The University of Hong Kong

Department of Computer Science

Final report

Final year project
Image matching lost and found system

Name          UID
Chu King Hei  3035693014

Supervisor
Dr. Luo Ping
Abstract

The lost and found procedure provides a platform for lost item owners to seek and retrieve the lost item seamlessly. Though the procedure has been widely adopted by transport and property management personnel, the processes of identification and classification of the items require lots of cost and manpower. The usage of image matching and classification models is an approach to boost the efficiency of the process. Therefore, this project is aims to provide a solution by implementing a interactive and user-friendly web application with image matching and classification model. The project is on schedule. Our team have completed the web application design and implementation that integrated with image matching and tagging models.
Acknowledgement

I would like to thank our supervisor Dr. Luo Ping for giving me continuous conveyance and support throughout the entire project.

In addition, I would like to thank our groupmates Anson and Oscar for their effort and contribution.

Lastly, I would like to thank Miss Mable Choi for continuously providing guidance on writing this report.
1 Introduction 9
1.1 Background and Motivation .................................. 9
  1.1.1 Overview of lost and found procedure ................. 9
  1.1.2 Issue on item identification ............................. 9
  1.1.3 Existing lost and found procedure in Hong Kong .... 10
1.2 Objective, approach and scope ............................... 11
1.3 Project contribution ......................................... 11
1.4 Outline of the report ....................................... 11

2 Methodology 13
2.1 Introduction .................................................. 13
2.2 Image identifying process .................................. 13
  2.2.1 Image matching .......................................... 13
  2.2.2 Image features extraction ............................... 14
2.3 Image classification: ResNet ................................. 15
2.4 Web Application ............................................ 16
  2.4.1 Frontend ................................................ 16
  2.4.2 Backend ............................................... 17

3 Experiments and Results 19
3.1 Introduction .................................................. 19
3.2 Image matching .............................................. 19
  3.2.1 Dataset ................................................ 19
  3.2.2 Selection of distance formulae ....................... 19
  3.2.3 Evaluation methodology: Score ........................ 20
  3.2.4 Selection of image classifier ......................... 20
3.3 Image classification ....................................... 22
  3.3.1 Dataset ................................................ 22
  3.3.2 Model training ....................................... 22
  3.3.3 Learning rate decay .................................. 22
  3.3.4 Data augmentation .................................... 23
  3.3.5 Response time ........................................ 25
3.4 Application design .......................................... 25
  3.4.1 Process status ....................................... 25
  3.4.2 Application workflow for item owner ............... 26
  3.4.3 Application workflow for operator .................. 27
List of Figures

1.1 Existing online form for reporting lost item by CityBus . . . . . . 10
2.1 Example of working mechanism of KNN . . . . . . . . . . . . . . . 13
2.2 Residual learning: a building block [1] . . . . . . . . . . . . . . . 15
2.3 Structure of web application [2] . . . . . . . . . . . . . . . . . . . . 17
2.4 API documentation generated by Swagger . . . . . . . . . . . . . . 18
3.1 Structure of Autoencoder . . . . . . . . . . . . . . . . . . . . . . . 20
3.2 Score of different implementations of classifier . . . . . . . . . . . 21
3.3 Accuracy of classification model without data augmentation . . . 22
3.4 The sample of data augmentation . . . . . . . . . . . . . . . . . . . 24
3.5 Accuracy of classification model with data augmentation . . . . . 24
3.6 Status of lost and found process . . . . . . . . . . . . . . . . . . . . 25
3.7 Application workflow for item owner . . . . . . . . . . . . . . . . . 27
3.8 Finalised workflow for operator to upload the image of item found 28
3.9 Home page of application for item owners . . . . . . . . . . . . . . 29
3.10 Page for declaring on whether the owner have an image of lost item 29
3.11 Convenient cropping tools for resizing the uploaded image . . . . 30
3.12 Website for displaying and selecting the correct image matching result . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
3.13 Screen capture of webpage for owner to search item by tag . . . . 32
3.14 Screen capture of webpage for the submission of contact informa-
tion form . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
3.15 Progress bar of item finding process on step 2 . . . . . . . . . . . . 33
3.16 Screen capture of the online status tracing service . . . . . . . . 34
3.17 Screen capture of email notification on case update . . . . . . . . 34
3.18 Screen capture of cropping tool for operator to upload the image 35
3.19 Screen capture of item specification form with prefilled category
tag . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
3.20 Screen capture of item specification form with tag creation . . . 36
3.21 Screen capture of requests with comparable information . . . . . 36
3.22 Screen capture of request detail page . . . . . . . . . . . . . . . . . 38
3.23 Screen capture of dashboard on web application . . . . . . . . . . 39
3.24 Screen capture of request list on web application . . . . . . . . . . 40
3.25 Requests list accessed by tablet with column hiding . . . . . . . . 41
3.26 Requests list with filtering by Status . . . . . . . . . . . . . . . . . 41
3.27 Requests list with filtering by Submission Date . . . . . . . . . . . 42
3.28 Requests list viewed in Compact Density . . . . . . . . . . . . . . . 43
3.29 Requests list viewed in Comfortable Density . . . . . . . . . . . . 43
3.30 Screen capture of exported requests list . . . . . . . . . . . . . . . 44
3.31 Screen capture of inventory web page . . . . . . . . . . . . . . . . 45
3.32 Screen capture of web page for managing Venues and Category
   Tags . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
3.33 Changing tag name to an existed name . . . . . . . . . . . . . . . 45
3.34 Screen capture of dashboard accessed by tablet . . . . . . . . . . 46
4.1 Implementation methodology of online learning . . . . . . . . . . . 48
List of Tables

