

## An Efficient Cloud based Image Target Recognition SDK for Mobile Applications

Haslina Arshad<sup>##</sup>, Lam Meng Chun<sup>#</sup>, Waqas Khalid Obeidy<sup>#</sup>, Tan Siok Yee<sup>#</sup>

<sup>#</sup>Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Selangor, Malaysia  
E-mail: \*haslinarshad@edu.ukm.my

---

**Abstract**— Smartphones have exploded in popularity in recent years, becoming ever more sophisticated and capable especially when these devices try to access the shared pool of computing resources provided by the cloud, on demand. Mobile services such as image target recognition SDK may enrich their functionality by delegating heavy tasks to the clouds as the remote processing. This paper proposes an image target recognition SDK based on cloud with the main goal of lightweight implementation on mobile devices based on processing performed over the cloud. In such circumstances, the focus of the proposed image target recognition SDK needs to be on effectiveness, robustness, and simplicity, while still preserving a high level of functionality (i.e. good recognition). Application areas involve android library development, pattern recognition, and web portal development. The applications mentioned in this paper bring an added value by being success stories for mobile cloud computing domain in general

**Keywords**— cloud computing; image recognition; feature extraction; mobile application

---

### I. INTRODUCTION

Mobile devices are increasingly becoming an essential part of day to day life as the most effective and convenient communication tools not bounded by time and place. Mobile users accumulate rich experience of various services from mobile applications and become a powerful trend in the development of IT technology as well as commerce and industry field. Technology is developing rapidly with the demands of changing times [1]. Cloud computing has been widely recognized as the next generation computing infrastructure which allows users to use infrastructure, platforms, and applications at low cost [2]. This paper presents cloud-based image target recognition for mobile application as an integration of cloud computing into the mobile environment. Mobile computing and cloud computing domains are converging as the prominent technologies that enable developing the next generation services based on data-intensive services [3]. Most images are digitized and kept in the cloud for better organization and management [4]. Cloud computing is a style of computing in which, typically resources on demand are provided over the Internet to users who need not have knowledge of, expertise in, or control over the cloud infrastructure that supports them. In addition, cloud computing enables users to utilize resources in an on-demand fashion. For this reason, integration of the mobile application with cloud computing can be rapidly provisioned and released with the minimal

management efforts or interaction between clients and service providers. This paper presents image target recognition SDK as a visual recognition solution based on the cloud where the novel and interesting elements of the presented approach are the integration of android library, pattern recognition and finally, integrating these solutions through a web portal.

### II. MATERIAL AND METHOD

Cloud computing is known to be a promising solution for mobile computing due to mobility, communication, and portability [5-8]. Mobile applications based on cloud enable mobile users to store or access large data through wireless networks. With the cloud, the users can save a considerable amount of energy and storage space on their mobile devices. Besides, application based on the cloud also helps in reducing the running cost for computer intensive applications that take long time and a large amount of energy when performed on the limited – resource devices. In addition, the cloud-based mobile application can make efficient use of the collected record from different users to improve the effectiveness of the services.

Although mobile application based on the cloud has many advantages for mobile users and service providers, because of the integration of two different fields, there are many technical challenges have been issues by the researchers. Research in [9] proposed an e-commerce platform focuses on data processing speed for the users. However, the

effectiveness of their applications mainly depends on the security of the users, customer satisfaction, customer intimacy and cost effectiveness. However, utilizing the cloud for mobile applications with the high storage capacity and powerful processing ability are previously proposed by [10], presents the benefits to enhance the communication quality. In this case, smartphone software based on the open-source JavaME UI framework and Jaber for the client were used. Later, through a web portal users can communicate with them. In addition, a contextual m-learning system based on mobile interaction in augmented reality environment platform claims the efficiency to integrate the mobile application with cloud [11]. Image Exchange utilizing large storage space in clouds for mobile users is the another mobile application based on cloud enables users to upload image after capturing which helps the users to save a considerable amount of energy and storage space because all the images are sent and processed on the clouds [12]. However, very few researchers previously work on mobile applications for target image recognition based on the cloud to integrate mobile applications with the cloud.

