An Approach to Checking Map Tiles by Annotation Recognition

Lei Ding [1,2], Hongping Zhang [1,2], Wei Huang [1,2], Wenhao Zhao [1,2],
Dejin Tang [1,2], Xinyan Zheng [1,2], Cong Wang [1,2], Heng Li [1,2], ZhenWang [1,2]

[1] National Geomatics Center of China, 100830 Beijing, China - (dinglei, zhanghongping, huangwei, zhaowh, tangdejin, xinyan_zheng, wangcong, liheng, wangzhen)@ngcc.cn

[2] Technology Innovation Center for Geographic Information Public Service, Ministry of Natual Resources, Beijing, China

Key words: Web Map Tile Service, Map Tile, Optical Character Recognition (OCR), Tiles Inspection

Abstract

As the capabilities of geographic information services increase, web geographic information services represented by internet mapping have become increasingly popular, widely used in various location-based services. It is precisely because of the surge in demand that higher requirements have been put forward for the updating and quality control of internet map data. The quality inspection of geographic vector data is relatively well-developed, with many inspection rules and automated processes, but there is a clear deficiency in the quality inspection of map tiles, which mainly relies on manual visual inspection, restricting the rapid updating and release of map tiles. To solve this problem, this paper proposes an automated map tile inspection method. First, web crawling technology is used to capture map tiles, and to avoid the problem of incomplete display of annotations across tiles, adjacent tiles are all captured and merged into one image; second, OCR technology is used for image text detection and text recognition, and finally, the newly added recognition results are stored in the database and compared and confirmed with the results in the database. The experiment proves that the method proposed in this paper is feasible and may greatly improve work efficiency, especially as the updating speed of online maps accelerates, an automated tile quality inspection method can greatly improve the updating speed.

1. Introduction

1.1 General Instructions

Web mapping has emerged as a prevalent method for disseminating online mapping content via the Internet. Popular services such as Google Maps, Baidu Maps, Amap, enable users to visualize cartographic data effortlessly through a standard web browser and an active Internet connection (García et al., 2012). In order to standardize this kind of map services, the Open Geospatial Consortium (OGC) developed the Web Map Tile Service (WMTS) interface standard. The standard is currently a major focal point in the development and advancement of web map services. It is a tile-based map service protocol that provides map data through a multi-level pyramid structure pre-generated tile collection. It provides a standard HTTP interface for requesting georeferenced map tiles from one or more distributed geospatial databases. It enabled server application can serve map tiles of spatially referenced data using tile images with predefined content, extent, and resolution (OGC, 2010). Map tile is the basis and core data of map service, that describe a unified framework for spatial orientation and basic data for online map visualization. Tile maps display geographic regions as a grid of identical tile images (McNeill et al., 2017), each tile image contains map annotations and symbols produced by symbolizing spatial data. Depending on the position and zoom level of the user's view, the client's browser dynamically requests and loads the corresponding map tiles, which are eventually combined to form a complete map image for the user. Depending on the position and zoom level of the user's view, the client's browser dynamically requests and loads the corresponding map tiles, which are eventually combined to form a complete map image for the user. The coordinate system, tile style, and background color of the map tiles are rendered during the tile production process, and cannot be modified by users.

Before providing map services, it is necessary to generate map tiles, which are typically rendered based on high-quality data. There have been many mature measures and detection methods for quality control in original vector data. In addition, quality inspection is also carried out before the tile is published online. However, despite being created from high-quality data, ensuring the accuracy of map tiles remains challenging due to various unexpected occurrences, such as tile loss and incomplete rendering, throughout the rendering, transmission, and uploading processes. To overcome this issue, the current approach to map tile quality inspection primarily relies on manual random sampling and visual checks before tile deployment. However, this requires significant manpower and time, leading to delays in tile deployment efficiency. This method fails to adapt to large-scale tile quality inspection and rapid data updates, potentially hindering the updating of online map tiles and compromising the quality of online maps.

