IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 18, 2018, accepted August 3, 2018, date of publication August 17, 2018, date of current version September 7, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2865490

Image-Matching Based Identification of Store Signage Using Web-Crawled Information

CHENYI LIAO[1], WEIMIN WANG[2], KEN SAKURADA[2], AND NOBUO KAWAGUCHI[1,3]

[1]Graduate School of Engineering, Nagoya University, Nagoya 464-8601, Japan
[2]National Institute of Advanced Industrial Science and Technology, Tokyo 100-8921, Japan
[3]Institutes of Innovation for Future Society, Nagoya University, Nagoya 464-8601, Japan

Corresponding author: Chenyi Liao (liao@ucl.nuee.nagoya-u.ac.jp)

This work was supported by JSPS KAKENHI under Grant JP17H01762.

ABSTRACT We address automatic matching of street images with relevant web resources to enable the identification of store signage in street images. Identification methods for signage usually involve image matching, which attempts to match query images to other similar viewings using pre-labeled copies from a target data set. Manual target data set, such as a fingerprinting database can ensure high-quality data but collected data must be fed manually, which significantly adds costs. Utilizing web-crawled information is a way for automatic data set generation at lower cost, however, imbalanced and noisy data can adversely affect identification accuracy. Our work aims to resolve these issues. We propose a signage identifier in Web-crawled information – SIWI. The SIWI includes a web image data set construction method, which can self-generate high-quality data sets through automated web-mining, including data filtering and pruning strategies, which effectively reduce the identification error caused by noise, imbalance, and insufficient data. Furthermore, by applying a Hybrid Image Matching method that combines the deep learning approach with the feature point matching to signage identification without Optical Character Recognition, it can handle arbitrary signage designs. Because there is no specialized training involved, the same process should also work for any other locations without manual adjustment. An experimental result achieves 91% accuracy in a real-life application, which confirms its effectiveness.

INDEX TERMS Web mining, data set generation, image matching, store signage identification.

I. INTRODUCTION

The challenge to match street images with relevant web resources is a crucial task with various potential applications. Identification of store signage in street images is a prerequisite for this task, as it provides an interactive user experience, such as a virtual shopping environment [1]. As illustrated in Fig. 1, visual information of a street image alone is not enough. It is also necessary to identify and link store signage found within street images to external web resources, and to do so in a way that is as accurate and cost-efficient as possible.

The most straightforward means to accomplish this is to manually annotate the relevant metadata. A technician can manually identify and label every store as a Point of Interest (POI) in the street images [2]. While simple, this process adds significant cost to the otherwise relatively cheap process of capturing the street image, which can be done reliably through automatic procedures. This also creates an ongoing maintenance issue, as the technician must manually re-adjust labels every time a store location or tenancy changes.

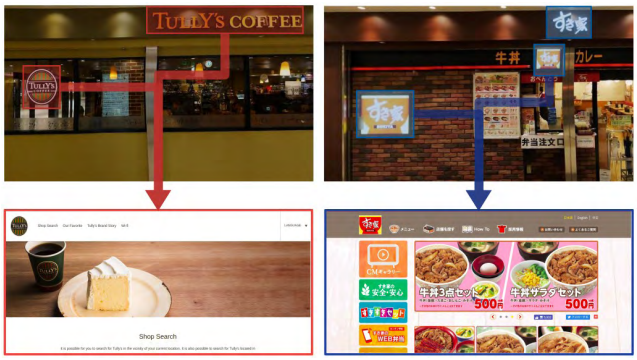

FIGURE 1. Signage is identified and linked to store websites. The user can access the websites to obtain information or order products from the store.

There exist a number of automated approaches that aim to resolve this issue. The web resource contains textual data and image data. Matching of web resources can be classified into the following two categories: textual-matching

based methods [3]–[7], and image-matching based methods [8]–[14]. Textual-matching based methods depend on an Optical Character Recognition (OCR) process, which focuses on recognizing text characters from signage and can then compare against a candidate store lexicon. This method is not suited for cases involving characters with special font types, non-characters, or characters from a language group different from what the OCR engine specializes in. The image-matching based method is dependent on specialized target datasets, which contains a group of pre-collected images, that it will then attempt to match the query image with. For signage identification, the images in the datasets need to be labeled with their associated store IDs. Manual dataset establishment such as fingerprinting [13], [14] can create well balanced and accurate target datasets, but they must be created and kept up to date manually. Deep learning methods [8]–[12] achieve human level accuracy recently but require millions of high quality training data. Web-mining [15] can be used to automatically assemble the target dataset from web resources but noise and a lack of sufficiently balanced data affect the accuracy of signage identification. A more detailed analysis of the limitations facing existing methods can be found in Section II.

In this study, we aim to address these limitations with an automatic store Signage Identifier in Web-crawled Information (SIWI), which requires only a list of store names and their corresponding web addresses as input. The list is manually obtained from the official websites of our experimental shopping mall. The street images from the same shopping mall are inputted without labeled signage, which is to be identified. In Section IV, we present an automatic web-image dataset collection method, which crawls images related to a store based on the provided URL and web-search images from the provided store names. The naive crawling process generates much unrelated information, such as irrelevant images from an image search engine or non-logo images from a store's website. We present several methods to deal with these problems: A two-step method (in Section IV-A) which first sifts out relevant storefront images from web-search images using VGG16 fine-tuning [12], and then crops out the desired signage patches using YOLO [16], [17]. Meanwhile, a statistics-based method (in Section IV-B) extracts webpage logos from websites. These two methods can effectively reduce the amount of irrelevant information obtained from data crawling. The same method used in Section IV-A is then used on street images to generate the query signage dataset in Section V.

In Section VI, we present a Hybrid Image Matching (HIM) method that combines the deeplearning approach with the feature point matching for signage identification. In Section VI-A, we present a pruning algorithm to link each signage patch from the query dataset to their candidate matching datasets. As each dataset is only a small subset of the total number of images obtained from the web-mining results, this effectively reduces the processing time required during the feature point matching process [18]–[20], which is based on RANSAC [21] and is detailed in Section VI-B. As a result, the signage in street images is output with identified store names and web addresses. In Section VII we evaluate how the SIWI works in a real-life scenario.

In this study, our contributions can be listed as follows:

1) We utilize web-mining to automatically generate target image datasets for signage matching at a low cost. Because the most web-crawled images contain much irrelevant information, and also because of the sheer number of possible combinations between web-crawled images and street images, we propose a series of data pruning and filtering methods. In our experiment, the method effectively reduce 83.46% and 98.86% of irrelevant images from 55,783 web-search images and 18,381 webpage images, respectively. A candidate matching dataset generation mechanism selects 390 target signage images for each matching and ensures that the matching process can be performed within reasonable computing times.
2) We propose a pure image-matching based method of store signage identification without involving OCR processes. This method overcomes the limitations that textual matching methods have regarding special font types, non-characters, and multilingual texts. Also, the proposed method dose not depend on a specialized training process. It is designed to be deployable to other various locations without any manually annotated training data.

II. RELATED WORK

In this section we investigate related literature about business or signage identification from in natural scenes. Because signage in a natural scene is typically matched with text or images in data resources, we organize the literature into groups of textual-matching based methods and image-matching based methods.

A. TEXTUAL-MATCHING BASED METHODS

Textual matching–based methods [3]–[7] use OCR engines, which convert signage into text. A local lexicon is generated in advance using store names and other relevant text information. The system then searches for the recognized text from within the local lexicon. When the text of a signage matches the text in the lexicon, the system assigns the corresponding store ID to the signage.

The performance of textual matching methods relies upon the performance of OCR engines. Under ideal conditions, in which the OCR engine can accurately recognize signage text from natural scenes, textual methods are closer to human behavior and better than image matching both in accuracy and speed as symbolical retrieval methods. However, in reality, signage from the natural scenes is not only always presented in an easy to recognize style. There is also a large percentage of signage with multi-language characters, specialized font, and non-character logo. The performance of OCR engines is significantly limited by multi-language and individual fonts.

It is also limited regarding non-character signage recognition. Therefore, there are obvious limitations when applying textual matching to signage identification tasks.

B. IMAGE-MATCHING BASED METHODS

Image-matching based methods attempt to match the query image with visually similar images in a target image dataset. It is important to choose appropriate image matching methods according to various types of target image datasets.

Fingerprint data collection is a method for target image dataset establishment. The main idea of fingerprinting is to identify user-uploaded street images by matching them against manually labeled fingerprint street images [13], [14]. Manually labeling the data is the most straightforward way to clean up a dataset. For example, in Xu *et al*.'s approach [13], three storefront images of each store are taken and labeled with store IDs manually to build the fingerprint database. When a user uploads an image that is taken by mobile phone to a server, the server retrieves the most similar signage from the fingerprint database and returns store information to users. The advantage of a fingerprinting method is that it can identify diversified target objects by matching them with a small number of labeled target images. Also, fingerprinting methods avoid many visual artifacts from natural scenes and can reach high identification accuracy because user-taken images and fingerprint images are usually shot from the same locations so that they share the conditions of the shoot and domains. This benefits feature point extraction-based image matching methods such as the Bag of Visual Word (BoVW). However, it requires manually collecting and labeling image data in advance, so the technique cannot automatically re-adjust labels when store location or tenancy changes.