### A. Vuforia Software Development Kit

Vuforia SDK uses computer vision-based image recognition technique and enabling capability of mobile applications and developers. Vuforia SDK is compatible with all major platform for software development: Windows, Unity 3D, Android, and iOS. Vuforia platform consists of Target Manager and License Manager. Target Manager allows the developer to create and manage targets and databases. It included both Cloud Target Database and Device Target Database. The developer needs to create a license key for their application from License Manager before creating an application. There is a disadvantage of Vuforia SDK. The pricing of Vuforia SDK is expensive, and the pricing is shown in Fig. 1.

	Starter	Classic	Cloud		
			Bronze	Silver	Gold
<b>Features</b>					
Image, Object, Cylinder, User-defined Target, Frame Marker, Text & Smart Terrain	✓	✓	✓	✓	✓
Cloud Recognition Service	✓	--	✓	✓	✓
Remove Watermark	--	✓	✓	✓	✓
<b>Pricing &amp; Usage</b>					
Device Recos	unlimited				
Cloud Recos	1,000 per month	--	10,000 per month	50,000 per month	150,000* per month
Cloud Targets	1,000	--	100,000*		
Price per App	\$0 Sign Up	\$499 one time fee Sign Up	\$99 per month Sign Up	\$399 per month Sign Up	\$999 per month Sign Up

Fig. 1 Pricing of Vuforia SDK

This research presents image target recognition SDK based on cloud integrating android library and pattern recognition. In the pattern recognition part, two features descriptor are experimented to find the better feature descriptor for feature description for image matching purpose and a cheaper solution for developers. The main aim of this research is to presents the newly proposed SDK

which shows the integration among clients and admin in terms with the mobile application based on a cloud with a cheaper solution.

### B. Feature Descriptor

Image recognition involves three important components. The first component is detection in which keypoint detectors are used to detect the keypoints of an object. The second component in the process of image recognition is a description. A descriptor is required to describe the keypoint detected in the description process. Descriptors can be divided into two types, which is vector descriptor and binary descriptor. SIFT and SURF are the examples of vector descriptor while BRIEF, ORB, BRISK and FREAK are the examples of binary descriptors. The next component of the image recognition process is matching. The keypoints of reference objects should be stored in the database in advance to match the points between the detected object with the reference objects. Once the features' keypoints have been detected, the image patch must be described. Descriptors of the keypoints must be built to identify and match keypoints across images. The description must be distinctive for each keypoint, but also need to be consistent under all viewpoints. One of the most famous keypoint descriptors is SIFT (Scale Invariant Feature Transform) [13] which detects keypoints based on Difference of Gaussians (DoG). Although SIFT was published in 1999, it still yields competitive results to state-of-the-art techniques. Apart from SIFT, several SIFT-like descriptors have been published, which involve some modifications, for example, ASIFT [14] and PCA-SIFT [15]. SURF (Speeded-Up Robust Feature) almost preserves the quality of SIFT but accelerates the gradient computations using integral images. To date, SURF descriptor is considered as the most popular replacement for SIFT. SIFT and SURF have successfully demonstrated their good robustness and distinctiveness in a variety of computer vision applications [16], [15], [14] (Yang et al. 2011, Nister & Stewenius 2006). However, the processing time for vector descriptors is still too high for real-time applications, especially those which run on limited computing power and memory capacity. Hence, binary descriptors aim to fill in this gap. With the rapid growth of real-time applications, binary descriptors which aim primarily at fast runtime and compact storage have become increasingly well-known. They show similar performance as SIFT-like descriptors, but at significant lower computational costs. The idea of binary descriptors is that each bit in the descriptor is independent and the Hamming distance can be used as similarity measure instead of Euclidean distance [18]. The four most recent and promising binary feature descriptors are BRIEF (Binary Robust Independent Elementary Feature) [19], ORB (Oriented Fast and Rotated BRIEF) [20] and BRISK (Binary Robust Invariant Scalable Keypoints) [21].

Binary descriptor compute the descriptor by comparing the intensity with 256 bits and 512 bits respectively. Binary descriptor goes over all the bits (pairs) and compares the intensity value of the first point in the pair with the intensity value of the second point in the pair. If the first value is larger than then second value, "1" is dominated in the string, otherwise "0" as formula below:

$$T(p; x, y) := \begin{cases} 1 & \text{if } I(p, x) < I(p, y) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Where  $I(p, x)$  is pixel intensity at point  $x$  and  $I(p, y)$  is pixel intensity at point  $y$ . Each binary descriptor can form by combining each bit as:

$$\sum_{1 \leq i < j \leq n} 2^{i-1} B(p; x, y), \quad (2)$$

The proposed BRIEF and ORB descriptor do not have a specific sampling pattern. The authors suggest learning the sampling pairs. BRIEF descriptor is not invariant to rotation and scale. The author claim that ORB descriptor is invariant to rotation and robust to noise [20]. BRISK descriptor's sampling pattern is composed out of concentric rings as Fig. 2. The pairs are divided into long-distance pairs and short-distance pairs. Long-distance pairs are used to determine the orientation and short-distance pairs are used for the intensity comparisons. The author claim that BRISK descriptor uses low computation cost, rotation, and scale invariance.