Aiming to address these drawbacks, this study proposes an approach to checking map tiles by annotation recognition based on the optical character recognition (OCR) (Mithe et al., 2013), specifically in PaddleOCR (Du et al., 2020). The OCR is a piece of technology that converts printed text and images into digitized form such that it can be manipulated by machine (Islam et al., 2017). And they have been widely used in various of application scenarios, such as office automation systems, factory automations, online educations, license plate automations, text scanning, map productions etc. (Du et al., 2020, Hu et al., 2020). The approach presented in this paper is based on PaddleOCR. PaddleOCR is an open-source Optical Character Recognition (OCR) library developed by Baidu. It is designed for text detection and recognition tasks and operates on the PaddlePaddle deep learning framework. PaddleOCR is highly regarded for its usability, efficiency, and support for multiple languages.

2. Map Tiles

Traditional paper maps are scaled according to a certain scale, after cartographic synthesis to optimize the display of content, can only provide geographic information a fixed level of map visualization effect. The web map breaks through the limitations of the traditional paper map, he supports multi-level map zoom display, in each level can be expressed in different levels of detail of geographic information. To provide a tiled web map service, the server pre-renders the map at a series of predetermined scales, utilizing a process of progressive generalization. These rendered map images are subsequently segmented into a grid of tiles. The OGC proposed the WMTS specification defines a kind of web map tile to express the specification.

The web map is a tile-based map service protocol that provides map data through a multi-level pyramid structure, describing a tile pyramid as depicted in Figure 1. Firstly, the tile map pyramid consists of multiple levels (Levels), each representing a zoom level of the map. The bottom level (level 0) represents the largest scale, i.e. the most detailed view of the map, and with each level up, the zoom level of the map increases and the view becomes more generalized. Secondly, each layer consists of a certain number of tiles that are uniformly distributed in geographic space. The tiles are usually square and have a fixed pixel size such as 256x256 pixels. Thirdly, within each level, tiles are organized in a matrix of rows and columns. The row and column indices of the tiles are computed from the top left tile (0,0). Fourthly, each tier has a different resolution, with the resolution decreasing as the tier increases. Correspondingly, the scale increases with increasing tiers, meaning that higher tier tiles cover a wider geographic area.

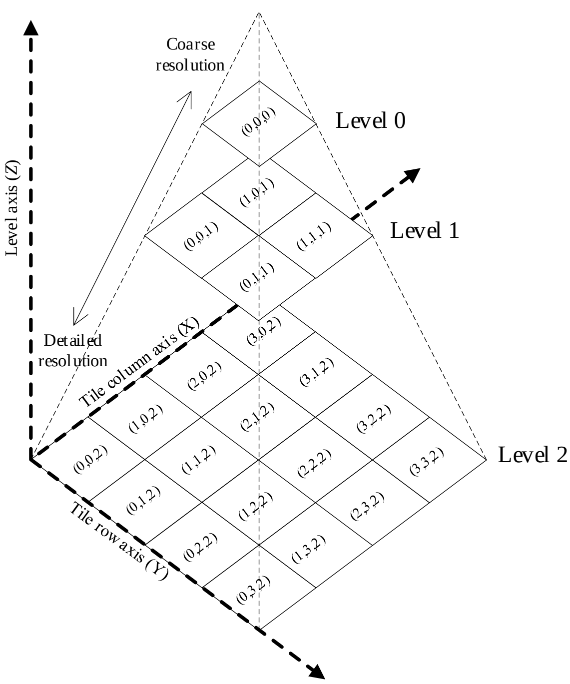

Figure 1. Tile matrix set representation.

