FASTScan: A Fast Mobile Image Recognition SDK with Local Database Using Convolutional Neural Network

Gamil Shamar, Siok Yee Tan, Haslina Arshad

Center for Artificial Intelligence Technology, Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Abstract—Image target recognition in mobile platform use cloud to perform heavy tasks as remote processing. Despite the advantages of using cloud, there are still challenges which are related to connection and time processing delay. The time will increase on sequential search in image database especially when the image label is not included in the database. This paper will present an efficient technique for image target recognition by recognizing images using a classified local database with local processing on mobile devices to reduce server dependency. Image recognition is performed using an updated local database which has been downloaded from the server and then training standard dataset using AlexNet model based on Convolutional Neural Network (CNN) to store class name in the database. The marker image captured by the mobile camera will be used in classification model to predict the image class. The technique will be tested Food101 dataset which contains 101-kind food photos and evaluated on mobile platform, iOS and Android. Compared to MyXScan, a cloud based image recognition SDK, this technique is three times more efficient in terms of time in fetching classified data and processing locally.

Index Terms—Mobile, Local Database, Image Recognition, CNN

I. INTRODUCTION

Mobile devices became necessary for everyday uses as tools of communication and lifestyle effectively and appropriately [7]. Increased capabilities of mobile visual computing application show good signs of the foreseeable future [5]. Mobile technologies are leading to a need of the clouds due to the demand of the applications for processing ability, storage size, and reduced power. Mobiles can access the shared resources supported by the cloud on request through on-demand approach [4]. This drives to an integration of the mobile application with cloud computing which can be fast supplied and released with less handling efforts or interaction between clients and service providers [6].

Even though newer mobile devices have high processing capabilities, feature support and storage, these devices still have restriction by relative lack of bandwidth, computing ability and power. For this reason, cloud computing comes up with solutions to spread these devices [2]. Many researchers have proposed solutions which were focusing on the integration between Cloudlet and Cloud to reduce the dependency on the Cloud where mobile devices do not need to contact the cloud server but instead only contact the Cloudlet [9]. There are still limitations of using Cloudlet since the coverage area of Cloudlet is very small and it faces some difficulties on scalability. [8] proposed a model to make the coverage area wider for the mobile users and allow them to do their requests services with lower cost in terms of power and delay but it still has problems in terms of the scalability.

Mobile services such as image target recognition may support their functionality by allowing the cloud to do heavy tasks as the remote processing [6]. Despite the use of clouds is useful, it will lead to challenges which are related to connection delay for each requested image to the cloud. MyXScan, a cloud-based image recognition SDK which works on Android platform integrated with web portal, has limitation of life time for each request to the cloud [1]. It also has a weak point of processing all images that captured by the mobile camera with not classified database. In this case the time will increase especially when the database is huge.

The aim of our work is to process mobile images data in order to recognize images with minimum processing delay. This work presents an image recognition solution with local processing on mobile devices and reducing server dependency. The main idea is developing an efficient and speedy technique based on the mobile environment (CPU, RAM, battery, speed). The interesting element is recognizing frames which have been captured by the mobile camera then using local data in case the mobile device has the updated database and then follow-up with the remaining processes of image recognition. Moreover, this work proposed a model for image classification to
predict the class of image marker before recognition process. This integration runs locally on iPhone or iPad and it is able to perform queries to the server for updating the new image records.

II. RELATED WORK

Image target recognition in mobile platform use cloud to perform heavy tasks as remote processing. Despite the advantages of using cloud, there are still challenges which are related to connection and time processing delay. The time will increase on sequential search in image database especially when the image label is not included in the database. In many research have been done on these two areas which trying to reduce the dependency on the server at every request and classify the database. In this literature review will discuss these solutions order by this research topics.