Deep learning methods [8]–[12] have achieved unprecedented success on object identification recently. Deep learning methods in particular are known for requiring large quantities of training instances, without which overfitting occurs. Existing datasets support high quality and large target datasets by manual labeling or data filtering. Wojna *et al*. [9] benchmarked their street sign identification with the French Street Name Signs (FSNS) dataset [22], which contains over one million labeled images of visual text. Movshovitz-Attias *et al*. [10] focuses on identifying store categories (restaurant, gas station, etc.) from street level imagery. They extract three million target images consisting of 2,000 categories from Google Street View. To ensure sufficient training data per category, they omit labels whose frequency is very low, resulting 1.3 million samples and 208 unique categories. Yu *et al*. [11] address the problem of detecting storefronts in street level imagery. They label about approximately two million panorama images through a crowd-sourcing system. Deep learning methods avoid many visual artifacts from natural scenes and can achieve human level accuracy. However, acquiring a large target dataset of high quality labeled data for training is a challenging task, and it is also impossible to ensure that the ready-made datasets can cover all scenes, as store identification is sensitive to location. This leads to unavoidable human intervention and specialized training to deploy the process is deployed to other locations.

Another feasible means for target dataset collection is by utilizing the web-mining technique. By ''recycling'' web resources, web-mining can dynamically establish local target dataset at lower costs. Zamir *et al*. [15] present a multimodal method that combines a textual matching model and an image matching model to automatically identify stores in an urban image. The image matching model also utilizes web-mining images from web resources. In the identification step, the two models separately generate Probability Distribution Function (PDFs) for each store. The two PDFs obtained from the textual matching and the image matching models are fused. The multimodal method is more robust than textual matching or image matching alone. The two models make up for each other's shortcomings. When the OCR engine has difficulty recognizing characters in a signage, the image matching model may locate a match from corresponding web-images. Likewise, when a storefront image does not exists on the web, the OCR engine may be able to recognize the store name directly. However, two issues remain unaddressed in this multimodal method. First, searching for web images by store names may generate a significant number of uninformative images (ones without a relevant signage). This work attempts to increase the proportion of relevant storefront images in search results by including specifying keywords such ''store name'', ''store name + city'' and ''store name + storefront''. Nevertheless, there remains a sizable percentage of uninformative images included in the search results. Noisy data affect the performance of store identification. Web-mining without data filtering is a limitation of this method. Second, a weighted average is employed during PDF fusion. Two weight values determine the relative reliance on either the textual matching model or the image matching model. The weights are obtained by training 50 labeled query images in the location. Therefore, the weights are specific to a given location. In other words, if the method is deployed to another location, its fusion weights must be re-trained. Continuous manual re-training remains an issue in the method.

Another way for target dataset collection is crowd-sourcing [23], [24] . Although the crowd-sourcing may produce noise, the costs are much lower than data collected by technicians. It can also ensure data up-to-date. However, for an explicit crowd-sourcing system, finding volunteers is costly. Web-mining can also be considered as an implicit crowd-sourcing. In this work, we challenge to obtain the web mining, which is an almost free strategy, and try to reduce the noise in the web-mining data as much as possible.

III. PROCESS OVERVIEW

The pipeline of the SIWI is shown in Fig. 2. The terminologies in this paper are defined in TABLE 1. The method is divided into three main components as follows:

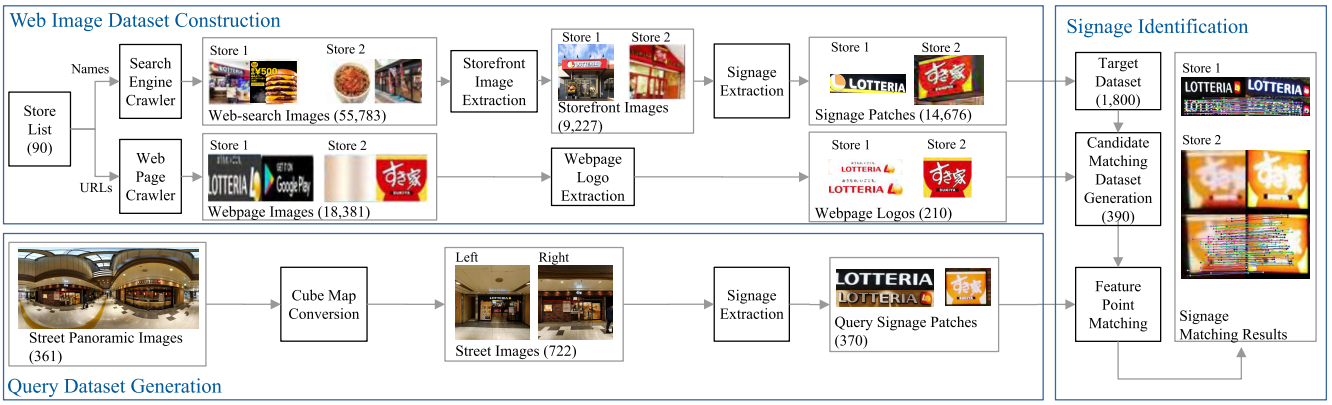

FIGURE 2. **Pipeline of the SIWI. The method is an image-matching based method, which automatically collects data from two Internet resources – the search engine and stores' websites using web-mining to construct a web image dataset as the target dataset. The Query Dataset is generated through extracting Query Signage from the Street Panoramic Images. The ultimate goal is to identify and assign the correct store ID to each Query Signage by matching them with target signage. The numbers in the brackets represent the numbers of each object.**

TABLE 1. **Terminology conventions used in this work.**

Noun	Description
Street Panoramic Images	The original panoramic images shot from the shopping street without store ID labeling.
Street Images	Images extracted from the *Street Panoramic Image. These contain Signage Patches that need to be identified with store IDs.*
Web Images	Images obtained from the Internet, which include both *Web-search Images and Webpage Images.*
Storefront Images	Images that show the storefront of a store. *Street images by definition all meet this criterion. However, the term Storefront Images when used in the context of this paper refers specifically to the subset of Web-search Images that also meet this criterion.*
Signage Patches	The portion cropped from *Street Images or Storefront Images that has the store's signage on it.*
Webpage Images	Images extracted from store webpages.
Webpage Logos	A subset of *Webpage Images that contains the representing logo of a store or company that owns the website.*

A. WEB IMAGE DATASET CONSTRUCTION

A store list including store names and URLs is required as input. The store list is easily obtained from the website of the mall or current geography information service such as Google Places [25]. An *search engine crawler* downloads web images from Google Images for each store using their store names as the keywords. Each web image is then assigned a store ID. These raw web images may include many irrelevant images such as product images. A *storefront image extraction* process attempts to extract only the storefront images from the original web-search images. Then, *signage extraction* is used to crop out signage patches from storefront images. At the same time, a *webpage crawler* downloads all webpage images from each store's official website. The signage patches and webpage logos thus generated from the basis of our target signage dataset. Detailed discussions of web image dataset construction are provided in Section IV.

B. QUERY DATASET GENERATION

In this work, query signage images is generated from street panoramic images in the query environment. Panoramic images are collected by a NavVis M3 Trolley [1].

The NavVis M3 Trolley is a fully comprehensive indoor data capture device. Designed for frequently indoor scanning and mapping work, NavVis can automatically generate 3D indoor maps using built-in LiDAR and panorama camera. However, the POIs on the 3D indoor maps must be manually annotated in this application.

A *cube map conversion* algorithm converts the street panoramic images into a cube-map of the six cardinal directions – left, right, front, back, top, and bottom. Because the shooting direction of NavVis is parallel to the passage, we use only the left and right images, which faces the storefront, as query street images. Using the same *signage extraction*, query signage patches are cropped from the street images. Detailed discussions of query dataset generation are provided in Section V.

C. SIGNAGE IDENTIFICATION

Once we have the target image database and query signage patches, the goal now becomes one of correctly identifying each query signage patch with its counterpart in the target image dataset so that it can be assigned a correct store ID. However, at this stage, there exist a large number of both target signage data and query signage patches. Directly matching each query signage to each image in the target signage dataset takes an unreasonable amount of processing

time. Therefore, we propose a Hybrid Image Matching (HIM) method, which create an individual candidate matching dataset for each query signage. This narrows the candidate matching images to only a relatively small number 390. It contains 180 signage patches (two signage patches from each store) extracted from 55,783 web-search images and 210 web logos extracted from 18,381 web images. Detailed discussions of signage identification are provided in Section VI.

IV. WEB IMAGE DATASET CONSTRUCTION

The web images are obtained from image search engines and the stores' official websites. They are called web-search images and webpage images, respectively. The goal in this section is to extract signage patches from web-search images and store logos from web-page images.