3.1 Accuracy of different distance formulae of KNN . . . . . . . . . . 19
3.2 Efficiency measurements of ResNet-50 and EfficientNet-B7 . . . . 21
3.3 Project schedule . . . . . . . . . . . . . . . . . . . . . . . . . . 47
**Abbreviations & Acronyms**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Program Interface</td>
</tr>
<tr>
<td>CNN</td>
<td>Convention Neural Network</td>
</tr>
<tr>
<td>KNN</td>
<td>K-nearest neighbor</td>
</tr>
<tr>
<td>ResNet</td>
<td>Residual Neural Network</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>UI</td>
<td>User interface</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Background and Motivation

1.1.1 Overview of lost and found procedure

It is common for losing item in public area. In 2021, there are more than 7 thousand lost item found and 5.6 thousand of reported lost item per month in Mass Transit Railway (MTR). However, there are only 30 to 40 percent of cases that the items were successfully found and returned, which is far from satisfactory.

Considering the operators from MTR spend 5 minutes on processing each of the cases, it requires 7 thousand hour to handles 84 thousand cases annually, which is costly and time-consuming.

In addition, the identification of lost item is a laborious process. Since the description of the object varies from person to person, only using text or verbally describing the outlook may not be sufficient to describe the object outlook. Therefore, the process of classification of the features, categories, and identification of the ownership of the lost item required lots of manpower.

Considering the low efficiency and success rate of the procedure, more attention should be paid towards enhancing the procedures.

1.1.2 Issue on item identification

The complexity of identification process intensified when the numbers of similar items increased, which is common in airports. In July 2022, Frankfurt Airport have received over 2,000 ubiquitous black suitcases in a month[3]. This causes the operators need to check through every similar items for every request to prevent mismatch of items before handling back to the owners. Moreover, operators require to collect additional information from the owner when the provided information is insufficient for classification and add another layer of complexity on the process.

The effect on the increase of complexity will result in more manpower, longer turnover time and lower return rate. Travelers has to retrieve their item later, or even losing them.

To improve existing procedure, we need to understand the issues from existing lost and found services.
1.1.3 Existing lost and found procedure in Hong Kong

Lost and found involves a series of procedures from searching or collecting the leftover items, matching with the description provided and updating processing status and return item to the owner.

Currently, the major transport operators including Citybus, Kowloon Motor Bus (KMB) and Hong Kong International Airport are manually handling the input from the item owners. In Fig. 1.1, we can find there are 11 categories of information required to fill in. For the column of Item Detail/ Other Information, allow the users to enter at most 750 characters, this showcase a very complexity and lengthy experience for owner. In addition, owners have to call customer service hotline, and have a long wait for a customer service agent to for replying the current application status, this further hinder user experience and added additional workload for the operators.

![Fig. 1.1. Existing online form for reporting lost item by CityBus](image)
To improve the user experience and reduce the manpower needed, our team have decided to integrate deep learning models with web application to streamline the entire process.

1.2 Objective, approach and scope

In this project, we aim to implement an application for lost and found system to provide automatic and efficient process. We will attempt to integrate a machine learning model into the application to automate the image matching and classification process. The scope of the project includes three parts, including image matching and image classification models as the backbone of the solution and a web application with server and database for both the operators and item owners to input the related information, confirm the matching result and trace the processing status of each case.

1.3 Project contribution

The image matching lost and found solution is anticipated to streamline the entire process for the public who lost any items, transportation and infrastructure personnel. This project is aimed to provide a reliable, efficient and user-friendly solution, which revolutionises existing time-consuming, labour-intensive lost and found process.

1.4 Outline of the report

The report consists of five chapters. The first chapter provides an introduction of the current lost and found procedure in Hong Kong and identify the issue of the system by reviewing from the existing services. In addition, it establishes the aims and showcases the importance of the our solution.

Chapter two introduces the methodology used in the project. The explanation of KNN and CNN for image matching, classification models and web application will be illustrated.

Chapter three reports the current progress and experiments of the project, we have completed the implementation web application that integrated with image classification and matching models. In addition, the chapter will display the detailed project schedule.

Chapter four states the future plans that we aim on the image classification
model.

Chapter five summaries the major accomplishments we have made on the entire project.
2 Methodology

2.1 Introduction

This chapter presents the models that we will use to build the image matching and classification model, including K-nearest neighbors (KNN) algorithm and Convolution Neural Network (CNN) and the web applications design for operators and item owners.

2.2 Image identifying process

2.2.1 Image matching

K Nearest Neighbor (KNN) which is a non-parametric is used for identifying similar images. In Fig. 2.1, the green and orange dots representing the item which belongs to category A and B respectively. Since the item with the same category is next to each other, by calculating the distance between any new data point between each point in the database can result in finding the first k closest neighboring point. By majority voting of the k-nearest data points, KNN determines the category of the new data point.

Fig. 2.1. Example of working mechanism of KNN

To utilize this model in the application, the initial step involves converting every image into a feature vector before processing. Afterward, the model calculates the distance between each of these feature vectors and the input image vector. Finally, the k nearest feature vectors are identified as the most similar images
and will be the output of the model.

Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Cosine Similarity (COSINE) are three choices for distance formula. $x^i$ denotes each pixel on an image stored in the database, and $y^i$ denotes each pixel on an input image from the user in the following formula.

$$RMSE = \sum (x^i - y^i)^2$$

$$MAE = \sum |x^i - y^i|$$

$$COSINE = \frac{X \cdot Y}{\|X\|_2 \|Y\|_2} \text{ where } \{x^1, x^2, ..., x^i \in X\}, \{y^1, y^2, ..., y^i \in Y\}$$

RMSE have a larger penalty upon error, which the penalty is scale with square while MAE only scale with linear. Cosine similarity compute the cosine of the angle of the formula, which is calculated by the dot product of the vectors divided by the product of the lengths. Further experiment is needed to determine which formula should be applied.