3. Web Scraping

Web crawling technology is an automated method of extracting information from the internet by simulating the behavior of a web browser. Web crawlers first identify the target website they want to crawl, then send HTTP requests to retrieve the website's HTML code. Next, the crawler parses the HTML code to extract desired information such as links, images, text content, etc. Finally, the extracted data is processed and stored for further analysis and application. The typical workflow of a web crawler includes: (1) Sending HTTP requests: Crawlers fetch HTML code by sending HTTP requests to websites. (2) Parsing HTML code: Crawlers use parsers tools to parse HTML code for extracting desired data. (3) Data extraction and processing: Relevant data is extracted from parsed HTML code, Crawlers need to clean and process extracted data to ensure its accuracy and completeness. Additionally, choosing appropriate storage methods based on requirements is essential for data persistence. (4) Storing data: The extracted and processed data is stored in databases, files, or other mediums for subsequent analysis and application. When conducting web crawling, Crawlers should set reasonable request frequencies to avoid unnecessary burden on the site and excessive pressure on target websites and minimize the risk of being banned.

The crawler target of this paper is the map service website, which can reasonably crawl the map service tiles without being blocked, and by reasonably crawling the map tiles, the speed of crawling can be controlled to ensure the stable operation of the website without affecting the usability of the website. The data obtained through this crawling process will only be used for scientific research and testing the feasibility of the methods described in this article. It will not be used for commercial purposes.

4. Optical Character Recognition

Optical Character Recognition (OCR) is a technology used to convert scanned images of printed or handwritten text into editable and searchable digital documents (OCR) (Mithe et al., 2013). OCR software recognizes individual characters in an image and converts them to machine-encoded text. OCR technology has been around for many years and is widely used in various applications, including document scanning, digitization, and automation. The OCR process typically involves several steps, including image preprocessing, character segmentation, feature extraction, classification, and post-processing. In the first step, the image is prepared by removing noise, adjusting contrast and brightness, and correcting distortions. In the second step, the characters are identified and separated from the background image. In the third step, features such as shape, size, and orientation are extracted from each character. In the fourth step, the characters are classified based on their features, and the corresponding text is generated. In the final step, the recognized text undergoes post-processing to correct any mistakes and improve the accuracy. OCR technology has many practical applications, such as converting paper-based documents into electronic formats, extracting data from forms and invoices, and recognizing handwriting. It can also be used in the fields of machine translation, natural language processing, and computer vision.

Paddle Optical Character Recognition (PaddleOCR) is an advanced OCR technology solution based on the PaddlePaddle deep learning framework (Du et al., 2020). It excels in recognizing printed, handwritten, and mixed text, supporting a wide range of languages and fonts, including Chinese and English. Paddle OCR is also adept at recognizing specific types of content such as tables, invoices, ID cards, and license plates, among others. The deep learning model within PaddleOCR has undergone extensive training and optimization, resulting in high-precision OCR recognition with exceptional accuracy across various complex scenarios. Its core functionality includes text detection and text recognition. Text detection is utilized to identify text regions within an image and isolate them, while text recognition transforms these detected text regions into editable and searchable digital documents. Paddle OCR offers a variety of rich API interfaces and SDK toolkits, simplifying the integration of OCR functionality into developers' applications. Furthermore, it can be deployed both on the cloud and locally, catering to a diverse range of scenarios and needs. In this article, the Paddle OCR toolkit for Python is employed to conduct offline recognition on map tiles. By harnessing the parallel computing capability of GPU, we enhance the speed and efficiency of the text detection and recognition process, resulting in faster and more accurate text recognition.

5. Architecture

5.1 Workflow

This study proposes a method for inspecting map tiles through annotation recognition, show as Figure 2. Firstly, establishing a sample database containing text, label position, and the confidence detected and recognized using PaddleOCR on online tiles. Secondly, get map tiles and comparing them with tiles in the sample database to identify changes. Thirdly, informing reviewers of the changes via email for manual verification.