A. WEB-SEARCH IMAGE COLLECTION

Web-search images are downloaded from a search engine by a crawler (Section IV-A1). The uninformative web-search images, such as images of products sold by the store, must be deleted. We wish to keep only the storefront images, which have the highest likelihood of containing the store's signage (Section IV-A2). After that, signage extraction (Section IV-A3) is used for cropping signage patches from storefront images.

1) WEB-SEARCH IMAGE CRAWLER

A search engine crawler automatically downloads web-search images using keywords. We use icrawler [26] to download web-search images from Google's image search engine. In regards to the choice of keywords, we adhere to Zamir *et al.*'s suggested method [15] of appending keywords such as ''store location'' to the store names to get a larger number of matching images. In addition, if a store name is represented in Japanese, we add its phonetic spelling (Romaji spelling) as an extended keyword using a morphological analysis tool (MeCab [27]). Note that the sequence of search results is informative. Search engines rank the search results based on correlation with keywords. Therefore, search results near the top have a higher relevance to the keyword. Effort should be made so that the relative sequences of all search results from all keywords are preserved when those results are merged into one dataset.

In this work, our experiment is carried out in shopping mall A in Nagoya, Japan. There are 90 stores in shopping mall A, all of which have their names listed conveniently for our use on shopping mall A's website. We limit our search results to the first 300 images larger than 200 × 200 pixels for each keyword. The first 300 photos are the most relevant to the keywords. Since we crop out the signage patches from the web-search images after this, the size setting ensures available resolutions. As a result, we obtain a total of 55,783 images from the 90 stores. The selection of keyword can be considered one of the factors, which can affect the number and quality of images in the web-search images dataset. We crawl web-search images using the ''store name'' and ''store name + shop'', respectively. We extract the first 20 web-search images from each store and isolate the storefront images in a classifier, which we discuss in section IV-A2. As a result, the keyword ''store name'' can obtain 604/1800 (33.56%) storefront images and the keyword ''store name + shop'' can obtain 772/1800 (42.89%) storefront images. We consider strict search conditions can obtain more accurate results but reduce the absolute number.

2) STOREFRONT IMAGES EXTRACTION

Fig. 3 illustrates an example of web-search images. The web-search images include both the useful storefront images (the top portion of the figure), which are likely to contain the store signage, and other images irrelevant to our task (the bottom portion of the figure). The storefront images should be extracted, and the irrelevant images should be filtered out.

The SIWI adopts a general storefront image classifier using a fine-tuned VGG16 [12] deep network. It is a binary classifier that predicts whether a web-search image falls into the storefront category or not. To ensure generality, we establish our training dataset from 55 stores in shopping mall B, which is another, unrelated shopping mall in Nagoya. All 55 stores in shopping mall B do not overlap with our experiment

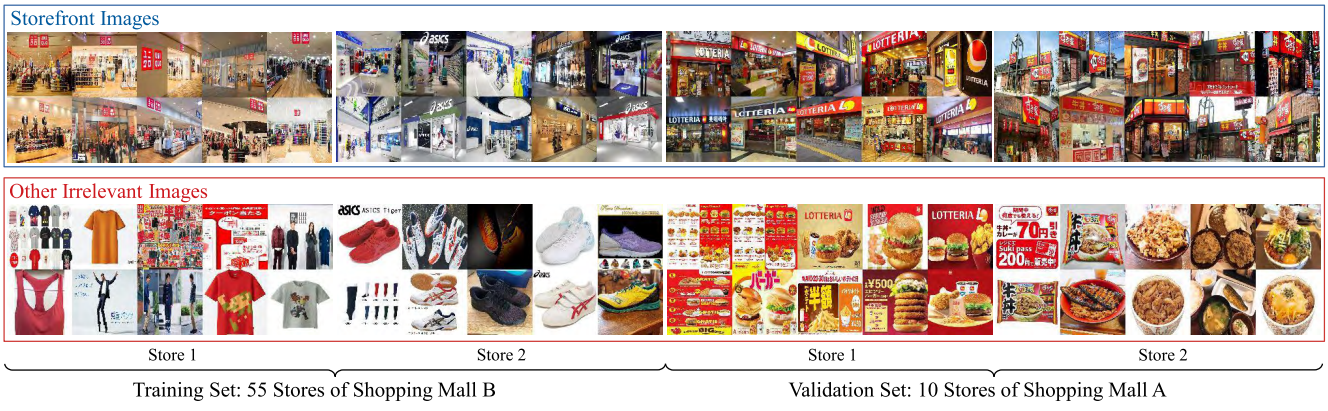

FIGURE 3. Examples of web-search images, arranged to highlight the difference between the desired storefront images and other irrelevant images.

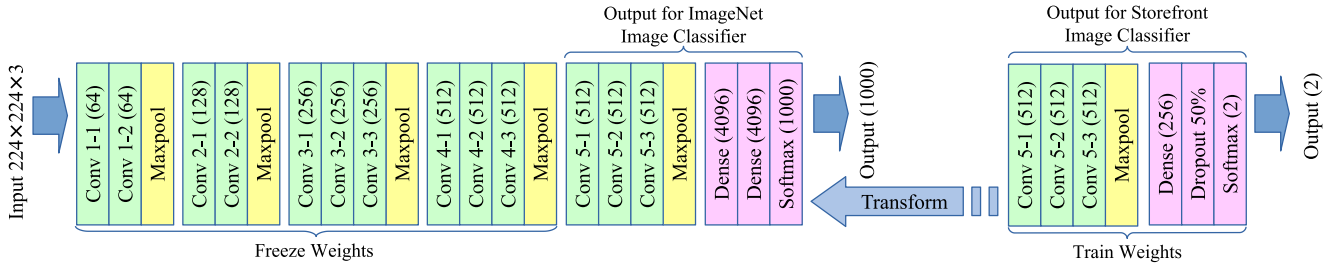

FIGURE 4. The network architecture for fine-tuning on VGG16. The weights of VGG16 are pre-trained in ImageNet. Layers from convolutional layer 5-1 to the output layer and are changed to the correct architecture. The weights on the first four groups are frozen and the weights from convolutional layer 5-1 are trained using in our dataset.

objects in shopping mall A. We first crawl for web-search images using those 55 store names as keywords via the same method explained in Section IV-A1. We then manually label 20 storefront images and 20 irrelevant images for each store. For validation purposes, we selected 200 storefront images and 200 irrelevant images belonging to 10 different stores from shopping mall A. Fig. 3 shows an example of the images in the training and validation sets.

The network architecture is shown in Fig. 4. The weight values of the VGG16 network are pre-trained using the ImageNet dataset [28], which has a 1,000 neuron softmax layer corresponding to 1,000 class outputs. We perform transfer learning using this pre-trained ImageNet. We alter the last two dense layers and the last softmax layer into a 256 consisting of a neuron dense layer, a 50% dropout layer, and a two neurons softmax layer. This adapts the output layer to correspond to our two desired outcomes: storefront images and other (irrelevant) images. During training, we freeze the first four out of VGG16's five convolutional layers, and only perform adaptations from the 5-1 layer to the output layer.

In the training stage, the batch size is set to 16 so that there are 138 iterations in one epoch. The performance of the storefront image classifier is evaluated on the validation dataset based on its precision, recall, and F1-measure, $F_1 = 2 \times (Precision \times Recall)/(Precision + Recall)$. The validation results for each epoch are shown in Fig. 5. The network converges on the 18th epoch with a precision of 98.48%, recall of 97.00%, and F1-score of 97.73%. The classifier thus effectively filters out irrelevant images from the web-search results. The effectiveness of storefront images extraction affecting signage results is evaluated in Section VII-B.

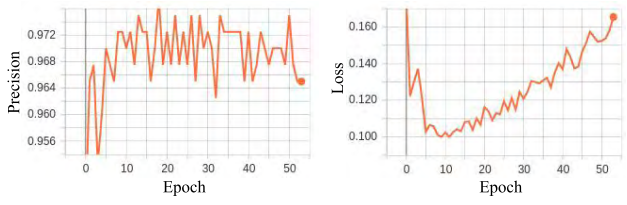

FIGURE 5. The validation accuracy (left) and validation loss (right) of the storefront classifier. Network converges on the 18th epoch with a precision of 98.48%, recall of 97.00%, and F1-score of 97.73%. The network converges on the 18th epoch. From the 19th epoch on over-fitting occurs, so that the loss increases on the validation set.

3) SIGNAGE EXTRACTION

Not all areas on a storefront image can be used to identify the store. Specifically, we are only interested in the store's signage. Signage patch extraction is used to crop signage patches from storefront images. We utilize a state-of-the-art object detector YOLOv2 [17] for fast store signage detection from storefront images.

As done with the storefront image classifier, we use the 1,100 storefront images from shopping mall B, as shown in Fig. 6 for generality purposes. We manually annotate 1,063 bounding boxes for signage patches in those 1,100 storefront images for training the YOLO. The initial YOLO network is pre-trained with the Pascal-VOC dataset [29]. We train using 82,000 batches with batch size of 64 (4,800 epochs) on the training set. The validation loss converges to less than 0.05.

