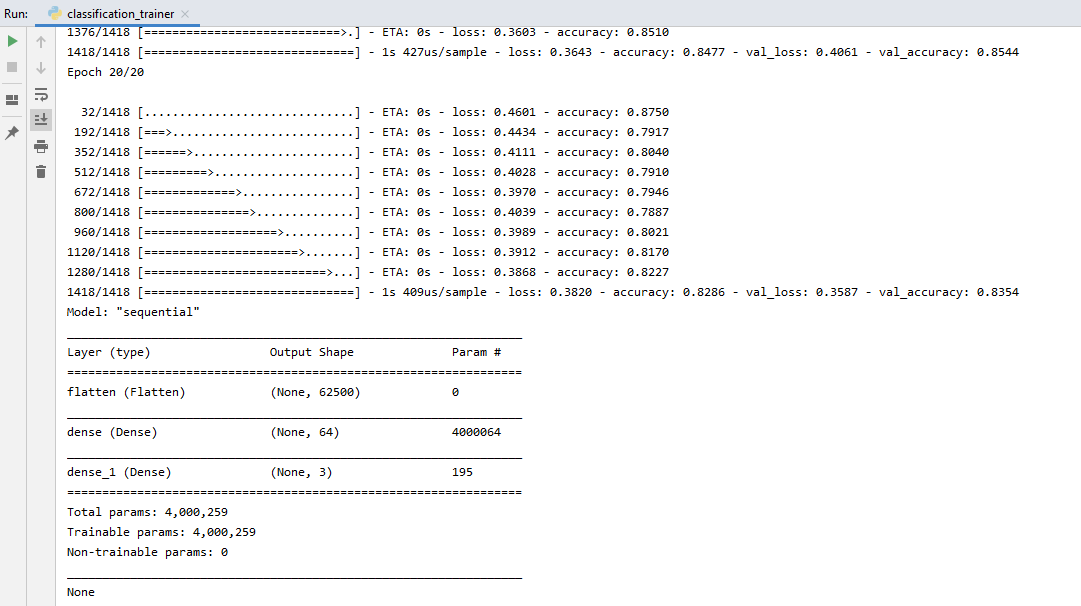
The Figure show the running of the Flask API Backend.



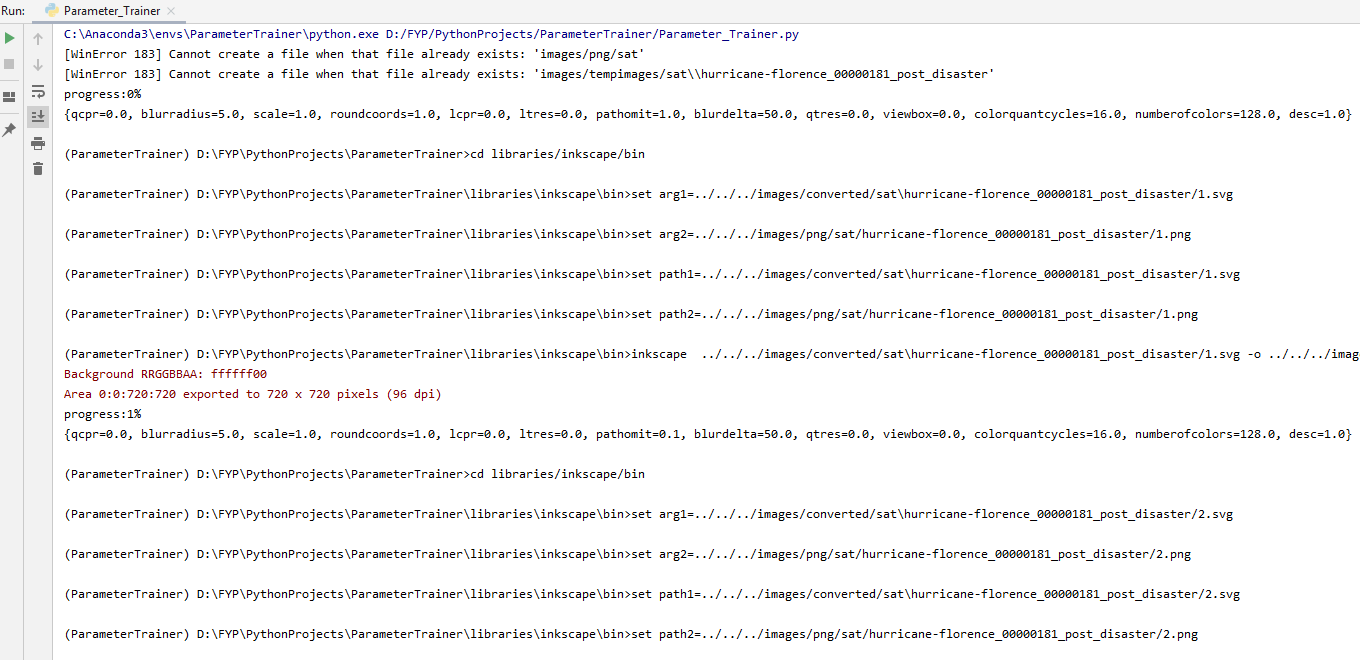
The Figure shows the response log created when responding to a request of the frontend



The figure shows the training of the image classification model



The figure shows the parameter trainer python file being executed to create the best fit parameter csv files from the data set of images per each classification.



## Chapter Summary

Initially the selection of technologies, programming language and tools required for the implementation of the project were discussed. A basic overview of the system was also described. Each component and the functionality that they implement in the system with screen shots of the code was then provided. Finally, to conclude the chapter screen shots of the systems UI and console outputs were depicted.

# C**hapter 8: Testing**

## Chapter Overview

This chapter discusses the testing process overview of the system and its results. The testing is carried out to test if functional and non-functional requirements are met and achieved by the implemented system. Initially, the testing goals and objectives are touched upon. The testing criteria is then mentioned. Black box texting is carried out on the main system and its results will be depicted. The image classification model will also be tested to get its error rate and accuracy. After functional testing is completely discussed. Non-functional testing according to the non-functional requirements mentioned in chapter 5 will also be elaborated on. The limitations of the testing will also be discussed as conclusion to the chapter.

## Goals and Objectives of Testing

This testing phase is carried out to ensure that the system implemented as described in Chapter 7 of this document meet the functional and non-functional requirements that were descried during the requirement specification stage of this project. This testing phase makes sure the following goals are achieved.

* Identifying bugs and fixing them.
* Validate and verify if all the functional requirements have been met by the system.
* Validate and verify if all the non-functional requirements have been met by the system.

## Testing Criteria

As the testing criteria of this testing phase, the following two criteria which are defined below will be used to identify and reduce the gap between the expectation and reality of the implementation.

**Functional Testing** – To test the functional requirements defined

**Non-Functional Testing** – To test non-functional requirements of the system. But elevate the overall experience of using the system.

## Functional Testing