CHAPTER 3

RESEARCH METHODOLOGY

3.1 Research Framework

The aim of this project is to visualize the state of urban green coverage in Malaysian city from the users' perspective with Green View Index (GVI). To achieve the aim, the research project is broken down into several phases that form the research framework.

In Phase 1, problem formulation is carried out to set up the foundation for research. It starts by determining the field of interest and the topic to continue research. It is then followed by literature review where the current state of research is analysed to form background understanding on the research topic and discover research gap. After that, the aim and objectives of the research is formulated and consolidated along with the defined scope of study area, which in this case being referred to the Johor Bahru city center.

After problem formulation, Phase 2 of data preparation is carried out at the beginning of the research project. In the preparation step, data is first collected, then pre-processed according to the need of the research. In the context of this research, the collected image data is converted into RGB format and masked to generate ground truth data for supervised learning.

Subsequently, the prepared data is used for models' development and evaluation in Phase 3. In this phase, Pixel Segmentation model is developed as the benchmark model to compare with the deep learning models developed to predict GVI in terms of accuracy. Finally, the predicted results are used for reporting and EDA in Phase 4. The generated insights are then presented on a dashboard as the final product.

The summarised diagram that displays the research framework is displayed in Figure 3.1 below.

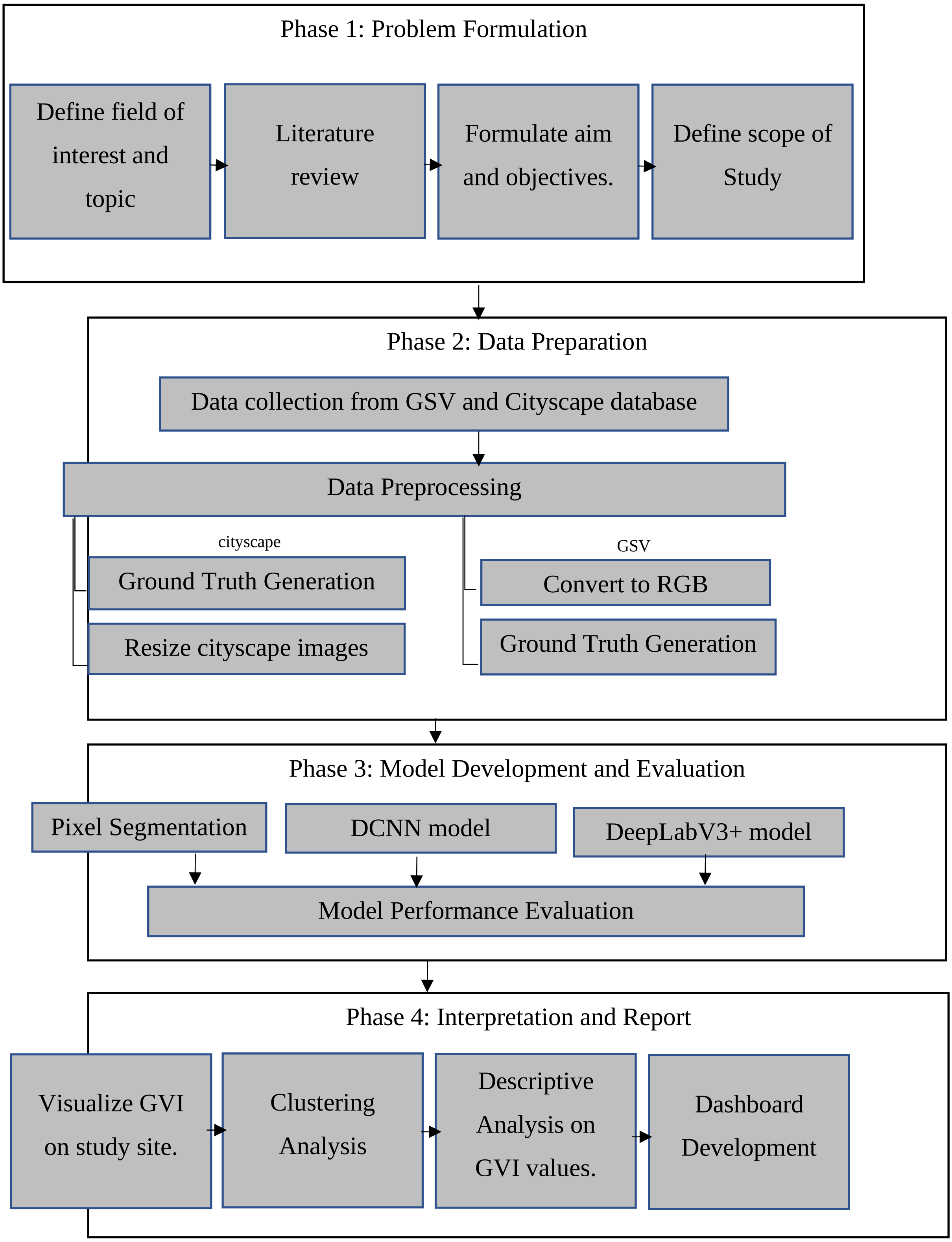

Figure 3.1 Project research framework

3.2 Data Preparation

To compute Green View Index (GVI) in target site, street view images in Johor Bahru City Center are needed. The data preparation for the project can be broken down into 2 parts, i.e. (1) sampling and retrieving street view data and (2) image preprocessing.

Besides, another set of street view data is collected from an open source dataset, Cityscape is also used to augment GSV collected from Johor Bahru City Center to improve the accuracy of model.

3.2.1 Data Collection

3.2.1.1 Sampling Street View Data

To limit the scope and resources required for the project in a manageable range, the methods used sampling of data collection points are improvised to fit within the Google Street View API framework. The image sampling processes are carried out in several steps:

1. **Decide on the study site and its boundary.**

The city center of Johor Bahru is selected as the study site as it has a high level of activity, movement and population density, making it a good representation of a Malaysian city suitable for our study. Due to the lack of official boundaries demarcating the Johor Bahru city center area, an expedient demarcation is made by using *Jalan Lingkaran Dalam* (dashed line) as an edge.

In order to retrieve street views from Google Street View using coordinates, the study site boundary has been defined as a circle with a radius of 1000m centered on the coordinates of Komtar JBCC. This boundary has been chosen based on several trials to ensure that it covers the busiest part of Johor Bahru city center. The boundary of the study site is illustrated in Figure 3.2 below.