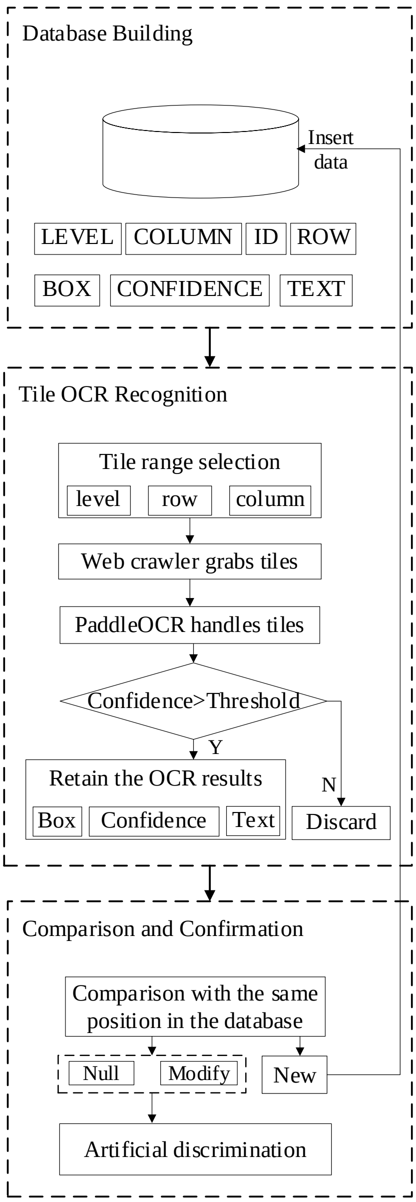

Figure 2. Map tile inspection process.

5.2 Database Building

This study defines a database for persistent storage of crawler recognition results. The attribute fields in the database include ID, TEXT, ROW, COLUMN, LEVEL, BOX, and CONFIDENCE, as shown in Table 1. ROW, COLUMN, LEVEL, and BOX together define the position of a tile. ID is a unique value assigned in the order in which the data is inserted into the database, and is the primary key of the table, which is assigned in a self-incrementing sequence. TEXT field stores the recognized text information, ROW represents the row number of the tile, COLUMN represents the column number of the tile, LEVEL represents the level of the tile, BOX is the coordinates of the text box detected, calculated from the upper-left pixel of the tile (0,0) in sequence, such as {{48.0,7.0},{87.0,7.0},{87.0,21.0},{48.0, 21.0}}, CONFIDENCE indicates the model's certainty about the correctness of the recognition results, representing the probability that the model considers the recognition result to be correct. Generally, a higher confidence value indicates a stronger belief by the model in the accuracy of the recognition result, making it more reliable. It is important to note that we do not store the retained map tiles; instead, we only store the results obtained from tile recognition and detection in the database.

Field	Type	Length
ID	VARCHAR	10
TEXT	TEXT	50
ROW	INTEGER	10
COLUMN	INTEGER	10
LEVEL	INTEGER	10
BOX	TEXT	50
CONFIDENCE	FLOAT	20

Table 1. Attribute fields

5.3 Map Tile Recognition

The map tile identification process mainly uses web crawler technology to request map tiles from the URL (http://t0.tianditu.gov.cn/cva_w/wmts?SERVICE=WMTS&REQUEST=GetTile &VERSION=1.0.0&LAYER=cva&STYLE=default&TILEMATRIXSET=w&FORMAT=tiles&TILEMATRIX={level}&TIEROW={row}&TILECOL={column}&tk=token) simulation of the map service, the tiles can be requested randomly or based on customized levels and ranges. The {x}, {y}, {z} represent the column number, row number, and level of the target map tile, respectively, which are values input by the user. The process is carried out without affecting the normal service of the target map website, and the crawling rate is strictly limited to ensure reasonable crawling data.

Since the map tiles are all 256*256 pixel resolution images, some map annotations span across two or more tiles, as shown in Figure 4, the annotations within the red bounding boxes span across tiles, while the annotations within the blue bounding boxes do not span across tiles. Therefore, it is necessary to preprocess the requested map tiles by merging the target tile with the surrounding eight tiles according to their positions into one image. The resolution of this image then becomes 768*768 pixels, which can prevent the loss of annotations on the target tile that may occur due to annotations spanning across tiles. As shown in Figure 3, the blue ones are the target tiles, and the white ones are the tiles surrounding the target tiles. The background of the annotated tiles is actually colorless and transparent; we have just added colors in the paper for illustration purposes.

