Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 45 No. 11
November 2024

Lost Sharing Web Service Based On Image Classification Model

Seok-WooJang[1], Jeong-Joon Kim[2*]

[1]Dept. of Software, Anyang University, Anyang-si, Gyeonggi-do, Republic of Korea

[2,*] Corresponding Author, Dept. of Software, Anyang University, Anyang-si, Gyeonggi-do, Republic of Korea

[1]swjang@anyang.ac.kr, [2*]jjkim@anyang.ac.kr

Abstract

Losing personal belongings is a common occurrence, yet the recovery rate of lost items remains alarmingly low. For instance, in South Korea, the recovery rate of lost phones is merely 3.2%, despite the efforts of mobile carriers and the police. Existing lost and found platforms, such as community-based applications, lack the necessary features for effective recovery, resulting in limited success. To address this issue, this paper proposes a lost and found sharing web service that utilizes image classification models to improve the identification and recovery of lost items. By enabling users to upload images and detailed information about lost or found belongings, this service enhances communication between users and increases the likelihood of successful recovery. The proposed system combines a user-friendly community forum with advanced image matching technology to better connect lost items with their rightful owners. The paper presents a detailed system design and discusses the potential impact of the proposed service on improving lost item recovery rates.

Key Words :Lost and Found System, Image Classification, Item Recovery, Community Sharing Platform, Lost Property Management

1. Introduction

Many people have experienced losing personal belongings at least once in their lives. If it's something trivial, losing it might not matter much, but if it's something valuable like a phone or wallet, people will spare no effort to find and lament their lost items. However, looking at the annual statistics, the recovery rate at Lost and Found Centers is alarmingly low. According to the Korea Association for ICT Promotion (KAIT), the 'Cell Phone Search Call Center' manages to reunite owners with about 30,000 lost phones annually, while the total number of loss reports received by the three major mobile carriers amounts to approximately 1.08 million, translating to a mere 3.2% success rate. This means that out of every 100 people who lose their phones, 97 are unable to recover them. Moreover, various community platforms such as the used goods trading applications 'Danggeun Market' and 'Facebook' have assumed the role of Lost and Found Centers. While these platforms leverage the ability of many people to quickly and easily share information, they do not align with the intended purpose of traditional Lost and Found Centers, leading to lower recovery rates. Thus, these community sites and applications have inadvertently become outlets for lamenting lost items [1, 2]. However, there is currently no dedicated lost property program available. Although the National Police Agency provides the 'Police Lost 112' service, it only handles information on lost items reported to the police, leaving out unreported lost items. Additionally, even for reported found items, there are challenges such as lack of images, detailed information, and difficulty in communication between users, making it easier for criticism rather than praise to be found among users.

The key to improving the recovery rate of lost and found centers lies in providing accurate and convenient information about lost items and facilitating effective communication with finders. It is essential to clearly convey the precise location where items were found and provide clear images of lost items. Utilizing a community forum format for lost and found centers where users can easily communicate and share information would yield more meaningful results. Moreover, leveraging image classification models to detect and match image data with specific items would facilitate convenient sharing of information about lost

items, potentially increasing the recovery rate of lost items.

Therefore, this paper aims to develop a lost and found sharing web service based on image classification models, allowing users to conveniently share information about found and lost items directly. The paper is structured as follows: Chapter 2 reviews related research, Chapter 3 describes system design and implementation, and Chapter 4 concludes by discussing the expected impacts of this study.

2. Related Research

2.1 Image Classification Model

An image classification model is a type of machine learning model provided by TensorFlow, considered a branch of artificial intelligence that improves automatically through experience. The study involves computer algorithms that automatically improve and are considered a field of artificial intelligence. The first key of the image classification model is representation and generalization. Representation is the evaluation of data, and it is classified as an image with ImageNet database labels. Generalization is the processing of data for unknown data and transfer learning. Another key concept of the image classification model is the scale of data, accuracy of data, data processing speed, and effective model design.[3]

2.2 Web Crawling

Web crawling refers to the act of fetching a web page and extracting data from it. It is essential for search engines, which use it to automatically search and index various information on the web. This process is also known as spider, bot, or intelligent agent. Instead of people manually searching for information on websites, computer programs continuously find and aggregate new web pages according to pre-set methods, using the found results to discover new information and add it to the index. While it efficiently retrieves vast amounts of data, it also has drawbacks such as the potential for manipulation of rankings through misuse of bot search capabilities or evasion of searches. Major search engines like Naver and Google utilize such bots for their operations [4, 5].