We evaluate the signage extraction based on precision, recall, F1-measure, and Intersection over Union (IOU). The IOU is defined as follows:

$$IOU(B', B) = \frac{\text{Overlap Area of B' and B}}{\text{Union Area of B' and B}} \quad (1)$$

B is the manually annotated, "correct" bounding area while B' is the bounding area output by the detector. A prediction is considered correct when its IOU exceeds an arbitrary threshold Θ. As shown in Fig. 7, Θ is set to 0.33 when the overlapping area is 50%.

We manually annotate 200 storefront images from 10 stores in shopping mall A as a validation set. When YOLO predicts the signage patch, it outputs a confidence value $\in [0, 1]$ for each bounding box. Results with a confidence value less than the threshold value are discarded. We iterate the experiment multiple times for each threshold value between 0 and 1, increasing it in 0.01 increments. The results are evaluated based on precision, recall, F1-measure, and average IOU as shown in Fig. 8. The optimal result of 87.45% precision, 76.42% recall, and 81.56% F-1 measure is obtained when the confidence threshold is set to 0.17. However, we desire a higher recall percentage in order to obtain more samples

FIGURE 6. Examples from the 1,100 manually labeled storefront images from shopping mall B (left two images). These are used as training data for YOLO [17]. The right two images are examples of signage patches extracted from storefront images by the trained YOLO.

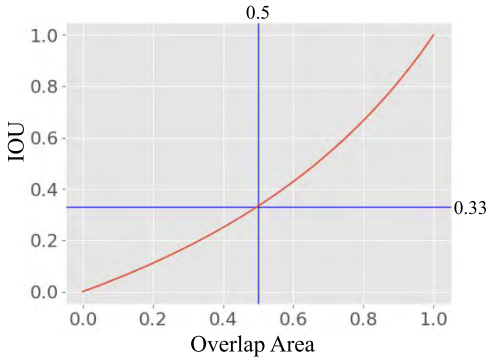

FIGURE 7. The relation between IOU and overlap area. When overlap is at 50%, the threshold of IOU is at 0.33.

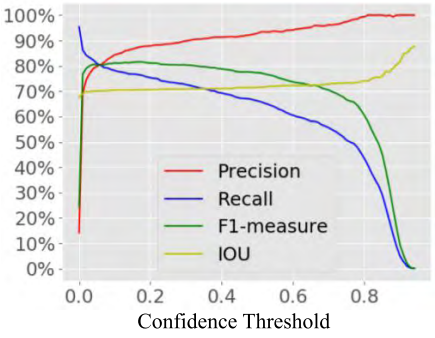

FIGURE 8. The chart plots the performance of YOLO on the validation set at different values of confidence thresholds.

for the signage dataset. Although this will also inevitably increase signal noise, the noise can be filtered out during the image matching phase as will be discussed later. We thus settle on a confidence threshold of 0.03. At this threshold, we can extract signage patches with a precision of 77.12%, recall of 83.18%, and F1-measure of 80.04%.

4) SUMMARY OF WEB-SEARCH IMAGE COLLECTION

Fig. 9 shows the resulting number of storefront images and signage patches for each store in the target location. The storefront image extraction process effectively prunes 46,556 (83.46%) irrelevant images from the 55,783 web-search images. The signage extraction process extracted 14,676 signage patches from the remaining 9,227 storefront images. The graph is sorted in descending order of number of signage patches. That number ranges from 3 to 657, indicating a significant imbalance. It may be reasoned that this is caused by famous stores such as Sukiya or UFJ Bank being much more well represented on the Internet compared to a relatively obscure local store. Another interesting phenomenaon is that a web search for a store providing food or services may simply yield more images relating to its storefront and signage in the first place. In comparison, if the name of a store is also the brand name for a particular brand of goods or clothes, the most common web-search images may then be more about its products rather than its storefronts. Fortunately, while these stores return fewer signage images, they usually also feature distinctive brand name logos on their websites. We thus introduce a webpage logos extraction in Section IV-B to compensate for this issue.

B. WEB-PAGE LOGOS EXTRACTION

There are 74 stores that maintain official websites in shopping mall A. In this section we propose an automatic method to extract store or brand logos from their official websites.

1) WEBPAGE IMAGE CRAWLER

The webpage image crawler downloads web-page images and their relevant data, including the image URL, and hyperlink from the web pages. The relevant data of web-page images is represented as:

$$(src, link) \in I_{i,j} \in w_{j,k} \in W_k \tag{2}$$

where W_k is website of k-th ($k \in [0, 73]$) store in the shopping mall A. $w_{j,k}$ represents j-th webpage in W_k. For the SIWI, we crawl the homepage and 10 sub-webpages for each web-site. Due to the different scale of each website, the maximum settings can effectively control the crawling time. The 10 sub-webpages are linked by hyperlinks from the homepage, excluding off-site links. $I_{i,j}$ is the i-th webpage image on the j-th sub-web-page. The *src* is the source URL of $I_{i,j}$. The *src* can uniquely identify webpage images on the Internet. The *link* is the hyperlink that the image links to.

The process for extracting $I_{i,j}$ from $w_{j,k}$ is as follows. An <img> element in the HTML document must be a

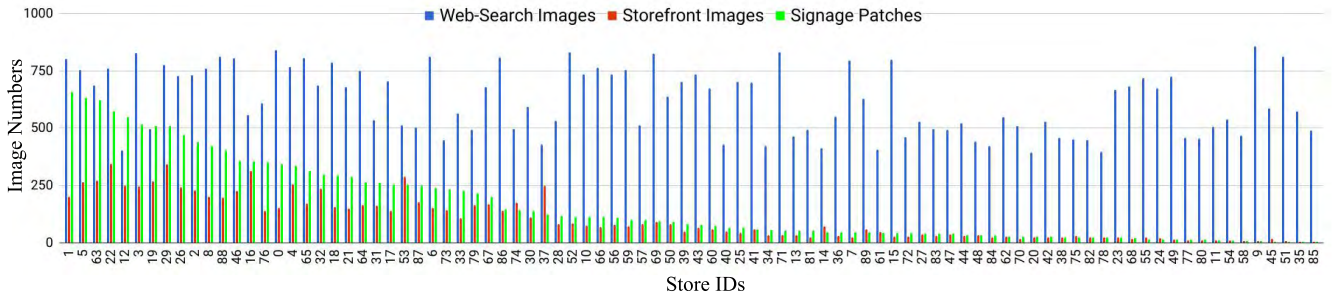

FIGURE 9. The number of web-search images (blue bars), storefront images (red bars), and extracted signage patches (green bars) from each store sorted by number of signage patches from left to right. Some stores tend to yield more signage patches than others and the resulting image sets are significantly biased in size. The imbalanced dataset significantly affects the performance of a training-based image retrieval method. We introduce a feature point matching method to resolve this issue.

webpage image. Moreover, some webpage images are also hidden in the element's background image attribute defined in its Cascading Style Sheets (CSS). The crawler scans all nodes in the HTML document and extracts the background image if the node has one. In other cases, a webpage image might be a PNG image with a transparent background. It relies on the background color of its parent nodes to form part of the color scheme of the logo. Therefore, when the crawler finds an image type capable of having transparency, such as a PNG, the crawler then scans all its parent nodes in the proper sequence to locate the one which has the same background color attribute in CSS, and converts the PNG image into JPG with the matching background color. The *link* of $I_{j,k}$ is extracted by scanning the image node and all its parents nodes in the proper sequence and finds the first hyperlink as the *link* of $I_{j,k}$.

Through the method above, we extracted 18,381 webpage images from the 74 store websites in shopping mall A. In Section IV-B2, we discuss extracting logo images from these webpage images.

2) WEBPAGE LOGOS EXTRACTION

Webpage logos are a small subset of webpage images. A website may have only one or two logos but thousands of other images. In addition, each website has its own distinct structure. More traditional approaches such as manual or tag-based methods [30], [31] that rely on information specific to a particular site or a particular class of sites cannot always guarantee success. Therefore, we propose a more general approach that is not affected by webpage structure and can effectively extract logos from almost all websites. We start by making the following observations regarding a logo's features:

- **Feature 1:** If a webpage image links to its homepage, then it is probably contains a logo image.
- **Feature 2:** If an image frequently appears in most of a website's webpages, then it probably is a logo.

Feature 1 usually applies to a logo located at the top left of most webpages. When the user clicks this logo, the browser redirects them to the homepage of the website. The same as storefront image extraction, to ensure generality, we test webpage logo extraction on 48 stores with official websites in the non-overlapping shopping mall B. In shopping mall B's dataset, 44/48 (92%) of website logos agree with this feature. Through feature 1 alone, we extracted 119 images from the 13,091 web-page images, including all of the 44 stores' logos.