Since KNN does not require training and the k number is not fixed and can be adjusted at any time, it gives high flexibility and elasticity to processing various types of lost items. We can expands the number of output, which is the k number when the users required more results.

### 2.2.2 Image features extraction

Nowadays, signal images are composed of millions of pixels, resulting in an increased computation cost for KNN. To determine the differences between images, the features, including the shape, size, color, and structure are more critical than performing direct comparisons on every pixels. Therefore, before feeding the image to the KNN, features should be extracted. Therefore, Autoencoder is an effective solution for feature extraction. Fig ?? shows encoder captures the image pixel and compresses it to an encoded feature. Then, the decoder reconstructs the image from the encoded feature. The training loss is calculated based on a pixel-wise difference between the original image and the inference image.
This whole process does not require any operator staff to monitor and only complete automatically by machine learning model and algorithm. In addition, the training for both autoencoder and KNN does not require any label data. Therefore, the training cost is low for gathering datasets.

2.3 Image classification: ResNet

There has been many research conducted on image classification tasks, and many models has been published AlexNet, VGG, and ResNet. In our project, we will use ResNet-50 as our model and modify it if necessary. Without a residual network, many models are susceptible to the gradient vanishing problem. This occurs when back-propagating the loss to optimize matrix parameters, but the effect on the parameters becomes smaller and smaller as it reaches higher layers. Ultimately, the loss becomes ineffective in optimizing the upper layers, which limits the accuracy of model. This problem is known as the gradient vanishing problem, and it becomes more severe as the number of layers of the models increases. To solve this issue, Microsoft introduced the residual network [1].

Fig. 2.2. Residual learning: a building block [1]

The key structure of residual network is shown in Fig. 2.2. The idea of the network is to provide a shortcut connection between some of the layers. The data of the upper layer will be transferred directly to a much deeper layer using identity mapping, and skipping all Fig. 2.2 the middle layers. Therefore, the loss can be back-propagated directly to the upper layer and hence reduce the effect of gradient vanishing problem. This makes the model ResNet-50 have a high accuracy in image classification, be easy to be optimized, and have a high
training speed.

The classification model is adopted to generated a category tag on each submitted image when the operator upload the image. Before adding the image to the KNN matching model, the tag is used to filter the image in database and boast the efficient of image matching processing by reducing the number of input images. In addition, this ensure the item owner who do not have an image of the item can search the target lost item based on the filtering result of category tag.

2.4 Web Application

Web application or web-based application is an application that operate on web browser or internet. [4] Instead of ordinary application that runs on the device’s operating system (OS). The Web or application server will execute the logic and response with specific output based on the user interaction though the internet. With an unique compatibility with any OS, users can access the web application with an up-to-date browser and internet connection, regardless of the type of device and OS that the users are using. This enables the operators to develop and maintain only one application across all operating systems instead of having one dedicated application for every OS.

The web application is used for lost item owners to submit request by uploading images to the server and operators to uploads information of items found, manage and analysis on the inventories, requests information. We adopt different frameworks in both backend and frontend for helping us to develop a high quality web application rapidly.

2.4.1 Frontend

We adopt React, a commonly used JavaScript library[5] for building the user interfaces. The Component-Based Architecture adopted by React allow us to create reusable and modular elements. React leverages a virtual Document Object Model (DOM), enabling faster rendering and more efficient updates of our application. In addition, React Router provides a seamless and declarative routing management for handling routing in the application.

Furthermore, we choose Material UI (MUI), a React-based user interface (UI) framework which provide pre-built components and design guidelines[6] for achieving better user interface and experience. Using MUI customize and flex-
ible components, we can create unique UI elements with responsive design to ensures the application is adoptable to varies devices and screen sizes. In addition, MUI provides a wide-variety of icons, typography, and color schemes for us to improve the UI.

2.4.2 Backend

We adopt Django REST framework[7] as the backend server of the system. Django provides us with an simple and flexible framework for building web applications. Through the Model-View-Controller (MVC) architecture, Django is able to clearly separates the system database, user interface and control logic of the system[7] and helps we to build a high quality web application efficiently. In addition, Django REST framework provides powerful packages for building up RESTful APIs, which enable the server can integrate with the frontend seamlessly. For instance, we use JWT package provided in the Django toolkit for authentication of admins and operators.

Fig. 2.3. Structure of web application [2]

Fig. 2.3 summarized the entire structure of the web application. With Django being the core of the application, Django utilises Object Relational Mapping (ORM) for communication between database, which handles the creation and
management of database model and queries.

The django REST framework will provide restful APIs to interact with the frontend. Upon receiving request from frontend web page, the django REST framework will deserialize the request, and pass to django for handling. If the request is related to any information in the database, for instance getting, setting or deleting any entries, django will communicate with database by ORM and return data to the REST framework. Finally, the REST framework serializes and sends back the data.

To keep a documentation on varies API that we have implemented in the server, we adopts Swagger[8] to generate the latest version while keeping record of previous versions of the document.

Fig. 2.4. API documentation generated by Swagger

Fig. 2.4 displays a section of the documentation of our application, which listed all the services provided by the server and the structure of the request and response object. This improves the efficiency of development by giving us a clear view on all the available services and their respective requirements.
3 Experiments and Results

3.1 Introduction

This chapter outlines the experiments we made on the image matching and classification models by evaluating the model performance and shows the result of web application design and implementation.