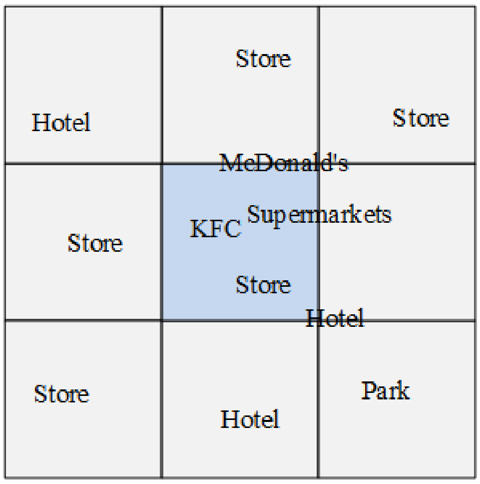

Figure 3. Annotation across tiles.

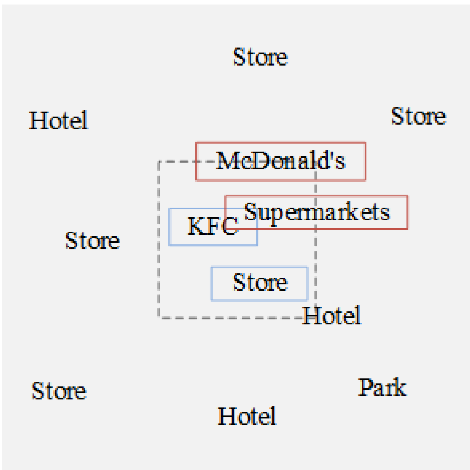

Figure 4. Map tiles merge.

Then, the open-source toolkit PaddleOCR is used to recognize text on the images. To improve recognition speed, we employ multi-threading technology to recognize multiple images simultaneously. The process in PaddleOCR includes text detection and text recognition. Text detection refers to marking the areas in the image that contain text with bounding boxes. The goal is to determine which areas of the image contain text for subsequent text recognition or other text processing tasks. Text recognition refers to converting the text content in the detected text areas into understandable text information. The result of the recognition for each image includes information such as box, text, and confidence. The box represents the coordinates of the bounding box area where each text is located in the image, this box is usually defined by the pixel coordinates {{x1,y1},{x2,y1},{x2,y2},{x1, y2}} of its four corners. There may still be issues with annotations being displayed incompletely across the edges of the tiles in the image. To avoid this problem, only the recognition results with coordinates in the central target tile are selected, with the x and y coordinate ranges being between [256, 512]. The text is the text information recognized based on the box range, and the confidence usually refers to the confidence score produced by the model for its recognition results. This is a value between 0 and 1, indicating the model's certainty or reliability in its predictions. Each detected text area is assigned a confidence value. This value reflects the model's level of certainty in predicting that area or text content. A higher confidence value usually means the model is more confident in the prediction result, while a lower confidence value may indicate that the model is less certain about the prediction result. We only retain recognition results with confidence greater than 0.9, filtering out those predictions that the model is less certain about. This helps to improve the accuracy and reliability of the recognition results.

Algorithm 1: The OCR of Map Tile
Input: Map tiles $\boldsymbol{D = \{d_1, d_2, \ldots d_i\}}$, Confidence threshold ∈
Output: OCR Results $\boldsymbol{C}$
1: Repeat the following steps until all tiles were recognized
2: Merge the target tile and its adjacent 8 tiles into one image.
3: Use PaddleOCR for text detection and text recognition on the image, get the boxes, texts and confidences.
4: Record the OCR Results $\boldsymbol{c_k}$ that confidence value is more than the threshold ∈;
5: **Return** OCR Results $\boldsymbol{C = \{c_1, c_2, \ldots c_k\}}$

