CHAPTER 4
ANALYSIS AND DESIGN

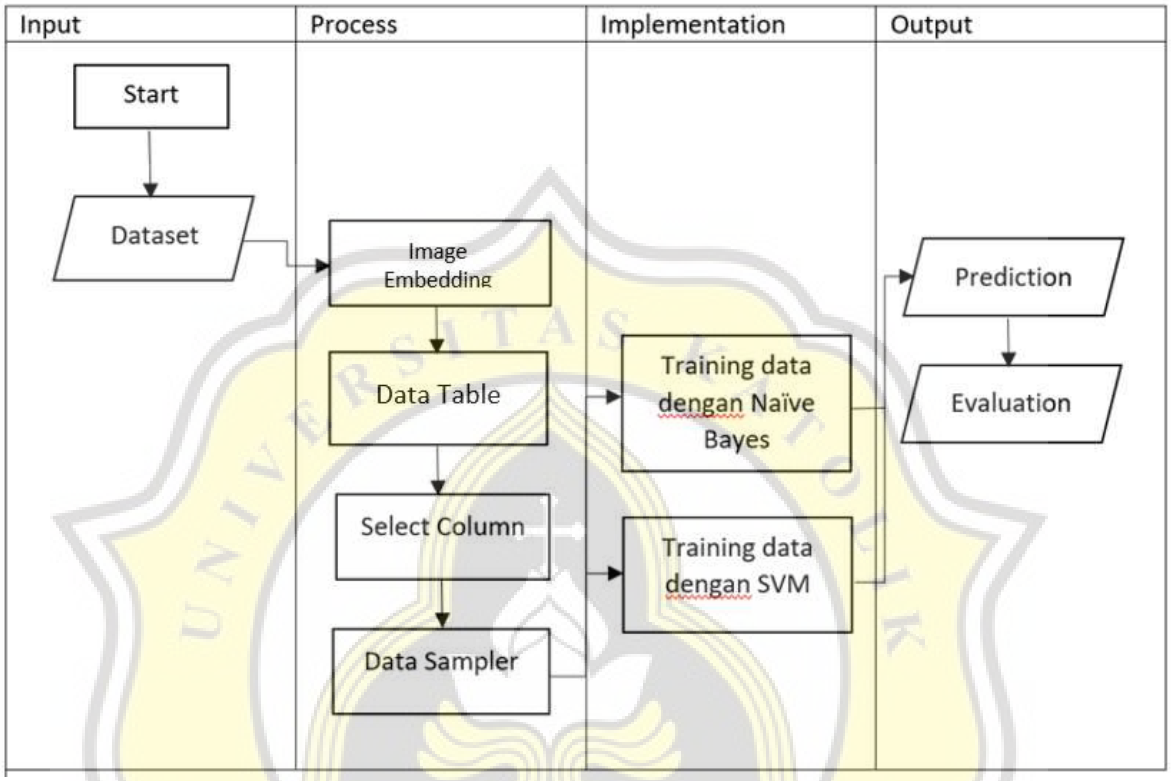

Table 4.1 The flowchart of the process

4.1 Data Collection

The dataset that will be used in this project is an image of a person wearing a mask and not wearing mask, as shown on figure 4.1. The dataset will be taken from the Kaggle.com website. Kaggle is a website where data scientists can collaborate, get inspiration, compete with one another, learn new concepts and coding techniques, and view examples of practical data science applications. There are numerous data sets that may be used for anything from something as straightforward as video game sales to something more intricate and significant like statistics on air pollution. Since this data is authentic and reliable, you may use it to train and test your models on tasks that will eventually benefit

to others. The dataset used has many variations, various image sizes, gender, skin color, pose, brightness, and facial expressions. There are more than 1300 datasets that have been collected in the past and dataset will be transferred to the "dataset" folder and divided into folders 0 and 1. Dataset of photographs of unmasked people can be found in Folder 0. The dataset in Folder 1 contains pictures of people wearing masks. However, in this case, we will run the program multiple times using the given datasets, testing them in descending order of size. Then, using the Upload Image widget under the Image Analytic add-ons, enter the Dataset into the Orange Data Mining software. But we must first install the Image Analytic add-ons in the Options>Add-ons option before we can utilize this widget.