2.3 Transfer Learning

Transfer learning refers to utilizing the capabilities learned by a neural network in one specific domain to aid in the learning of a neural network used in a similar or entirely new domain. In the context of image classification, networks like ResNet or VGG consist of CNN layers at the front end. These CNN layers are adept at extracting features from images, initially identifying shapes, then patterns, and finally higher-level features. This ability of neural networks to extract image features can be leveraged across different fields. Specifically, transfer learning involves using the feature extraction capabilities of highly performing networks like ResNet or VGG, which have been trained on tens of thousands to millions of images. Typically, only the last output layer, often a linear layer, is modified and retrained for the new task. This allows the network to adapt its learned features to the new data, thereby speeding up training and improving performance [6, 7].

3. System overview

The interface for implementing the web page was developed using JSP. Apache Tomcat v8.5 served as the web server, MySQL 8.0 was used as the database, and Eclipse 4.14 was the IDE used for development.

The web page system architecture for this service is depicted in Figure 1.

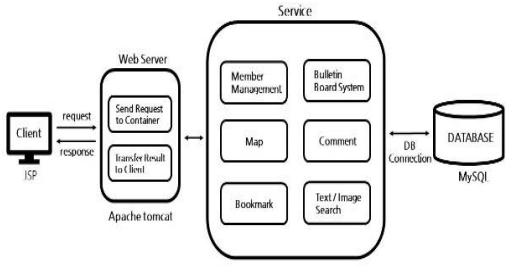


Fig. 1. Web Page System Diagrams

As seen in Figure 1, the web page is structured around membership management and bulletin board functionalities, enhanced with features for more detailed sharing of lost items, including road address and image search capabilities.

To register as a member on this web page, users need to provide their ID, password, name, gender, and email. To prevent duplication of IDs during registration, the userID field has been set as a primary key in the user table. The specification of

the user table responsible for membership functions is shown in Figure 2.

▶	1	userID	varchar	varchar(20)	PRI	NO
	2	userPassword	varchar	varchar(20)		YES
	3	userName	varchar	varchar(20)		YES
	4	userGender	varchar	varchar(20)		YES
	5	userEmail	varchar	varchar(50)		YES

Fig. 2. user table specification

The bulletin board is divided into a "Lost and Found Board" for sharing lost items and a "Free Board" for freely writing posts. Logging in is required to write posts on any board. Each post consists of a title, content, author, posting time, and optionally an attached photo. Figure 3 shows the specification of the bbs table responsible for bulletin board functionality

▶	1	boardID	int	int(11)	YES
	2	bbsID	int	int(11) PRI	NO
	3	bbsTitle	varchar	varchar(50)	YES
	4	userID	varchar	varchar(20)	YES
	5	bbsDate	datetime	datetime	YES
	6	bbsContent	varchar	varchar(2048)	YES
	7	map	varchar	varchar(30)	YES
	8	bbsAvailable	int	int(11)	YES

Fig. 3. bbstable specification

In the Lost and Found Board, an address feature was implemented to specifically indicate the location where an item was found or lost. Users can either type the address directly or use the address search button to find the address. This was implemented by obtaining an open API from the road name address site operated by the Ministry of the Interior and Safety.

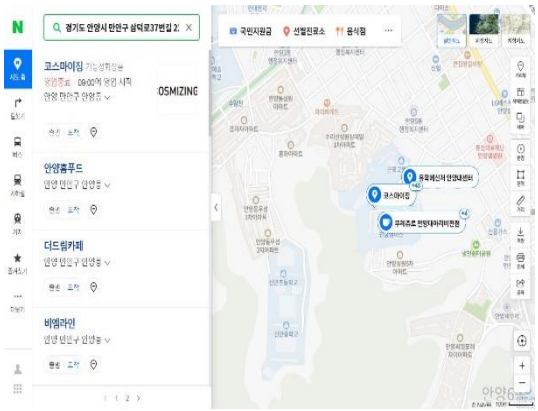


Fig. 4. NaverMap

When you enter a keyword for the address you want to input, a list of related addresses is displayed. Selecting the desired address will input that address. When you click the address section in Figure 4, a Naver map window with the searched address is generated.