3.2 Image matching

3.2.1 Dataset

To evaluate the performance of KNN, CIFAR10 dataset is used. The CIFAR10 is a dataset that contains 60,000 32x32 color images in 10 classes.[9] The dataset have separated the 60,000 images to 50,000 training images and 10,000 testing images.

3.2.2 Selection of distance formulae

We have used the dataset to evaluate KNN model with the MAE, RMSE and COSINE. Table 3.1 summarizes the accuracy of three of the distance formula on 10,000 testing images of CIFAR10.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN with RMSE</td>
<td>35.39%</td>
</tr>
<tr>
<td>KNN with MAE</td>
<td>38.59%</td>
</tr>
<tr>
<td>KNN with COSINE</td>
<td>48.30%</td>
</tr>
</tbody>
</table>

From Table 3.1 we can find that among all three of distance formula, COSINE performs the best, which have an accuracy of 48.30%, more than 10% higher than MAE.

After examining the result, we finds that since data CIFAR10 consits of different coloured background, both the MAE and RMSE has a huge bias towards the background of the images. This induced the low accuracy of two model. Therefore, we decide to opt COSINE as our distance formula.
3.2.3 Evaluation methodology: Score

Unlike image classification, for image matching it is difficult of us to evaluate a model by simply analysis through the accuracy of the result. Therefore, we defines a score calculating methods based on the probability of the recognized similar images. By utilising the tags provided on CIFAR10 dataset, and calculate the probability of tags from images output by the models that equals to the tag from the input image. For instance: if we input an input with tag Ship to the KNN model, while the k number is equals to 5, the models output 3 images with tags Ship, while other 2 are incorrect, and the score is 3 divided by 5, which equals 60%.

3.2.4 Selection of image classifier

Originally, we have tried to design an Autoencoder (Fig. 3.1) for the task. The encoder is using 3 layers of Convolution Block, which includes Batch notarization, convolution stride 2 and ReLU as the activation function and 2 linear blocks for generating the image features. However, the accuracy using the encode is still undeniable, which have 48.9 % only.

Therefore, to further improve the accuracy, we turn our focus to building the feature extractor for providing the input of KNN with COSINE. We have extracted the CNN layers, by removing the Fully-connected layer of ResNet-50 and EfficientNet-B7 to formulate these classifier to a feature extractor.

We have tested Plan KNN, KNN with Auto-encoder and classifiers based on...
ResNet-50 and EfficientNet-B7 with CIFAR10 dataset with different numbers of k on KNN and evaluate their performance based on the score defined in Section 3.2.3. The result is summarized on Fig. 3.2 below.

![Score of different implementations of classifier](image-url)

**Fig. 3.2.** Score of different implementations of classifier

We have discovered that using classifier based on ResNet-50 and EfficientNet-B7 gives us a huge score boost on all the number of k, which has over 40% of score improvement when comparing to Autoencoder and plain KNN. This showcase these classifiers can output more result with same categories and should have better ability for matching similar images.

In order to choose among ResNet-50 and EfficientNet-B7, we have referenced to the number of parameters and FLOPs to ensure the model is efficient enough to power the application with low latency.

**Table 3.2.** Efficiency measurements of ResNet-50 and EfficientNet-B7

<table>
<thead>
<tr>
<th>Model</th>
<th>No. of params</th>
<th>GFLOPS</th>
<th>Latent size</th>
</tr>
</thead>
<tbody>
<tr>
<td>EfficientNet-B7</td>
<td>66.3M</td>
<td>37.75</td>
<td>2,560</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>25.6M</td>
<td>4.09</td>
<td>2,048</td>
</tr>
</tbody>
</table>

From Table 3.2, we found that ResNet-50 is more efficient than EfficientNet-B7 as it use lower params, FLOPs and with smaller latent size, coupled with the better performance of ResNet-50 on the testing on real data, we have selected ResNet-50 be our classifier of the model.
3.3 Image classification

3.3.1 Dataset

We have extracted 98 categories of commonly lost items from the ImageNet-1k[10] dataset, including laptops, mouse, water bottle and so on for classification, each category contains 1,300 images. In addition, we have randomly extracted 85% of the images for training, while the remaining 15% of the images are for testing.

3.3.2 Model training

Compared to the traditional Stochastic Gradient Descent (SGD), using Adaptive Moment Estimation (Adam) optimiser and normalise the image before training to ensure the model is able convergence effectively even with a large dataset, which is crucial for applying the model for online learning. After multiple trials, we have found that using batch size of 128, with an initial learning rate $10^{-4}$ gives out the best result and we have adopted it.

3.3.3 Learning rate decay

At first, we directly resize the images in the dataset to 224x224 as the input of the model, and as a result, the model has significantly overfitted to the background of the image after 12 epochs.

![Fig. 3.3. Accuracy of classification model without data augmentation](image-url)
To mitigate overfitting, we implement learning rate decay to reduce the learning rate as the training progresses. Fig. 3.3 summarize the test and verification accuracy of the model. The top-1 verification accuracy slightly increases 1.20% from 50.53% on a model without learning rate decay to 51.73% with a learning rate decay of 0.0125. However, the verification accuracy is still undesirable, and we decide to work on inspecting the input images.

3.3.4 Data augmentation

After a walk-through with the input images, we discover instead of the target lost item, the background covers a large proportion of the image and the model misclassified the image based on the background of the image when we directly resize the image to 224x224 (Fig. 3.4). Therefore, we decide to complete a data augmentation aim to increase the proportion of the target lost items in the input image.