Table 2. The OCR of map tile algorithm

5.4 Comparison and Confirmation

Retrieve records from the database that match the target tile's ROW, COLUMN, LEVEL, and BOX information for comparison. If the TEXT is the same, there is no change. If the recognized TEXT is different, send an alert message for manual confirmation. If there is no position information for the target tile, update the database with the recognized information, which is always the case during the first run. If the recognized tile annotations are all empty, but the database has recognized text for that tile, it can be determined that these tiles are missing, and an alert message should be sent for manual confirmation. Changes detected during online map inspections will be sent to reviewers via email. The email will mainly include the location information of the tiles, details of the changes, and the reasons for any errors. Upon receiving the email, reviewers will manually confirm the changes. Any erroneous information will be promptly corrected in the map tiles.

6. Implementation

6.1 Data Preparation

An experiment was implemented to assess the effectiveness and feasibility achieved by our proposed approach. In order to avoid possible impacts on online map sites, we use local data to publish WMTS services to simulate map sites. The experimental dataset used for map service publishing in this paper is from the Open Street Map (OSM), which includes POI data of types such as schools, hospitals, restaurants, stations, hotels, parks, etc., which are in the WGS 84 coordinate system. In the experiment these data were made into map tiles to be published through a map service tool. The map tile projection uses the Mercator projection.

In order to be able to save the results of subsequent OCR recognition, we used a PostgreSQL database to create a database of OCR results. The attribute fields in the database include ID, TEXT, ROW, COLUMN, LEVEL, BOX, and CONFIDENCE. Based on the PostgreSQL database, which is an open-source and free software with good performance. PostgreSQL is a powerful, open-source object-relational database system with an emphasis on extensibility and standards compliance. It is designed to handle a range of workloads, from single machines to data warehouses with many terabytes of data. PostgreSQL provides a robust and reliable platform for many different application areas, including web and mobile applications, geospatial databases, and enterprise systems.

6.2 Map Tile Recognition

In this experiment, we built an automated tool using the Python 3.8 environment for web crawling and tile recognition. For the web crawling part, we used the `requests` package, which is a very popular HTTP library in Python that allows you to send various HTTP requests to interact with web services. Known for its concise API and ease of use, this library makes it very simple to make network requests. For the tile recognition part, we used the PaddleOCR package. To support GPU acceleration, we installed the paddlepaddle-gpu version, which is the GPU-accelerated version of PaddlePaddle. By leveraging the parallel computing power of the GPU, paddlepaddle-gpu can significantly speed up the training and inference of deep learning models. The url for the map service address is *http://192.168.10.154:8080/rendering?z={z}&x={x}&y={y}&prj=3857&imageTpye=png&tileSize=256*, only need to alter the numerical values of *{x}, {y},* and *{z}* to fetch the specific tile. For example, when *z=17, x=108782, y=53206,* the tile obtained is shown in Figure 5. It includes both Chinese and English map annotations. Additionally, to accelerate the processing speed, the experiment adopted a parallel processing strategy, achieving an average speed of processing 5 tiles per second on a regular office personal computer.

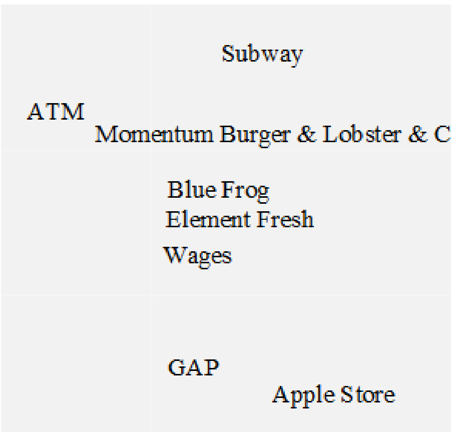

Figure 5. A map tile.