Each member can bookmark posts. The bookmark feature allows members to save and manage posts they want to keep. By clicking the heart button in the top right corner of a post, the post is bookmarked to their account, and the heart button becomes active. Bookmarked posts can be viewed in the membership management tab after logging in.

One of the issues mentioned in the introduction regarding existing lost and found services is the lack of communication between finders and losers. To address this problem, this service implements a comment feature that allows users to communicate with each other.

Functionality, This web service provides an image search functionality. If a user has a photo of the item taken before it was lost, they can use the photo to search for their lost item.

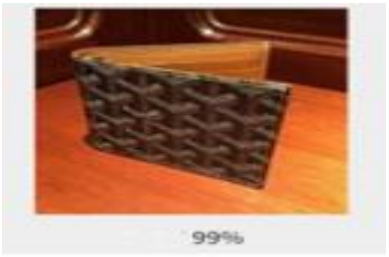


Fig.5. Image search

Figure 5 is an example of the image search functionality in this web service. Users can upload a photo of the lost item they are looking for by clicking the upload button at the bottom right. Once the image is uploaded, it is analyzed, and the results are displayed.

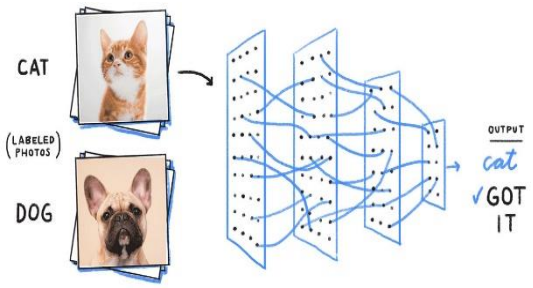


Fig. 6. Result of image classification

Figure 6 shows the console window of a page where a random wallet photo has been uploaded. In this project, the image classification model was implemented using seven datasets, and it can be observed that the wallet category has the highest probability at 99%. After the image analysis is completed, clicking the "View Results" button initiates a search based on the uploaded photo. A detailed explanation of the image search implementation will be provided in the next chapter.

```
// Load the model.
mobilenet.load().then(model => {
// Classify the image.
model.classify(img).then(predictions => {
console.log('Predictions: ');
console.log(predictions);
});
});
```

Fig. 7. Image classification model code oftensorflow

4. Implementation

To implement the image search system, this paper uses the TensorFlow image classification model. Web crawling was conducted using Python version 3.8 and Anaconda version 4.8.3, and Teachable Machine was utilized as the transfer learning model. The image search system in this paper consists of four components: the 'TensorFlow Model,' the 'Image Classification Model,' 'Web Crawling,' and 'Transfer Learning.'

The image search system is designed to help users share lost items more effectively by allowing them to search for lost or found items using images. The system utilizes TensorFlow.js image classification models to categorize items in images and express the categories with a confidence score. Focusing on statistically common lost items—'phone,' 'wallet,' 'AirPods,' 'ring,' 'backpack,' and 'handbag'—a dataset of 100 images per item was collected using Python and Anaconda for web crawling. This dataset was then used with the Teachable Machine image classifier tool to further train the model on these seven categories of lost items, resulting in an image classification model specifically tailored for lost item detection.

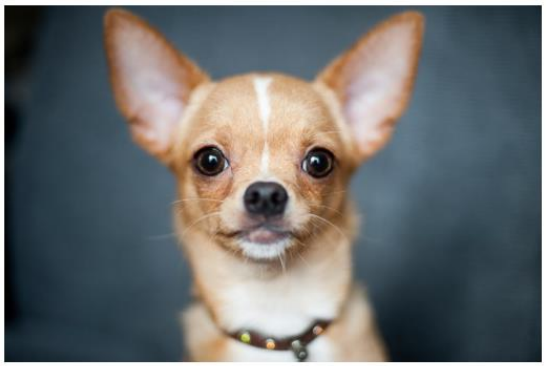
TensorFlow is a core open-source library that aids in developing and training ML models. It offers a comprehensive and flexible ecosystem comprised of tools, libraries, and community resources, enabling the implementation of cutting-edge technology in ML and the easy building and deployment of ML-integrated applications. TensorFlow provides libraries tailored to the user's development environment, including Colab, JavaScript, mobile and IoT, and production environments. For the development of this system, the JavaScript model, TensorFlow.js, was used to train and deploy models in a web-based environment through browsers and Node.js.