To increase the proportion of the items, we will first resize the image to 256x256, then centre crop the image to 224x224 (Fig. 3.4). In addition, we also implement random flip horizontally and random rotation to increase the randomness of input images and aim for reducing overfitting of the model.

Fig. 3.5 summarize the accuracy of models after adding data augmentation under different learning decay. The top-1 verification accuracy has a large, 5.845 increase from 51.73% on the model with a learning rate decay of 0.0125 and no augmentation to 58.57% with learning rate decay of 0.0125 and data augmentation.
Fig. 3.4. The sample of data augmentation

Fig. 3.5. Accuracy of classification model with data augmentation
Apart from the accuracy, the response time is another crucial factor affecting the server performance and user experience.

### 3.3.5 Response time

We have tested the response time of the classification model by inputting 1000 photos for classification on a server which only support CPU. By taking arithmetic average of the response times, the model required an average of only 0.6 seconds for classifying the image. This showcase our model is capable to classify promptly and provide a responsive result to the end users.

### 3.4 Application design

We can generalise the process by two scenarios, the owner finds its item upon submission of request and the opposite, not match item upon submission since the item is not return to the lost and found center and store in the database at the moment.

To establish a faultless design on the lost an found process, apart from the verification completed by lost items owner, the operators should double check on the correctness of submitted request for fraud prevention. Therefore, we have design an application that ensure the consensus are met on both side at all scenarios.

#### 3.4.1 Process status

For generalising the process of lost and found and providing the service of status tracking, we define four statuses for the entire process starting for two scenarios, from the submission of request to the retrieval of lost item, as shown in in Fig. 3.6.

![Fig. 3.6. Status of lost and found process](image-url)

Generally, when the owner submit the application, the application status will
set to Pending until both parties confirms the uploaded images and respective information. If both parties agrees, the status is shifted to Approve, indicating the application has been confirmed and the owner can come to the lost and found center to collect the item. Once the item is collected, the status is changed to Return, indicating the item has successfully returned to the item owner, which marks the completion of the process cycle. If the consensus does not meet, it indicates the image of item found is not belong to the item owner, or the provided information from the owner is incorrect, and the status is changed to Reject.

Owners can track the latest cast status and updates by entering a unique reference number on website, which is provided upon case submission at any time on the web application.

For implantation of the application, we need to deep into the design of workflow for both the operators and item owners on both scenarios and consider the services provided by the backend server and the reaction, behaviour and user interface on frontend.

### 3.4.2 Application workflow for item owner

Fig. 3.7 has summarized the workflow of the item owners. In case of the owner has the image of the lost item, the photo will then pass to the server for image matching with the KNN with classifier as indicated in 2.2.1 and return a list of similar images. Otherwise, if the owner do not have the item image, the owner will enter category tags to the system and the server will return a list of images that match with the tags.

Once the owners received the images, the owner can select and verify the image is belongs to him and provide related information, including the venues and date of item lost. If the owner is able to find and verify the image that matches with his item, the application will sent to the operator for verification. Once the operator verify the request, since both parties have verified it, the request status will now change to Accept. In case of the owner can not find the image that matched with the item, the request will store in database, and the owner will up notified when then similar item is found by the operator.
3.4.3 Application workflow for operator

Fig. 3.8 shows the finalized workflow for operator to upload the item image. When the operators upload an image, a tag will be generated and auto-filled by image classification model as mentioned in 3.3. After filtering with date, venue and tags of the items, if there is any images satisfy the filtering requirements, the images with pass to the image matching model to find whether there is any image matched with the image submitted by owners. If there is any request that the image is matched and the information is verified by the operator, the server will send an email with the case specific URL to the owners, which direct owners to the page of confirming correctness of item found. Once user confirms the result, since consent between two parties (operators and owners) is met, the matching process is completed and the owner is able to collect the item.

When the request do not satisfy the filtering requirements or disapprove by either operators or the requester, this indicates the item still has not reported by any owner and the case will store in database for item owner to search by
Apart from the design of the application, our team completes the implementation of the web application for owners to report and track the status, and for operators to upload image, review requests, and complete analytic.

### 3.5 Web application for item owner

We implement the one interactive web application for both admin and item owners. For accessing the administrative page, the admins need to login at the home page of the system with preregistered credential.

When the owners first visit our page, the users are directed to home page of the system as indicated as Fig. 3.9. To ensure clarity and accessibility of the application, for pages required for selection of service (Fig. 3.9, 3.10), the options available are displayed at the middle of the page to ensure the users can
simply identify and navigate to the related service.

![Lost And Found System](image)

**Fig. 3.9.** Home page of application for item owners

The web application offers two main features for item owners, which are lost items finding as mentioned in Section 3.5.1 and status checking on Section 3.5.2.

### 3.5.1 Items finding

Once the owner opted for "Find my lost items" at the home page, the owner is required to identify whether he have the item of lost item (Fig. 3.10).

![Page for declaring on whether the owner have an image of lost item](image)

**Fig. 3.10.** Page for declaring on whether the owner have an image of lost item

### Image matching

If the owner clicks yes, indicating the owner has the image of item, the owner will be directed to the image matching lost and found system as shown in Fig. 3.11 for cropping and upload the image to the backend database.
Once the owner cropped and submitted the image, the image is forwarded to the server for image matching with model mention in Section 2.2.1. The model will output a list of probability of similarity of the images of item found in the database. The similar images will be display to the owner by descending order of the similarity as shown in Fig 3.12.

When the owner move the cursor on one of the image of items found, the image is magnified for user to identify the smaller features of the image clearly and enhance the clarity of the application.
We have optimised to speed up the matching process by storing the latent factor in our database upon the submission of an image. Therefore, the server does not need to compute the latent factor for every request, which improves the efficiency of the process, especially when there are many images in inventory.