For the remaining 4/48 websites, which disagree with feature 1, we use the statistics-based feature 2 to extract the logos. Feature 2 is inspired by Li *et al.*'s approach [32], which sought to eliminate noisy elements from web-sites. They note that noisy blocks usually share some common content and presentation styles. Therefore, the repeating elements in a webpage, which are usually uninformative content or noise. While a website's logo can hardly be considered noise, it does share the characteristics of being fairly repetitive and uninformative. The frequency of an image's appearance on a website could thus be used as a potential means to identify it as a logo. We calculate the frequency of a web-page image using the following formula:

$$f_{i,k} = \frac{\sum_{j=1}^{|W_k|} C(I_{i,j}, w_{j,k})}{|\{w_{j,k} : w_{j,k} \in W_k\}|}$$
$$C(I_{i,j}, w_{j,k}) = \begin{cases} 1 & I_{i,j} \in w_{j,k} \\ 0 & I_{i,j} \in w_{j,k} \end{cases} \quad (3)$$

where the frequency $f_{i,k}$ is calculated similar to a Term Frequency (TF) [33]. The$|\{w_{j,k} : w_{j,k} \in W_k\}|$ is the number of webpages $w_{j,k}$ in the website W_k. The $\sum_{j=1}^{|W_k|} C(I_{i,j}, w_{j,k})$ is the number of webpages $w_{j,k}$, which include $I_{j,k}$. Function C returns 1 if the webpage image $I_{i,j}$ is in the webpage $w_{j,k}$, and otherwise returns 0. The range of frequency is $f_{i,k} \in (0, 1]$.

For determining the threshold on $f_{i,k}$, we investigated all 48 websites and manually labeled webpage logos in all the webpage images. Figure 10 shows the histogram of $f_{i,k}$ on all 48 websites. Because of the large amount of low-frequency images that range from 0 to 0.2, we scale the Y-axis into 1000. It can be seen that all webpage logos (the orange bars) have a $f_{i,k}$ larger than 0.8. Therefore, we set a threshold $f_{min} = 0.8$ for $f_{i,k}$ to extract webpage logos. For the four websites which disagree with feature 1, we extract

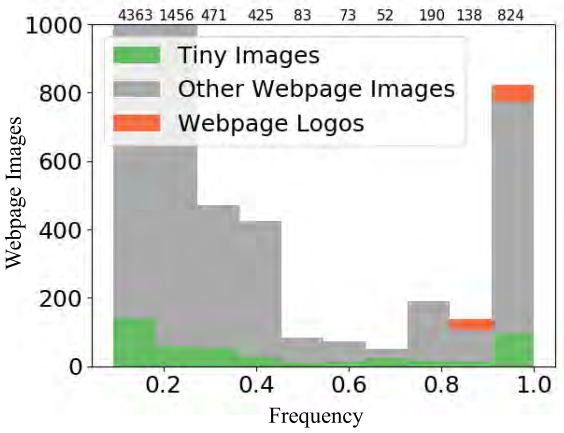

FIGURE 10. The histogram of webpage image frequency $f_{i,k}$ on 48 stores' websites. The numbers on the top of the figure indicate the amount of all web images. Since most logos appear in high frequencies they can be narrowed down simply by setting a threshold on $f_{i,k}$.

82 webpage images including all logos of the four websites through feature 2. However, some noise, such as the tiny buttons and placeholders, are also extracted by feature 2 as shown in Fig. 11. We thus set a size threshold to remove these irrelevant webpage images, as shown in Fig. 10.

FIGURE 11. The green bars in Fig. 10 represent image elements such as a placeholder or a tiny button, which can sometimes occur with a frequency approaching actual store logos. However, most are also significantly smaller than logos, so they can be easily filtered out based on size.

We also applied the above webpage logo extraction on the shopping mall A, and we obtained 18,381 webpage images from the 74 store websites. 70/74 (95%) of website logos agree with feature 1. Through feature 1 alone, we extracted 165 images from the 18,381 web-page images, including all 70 stores' logos. For the 4/74 websites which disagree with feature 1, we extracted 45 webpage images including all logos of the four websites through feature 2. In the end, we extracted 210 images from the total 18,381 webpage images. All logos of the 74 stores at shopping mall A are included in these 210 webpage images.

3) USING FAVICON AS WEBPAGE LOGOS

We also discuss using favicons as webpage logos. A favicon (short for favorite icon), which is displayed in the browser's address bar, is usually as a webpage logo. In the shopping mall A, 47/74 (64%) websites have their unique favicons. The favicon is typically loaded via a fixed HTML tag, which can be accurately extracted in CSS selector such as a "link[rel = shortcut icon]". This rule-based extraction method can avoid noise. However, due to size limitation, a favicon always illustrates a general image of the webpage. As shown in Fig.12, abbreviated favicons are not suitable as webpage logos. Available 16 favicons can be replaced by matching with webpage logos. Based on the above considerations, we did not adopt favicons as the webpage logos.

FIGURE 12. The left row shows the example of signage in the natural scene. The middle row shows corresponding favicons. The right row shows the example of webpage logos extracted from webpage images. There are 31/47 (66%) such favicons that are not suitable as webpage logos in the shopping mall A.

V. QUERY DATASET GENERATION

The query signage is the storefront signage extracted from panoramic images. Panoramic images are collected by NavVis [1], as shown in Fig. 13. Removing panoramic images shot in the basement, parking, and other unrelated areas, we obtained 361 panoramic images. We converted the panoramic image into a cube map with six orientations – left, right, front, back, top, and bottom. The shooting direction of NavVis is parallel to the passage, so we use only the left and right images as the query street images, because only those images face the storefronts. Using the same signage extraction discussed in Section IV-A3, the 370 signage patches are extracted from the storefront as query signage patches. The goal is to identity and label the correct store ID for each query signage patch.

VI. SIGNAGE IDENTIFICATION

The traditional methods apply feature vector techniques on signage identification. BoVW or deep learning-based image retrieval methods convert the query signage patches and target signage patches into fixed length feature vectors. They then match query signage patches with target signage patches based on the similarity on their feature vector. Such method allows the quick retrieval of thousands of images in a dataset.

However, relative to a manually created image dataset, the self-generated dataset through web-mining contains more noise and imbalanced samples, which significantly affect the accuracy of image retrieval methods. For resolving this issue, we considered replacing feature vectors with feature point matching, which matches each query signage patch with all

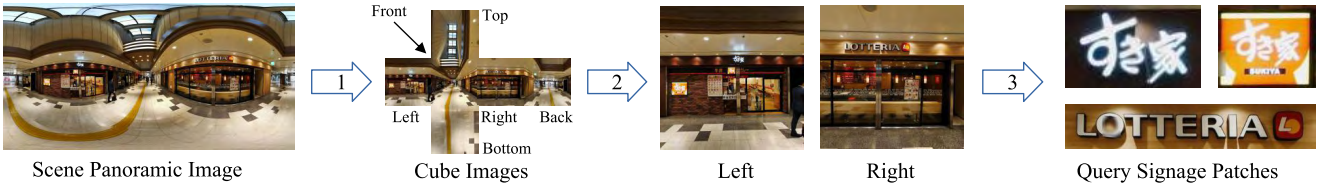

FIGURE 13. 1. A cube image including the six cardinal directions is converted from a street panoramic image. 2. The left and right images are extracted as street images. 3. Query signage patches are extracted from the street images.

target signage patches based on corresponding feature points. Such feature point matching method can compensate for a noisy and unbalanced dataset because the correct query signage and target signage pair usually has more corresponding feature points than an incorrect one, regardless of image noise.

Computing time is the main consideration when applying feature point matching on a large-scale. Matching a query signage with all the thousands of target signage patches requires an unreasonable computing time. Therefore, we need to generate a candidate matching database, which is a smaller subset of all target signage patches, for each query signage. The above-mentioned feature vector method is well suited for this task, because even under ideal circumstances the output from feature vector methods needs to undergo some form of further screening, as the correct pair does not always appear as the pair with the highest vector similarity score. These two methods, when used in tandem, serve well to complement each other's disadvantages. Because the method combines the deep learning approach with the feature point matching for signage identification, we name this a Hybrid Image Matching (HIM) method.

A. CANDIDATE MATCHING DATASET GENERATION

For each query signage extracted from street images in Section IV-A3, we generate an individual candidate matching dataset. As shown in Fig. 14, the dataset consists of both signage patches (from section IV-A) and webpage logos (from section IV-B). For the signage patches, we extract the first 20 signage images for each store. As discussed in Section IV-A1, the sequence of the search results is informative. Search engines rank the search results based on their correlation with keywords. We therefore take only the first 20 signage patches from the search results of each store. This brings the total number of candidate target signage patches to 1,800. Next, we run these 1,800 signage patches, along with the query signage, through a RESNET50 [8] process. As shown in Fig. 15, a top-n experiment explains it is easier to get more accurate results with a smaller n using RESNET50 feature. This RESNET50 process is pre-trained by ImageNet to convert each input image into an array of 2,048 dimension feature vectors. The cosine similarity between two sets of feature vectors is highly correlated to the image similarity of their parent images. By selecting only the top two target signage patches from each store based on

(c) Matching the query signage patch with each target signage in its Candidate Matching Dataset

FIGURE 14. An example of The HIM.Generating candidate matching dataset (b) for each query signage (a). (c) matches the query signage patch (a) with each target signage patch in its candidate matching dataset (b), and then selects the inlier maximum computed by RANSAC.