A. Cloud-Based Mobile Image Recognition

Previous researches have proposed cloud-based mobile image recognition with different purposes. The goal is lightweight implementation on mobile devices based on processing performed over the cloud. The disadvantages of this method are related with the time processing specially when the network connection is slow. Time is also lost when each frame has to make separate request to the server continuously. MyXScan is a cloud-based image target recognition for mobile application and it has been proposed as integration of cloud computing into mobile environment [1]. The main idea of MyXScan is scanning image in Android application then sending the image features to the cloud for recognition. After that, the cloud will match all images features in the database with input images features to find the best matcher. Finally, the result will be sent to mobile application as JSON format and view it over user interface. Even though this SDK offers a cheaper solution compared with others, it is still facing challenges related with the request time to the server and handling all images to do the whole process.

B. Integration Between Cloudlet

Many supporting models have been proposed to improve the efficiency of accessing resources with minimal cost and delay. [9] presented a secure framework which is integrated between Cloudlet and Cloud. In this model, it does not require to reach the server cloud and instead contact the Cloudlet. They presented a master cloudlet management to settle the exchange the information between the Cloudlet and the Cloud. In health care there are also a lot of services suppliers need to improve their services for long term. Hence, a multi-layer computing model integration of Local Cloud layer and Enterprise Cloud layer has been proposed [10]. The goal is to handle data which has been collected to make cases. These cases will help to find abnormal situation within observed data.

Even though the Cloudlet presents high interest as a Cloud solution, there is still a limitation. A system was proposed based on the limitation of cloudlet is an effort to combine the Cloudlet ability into the Mobile Edge Computing. This model aims to expand the converge area for the mobile users in which the users can do their requests services with lower cost in term of power and delay [8]. The main goal of this system is to address the challenges and limitations of Cloudlet which is two points. The first one is the coverage area regions of Cloudlets are very small and need to widen the regions for accessing the resources. The second one is Cloudlets have critical issues when it comes to scalability.

C. Mobile Image Recognition using Convolutional Neural Network

Due to the recent progress of the studies on deep learning, convolutional neural network (CNN) based methods have outperformed conventional methods with a large margin [11]. Therefore, CNN-based recognition should be introduced into mobile object recognition. However, since CNN computation is usually performed on GPU-equipped PCs, it is not easy for mobile devices where memory and computational power is limited. In this work, they explore the possibility of CNN-based object recognition on mobile devices, especially on iOS and Android devices including smartphones and tablets in terms of processing speed and required memory. They mainly used the augmented Food101 dataset which is a 101-class food categories dataset containing at most 1000 food photos for each food class, because the objective of this work is implementing practical CNN-based recognition engines. There was no image recognition process in this work, they just make classification for target images and test the result based on the accuracy and time.

III. PROPOSED WORK

This approach contains two main components which are image recognition process in local database and using CNN model to classify the image. First, a training Food101 dataset based on AlexNet model in CNN which is handled using Caffe (Krizhevsky, Sutskever, and Hinton 2012). This model has been implemented using the Nvidia GPU on PC. The expected class name of target image will be stored in the database. This database was created from images that have uploaded to the server through the web portal (www.myxscan.com). In this portal we do two main things, first we classify the class of uploaded image then store the class name to the database. Second, we extract image features and store them to the database as JSON format. In mobile platform, the image class will be predicted from the class name by using the same model. The image is then used in image processing process.

A. Predict Image Class using CNN Model

In Figure 1, image database contains a group of classified images which is uploaded to web portal by users and then downloaded to the mobile synchronously. For this reason, CNN Model have been used to get the class of query image.
A set of relevant image descriptors to the query image is returned. Those images with high score will enter image recognition process first. Otherwise, if there are no images returned, so no need to perform the rest of the steps will not be performed.

**B. Pattern Recognition**

The main components in the pattern recognition process are feature descriptors which are ORB or BRISK descriptors. While users upload target image in the web portal, the target image will be rated in the range from zero stars to five stars. The rating method is based on two measures in target image, which are the number of features and the contrast. Throughout the real-time image recognition, the image that have been retrieved as similar image will also be extracted using BRISK descriptor, and then the feature's data will be handled with the local recognition process. In Figure 2, if the captured image’s features are matched, then the image details will show in users’ UI.

**C. iOS Application Development**

The iOS application is developed to test the proposed technique. Development of iOS application includes four phases which are:

- Perform the image recognition process locally and show the result on user interface.
- Use CNN model to predict the class name of the query image.
- Matching only the highest probability with marker image to get the exact image.
- No image match status.
- Match status.

