Character Recognition API IME

Technology Reports

Highly Accurate Character Recognition Technology

With the spread of smartphones and cloud-based services, services able to extract characters from photographs and images are also showing signs of expansion. However, extracting characters from photographs of scenery is a difficult issue, and there is room for improvement, especially in recognizing Japanese. Thus, we have developed technology that implements highly-accurate character recognition from scenery photographs. We have provided a service using this technology, created a platform infrastructure, and published an API. Our goal is to develop open-innovation services through this infrastructure.

Service & Solution Development Department
Takafumi Yamazoe
Tetsuo Sumiya
Atsushi Iwasaki

1. Introduction

Lately, it has become possible to provide services relying on high-level processing through the spread of smartphones and cloud-based services. Some very interesting services that provide various conveniences by recognizing characters within photographs and images have begun to appear on the market. However, it can be very difficult to extract characters correctly when there are overlapping adverse conditions in the image besides characters, such as buildings and other scenery, shadows, moiré[*1] patterns and reflections.

By introducing language processing, we have developed a character recognition technology that is able to extract characters accurately even under adverse conditions. We have also combined this technology with a large language database to create a platform[*2] infrastructure.

In this article, we describe this character recognition technology as well as an Android™[*3] application created using the technology. We also mention initiatives to publish an Application Programming Interface (API)[*4] which will enable ordinary program developers and content providers to use this infrastructure in developing their applications and Web services.

2. Character Recognition Technology

The new character recognition technology is shown in **Figure 1**. It consists of the sequence of processes: (1) Character region detection, which finds regions in the image that could be characters, (2) Character recognition, which examines what characters might appear in these regions and (3) Language processing, which corrects or eliminates results from among the character recognition results.

*1 **Moiré**: A new striped pattern emerging from overlapping well-ordered striped patterns, due to differences in periodicity between the stripes.