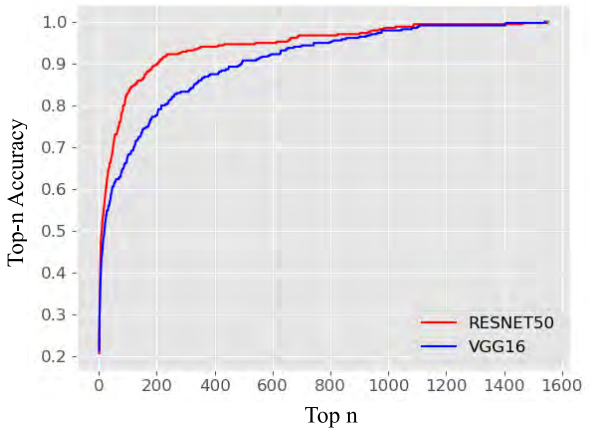

FIGURE 15. A top-*n* experiment on ground-truth using VGG16 feature and RESNET50 feature, respectively. We manually create a ground-truth that mapping 370 query signage patches to 1,800 candidate target signage patches with corresponding store IDs. For each query signage patch, we sort all 1,800 target signage patches according to the cosine similarities in RESNET50 feature and VGG16 feature, respectively. If the first n similarities can get the correct m correspondences,we use $m/370$ as the accuracy. As a result, when $n = 1$, we get the accuracy of 27.74% in RESNET50 and 29.04% in VGG16. When n = 10, we get 50.00% in RESNET50 and 44.01% in VGG16. When n = 100, we get 82.93% in RESNET50 and 67.96% in VGG16. When n = 200, we get 89.82% in RESNET50 and 77.54% in VGG16.

cosine similarity of their feature vector to the query signage, we further narrow down the number of candidate signage patches to just 180, which contains two signage patches from each store. As for the webpage logos, we simply insert all 210 logos, which we extracted from 18,381 web images, with their store IDs into the candidate matching dataset. Therefore, for each query signage, the total number of image matching steps is reduced to a maximum of 390, which represents reasonable computing times.

B. SIGNAGE MATCHING

All the target images in the candidate matching dataset are already labeled with their corresponding store ID. By correctly matching query signage patches with the appropriate target signage patches, the correct store ID for a query signage can be obtained.

Matching algorithms extract common visual feature points between the query signage and the target signage. Traditional matching algorithms, such as SIFT [18] and ORB [19], extract feature points with their descriptors, which are represented by fixed vectors, independent from source image and target image. The feature point matches are then extracted by computing the distance of feature point descriptors. Because of this independent feature point extraction, traditional matching algorithms ignore the transformations and deformations from the source image to the target image. This may work well on images shot in the same scene and domains, such as an image mosaic task. However, in our work, the query signage and target signage are mostly shot from different scenes. In addition, the signage shot by a camera and the webpage logo as a Computer Graphics (CG) belong to different domains. The algorithm may thus extract different key-points even though it is processing the same locations.

We use DeepMatching [20] to match a query signage with its candidate matching dataset as shown in Fig. 16. DeepMatching is a matching "in the wild" algorithm that explicitly handles non-rigid deformations through bounds on deformation tolerance. For example in Fig. 16, SIFT and ORB matching have a challenging time dealing with images from different domains or different shooting conditions. DeepMatching sidesteps these concerns.

The challenge in applying DeepMatching is its computing time. DeepMatching computes a correlation pyramid in convolution for each non-overlapping 4×4 atomic patch in the entire image. The larger an image is, the more of these patches it will have to process, thus increasing computing time. Specifically, the computing time increases quadratically with respect to the width (W) and height (H) of an image. As shown in Fig. 17, matching two square images of 300×300 pixels each will take about 1s. If those two images are only 200×200 pixels, then it would only take about 0.2 seconds. Restricting the size of images can effectively reduce computing time. We propose a resizing strategy as follows:

$$W = \begin{cases} 200 & W > H \\ 200 \times \dfrac{W}{H} & W < H \end{cases}$$
$$H = \begin{cases} 200 & H > W \\ 200 \times \dfrac{H}{W} & H < W \end{cases} \quad (4)$$

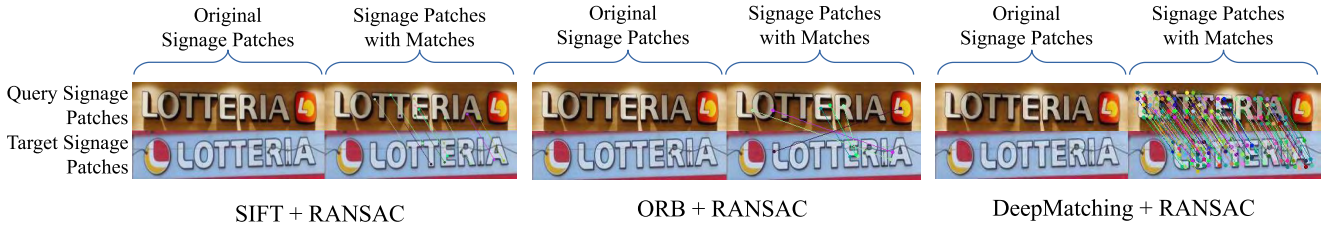

FIGURE 16. A demo of SIFT matching, ORB matching, and DeepMatching on logo signage and textual signage. The top image patch is the query signage. The bottom image patch is the target signage from web-search image. The left patch shows the original images. The right patch is the images with matches. The matches of SIFT is better than ORB. DeepMatching is better to handle image matching under the different conditions or domains.

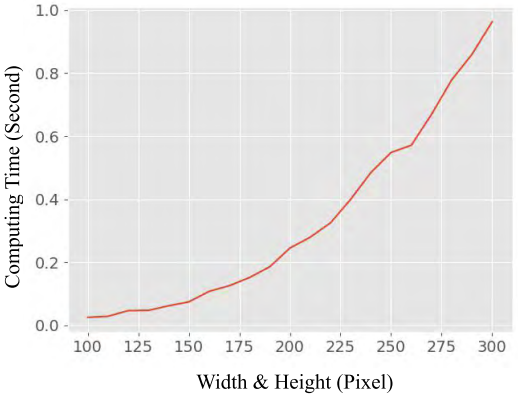

FIGURE 17. Computing time for various image sizes. We match two square images of the same size and gradually increase their dimensions. The computing time increases quadratically with the increase in dimension size.

where W is the width of the image and H is the height of the image. We resize the larger of either W or H to 200 pixels, which ensures 25 atomic grids along the long side, and then scale the short side accordingly. If, however, this results in the short side being reduced to less than 32 pixels, the bottom levels in the correlation pyramids may cease to be discriminative. DeepMatching will then not output any matches. We therefore restrict the scaling of the short side to a minimum of 64 pixels.

Furthermore, because DeepMatching computes the descriptor as a histogram of oriented gradients, a dark character with bright background cannot be matched with a bright character with dark background, as shown in Fig. 18. Although they have the same contours, their gradient orientation is reversed. Therefore, we match the query signage with both the originals and inverses of the target signage patches. The results are saved independently along with the rest of the candidate results.

After matching a query signage to all target signage patches in its candidate matching dataset, we propose a measure to determine a matching result as the correct one. Not all matches from the image matching algorithm are informative. The informative matches must be extracted through projection transformation. We consider that signage patches are two dimensional objects, so we can transform the query signage into the target signage by linear projection, or homography.

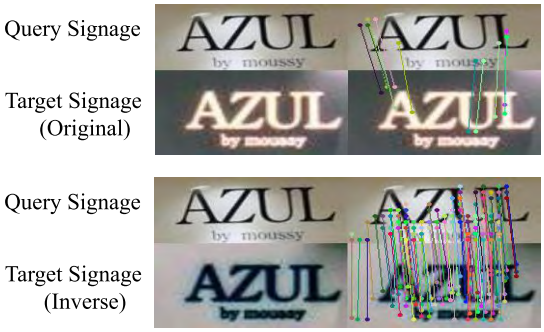

FIGURE 18. Dark character with bright background cannot normally be matched with a bright character with dark background, but can be matched with the inverse image of it.

This homography can be computed by RANSAC according to corresponding matching points [34]. If a matching point agrees with its homography, it is considered an inlier point. The correct image should have a higher number of inlier points than incorrect images. Therefore, for each query image, we sort all of its target signage matching results based on the number of inlier points and extract only the first one as the correct target signage. We then assign that signage's store ID to the query signage. We thus achieve our initial goal of assigning a store ID to all signage patches in the street images. In Section VII, we evaluate the accuracy of the SIWI.