**IV. EXPERIMENT RESULTS**

The experiment has been measured in iOS platform using iPhone X. The result shows the improved technique able to achieve high matching accuracy up to 98% and the processing time is very fast compared to previous work; MyXScan. The iOS app has been implemented in “FastScan” application which are shown in Table 1. The experimental result has been done randomly with different labels of food names and database contains 101 rows only. Also comparing the processing time of mobile image recognition in different scenarios. As it shown these scenarios occur in the cloud without classification, cloud with classification, local without classification and local with classification. From Table 1, the local processing with classification is the obtained the shortest computation time compared the other techniques.

<table>
<thead>
<tr>
<th>Class Label</th>
<th>Cloud without Classification (time second)</th>
<th>Cloud with Classification (time second)</th>
<th>Local without Classification (time second)</th>
<th>Local with Classification (time second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>beef tartare</td>
<td>15.68</td>
<td>2.43</td>
<td>0.40</td>
<td>0.11</td>
</tr>
<tr>
<td>baklava</td>
<td>16.53</td>
<td>2.04</td>
<td>0.25</td>
<td>0.14</td>
</tr>
<tr>
<td>macaroni and cheese</td>
<td>16.32</td>
<td>3.58</td>
<td>0.39</td>
<td>0.17</td>
</tr>
<tr>
<td>Tuna tartare</td>
<td>16.61</td>
<td>2.36</td>
<td>0.36</td>
<td>0.12</td>
</tr>
<tr>
<td>Pad thai</td>
<td>15.91</td>
<td>1.63</td>
<td>0.43</td>
<td>0.11</td>
</tr>
<tr>
<td>Deviled eggs</td>
<td>15.81</td>
<td>2.03</td>
<td>0.39</td>
<td>0.11</td>
</tr>
<tr>
<td>waffles</td>
<td>16.43</td>
<td>2.05</td>
<td>0.34</td>
<td>0.11</td>
</tr>
<tr>
<td>sashimi</td>
<td>18.47</td>
<td>1.92</td>
<td>0.41</td>
<td>0.11</td>
</tr>
<tr>
<td>Cheese plate</td>
<td>13.00</td>
<td>2.88</td>
<td>0.38</td>
<td>0.11</td>
</tr>
<tr>
<td>Prime rib</td>
<td>14.23</td>
<td>2.12</td>
<td>0.37</td>
<td>0.12</td>
</tr>
<tr>
<td>Spring rolls</td>
<td>14.02</td>
<td>2.15</td>
<td>0.20</td>
<td>0.10</td>
</tr>
<tr>
<td>Hot dogs</td>
<td>14.94</td>
<td>2.00</td>
<td>0.36</td>
<td>0.09</td>
</tr>
<tr>
<td>Baby back ribs</td>
<td>19.73</td>
<td>2.16</td>
<td>0.41</td>
<td>0.11</td>
</tr>
<tr>
<td>Lobster roll sandwich</td>
<td>10.88</td>
<td>2.80</td>
<td>0.38</td>
<td>0.13</td>
</tr>
<tr>
<td>cannoli</td>
<td>18.52</td>
<td>3.59</td>
<td>0.21</td>
<td>0.10</td>
</tr>
<tr>
<td>Average</td>
<td>15.81</td>
<td>2.38</td>
<td>0.35</td>
<td>0.12</td>
</tr>
</tbody>
</table>

**V. CONCLUSION**

The client and server processing delay will increase based on the connection speed and number of images in the cloud database. In addition, more wasting time will occur when the database isn’t classified or categorized so the
recognition process will be performed sequentially until it finds the targeted image. For this reason, the whole process of image recognition in mobile needs more improvement either on fetching the data from the cloud or the structure of the image database. Hence, this work has developed an image recognition SDK using local database to minimize processing delay and reduce server dependency and build CNN model to classify images and then store the label in the database to fetch only relevant images.

VI. ACKNOWLEDGEMENT

The authors would like to thank all those participated in this work as part of the project sponsored by research university grant Universiti Kebangsaan Malaysia (Inovasi-2017-007).

REFERENCES