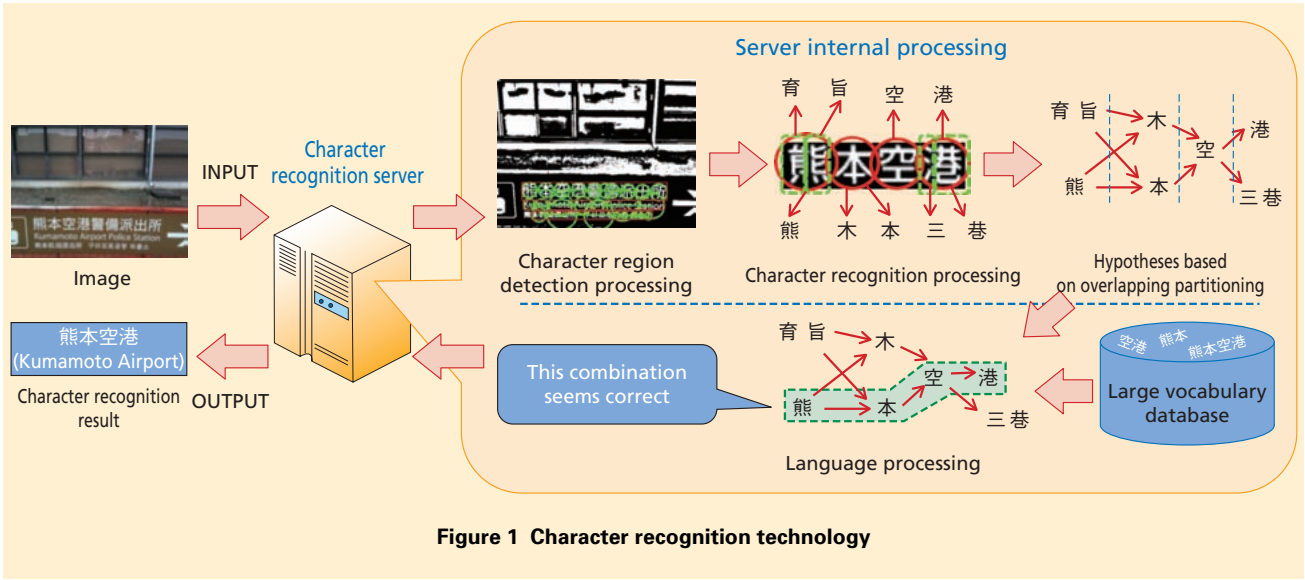


Figure 1 Character recognition technology

2.1 Detection of Character Regions

This technology is intended to handle not only images scanned from documents, as are generally used for OCR, but also images of scenery, where the positioning and size of characters is not uniform. Thus, it must first decide what sort of characters appear in the image from the patterns of shapes in the image and various characteristics of those shapes. In detecting these shape patterns, we place a priority on reducing the number of correct patterns that go undetected (false negatives), even if the number of incorrect patterns detected (false positives) increases. This is because we eliminate incorrect results in later processing.

Images of scenery can contain shapes that resemble characters, such as window frames or the leaves and branches of a tree, so we narrow the scope of character candidate shape patterns using the positioning and orientation of the patterns.

2.2 Character Recognition

Next, we correct the reduced list of character region candidates for distortion and extract the regions as strings of characters. We then divide the strings into individual characters and study what characters they may represent individually. Dividing into individual characters can be difficult, for example, distinguishing between an "m" and the sequence of "r" and "n" appearing without a gap. Thus, we perform our recognition using overlapping character-partitioning position hypotheses.

2.3 Language Processing

The result of character recognition obtained in this way is a complex set of hypotheses consisting of multiple character partitions and positions, with a large number of result candidates for each character. For scenes consisting of patterns resembling characters, it also includes results from falsely detected regions. From this character recognition result with complex structure and including errors, we correct detected words and eliminate results that are meaningless as words using language processing. For example, in the case of Fig.1, for the "熊" in "熊本空港" the results included "育" and "旨", by erroneously partitioning it into the left and right parts. For "本," it also determined that "木" was more correct than "本" In such cases, the most likely possibility from among all pattern combinations can be selected using language processing.

For language processing, we use a large scale vocabulary database that we are gathering ourselves, which supports

*2 **Platform**: A shared infrastructure. Here, we mean a system providing functionality to various connected applications or Web services.
*3 **Android™**: A software platform for smartphones and tablets consisting of an operating system, middleware, and major applications. A trademark or registered trademark of Google Inc., United States.
*4 **API**: An interface that provides functionality of an application program. Allows developers to develop programs by combining control logic with API calls.

over one million words and includes the latest vocabulary.

3. Providing a Character Recognition API

We have developed the character recognition technology and large scale vocabulary database described above into an infrastructure platform and have published an API for enterprise and ordinary developers [1]. This API includes a line-image recognition API and a scene-image recognition API.

- Line-image recognition API

  The line-image recognition API obtains words recognized from the image of a single line of characters extracted from a larger image. Character recognition from a line image is fast, and returns a result within approximately three seconds.

- Scene image recognition API

  The scene-image recognition API extracts characters from a scene that contains characters and returns the recognized words together with information such as their position in the image. Recognition processing for a scene image requires complex computation, so obtaining the result can take approximately two minutes.

The system architecture of the character recognition API is shown in **Figure 2**. Processing load is balanced among all servers using a load balancer*5. Line-image character recognition requires relatively less processing, so a result can be returned quickly even when the request load is high. Scene-image character recognition requires longer processing times, so the scene character recognition API does not maintain the session until processing has completed, but closes it after receiving the request. Then, the application using the character recognition API can query the processing server directly regarding the status of the request. This asynchronous structure allows for a highly scalable implementation.

This API allows developers to build various types of mashup services (navigation, dictionary, translation, travel, goods management, etc.). Many new services will be created through provision of this API, showing the spread of character-recognition services. Also, by gathering large amounts of image data, we hope to improve the accuracy of character recognition.