(a) (b)

Figure 4.1 (a) example dataset for without mask , (b) example dataset for with mask.

4.2 Implementation

Table 4.1 presents steps about the process. The procedure is broken down into 4 steps: input, process, implementation, and output. The Orange Data Mining application will be employed in this project. The University of Ljubljana created the Orange application with the goal of presenting a method to analyze data graphically without the requirement for prior programming knowledge. Regression and classification tasks can be completed with orange data mining tools. In this project, current datasets will be processed using categorization techniques. The act of classification is a technique for grouping a number of unlabeled items into a number of distinct

classes. An attempt is made to analyze the relationship between a group of feature variables and a target variable during classification. The target variable in classification is of the type category. The value of the target variable is the obvious distinction between classification and regression. The target variable for categorization must be either a discrete value or a category. Later, the newly classified data will be put into a category with the target variable. Binary classification and Multi-class classification are the two categories into which classification in machine learning is separated. If the target variable only has two categories, such as 0 and 1, Yes and No, X and Y, and so on, binary classification is used. In contrast, the target variable of the multi-class classification has more than two categories. We shall apply the binary classification method in this instance. In binary classification, several different algorithms are applied. Logistic Regression and Accept Vector Machines are the two that were created with solely binary classification in mind (i.e., they do not support more than two class labels). Nearest Neighbors, Decision Trees, and Naive Bayes are a few additional algorithms. At Orange, we'll employ a widget system for data mining. Each widget has a distinct purpose and can produce or receive data.

The primary widget for image analysis in Orange is the Image Embedding widget. The widget downloads the picture table and sends the data to the server to be embedded in a machine learning algorithm-friendly format. The widget receives number vectors from the server after pushing photos through a deep neural network that has already been trained. Since embeddings are computer-generated vectors of numbers, they are typically difficult for people to understand (also, the network that infers numbers is learnt automatically from labelled images). One aspect of the image is represented by each number. The neural network will receive the image as input, and the output of the penultimate layer of the neural network will be used as an embedding. Data must be added to the Image Embedding widget in order for Orange to submit the image to the server, which will then put it through a deep neural network that has already been trained, such as Google Inception v3. Convolutional neural networks are the foundation of the deep learning model known as Inception V3 that is used to classify images. The Inception V3 is a better version of the Inception V1, a foundational model that was first released as Google Net in 2014. It was created by a Google team, as the name suggests. Continuing from the Data table. Deep networks are frequently developed with a particular goal in mind. For instance, Google Inception v3 can categorize photos into one of 1,000 image classifications. We can simply disregard the suggested classification and

utilize the network's penultimate layer, which has 2048 nodes (numbers), to represent the image vectorially. To ensure that there are no errors or missing data during processing, the data that has been input will first be verified to see if it is correct or not using the Data Table widget. Using the Select Columns widget, data domains can be manually arranged. Users can choose which attributes and how to use them. For instance, a discrete set of attributes and class attributes would make up the domain for constructing a classification model. The appropriate attribute in this situation is "Category." Numerous data sampling techniques are used by the Data Sampler widget. Sample and supplementary datasets are the result (with examples from input sets not included in the sample dataset). After selecting Sample Data and entering the input dataset, the output is processed. Data will be used in the training of the Naive Bayes and SVM models for the implementation phase.

A machine learning algorithm for classification issues is naive Bayes. Its foundation is the Bayes probability theorem. It is employed for classification requiring large training data sets with many dimensions. It is well-known for both its efficacy and simplicity. With the Naive Bayes algorithm, creating models and making predictions happens relatively quickly. When used with vast amounts of data, this approach is advantageous, however it performs poorly when selecting attributes. Naive Bayes is a simple probability-based classification method that is intended to be employed with the assumption that the explanatory factors are independent. In this technique, probability estimation is given more importance than other aspects of learning. This method has the advantage that it can estimate the estimated parameters required for the classification process with a small amount of training data. Only the variance of a variable within a class, rather than the whole covariance matrix, is required to determine the classification because it is presumed to be an independent variable. The correctness of the Naive Bayes algorithm and its high speed when used to a large number of datasets are additional benefits of the algorithm. The error rate is also lower when the dataset is large.