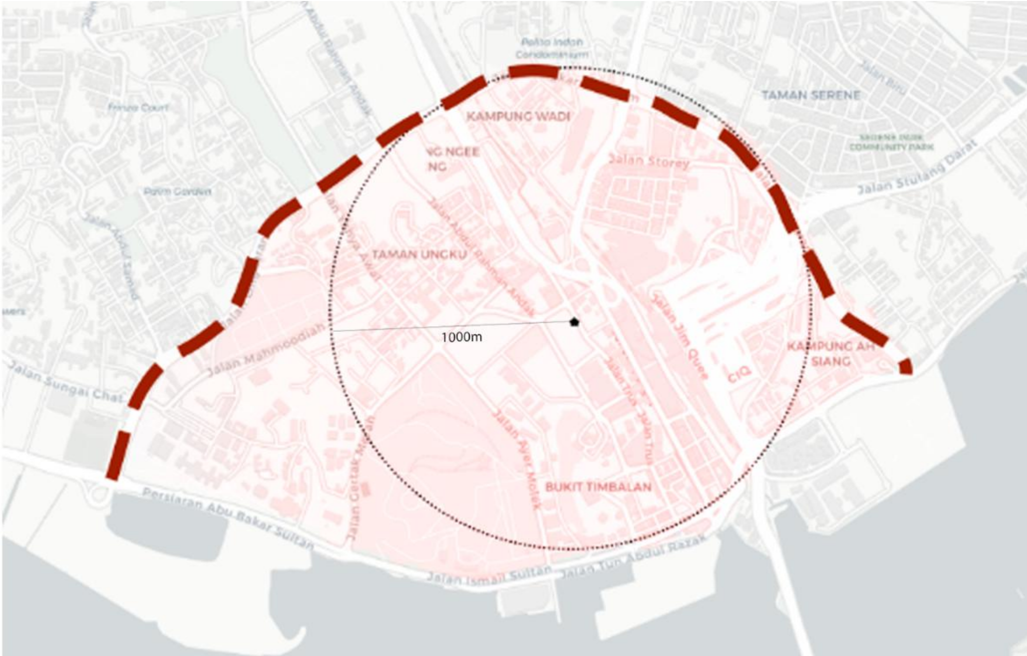

Figure 3.2 Study site plan with boundaries.

2. Sampling random coordinates on streets within the boundary.

The random coordinates are defined by using Pythagoras theorem, following Equation (3.1) to ensure that they are located within the 1000m circular boundary. The center latitude and center longitude represent the latitude and longitude of the center of study site, Komtar JBCC; latitude and longitude represent any random coordinate generated. The radius of the site boundary is converted to degrees of latitude at the equator by dividing it by the approximate distance of 111km per degree of latitude.

$$latitude \leq center\ latitude\ \pm \sqrt{radius^2 - (longitude - center\ longitude)^2} \tag{3.1}$$

Equation (3.1) shows the formula used to limit random sample points within 1000m radius from the canter coordinates *(Komtar JBCC).*

3. **Retrieve GSV images from the sampled coordinates.**

Various GSV parameters discussed in the literature review, e.g. GVI4, GVI, GVI8, GVI16 are explored from the perspectives of zoom, distortion, overlap of images and cost to find out the best configuration to use for our task of GVI computation. The value for each parameter is determined in place in the following sequence.

i. **Field of view**. The field of view of GSV images are set to the minimum of 65 to minimize the proportion of noise included in images collected, e.g. road and sky that are not typically in our field of focused vision unless special attention is given to.

ii. **Headings**. Based on the field of view used, the headings of visuals decide the images retrieved from a single location excluded parts of the view or overlapped with one another. With trial and error, 6 pictures facing different directions of 0°, 60°, 120°, 180°, 240°, 300° to capture every view that one might see while minimizing overlap as illustrated in Figure 3.3.

iii. **Pitch**. A slight pitch of 10 degree upwards is applied to reduce the proportion of road being captured in the image.

The parameters set is shown as follows, with a for loop used to take multiple images of different headings.

```
params = {
        'key': api_key,
        'size': '1280x1280',
        'fov': 65,
        'location': f'{latitude},{longitude}',
        'heading': 0,
        'pitch': 10
    }
```

Figure 3.3 Sample view collected from a sample point at (1.4574946, 103.7690224)

In total, 6385 images from 1065 sample locations were selected. The street view images are saved as RGB format and their metadata, latitude, longitude and heading are stored as a csv file for possible future use. Table 3.1 displays the first few rows of metadata.   The selected sample locations are plotted on a map (Figure 3.4) to visualize the density of sampling.

Table 3.1 The metadata of sampled images.

	lat	Lng	heading
0	1.46143	103.7624	300
1	1.461554	103.7623	0
2	1.461554	103.7623	60