**Searching by tags**

If the owner does not have the image, they will be directed to the webpage as shown in Fig. 3.13 for searching images based on category tags.

The user can search for the image of items found by searching and enter the category tags that are predefined by the operators. Since there may be some categories with similar meaning, for instance: notebook and laptop, the owners can search for multiple tags. The webpage will display all the images that satisfy any one of the input categories tags with the order starting from the image of the more recently found item. The owners can select the target item by simply clicking the respective image.
Providing contact information

After selecting the target image, or clicking the "I Do Not Found My Item" button, the application progress will shift to the last section, to provide contact information. Comparing to existing form of Citybus as mentioned in Section 1.1.3, our application do streamlined to information need to input from 14 text fields to the necessarily contact information, location of losing the item and remarks only as shown in figure 3.14.

Fig. 3.14. Screen capture of webpage for owner to search item by tag

Fig. 3.14. Screen capture of webpage for the submition of contact information form
Once the form is submitted, a notification bar will pop out on the top right hand corner to notify the owner that the application is completed. This ensure the application gives clear feedback to the owner, and the owner understands the application process is completed. Then user will be directed to the status page as indicated in Section 3.5.2.

**Status Bar**

To ensure the clarity and easy navigation of the application process, a progress bar is displayed on the top of the page as shown in Fig. 3.15.

![Fig. 3.15. Progress bar of item finding process on step 2](image)

The progress bar display all the steps of request processing and the indicated the previous steps with a tick icon, current step with black color icon and text and future steps as grey color. This allow the owner navigates the entire process by identifying the previous, current and future steps of the request process and gives an estimation on the time and information needed for the application.

### 3.5.2 Status tracing

Our application ensures the owner can trace the case status by status checking on web page and email notifications. By inputting the unique reference number to the application or after submitting the web page, the owner will able to check the latest status and update time of the case. Owner is able to simply copy to reference number of the case by click the copy the typography located on the right of the number and allow the user to easily copy and save up the reference number.

When there is any change of status, for instance, when the operators found an item that is related to the application, the server will send an email to the registered email address of the owner. The email contains a URL which direct owners to the web page containing the case status or action to be completed.
3.6 Web application for operators

The web application is aim to provide an interactive and user friendly features for operators to complete three main features, including images upload, requests review and analytic on cases information after login to the system with preregistered credential.

3.6.1 Submission of image

Upon receival of lost items, the operators can efficiently upload the image to our system by the cropping tool provided as shown in Fig. 3.18, so that operators
can skip the progress of preprocessing the upload image on other software.

Fig. 3.18. Screen capture of cropping tool for operator to upload the image

Once the operators have uploaded the image, the image will send to the server for generating the categories tags by the image classification model as mentioned in Section 2.3. Since the response time of the model is only around 0.6 seconds, without a lengthy delay, the operator can smoothly view the item specification form with prefilled category tag generated by the model, in this case the tag is mouse as shown in Fig. 3.19. The prefilled tag reduced the manpower needed for classification and streamline the process.

Fig. 3.19. Screen capture of item specification form with prefilled category tag
The form only required the operator to fill in 3 necessary information, including location, lost item found, which saved up many time for providing text description of the image.

![Fig. 3.20. Screen capture of item specification form with tag creation](image)

The operator can search from a set of predefined category tags and a venue by searching on a drop down list. In case of the desired tag or venue is not found, the operator can simply add by inputting the name and click enter to create a new one as shown in Fig 3.20. Once the tag is successfully created, a confirm message on the top right corner of the webpage is displayed.

In case of there is any submitted request with comparable information, including: date and venue of lost, the system will display all the similar request as shown in Fig. 3.21.

![Fig. 3.21. Screen capture of requests with comparable information](image)

The server complete image matching with same model mentioned on Section
2.2.1, using the image uploaded by operator to match with the images from filtered requests in the database. The frontend have display the requests and the images according to the similarity output by the model. Therefore, on Fig. 3.21, the most similar image, the mouse is located on the top left corner of the output and the least similar image, the water bottle is located on the right.

The operator can confirm that the image is matched with the selected cases clicking the image, and an email will send to the requester automatically as mentioned in Section 3.5.2. By clicking the URL included in the email, the requester is able to confirm the item that received by operator belongs to him. If the requester verified the item is belongs to him, consent of both parties are met, the status now changes to Accept. Otherwise, the case is stored in the database for owner to search.

3.6.2 Request review

Once the owner submits a request, the operator can view the request detail by clicking the Pending Request Section on Dashboard or the information button on the Request List that will be mentioned in Section 3.6.3 and 3.6.4 respectively.

The request detail page (Fig. 3.22) displays that all the case information with a simple and concise user interface with the help of icons to represent the idea of the fields.

To simply the comparing process, the page has integrated with a smart comparing system for automatically compare the difference between the information, including the date of lost, category tags and lost venue, uploaded by the requester and operator. Since one entry may contain more than one values, for instance category tags, an entry is defined as unmatched if all requester input values do not match with one of the value uploaded by the operators.

The system issues a warning massage on the top of the page if one or more entries are unmatched and highlight the unmatched entry in yellow color as shown in Fig. 3.22. This ensures operator can conveniently spot the difference at a glance and save up the time required for manually comparing the information on reviewing the requests.

On the second row of the page, requester information and request status are listed concisely for identification of owner and current progress of the case.

For the convenience of comparing the difference among the information sub-
mitted by requester and operator, the detailed information submitted by both parties is displayed side by side when the page is visited by computers or tablets on the third row of the page. Therefore, the operator can easily compare every entries horizontally.