Support Vector Machine is referred to as SVM. SVM is used to categorize objects based on their features. SVM is among the most accurate classifiers available. SVM was created to address the binary class issue. By dividing the difficulty of multiple classes into pairs of two classes, such as one-against-one and one-against-all, it resolves the issue. While this is happening, the SVM (Support Vector Machines) widget executes input to higher-dimensional feature space mapping. SVM widgets can be used for regression and classification applications. SVM used a

hyperplane to partition the class. A function that can be used to divide classes is called hyperplane. A support vector is the outer data object in SVM that is closest to the hyperplane. The most challenging objects to categorize are those known as support vectors because of how closely their position resembles that of other classes. Some of the SVM's own Kernels include linear, polynomial, sigmoid, and RBF. The correctness of the output depends on the kernel choice. When data is linearly divided, a decent kernel function to utilize is SVM kernel linear. Non-linear kernel functions like polynomials are ideal for problems when all training data is normalized. When the data cannot be separated linearly, the RBF kernel is a kernel function that is utilized. When using the RBF kernel, the cost and gamma parameters will be optimized. Sigmoid Kernel The majority of neural networks like it. This kernel function functions as an activation function for neurons and is comparable to a two-layer perceptron model of the neural network.

4.3 Evaluation

The model's predictions for the data are then displayed in the Predictions widget. A dataset and one or more models are both accepted by the Predictions widget. All of the AUC, CA, F1, Precision, and Recall results from the Naive Bayes and SVM algorithms will be shown during this phase. A performance indicator for machine learning classification issues when the output can be two or more classes is the confusion matrix. A table containing four possible combinations of expected values and actual values is the confusion matrix. In the confusion matrix, the classification process' outcomes are denoted by four terms: True Positive, True Negative, False Positive, and False Negative. The Confusion Matrix widget will also be helpful for displaying the percentage between expected and actual classes. The number and percentage of instances between expected and actual classes are provided by the confusion matrix. The proper instances are sent into the output when the matrix's elements are chosen. We can see the exact occurrences and how they were misclassified in this way. In the meantime, a test's true positive rate and false positive rate are plotted using the ROC Analysis Widget. ROC Analysis is for analyzing performance measurement tool data to categorize issues and establish a model's threshold.

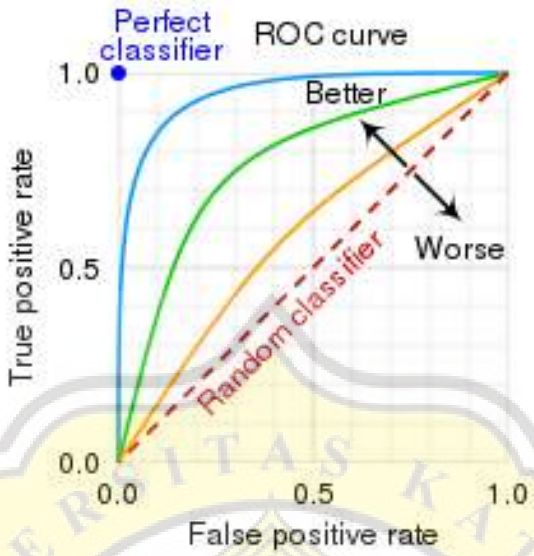

Figure 4.2 example of ROC Analysis.

The technique's name comes from the fact that it was initially created for users of military radar sensors beginning in 1941. With the aid of ROC analysis, it is possible to choose potentially optimal models and exclude less-than-ideal ones without first establishing the cost context or the class distribution. A direct and natural connection can be made between cost-benefit analysis and diagnostic decision-making through ROC analysis. Because it compares two operating characteristics (TPR and FPR) when the criterion changes, the ROC is also known as a relative operating characteristic curve.The ROC Analysis widget displays the ROC curve and associated curvature for the tested model based on figure 4.2 . The image above depicts which curve is better and which is worse in a classifier. It acts as an average of assessments of various classification models. The curve shows the ratio of true positives to false positives, with the true positive rate on the y-axis (sensitivity) and the target's probability on the x-axis (1-specificity; target probability = 1 when true value = 0). The classifier is more accurate the more closely the curve resembles the left bound and then the upper bound of the ROC space. Based on the costs of false positives and false negatives, the ROC Analysis widget may also determine the appropriate classifier and threshold.