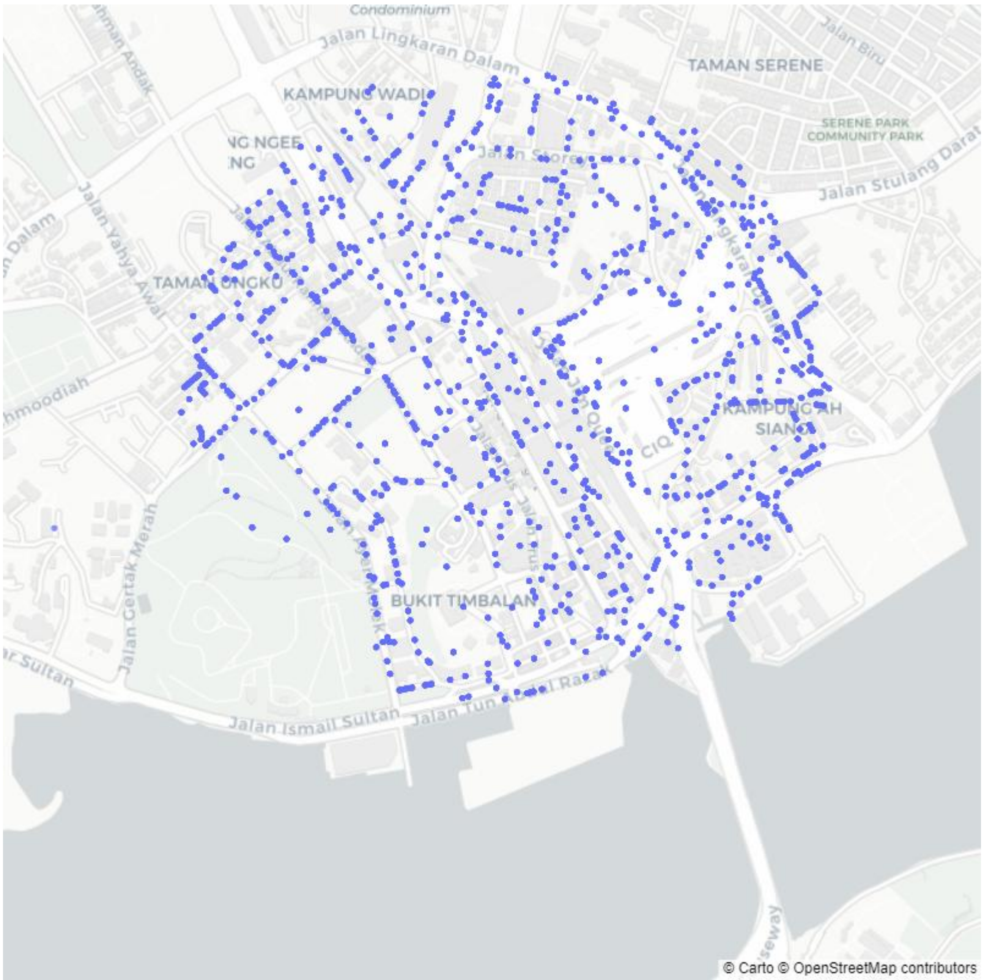

Figure 3.4 Map plot of sampled locations

3.2.1.2 Cityscape Data Collection

Cityscape dataset is downloaded from https://www.cityscapes-dataset.com/downloads/. From the site, two zip files: *gtFine_trainvaltest.zip (241MB) [md5]* consisting of segmented street views labelled data and *leftImg8bit_trainvaltest.zip (11GB)* *[md5]* consisting of street view images are downloaded. Both zip files have complementary image pairs in train (2975 images), val (500 images), test folders (1525 images) respectively. Nevertheless, it is worth noting that the test folder in the segmented street view labelled data are null sets that cannot be used.

The dataset segmentation consists of 30 label classes. In the data processing section, 30 label classes are to be converted to only 2 classes of green view or non-green view. Figure 3.5 shows an example of how Cityscape dataset's street view image and its complementary looks.

Figure 3.5 The street view image and its complementary segmentation label image from Cityscape dataset.