TensorFlow.js offers a variety of configuration options. For users familiar with concepts such as tensors, layers, optimizers, and loss functions, TensorFlow.js provides flexible building blocks for neural network programming in JavaScript.

TensorFlow.js provides open-source models for tasks such as image classification, object detection, body segmentation, and pose estimation. Among these, we used the image classification model, which classifies and labels the image.

The TensorFlow image classification model "MobileNet" is a compact, low-latency, low-power model that is parameterized to meet the resource constraints of various use cases. It can be built for classification, detection, embedding, and segmentation, similar to other popular large-scale models like Inception. MobileNet balances trade-offs between latency, size, and accuracy while favorably comparing to popular models in the literature. This model does not require prior knowledge of machine learning and can take browser-based elements (img, video, canvas) as inputs, returning an array of the most likely predictions and their associated confidence levels.

Using the getElementById code, the image element is retrieved, and the MobileNet model is loaded. Next, the image is classified, and the most probable predictions along with their accuracies are predicted and logged to the console as an



array.

Fig. 8. Image classification model of tensorflow

As shown in Figure 8, the image classification model identifies, classifies, and labels the image accordingly. For instance, it classifies the Chihuahua image in Figure 8 with the Chihuahua

label and displays the accuracy. The TensorFlow.js image classification model is trained with a database of various labels, allowing it to classify items with relatively high accuracy when users provide appropriate photos. However, there are issues with using the raw data directly. To classify lost item images, labels for those images are necessary, but the current model may have missing labels or show low accuracy.

Before additional training of the lost item images, a dataset for statistically common lost items such as 'phone', 'wallet', 'AirPods', 'backpack', 'ring', 'handbag', and 'wristwatch' was needed. Therefore, a crawling task for lost item images was carried out. Google Image Search Engine was selected as the target for crawling, and the data was loaded in image document format to download image files. The crawling task was performed using Python and Anaconda, as they offer excellent usability and simplicity for libraries used in crawling.

```
1   pumjong = {
    "lost": ["cell phone", "wallet", "watch", "airpod", "handbag", "backpack", "ring"],
    }


    def crawling(target_name):
    driver.get("https://www.google.co.kr/imghp?hl=ko&ogbl")
    elem = driver.find_element_by_name("q")
    elem.send_keys(target_name)
    elem.send_keys(Keys.RETURN)
    # (Seconds) Increase this number if your network is slow
    SCROLL_PAUSE_TIME = 5
    NUMBER_OF_PICTURES = 100 # Increase this number if you want to get more pictures
    # Get scroll height
    last_height = driver.execute_script("return document.body.scrollHeight")
```

```
1     count = 0
    while count < NUMBER_OF_PICTURES:
    # while True:
    # Scroll down to bottom
    driver.execute_script(
    "window.scrollTo(0, document.body.scrollHeight);")
```

Fig. 9. def functioncode needed to build a dataset.

The following steps outline the process to download folders for seven types of lost items: 'phone,' 'wallet,' 'AirPods,' 'backpack,' 'ring,' 'handbag,' and 'wristwatch.' First, the items are specified, and the information is sent to the Google Image Search server via URL. To download the image files, the crawling function designates the Google Image Search window using the driver.get key. Since searching with English names yields more accurate data results, the key values are set to be collected in English. The items are searched using the name tag "q" in the Google Image Search window, and the key values are sent. To ensure smooth crawling even with a slow network, SCROLL_PAUSE_TIME is set to 5. Since 100 images are downloaded for each item,

```
    driver = webdriver.Chrome(ChromeDriverManager().install())
    for key in pumjong:
    os.makedirs(key, exist_ok=True)
    os.chdir(key)
    for val in pumjong[key]:
    os.makedirs(val, exist_ok=True)
    os.chdir(val)
    crawling(val)
    os.chdir('..')
    os.chdir('..')
    driver.close()
```

NUMBER_OF_PICTURES is set to 100.

Fig.10. Drivercode

To execute the crawling function, the chromedriver.exe file is required. Therefore, you need to install ChromeDriver and set it up. Then, create directories using the predefined key names. Since the key name is designated as "lost items," you will access the lost items folder and create subdirectories for

```
pip install selenium
python crawl_google_image_py
d:
cd Coding\teensorflow.js\crawling
python crawl_google_image_py
```

each value before starting the crawling task.