VII. EVALUATION EXPERIMENT

A. ACCURACY EVALUATION OF SIWI

In this section we verify the accuracy of the SIWI. Fig. 19 illustrates a complete example of the aforementioned signage identification method in action. Two image patches ("LOTTERIA" and "SUKIYA") are cropped out from street images as query images. The "LOTTERIA" patch is matched with a similar looking signage patch, which is separated and cropped it to size from thousands of other web search images. The "SUKIYA" image is matched with the official "SUKIYA" logo, which our logo extraction method effectively extract from its website. In this way, we processed all 370 signage patches extracted from panoramic images discussed in Section V, and compared the resulting list of matched store IDs with a manually-produced ground truth to verify its accuracy.

Table 2 presents the results of the accuracy evaluation. The SIWI correctly matches 337 of the total 370 signage patches

TABLE 2. The evaluation result on signage identification. The numbers in the brackets represent the accuracy on signage in each pattern. The numbers out of the brackets represent the number of correctly predicted signage patches.

Target Signage \ Query Signage	321 Signage Patches of 74 Stores with Websites	49 Signage Patches of 16 Stores without Websites	370 Signage Patches of All 90 Stores
Web-page Logos from Stores' Websites	182 Signage Patches (57%)	–	–
Signage Patches from Web-Search Images	266 Signage Patches (83%)	39 Signage Patches (80%)	305 Signage Patches (82%)
All	298 Signage Patches (93%)	39 Signage Patches (80%)	**337 Signage Patches (91%)**

FIGURE 19. An example of the signage identification process. Two image patches ("LOTTERIA" and "SUKIYA") are cropped out from street images as query images. The "LOTTERIA" patch is matched with a similar looking signage patch, which is separated and cropped it to size from thousands of other web search images. The "SUKIYA" image is matched with the official "SUKIYA" logo, which our logo extraction method had effectively extracted from its website.

TABLE 3. The evaluation result on store identification.

Query Signage / Target Signage	74 Stores with Websites	16 Stores without Websites	All 90 Store
Web-page Logos from Stores' Websites	40 Stores (54%)	–	–
Signage Patches from Web-Search Images	63 Stores (85%)	12 Stores (75%)	75 Stores (83%)
All	71 Stores (96%)	12 Stores (75%)	**83 Stores (92%)**

from 90 stores, achieving an accuracy of 91%. When only the webpage logos are used as the target signage set, the result is 182 correct matches (57%) of the total 321 signage patches from 74 stores that have a website. On the other hand, if only web-search images are used on the same 321 signage set, the correct results are 266 (83%). Combining these two methods raises the overall accuracy to 298, which is 93%. This illustrates the benefits of combining several different sources of sample images for improving the overall matching result.

This method can also be used in annotating POIs with store IDs on the 3D maps established by NavVis. NavVis can automatically establish the 3D structure of a shopping mall, as long as the locations and directions of panoramic images are known. For a given store, if more than half of its signage patches can be correctly identified, we consider the ID of the store to be correctly predicted. The result is presented in Table 3. Where 83 of the total 90 stores are correctly identified by the SIWI, which is an accuracy of 92%.

B. COMPARISON WITH OTHER ALGORITHMS AND INDIVIDUAL STEPS EVALUATION

A comparative experiment should be performed based on a common dataset, but there currently are no available open datasets for evaluating store signage identification. We therefore conducted a comparative experiment with the same dataset from shopping mall A, using methods similar to existing approaches and compared the results with the SIWI. Additionally, we evaluate the effectiveness of individual steps of the SIWI. Detailed experimental results are shown in Table 4.

OCR approaches [3]–[7], [15] are currently the most widely-used approaches to signage identification. In experiment 1, we evaluate an OCR-based signage identification. Because OCR engines are sensitive to language and most are configured to work in an English environment, we extract only the English signage patches from the 90 stores. Using the TESSERACT [35] OCR engine on those English storefront signs referencing [7], 29% of texts on them can be correctly recognized. Fig. 20 shows several signage patches with special fonts and extreme illumination conditions, which significantly affect the performance of an OCR engine. We consider OCR-based methods to be restricted in their applications, as they only operate properly within a rigidly defined environment.

Next, we evaluate training-based image matching methods, which include BoVW methods [13], [15] and deep learning-based methods [8], [12]. These methods all require a training set. We first select the top 20 search results per shop and all

TABLE 4. We compare the SIWI 9 with other methods 1 to 6. In Experiment 1, we verify a textual-matching based signage identification method on English street signage dataset using Tesseract OCR engine [35], which is used by approach [7]. In Experiments 2 to 4, we verify image classification methods including BoVW [13], [15] and VGG16 fine-tuning [12], to classify all 370 query street signage patches into store IDs. The classifiers are trained by the original "noisy" web image dataset and manual "clean" web image dataset, which we crawled from the Internet, respectively. Also, from Experiment 6 to 9, we evaluate individual steps of the SIWI. By comparing Experiments 6 and 9, we verify the effectiveness of storefront image extraction introduced in section IV-A2. In Experiments 7, 8, and 9, we compare the accuracies, which utilize ORB [19], SIFT [18], and DeepMatching [20] as the matching method respectively. The result explains the DeepMatching is suitable for matching street images and web images, which we discussed in section VI-B. Experiment 9 shows the result of the SIWI, which achieve the highest accuracy.

#	Identification Method	Query Signage Dataset	Training Signage Dataset	OCR Tool or Classification Model	Accuracy
1	Textual Matching	302 English Signage Patches	–	Tesseract OCR Engine [35]	29%
2	Image Classification	All 370 Signage Patches	Original "Noisy" Web Image Dataset	BoVW [15]	39%
3				VGG16 Fine-tuning [12]	55%
4			Manual "Clean" Web Image Dataset	BoVW [15]	47%
5				VGG16 Fine-tuning [12]	64%
			Target Dataset Generation Steps	**Utilized Matching Method**	
6	SIWI	All 370 Signage Patches	✗Storefront Image Extraction ✓Signage Extraction ✓Web-page Logo Extraction	DeepMatching [20]	84%
7			✓Storefront Image Extraction ✓Signage Extraction ✓Web-page Logo Extraction	ORB [19]	48%
8				SIFT [18]	64%
9				DeepMatching [20]	**91%**

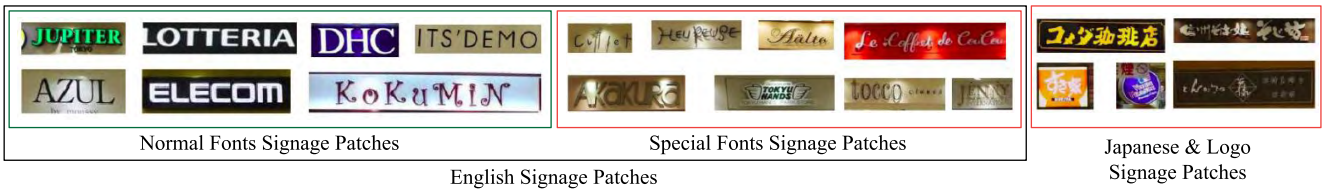

FIGURE 20. Using OCR-engine TESSERACT on query storefront signs. The OCR engine can recognize normal English fonts as shown on the left. Special fonts (middle) and multi-language characters cannot be recognized correctly.

210 webpage logos and used these as our first training set to simulate the noisy environment prior to the candidate matching dataset generation process discussed in Section VI-A. We then manually filtered out noise from the signage patches and webpage logos, and selected 10 "clean" target signage patches from each store ID to construct our second training dataset. If a store ID has less than 10 "clean" target signage patches, we over sample the training images to 10 images, as suggested by Paulina and David *et al*. [36]. This improved dataset simulates the situation after the candidate matching dataset has been generated. We supplied the two training datasets to both the BoVW and VGG16 fine-tuning schemes and evaluated their accuracy respectively.

For BoVW Experiments 2 and 4, we extract descriptors from query signage patches and target signage patches in SIFT. All descriptors are clustered by k-Means into 2,000 clusters as the visual words. The histogram of signage patches are computed by descriptors on each word. The store IDs of query signage patches are predicted by a SVM [37], which is trained by BoVW histogram of the target signage set. The BoVW model can achieve an accuracy of 39% with the "noisy" dataset and an accuracy of 47% with the "clean" dataset.

The VGG16 fine-tuning Experiments 3 and 5 use the same transformation as shown in Fig. 4. The output softmax layer is set to 90 neurons, which is fitted to our 90 store IDs in shopping mall A. The weights are initialized by ImageNet. The VGG16 fine-tuning can achieve an accuracy of 55% with the "noisy" dataset and an accuracy of 64% with the "clean" dataset.