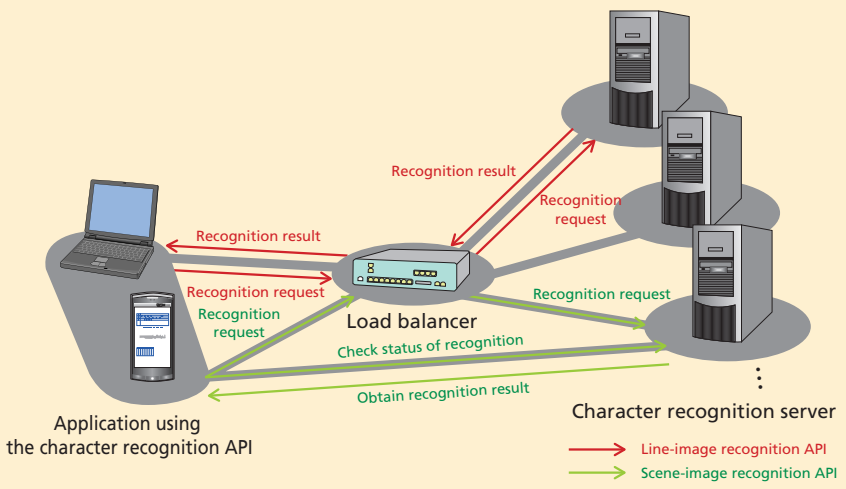
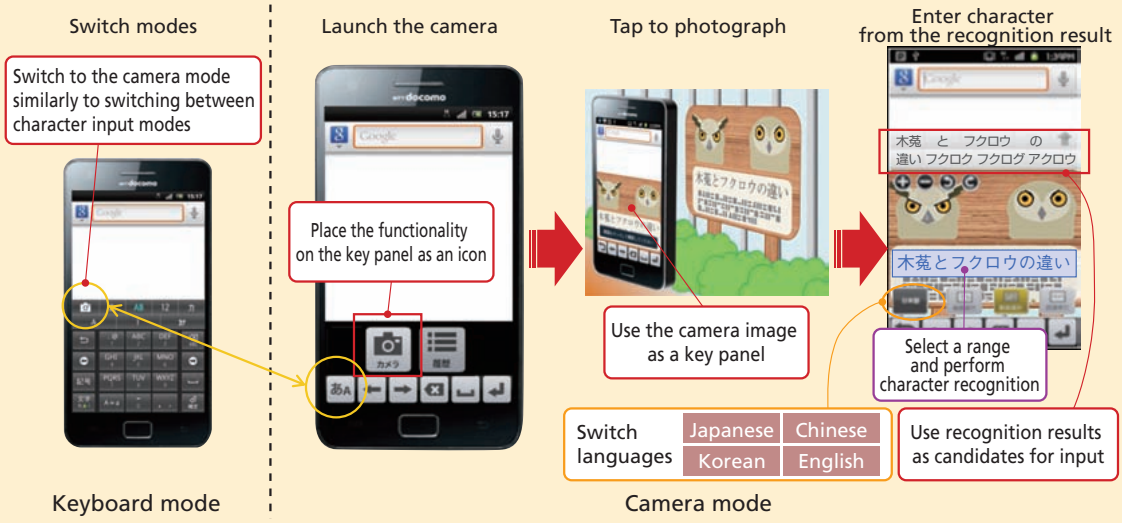


Figure 2 Character recognition API system structure

4. Example Service Using the API

As an example of a service using this API, we developed a photographed character Input Method Editor (IME) for Android. Existing IMEs use a keypad or QWERTY keyboard for input, but this application allows the camera or an image gallery to be used for input. The results of character recognition are shown in the same field that candidates for ordinary predictive input methods use, and words from the character-recognition results can be used directly as character input. The user can select a region of the image containing a single line of character, which is then recognized using the line-image recognition API (**Figure 3**). Alternatively, images can be registered in an image gallery, where they are processed to detect words, and these can then be used for input through the photographed-charac-

*5 **Load balancer**: A device that centrally manages external requests and transmits them to servers with equivalent functionality. Used to distribute load over the servers.



Google™ : Google™ or its icon is a trademark or registered trademark of Google Inc., United States.

Figure 3 Photographed character input (using camera)

ter input key panel (**Figure 4**). This feature is implemented using the scene image recognition API.