Fig. 11. Crawling execution code

After completing the Python crawling code, you need to switch to the Anaconda development environment to execute the code. Run the

Anaconda Prompt and navigate to the directory where you wrote the Python crawling code. Then, execute the crawling file using the command python crawl_google_image.py. However, the first time you run it, it may not execute correctly because the Selenium library is not installed. Therefore, install the Selenium library by running pip install selenium, and then execute the crawling file from the directory where you wrote the Python crawling code.

Fig. 12. Result of lost and found data set crawling

Once the crawling task is completed, check the image files in the generated directories to ensure accurate classification by the image classification model. Identify any unsuitable photos and delete them, keeping only the appropriate ones.

Sophisticated deep learning models have millions of parameters and training them from scratch often requires extensive computing resources and data. Transfer learning is a technique that builds new models by reusing parts of a model that has already been trained on related tasks. By leveraging the pre-existing knowledge of models trained to recognize thousands of different objects within images, transfer learning allows the detection of specific image classes with much less training data than originally required for the base model.

To build a custom image classifier that trains in the browser using TensorFlow.js, the Teachable Machine development tool is used. Teachable Machine is a web-based tool designed to quickly and easily create machine learning models, and it is widely used for transfer learning image classification. The pre-collected files are grouped into classes or categories, and the model is trained and exported.

The model is then trained to classify these categori es. After training, the model code is exported for t esting.

Fig 13. Result of cell phone learning

Fig.14. Result of cell backpack, watch learning

Using the model trained with Teachable Machine to classify the seven categories of lost items, the results show that the model can classify new images, not just the sample images used in training, with high accuracy.

5. Conclusion

In this paper, we proposed a method for users to search for lost items easily and efficiently by using a high-accuracy image classification model extracted from Teachable Machine instead of the basic TensorFlow image classification model. While commonly used transfer learning methods such as the TensorFlow image classification model and object detection model are effective, this paper utilized a modified version of the image classification model specifically designed for lost item categorization.Experimental results using the TensorFlow image classification model with datasets of seven categories of lost items (ring, watch, wallet, phone, AirPods, handbag, backpack) showed that the model failed to classify handbag, ring, and AirPods images. Although it successfully classified phone, wallet, backpack, and watch images, the accuracy was below 70%, indicating low performance in terms of accuracy.On the other hand, the transfer learning approach demonstrated generally excellent accuracy even with new images not included in the sample set. The trained lost item classification model was able

to effectively classify and search for items based on their categories. This functionality allows users to conveniently search for various lost item categories in posts, enhancing the quality of service provided to them. Moreover, the free and convenient sharing of lost item information can significantly increase the recovery rate of lost items.

References

[1] Se-Jung Park, “Call me if you pick up AirPods in your neighborhood.”,Herald economy. on News comprehensive, March 2021.DIO:http://mbiz.heraldcorp.com/view.php?ud=20210316001111

[2] Young-Joo Kim, “There was a reason to find the owner of only 3 out of 100 lost phones.”, News specializing in telecommunication broadcasting, Mar August 2016. DOI: https://m.etnews.com/20160817000453

[3] TensorFlow. “Tensorflow image classification tutorial.”, Google Developers Site Policies., Vol. 2021-08-16 UTC.DOI: https://www.tensorflow.org/?hl=ko

[4] IT-Wanderer “Crawling for data collection Part 1: What is Crawling?”, Crawling posting for series/data collection, No. 1, March 2019.DOI: https://jcdgods.tistory.com/317

[5] Sae-Rom Kim, “No more boring searches! "Crawling" that collects information quickly.”, The Ministry of Science and ICT blog, March 2020.DOI https://scienceon.kisti.re.kr/srch/selectPORSrchArticle.do?cn=JAKO201926358474432&dbt=NART

[6] So-Young Jeong, Min-gyo Jeong “Pedestrian Classification using CNN's Deep Features and Transfer Learning”, Journal of Internet Computing and Services, v.20 no.4 , pp. 91-102, Sep 2019.

DOI:https://scienceon.kisti.re.kr/srch/selectPORSrchArticle.do?cn=JAKO201926358474432&dbt=NART

[7] TensorFlow. “Transfer learning image classifier.”, Google Developers Site Policies., Vol. 2021-08-16 UTC.DOI: Transfer learning image classifier.