On the bottom of the page, there is a section for selecting the approval of request. The operator can either accept or decline the request based on the information listed. The application status on both operator and requester side are updated to Rejected or Accept accordingly. The requester may check the updated status though email or webpage for the updated status as mention in Section 3.5.2.

Apart from the two main functions that is used by the operators, we have implemented four webpages for data analytic, inventory and venues category tags management.

3.6.3 Dashboard

As the homepage of web application for the operators, dashboard is designed for them to conveniently get the statics of lost and found procedures, including the number of pending request awaiting for process, top category tags and losing spots of lost items as shown in Fig. 3.23.

![Fig. 3.22. Screen capture of request detail page](image)
On the top of the dashboard, the numbers of requests and received lost items of current month is displayed for giving an overview of this month situation to the operator.

In addition, the number of pending requests, with the latest case ID and update time is displayed on the left of second row. By clicking the pending request section, the web application will direct the operators to the detail information of latest pending request, to ensure they can conveniently and rapidly review and approve the request.

In the section of top 5 Lost items Hot Spots, it display the 5 places with the most lost items found with an interactive bar chart, when the mouse cursor is located on one of the bar, the chart will display the quantity of item lost and the location for adding simplicity on data reading. This notify the operator on the Black Spots of losing item and encourage the operators to come up with solutions to reduce the number of item lost in the specific regions.

Moreover, the Popular Tags section display number of items lost in 8 most commonly lost category tags in descending order as default for viewing the most commonly lost items at a glance and get to know the current trend of the categories of item lost. A search bar is added in this section for the operators to rapidly access the number of items lost of the target category tag to improve the user experience.

Fig. 3.23. Screen capture of dashboard on web application
Finally, the Remark by Staffs section provides a space for communication among the operators. The operators can send and receive broadcast messages among themselves to communicate on any updates of the cases. This section acts as a notice board for noting the important remarks and saving up the time needed for checking the updates and communicating through other channels.

### 3.6.4 Requests List

Request List page lists out submission date, reference number, requester information, request status and last modified timestamp of all requests by the lost item owners in an interactive table format as shown in Fig. 3.24.

![Screen capture of request list on web application](image)

**Fig. 3.24.** Screen capture of request list on web application

To access the complete information of a specific request, the operator can simply click the respective information icon on the right of the row, which directs the user to the page of Request Review as mentioned in Section 3.6.2.

On the top of all tables on the web application, the button of four useful features, hide columns, filter rows by value, display density and export as CSV file are listed.

To hide or display a column, operators can click on the COLUMNS icons and selects the target columns wants to hide or display. There is a search filed for the operators to quickly search for the target columns and a set of HIDE ALL.
and SHOW ALL button to act as express key to hide or display all column on
the table as shown in Fig. 3.25 to customize the column wanted to display.

Fig. 3.25. Requests list accessed by tablet with column hiding

This design is especially important for tablet or mobile phone users, as it can
display the necessary information according to the preference of the operator
and prevent the needs of scrolling horizontally to find out all the data in a row.
In addition, similar to Excel and Google Sheets, our table offers row filtering
features for searching the target results as shown in Fig. 3.26.

Fig. 3.26. Requests list with filtering by Status

For every object type, for instance: TimeStamp and String, we have designed
a filtering field that suits for the type. To filter the submission or last modified timestamp, we have implement an interface with a calendar and time selector as shown in Fig 3.27. The operators can filter the cases that are on, before or after the selected timestamp.

For filtering by the status of requests, the operators can opt from is, is not or is any of for the filtering operator and select the status from drop down list. For instance, when the operator wants to view the cases which have successfully processed, he can filter the Status by is any of Return and Accept as shown in 3.27.

![Fig. 3.27. Requests list with filtering by Submission Date](image)

The filtering features provides an flexible searching features for operator to search based on the different data types on all the columns of the image.

Furthermore, in order to provide a comfortable view to satisfy all users preferences, we have implemented the density features, to allow the users to set the width of every row of the table.

Fig. 3.24 on last page has showcase the default density of the table, the height is set to middle among the three density option and design to suit majority of user preference.

To view more rows in a single page for convenience comparison, users can select the Compact density as shown in Fig. 3.28. The height of each row is reduced for displaying more rows on same page.

To enjoy a more comfortable view of the table, users can select the Comfortable Density as shown in Fig. 3.29. The height of each row is increased for enlarging the text and the padding of each row in the table.

Finally, for the simplicity of data analytic, visualisation and exporting the request information, we have implements export to CSV feature for the user to
view the list in Google Sheet or Microsoft Excel shown in Fig. 3.30. By simply clicking the export button, the data from the table will be phrase as a CSV file and downloaded on user device.

3.6.5 Inventory

The inventory page display all the lost items found by the operators as shown in Fig. 3.31 for the operators to simply check though all the images lost items. The received lost items can be classified as two categories, Send: items that have been returned to the owners and Inventory: items that have not returned to owners. Our system can help the user to filter out the items by clicking the inventory or send button to display the items belong to the respective category only.

To further filter the inventory to seek for a specific item, operators can filter based on the date of found, venues and one or more categories tags. This provides flexibility for the operators to view their targeted items only.

3.6.6 Manage Venues and Category Tags

Since the venues or category tags may changes over time, there is a necessarily for changing or deleting the venues and category tags. Therefore, we design a
Fig. 3.30. Screen capture of exported requests list

web page for managing the venues and category tags as shown in Fig. 3.32.

The page has clearly displayed the tags and venues in a table format, similar to the requests list on Section 3.6.4, the tables provides the features of hiding columns, filter by values, changing density of rows and export as CSV.