There were many important parameters in the OCR of map tile algorithm. The value of ∈, *Row Range (x), Culoum Range (y), Level* were given as shown in Table. 3. In this experiment, the value of ∈ was 0.9, which means that only recognition results with a confidence level greater than 0.9 will be retained. The *Row Range (x)* set in the experiment is from 53185 to 53219, the *Culoum Range (y)* is from 108759 to 108804, and the tile level is a constant value of 17. This indicates the scope and level of the test data. This means that the range and level of test data, which is chosen arbitrarily, is not specialised. In this area, there are a total of 1,530 tiles, and it should be noted that some tiles do not have annotations.

Parameter	Value
∈	0.9
Row Range (x)	[53185, 53219]
Culoum Range (y)	[108759, 108804]
Level	17

Table 3. Experimental parameters

6.3 Results

We conducted OCR recognition on the tiles of the selected experimental area for this paper, and filtered out the recognition results. The recognized results were stored in a database, and Table 4 lists some of the recognition results and fields. After the first run of the program, a total of 500 recognition entries were obtained. POI data of various types, such as hotels, restaurants, hospitals, schools, shops, and coffee shops, can all be recognized. The "NAME" column represents the actual field values of the data, while the "TEXT" column contains the results of the OCR recognition. As shown in Figure 6 and Table 5, among the 300 recognition results, there are only 18 with a confidence level below 0.9, which accounts for 6% of the total. There are 282 recognition results with a confidence level above 0.9, accounting for 94% of the total. Among these, 148 have a confidence level above 0.99, which represents 49.33% of the total. In the experiment, it was found that 94% of the recognition results had a confidence level greater than 0.9, indicating that the PaddleOCR model has a high degree of accuracy and reliability, and users can have a high level of trust.

Id	X	Y	Name	Text	Conf-idence
01	108767	53211	Yun cafe	Yun cafe	0.971
02	108767	53201	Wages	Wages	0.998
03	108767	53201	Blue Frog	Blue Frog	0.933
04	108789	53209	Xihua Hotel	Xihua Hotel	0.981
05	108789	53212	Jinrunfa	Jinrunfa	0.999
06	108790	53202	Xiaoyin School	Xiaoyin School	0.999
07	108790	53203	GAP	GAP	0.998
08	108770	53196	Apple Store	Apple Store	0.993
09	108782	53206	Small cafe	Small cafe	0.987
10	108777	53208	Grand Cozy Hotel	Grand Cozy Hotel	0.995
…	…	…	…	…	…

Table 4. The OCR results in the database

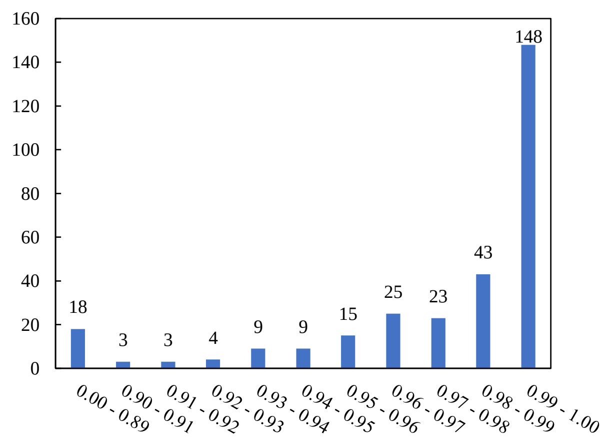

Figure 6. Confidence histogram.

Confidence	Percentage
<0.9	6%
[0.9,1)	94%
[0.99,1)	49.33%

Table 5. Confidence percentage

To further test the recognition capability of the method proposed in this paper for identifying changing information, we updated 50 annotation entries as test data. We supplemented the original 'name' field with random Chinese characters or English letters to form a new 'name', such as changing "Yun cafe" to "Yun cafe b", to verify the model's ability to recognize the changed information. The experimental results show that out of 50 updated name fields, 49 were recognized, accounting for 98%. This proves that the method proposed in this paper is feasible.