3.2.2 Data Preprocessing

3.2.2.1 Data Pre-processing for GSV

To develop a deep learning model, labeled data is needed. In this case, the data for model development consists of images and the labels correspond to the presence or absence of vegetation in those images. There are two types of labeled data needed:

1. Extract vegetation pixels from the images. To create the labels, binary masks are applied to the images, where vegetation is represented by white pixels and all other areas are represented by black pixels. This helps to simplify the images and make the training process more efficient. A sample is shown in Figure 3.6.

Figure 3.6 A sample of the street view image and its target mask.

2. GVI at every sample point used for model development. To develop models that can accurately predict GVI, GVI calculated from the manually extracted vegetation pixels are used as the ground-truth. GVI is calculated with Equation (3.2).

$$GVI = \frac{\sum_{i=1}^{n}\sum_{i=1}^{M} Area_g}{\sum_{i=1}^{n}\sum_{i=1}^{M} Area_t} \tag{3.2}$$

Equation (3.2) shows the formula to calculate GVI.

At the end of the data preprocessing, input data for model development is prepared: street view images represented in *Numpy* array format (training data), the true GVI of each street view image used for training and testing (target data).

3.2.2.2 Data Pre-processing for Cityscape Data

The preprocessing for cityscape data can be broken down into steps below:

Firstly, load all training data and validation data from *gtFine_trainvaltest.zip (241MB) [md5]* (segmented street views labelled data) and *leftImg8bit_trainvaltest.zip (11GB) [md5]* (street view images) as *NumPy* array. Then, transform the array into RGB format by extracting only the first 3 channels. Both resulted *NumPy* arrays should have the shape of (number_of_images, height, width, number_of_channels), which in this case are (number_of_images, 512, 1024, 3) and (number_of_images, 512, 1024, 3).

To appropriate cityscape data for GVI computation, a change of the multiclass segmentation into binary segmentation of green view and non-green view is needed. It is done by determining the unique colours in the label images, then change values of the colour pixels with labels representing green view ('vegetation' and 'terrain') to [1,1,1] and other pixels to [0,0,0]. The generated binary segmentation data is then converted into array with only 1 channel with shape (number_of_images, 512, 1024) to save memory and makes GVI computation more convenient.

Finally, the arrays representing street view images and binary label data is clipped into a square shape of size (512, 512) so that it has the same square layout as the GSV images. GVI of the images are generated as the ground truth by summing up the values in the binary label data array divide by (512*512).

3.2.3 Splitting Data for Model Development

In the *Treepedia* study published by Cai et. al, 320, 80, and 100 google street view images are used for training, validation and testing respectively. This serves as a benchmark for the number of images used for model training in this project. As the model deployed was published on github: billcai/treepedia_dl_public: Treepedia 2.0: Deep Learning Based Large

Scale Quantification of Urban Canopy Cover (github.com), we can use transfer learning to train the model for the project with relatively fewer data.

Thus, in this project, a slightly lower number, 280 GSV images are used for training to reduce the training time, while 80 and 80 images are used for validation and testing respectively due to the large amount of time needed for manual data labelling for ground truth generation.

As for the cityscape dataset, 2975 images are used for training and 500 are used for validation. As it is only used for model pretraining, test data is not required.

3.3 Model Development for GVI Prediction

In this project, the classical Pixel Segmentation method proposed by Li et. al., 2017 is benchmarked against two supervised deep learning models DCNN end-to-end model and DeepLabV3+ model.

The Pixel Segmentation model is used as the benchmark as it is the most widely adopted approach used to compute GVI of street view images. Besides, compared with supervised deep learning models, the Pixel Segmentation approach comes with several major benefits, e.g. greater interpretability, less susceptible to human induced error in the data preprocessing stage, no need of time and computation resources for model training.

Due to the benefits mentioned above, deep learning models need to predict GVI with higher accuracies to show that they worth more than what they cost.