By clicking the information icon on the right of the table, the user will be directed to the inventory page in Section 3.6.5 that filtered by the specific venues or tags. This feature is created for convenience on seeking for the image that related to the target.

To change the name of tags or venue, operator can simply click on the name and change the value. To prevent confusion and inaccuracy static case by duplicated name, prior updating the name to database, the system checks and provides warning if there is an entry with identical name as shown in Fig. ??.

If the name is not taken, the system will ask the operators to confirm the action by listing out the changed or deleted names. Once the operators have confirmed and the database operation is completed, user will receive a confirmation on the top right corner.
Fig. 3.31. Screen capture of inventory web page

Fig. 3.32. Screen capture of web page for managing Venues and Category Tags

Fig. 3.33. Changing tag name to an existed name
3.6.7 Interactive web design

Since the operators may access the web application through computer, tablets or mobile phone, to ensure the UI of webpage fits into different screen size, we have implemented interactive webpage design on all pages.

When the operators is accessing the webpage with tablet or phone, the number of components displayed on a row will be reduced to increase the screen size assigned for the components. For instance, when the operators visits the Dashboard as state in Section 3.6.3 with tablet, some component will occupy the half of the row instead of one-third of the row on computer as shown in Fig 3.34. This ensures the operators can clearly obtain information on every pages though any devices.

![Dashboard accessed by tablet](image)

**Fig. 3.34.** Screen capture of dashboard accessed by tablet
3.7 Project Schedule

Table 3.3 covers the project schedule. The implementations of web application, matching and classifying models is completed on April, which meets the initial project schedules. The immediate next step for our team is to prepare on the project exhibition on May.

Table 3.3. Project schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Details</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 Oct 2022</td>
<td>Completion of Detailed Project Plan and Project Web Page</td>
<td>Completed</td>
</tr>
<tr>
<td>2 Oct 2022 to 30 Dec 2022</td>
<td>Implementation of the image matching machine learning model</td>
<td>Completed</td>
</tr>
<tr>
<td></td>
<td>Analysis on the performance of model by performance metrics</td>
<td>Completed</td>
</tr>
<tr>
<td></td>
<td>Modification and tuning of the model based on the performance metrics</td>
<td>Completed</td>
</tr>
<tr>
<td></td>
<td>Implementation of the backend database for storing the user request and the record of received items</td>
<td>Completed</td>
</tr>
<tr>
<td>9 Jan 2023 to 22 Jan 2023</td>
<td>Completion of preliminary implementation and detailed interim report</td>
<td>Completed</td>
</tr>
<tr>
<td>23 Jan 2023 to 31 Jan 2023</td>
<td>Preparation on the first presentation</td>
<td>Completed</td>
</tr>
<tr>
<td>1 Feb 2023</td>
<td>First presentation</td>
<td>Completed</td>
</tr>
<tr>
<td>23 Jan 2023 to 10 Apr 2023</td>
<td>Application for users to submit and tracking the application status</td>
<td>Completed</td>
</tr>
<tr>
<td>10 Apr 2023</td>
<td>Application for the operator to monitor and view the application details and the status of lost items</td>
<td>Completed</td>
</tr>
<tr>
<td>10 Apr 2023 to 18 Apr 2023</td>
<td>Completion of finalized tested implementation and Final Report</td>
<td>Completed</td>
</tr>
<tr>
<td>3 May 2023</td>
<td>Project Exhibition</td>
<td>In Progress</td>
</tr>
</tbody>
</table>
4 Future Plan

4.1 Image classification model

4.1.1 Online Learning

To ensure our model be able to learn autonomously and adopt to fast-changing data, our group plans to adopt online continual learning to keep updating the predictor for the data input after the deployment. To avoid class imbalance, a new type will be added only when at least 10 lost items with that type are found.

We have designed the methodology for applying the model with online learning by updating the model on a based on fixed-time interval (Fig. 4.1). We will input the photos added between the time interval to the top-layer of the model for fine-tuning. In case of there is a new category of lost items added, append a new dimensionality of as the output of the fully-connected layer of the network.

![Implementation methodology of online learning](image)

The difficulty of this update is when a newly added category will have a limited number of photos and induce an issue of class imbalance. We will resolve this
issue by upsampling the photo of new categories by Synthetic Minority Over-Sampling Technique (SMOTE) [11].

4.1.2 Multiple tagging results

Since the current classification only return the result with highest similarity, only 1 classification result is provided. Based on testing completed by friends, operators needs to manually fill in some more tags for classifying the issue, and color is the most commonly added tags. Our team is planning to develop an model to determine the color of the lost item and add it to the catagory tags to streamline the process.
5 Conclusion

Since the current lost and found process rely heavily on manually classifies and matches the items, our team have decided to bring up a cost-effective and efficient solution for the transport and property management operators and delivers a better user experience for the item owners. The project is aim to implement web application based on image matching and classification models to simplify the entire process. The project is expected to improve the user experience and reduce the operational cost of the process by delivering a reliable, efficient and user-friendly solution to the industry.

We have completed the application design and implementation though analysing the workflow for both operators and item owners. In addition, we have experimenting implementing the matching model using plain KNN, KNN with AutoEncoder and KNN with image classifier by ResNet-50, and adopted ResNet-50 as the classifier due to the performance and efficiency. Furthermore, we have completed the training of image tagging model with data augmentation and learning rate decay. Currently the application alongside with the models are ready to use, the stumbling block awaiting us to surmount is to implement online learning on the classification model with high accuracy. We will analyse the validation accuracy graph and fine tune the dependencies of models to achieve high accuracy of model.
References