Id	Test Name	Text	Confidence
01	Yun cafe	Yun cafe	0.971
02	Zhongxing Hotel	Zhongxing Hotel	0.998
03	Xueshiyuan Hotel	Xueshiyuan Hotel	0.933
04	Xihua Hotel	Xihua Hotel	0.981
05	Jinrunfa	Jinrunfa	0.999
…	...	…	…

Table 6. Result comparison

6.4 Discussion

In this experiment, we use OSM data most test data to make map tiles, simulate a map service website, crawl the map tiles by web crawler, use OCR tool for tile annotation recognition, and save the recognition results into the database. The experimental results are normal, 94% of the text recognition with a confidence level of more than 90%, for the updated annotations can also be found accurately, the experimental results verify the feasibility and effectiveness of the map tile detection and recognition method proposed in this paper. In addition, the experiments provide an automated tile identification scheme that provides a new path for online map tile quality control, where correct map annotation is very important in geographic information services no matter when and where.

7. Conclusions

Aiming to overcome online map tiles check drawbacks, this study proposes an approach to checking online map tile based on PaddleOCR model. The contribution of this study is hence three-fold: (1) this study is the first application of OCR technology to text detection and text recognition of map tile annotations, creating a new way for automated quality control of map annotations; (2) this approach proposes a map annotation recognition method based on PaddleOCR model, which can realize automated text extraction and comparison to ensure the accuracy and consistency of map data; (3) there are many advantages in map tiles check using this method, such as improving efficiency and speed up, especially when dealing with large scale map data.

References

Ding, L., Huang, W., Zhang, H., Tang, D., Zha, Z., Zheng, X., Wang, C., Wang, Z., and Li, H.: An approach to updating vector features by clustering algorithm, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLIII-B4-2022, 607 – 612, https://doi.org/10.5194/isprsarchives-XLIII-B4-2022-607-2022, 2022.*

Du, Y., Li, C., Guo, R., Yin, X., Liu, W., Zhou, J., Bai, Y., Yu, Z., Yang, Y., Dang, Q., & Wang, H. (2020). PP-OCR: A Practical Ultra Lightweight OCR System. *ArXiv, abs/2009.* 09941.

García, R., de Castro, J. P., Verdú, E., Verdú, M. J., & Regueras, L. M. (2012). Web map tile services for spatial data infrastructures: Management and optimization. Cartography-A Tool for Spatial Analysis, 26-48.

Hu, X., Ding, L., Shang, J., Fan, H., Novack, T., Noskov, A., & Zipf, A. (2020). Data-driven approach to learning salience models of indoor landmarks by using genetic programming. *International Journal of Digital Earth*, 13(11), 1230-1257.

Islam, N., Islam, Z., & Noor, N. (2017). A Survey on Optical Character Recognition System. *ArXiv, abs/1710.05703*.

McNeill, G., & Hale, S. A. (2017, June). Generating tile maps. *In Computer Graphics Forum* (Vol. 36, No. 3, pp. 435-445).

Mithe, R., Indalkar, S., & Divekar, N. (2013). Optical character recognition. *International journal of recent technology and engineering (IJRTE)*, 2(1), 72-75.

Nguyen, T. T. H., Jatowt, A., Coustaty, M., & Doucet, A. (2021). Survey of post-OCR processing approaches. *ACM Computing Surveys (CSUR), 54(6), 1-37.*

OpenGIS Web Map Tile Service Implementation Standard Open Geospatial Consortium document OGC 07-057r7

Patel, C., Patel, A., & Patel, D. (2012). Optical character recognition by open source OCR tool tesseract: A case study. *International journal of computer applications, 55(10), 50-56.*

Singh, A., Bacchuwar, K., & Bhasin, A. (2012). A survey of OCR applications. *International Journal of Machine Learning and Computing, 2(3), 314.*

Zhang, H., Du, M., Huang, W., Ding, L., Tang, D., and Jiang, J.: Research of vector tile construction technology based on apache sedona, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLIII-B4-2022, 639–644, https://doi.org/10.5194/isprs-archives-XLIII-B4-2022-639-2022, 2022.*