3.3.1 Pixel Segmentation Method

Firstly, separate bands of red, green and blue are extracted from the RGB images. As we wish to find pixels that have higher green band values than the other two bands, differences are computed by subtracting the red band and blue band from the green band respectively. If both differences are larger than 0, they are multiplied to form a new Difference image. The generated Difference image is a grayscale mask that represents the number of differences between the green band of a pixel with the other two bands. The lighter the pixel or the higher its value, the 'greener' the pixel appears in the original picture and vice versa.

To eliminate noise, a threshold is assigned such that selected pixels with value larger than threshold are assigned to value of 255 and those below assigned 0. The outcome classifies the selected pixels as vegetation, while the others as not vegetation. After this, the GVI of an image can be easily computed from the classification outcome with Equation (3.2). Figure 3.7 simplifies and illustrates the whole process.

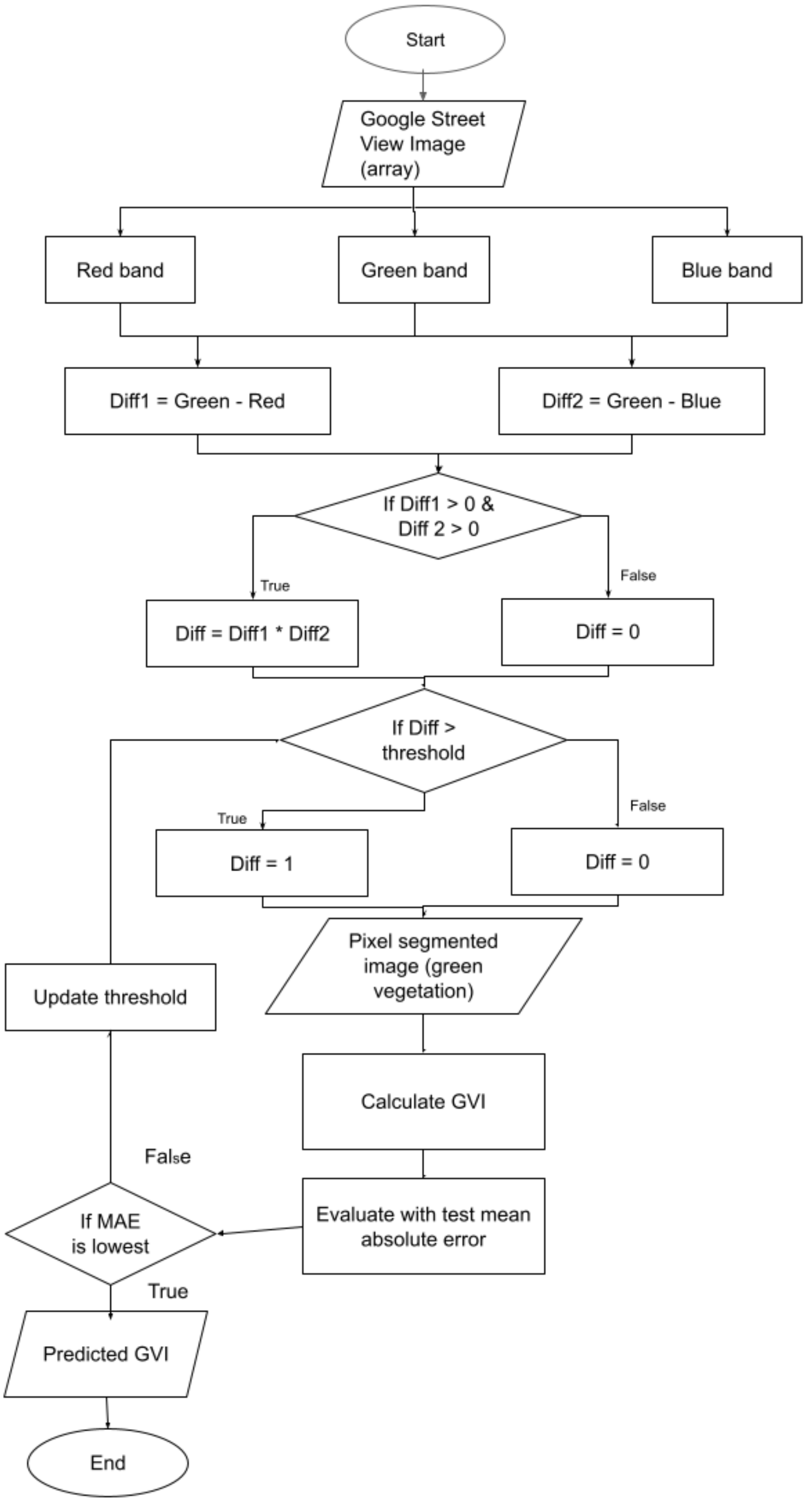

Figure 3.7 The process of Pixel Segmentation for green vegetation extraction and GVI calculation.

3.3.2 Deep Learning Models

Deep learning models developed and studied in this project are supposed to be the models shown to have the highest performance in the literature review, i.e. DCNN end-to-end model with mean absolute error of 4.67% and HRNet-OCR with mean IOU of 80.6%. Nevertheless, due to the issues and complexity to retrieve HRNet-OCR model from the suggested repository, the DeepLabV3+ model with the mean IOU of 78.37% is used to predict GVI.

The DCNN end-to-end model is downloaded from the Github repository uploaded by Cai: billcai/treepedia_dl_public: Treepedia 2.0: Deep Learning Based Large-Scale Quantification of Urban Canopy Cover (github.com), while DeepLabV3+ structure is set up with code provided by Tensorflow keras site: https://keras.io/examples/vision/deeplabv3_plus/. Nonetheless, as the output of DeepLabV3+ model is pixel-segmented image which is different from that of GVI prediction generated from DCNN end-to-end model, several Dense layers are connected to the output of DeepLabV3+ model, with the last layer of neuron having a sigmoid function as activation function to change the output to GVI values between 0 and 1 so that both deep learning models can be compared on the same level.

The trained DCNN model was pre-trained on ImageNet dataset, then trained on Cityscapes dataset, and finally on 320 GSV images collected by Cai et. al. For DeepLabV3+ model, pretrained ResNet50 retrieved from Keras is used as the backbone of its encoder module and trained on the preprocessed cityscape data. Then transfer learning is done on both pre-trained models (DCNN end-to-end and DeepLabV3+ model) by fine-tuning it based on local street view images collected. The trained model is then used to directly predict the GVI values of GSV images. Figure 3.8 showcases the process to develop deep learning models to predict GVI in this project.