By observing the pairs of Experiments 2 and 4, or Experiments 3 and 5, it can be seen that the noise from web-mining results reduces accuracy by 8%. Other than noise, a lack of sufficiently balanced training datasets can also adversely affect traditional signage identification methods. However, noise and a lack of quality data is to be expected when dealing

with automated web mining. Comparing with Experiment 9, we believe the SIWI is equipped to tackle these problems effectively.

We now verify the effectiveness of the individual steps. Experiments 6 and 9 verified the effectiveness of the storefront image extraction process discussed in section IV-A2. In this experiment we skip the storefront image extraction and went directly to the signage extraction, which extracts 58,923 signage patches from the 55,783 web-search images. We then kept the first 100 signage patches per store. From Section VI-A, the RESNET50 feature is used to find the two most similar signage patches for a query street signage to generate the candidate matching dataset. Without storefront image extraction, the noise in the target dataset increases. The accuracy drops to 84%. It can be seen that the storefront image extraction process effectively filters irrelevant images from a large set of web-search images. Using storefront image extraction improves the SIWI by almost 7% over other methods.

Experiments 7, 8, and 9 evaluated the effectiveness of different feature extraction methods in dataset generation method in the SIWI. In Experiment 6, which features the ORB [19] method, the best matches are found from the top 70% of the nearest pairs between feature point descriptors extracted from query images and target images. In Experiment 7, which features the SIFT [18] method, we apply the ratio suggested by Lowe's paper (0.7) [18] on the threshold to find the best matches. The available matches, which accept a homography, are extracted from the best matches in RANSAC.

As a result, the SIFT method proves to be about 15% more accurate than the ORB method. We think that both SIFT and ORB will work on images shot from the same scene. However, in our task, the scenes and web images are mostly obtained from different views or belong to different domains. This significantly affects feature extraction. This issue is resolved by utilizing Deepmatching [20], which is also useful for matching street images and web images.

VIII. CONCLUSIONS

Our work proposes SIWI - an image-matching based automatic signage identification method using web information. Compared to existing textual-matching based methods, this method is not limited to just text or character-based signage. It is able to process any arbitrary signage design. In addition, the SIWI replaces manual data collection with an automated web-mining and data sifting process, which can self-generate entire matching databases from very simple inputs. And applies data pruning and filtering methods, so it effectively reduces the number of images selected for matching, and ensure that the process can be performed within reasonable computing times. Feature point image matching is also not dependent on specialized training, thus making this a highly portable system that should work for other locations with no fine tuning required.

An inherent limitation of the SIWI lies in its dependency on web search results and webpage images for data. If a target store does not have a signage that features images available on the Internet, then it cannot be identified by this method. Additional research should focus on methods other than web-mining through search engines and websites.

REFERENCES

[1] *NavVis*. Accessed: Mar. 6, 2018. [Online]. Available: http://www.navvis.com/

[2] *NavVis IndoorViewer Demo*. Accessed: Mar. 6, 2018. [Online]. Available: https://youtu.be/Pb9Hf0Y6gv8?t=50s

[3] R. Meng, S. Shen, R. R. Choudhury, and S. Nelakuditi, "AutoLabel: Labeling places from pictures and websites," in *Proc. Ubiquitous Comput. (UbiComp)*, 2016, pp. 1159–1169.

[4] J. Park *et al.*, "Automatic detection and recognition of Korean text in outdoor signboard images," *Pattern Recognit. Lett.*, vol. 31, no. 12, pp. 1728–1739, 2010.

[5] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2963–2970.

[6] X. Chen, J. Yang, J. Zhang, and A. Waibel, "Automatic detection and recognition of signs from natural scenes," *IEEE Trans. Image Process.*, vol. 13, no. 1, pp. 87–99, Jan. 2004.

[7] S. S. Tsai, H. Chen, D. Chen, V. Parameswaran, R. Grzeszczuk, and B. Girod, "Visual text features for image matching," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2012, pp. 408–412.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[9] Z. Wojna *et al.*, "Attention-based extraction of structured information from street view imagery," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 1, 2018, pp. 844–850.

[10] Y. Movshovitz-Attias, Q. Yu, M. C. Stumpe, V. Shet, S. Arnoud, and L. Yatziv, "Ontological supervision for fine grained classification of Street View storefronts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1693–1702.

[11] Q. Yu *et al.* (Dec. 2015). "Large scale business discovery from street level imagery." [Online]. Available: https://arxiv.org/abs/1512.05430

[12] K. Simonyan and A. Zisserman. (Sep. 2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[13] H. Xu, D. Zhao, J. An, and L. Liu, "Indoor shop recognition via simple but efficient fingerprinting on smartphones," in *Proc. Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, 2016, pp. 27–31.

[14] J. Song, S. Hur, and Y. Park, "Fingerprint-based user positioning method using image data of single camera," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, 2015, pp. 13–16.

[15] A. R. Zamir, A. Dehghan, and M. Shah, "Visual business recognition: A multimodal approach," in *Proc. ACM Int. Conf. Multimedia*, 2013, pp. 665–668.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[17] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2016, pp. 6517–6525.

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2564–2571.

[20] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "DeepMatching: Hierarchical deformable dense matching," *Int. J. Comput. Vis.*, vol. 120, no. 3, pp. 300–323, 2016.

[21] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[22] R. Smith *et al.*, "End-to-end interpretation of the french street name signs dataset," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 411–426.

[23] X. Teng, D. Guo, Y. Guo, X. Zhao, and Z. Liu, "SISE: Self-updating of indoor semantic floorplans for general entities," *IEEE Trans. Mobile Comput.*, pp. 1–14, 2018. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8307177/
[24] S. Fang, C. Liu, F. Zhu, and E. J. Delp, "cTADA: The design of a crowdsourcing tool for online food image identification and segmentation," in *Proc. IEEE Southwest Symp. Image Anal. Interpretation*, 2018, pp. 1–4.
[25] *Google Places*. Accessed: Mar. 6, 2018. [Online]. Available: https://developers.google.com/places
[26] *Icrawler*. Accessed: Mar. 6, 2018. [Online]. Available: https://github.com/hellock/icrawler
[27] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to Japanese morphological analysis," in *Proc. Empirical Methods Natural Lang. Process. (EMNLP)*, 2004, pp. 1–8.
[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 248–255.
[29] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2014.
[30] J. San, "Effective Web data extraction with standard XML technologies," in *Proc. 10th Int. Conf. World Wide Web*, vol. 39, 2002, pp. 689–696.
[31] S.-H. Lin and J.-M. Ho, "Discovering informative content blocks from Web documents," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2002, pp. 588–593.
[32] L. Yi, B. Liu, and X. Li, "Eliminating noisy information in Web pages for data mining," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2003, pp. 296–305.
[33] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *J. Documentation*, vol. 28, no. 1, pp. 11–21, 1972.
[34] E. Vincent and R. Laganiere, "Detecting planar homographies in an image pair," in *Proc. 2nd Int. Symp. Image Signal Process. Anal., Conjunction 23rd Int. Conf. Inf. Technol. Interfaces. (ISPA)*, 2001, pp. 182–187.
[35] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, 2007, pp. 629–633.
[36] D. Masko and P. Hensman, "The impact of imbalanced training data for convolutional neural networks," Degree Project Comput. Sci., KTH Roy. Inst. Technol., Stockholm, Sweden, 2015, pp. 1728–1739.
[37] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.

CHENYI LIAO received the B.S. degree from the University of Science and Technology Beijing, China, in 2009, and the M.S. degree from Chubu University, Japan, in 2013. He is currently pursuing the Ph.D. degree with the Graduate School of Engineering, Nagoya University, Japan. Since 2017, he has been a Researcher at the Institutes of Innovation for Future Society, Nagoya University. His research interests are in data mining and data processing.

WEIMIN WANG received the B.S. degree from Shanghai Jiao Tong University in 2009, the M.S. degree from Osaka University in 2012, and the Ph.D. degree from Nagoya University in 2017. From 2012 to 2014, he was involved in the digital and analog circuit design at NF Corporation. He held a post-doctoral research position at Nagoya University for six months. He is currently a Researcher at the National Institute of Advanced Industrial Science and Technology.

KEN SAKURADA received the B.E., M.E., and Ph.D. degrees from Tohoku University, Japan, in 2009, 2011, and 2015, respectively. He was an Assistant Professor with the Graduate School of Engineering, Nagoya University, from 2016 to 2018. He has been a Senior Researcher at the Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology. From 2013 to 2014, he was a Visiting Researcher at Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. From 2012 to 2014, he was a recipient of the Research Fellowship of the Japan Society for the Promotion of Science.

NOBUO KAWAGUCHI received the B.E., M.E., and Ph.D. in computer science from Nagoya University, Japan, in 1990, 1992, and 1995, respectively. Since 1995, he has been an Associate Professor with the Department of Electrical and Electronic Engineering and Information Engineering, School of Engineering, Nagoya University. Since 2009, he has been a Professor with the Department of Computational Science and Engineering, Graduate School of Engineering, Nagoya University. His research interests are in the areas of human activity recognition, smart environmental system, and ubiquitous communication systems.