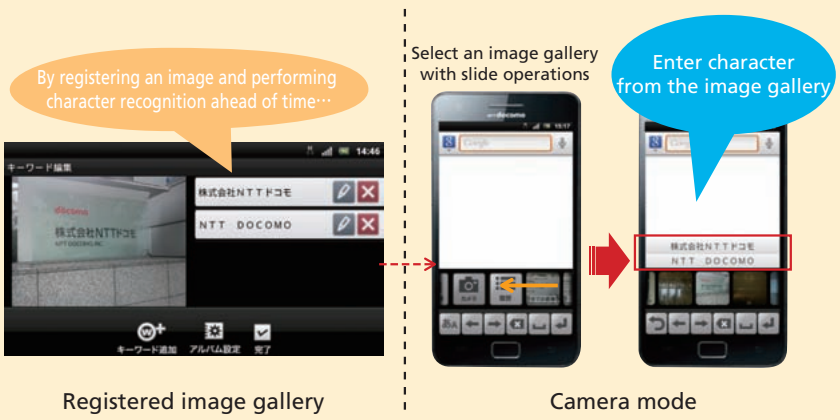


Figure 4 Photographed character input (using gallery)

5. Example of Application in Terminal Applications

As an application example, we developed a restaurant menu translation application using the language processing features of the character recognition API. An overview of this application is shown in **Figure 5**. Language processing is used to detect and correct names of dishes, and by implementing these functions on the terminal, we were able to implement real-time operation that does not require communications. The application reads a menu in a foreign language or Japanese, and immediately translates it into Japanese or a foreign language. If, while on vacation, the user cannot read a menu (in Korean, Chinese, English or Japanese) he/she can simply display the menu in this application with the camera, and a Japanese or a foreign language translation is displayed on the same screen. The user selects words to translate on the menu, positions them within a frame in the middle of the screen, and the translation result is displayed immediately. The words, before and after translation, can also be used to search the Internet by simply tapping a Share button. In the future, we plan to implement a mobile service that will allow users to see Japanese or a foreign language translations of character that is difficult to input, such as on a sign or on products

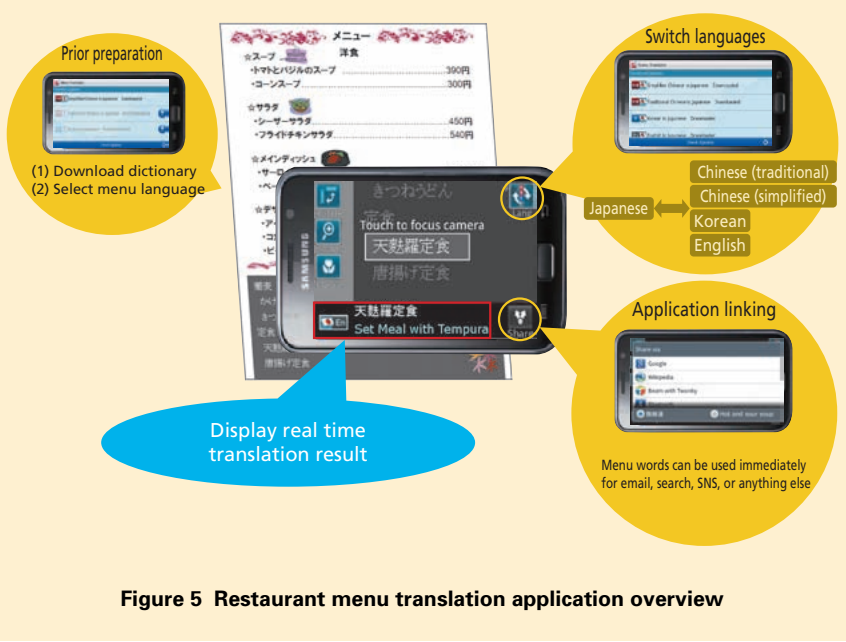


Figure 5 Restaurant menu translation application overview

while shopping overseas, simply by displaying it on their mobile terminal screen using the camera.

6. Conclusion

We have developed technology that enables characters to be extracted from images and photographs easily, by building it into an infrastructure. By applying this technology and infrastructure to photographs taken using a mobile phone camera, we have further promoted advances in mobile services.

In the future, we are planning to improve performance by gathering image data on a large scale, to add functionality such as support for handwritten character and additional languages, and to study application of the infrastructure we have developed for various services.

REFERENCE

[1] NTT DOCOMO: "Character recognition API."
http://recognize.jp/