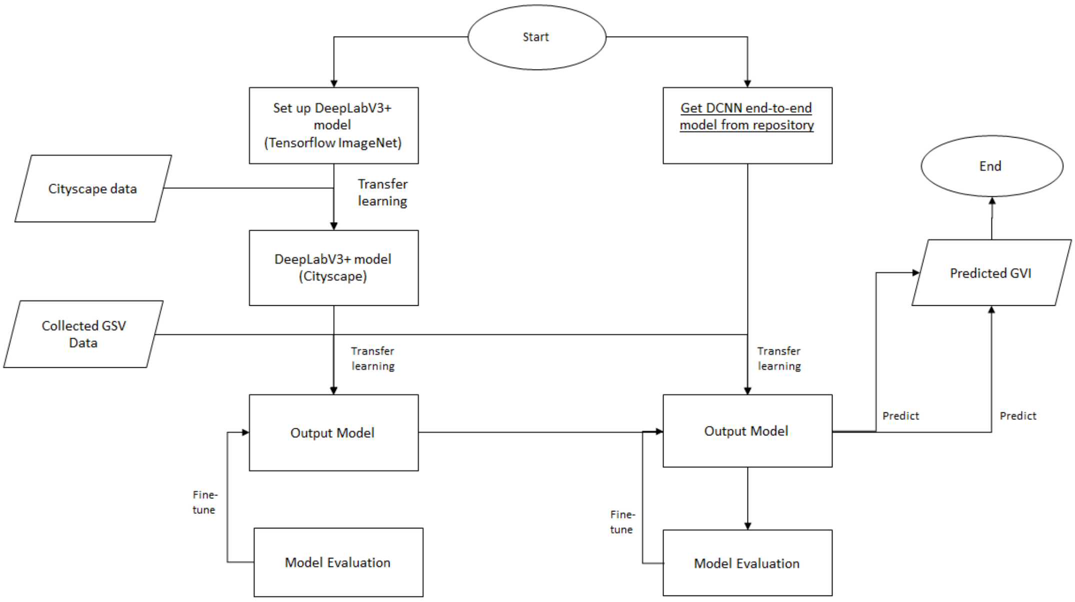

Figure 3.8 The process to develop deep learning models to predict GVI.

In the case of deep learning model training at each step of model fine tuning, hyperparameter tuning plays an important role. For each model, a set of hyperparameters are tuned.

- DCNN end-to-end model: learning_rate = [0.0001, 0.001, 0.01], batch_size = [16, 32, 64]

- DeepLabV3+ model : learning_rate = [0.0001, 0.001, 0.01], batch_size = [8, 16, 32], number of trainable layers = [4, 8, 12]

RandomSearch method in Keras-tuner package is used to automate the search of best hyperparameters for each model training. Nonetheless, due to memory limitations, manual fine tuning is also required in most cases.

3.3.3 Evaluation of Models

The final outputs of the models are all predicted GVI values between 0 and 1. The accuracy of the prediction models are defined by its error rate: the lower its error rate, the higher its accuracy. With the 80 images from test data, following the metrics used by studies covered in Chapter 2, mean absolute error (MAE) in percentage, 5-95 percentile of MAE, Pearson correlation coefficient with the true GVI values (r), are used as the parameters to assess the error rates of the proposed prediction models. Smaller values for MAE and r value closer to 1 represents higher accuracy. Meanwhile, 5-95 percentile of MAE is used to measure the centrality and spread of the error. For this metric, the desired outcome is for the spread to be small and centered on 0.

Other than accuracy, it is also important to find out underlying biasness in models. Thus, bias analysis for models is carried out by plotting error (prediction – true GVI) against the predicted GVI values. This allow us to look into the bias of the models at different levels of GVI. Besides, the frequency of underestimation and overestimation at different predicted GVI levels are documented and discussed as well.

Other than accuracy, the models are also evaluated by the inference time. The time taken to predict 1000 GVI values for each model is recorded.

3.4 Dashboard and Report

The results generated from the GVI predictions are used to understand how often the people in the site can view greens in their daily lives. Thus, further exploratory analysis on the patterns underlying distribution of GVI predictions in Johor Bahru city center is carried out. Moreover, a dashboard is built to provide an interface to the findings of this project.

3.4.1 Clustering Analysis

Clustering analysis is essential to help us to figure out the underlying patterns of GVI distribution in the site. Nonetheless, it is no trivial task to know which clustering method to apply on the GVI predictions on site. In this case, there are several parameters to experiment with to find out the best clustering approach:

1. Clustering model: partition-based clustering (Kmeans) and hierarchical clustering (agglomerative clustering)

2. Number of clusters

The clustering results are evaluated based on both silhouette score and inertia to find out the best approach to clustering.

Exploratory data analysis is carried out on each cluster generated to find out the characteristics of each cluster and thereby probable solutions that can improve GVI in the target site.

3.4.2 Dashboard Development

The dashboard provides a user-friendly interface to access the findings from the project. The developed dashboard should include 2 main components:

1. Allow the user to gain understanding on the prediction models developed for GVI prediction. This makes sure that the users of the dashboard are informed about the underlying bias and errors that might present in the prediction results.

2. Allow users to learn about the GVI distribution and their underlying patterns on site easily.