### C. Research Methodology

This work proposes an image target recognition SDK based on a cloud with the main goal of lightweight implementation on mobile devices based on processing performed over the cloud. The overall research methodology is proposed for this work is based on the stages mentioned in Fig. 5. The main phases of the research methodology are the identification of the problems by analysing the literature and the trends in the industry, the design of the proposed image target recognition SDK, its evaluation and release. The contribution of this work can be applied to both; the scientific body of knowledge as well as to the industrial applications. The developmental methodology is divided into three main phases as shown in Fig. 6. At first Android, Library is developed in order to transfer data to the server. Based on the data matching results, the resultant matching data are sent to the user interface to perform various actions. These actions include Show Toast Message, Display an Image or 3D model, etc. In pattern recognition stage, various kinds of testing are performed in this stage in order to measure features in the image. Finally, a web portal is developed for the user and admin in the form of a dashboard to use the SDK.

### D. Android Library Development

Android library involves four aspects which are:

- Transfer data in real-time from phone
- Receive data to user interface
- No image match condition
- Match condition

This research used `HttpAsyncTask` class to transfer the data in real time from the phone. This class uses HTTP protocols to post the real time frame data on the server shown in Fig. 3. The results are received in JSON format and send to the user's UI to perform various actions.

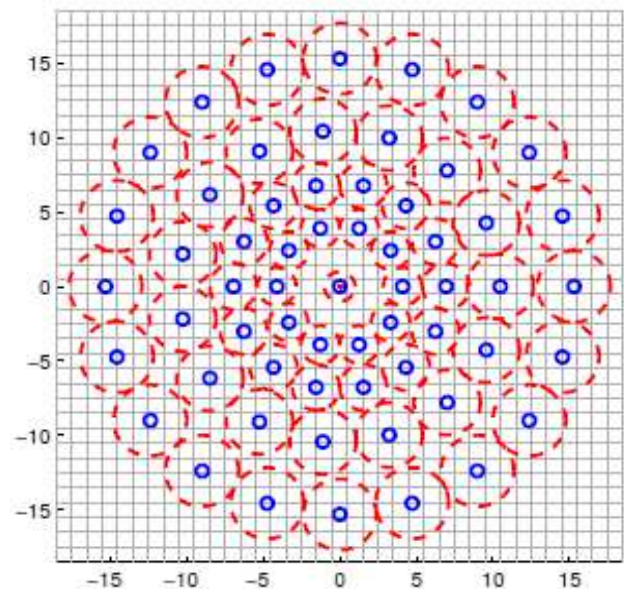


Fig. 2 Sampling pattern of BRISK descriptor

```
// 1. create HttpClient
HttpClient httpClient = new DefaultHttpClient();

// 2. make POST request to the given URL
HttpPost httpPost = new HttpPost(url);

String json = "";

// 3. build jsonObject
JSONObject jsonObject = new JSONObject();
jsonObject.accumulate("Image Data", newString());
/*
 * jsonObject.accumulate("country", newString+" Malaysia");
 */

// 4. convert JSONObject to JSON to String
json = jsonObject.toString();
```

Fig. 3 Example of HTTP protocol

A special class called `JsonResponse` is created which handles the results and sends back to the user interface shown in Fig. 4.

```
JSONObject jsonObjectResponse = (JSONObject) new JsonParser()
    .parse(result);

String ResponseStatus = jsonObjectResponse.get("Status").toString();
String ResponseMessage =
    jsonObjectResponse.get("Message").toString();
String ResponseFileName =
    jsonObjectResponse.get("FileName").toString();
String ResponseFileID =
    jsonObjectResponse.get("FileID").toString();
Log.e("Live Json Result", "Status: "+ ResponseStatus);
```

Fig. 4 JsonResponse class

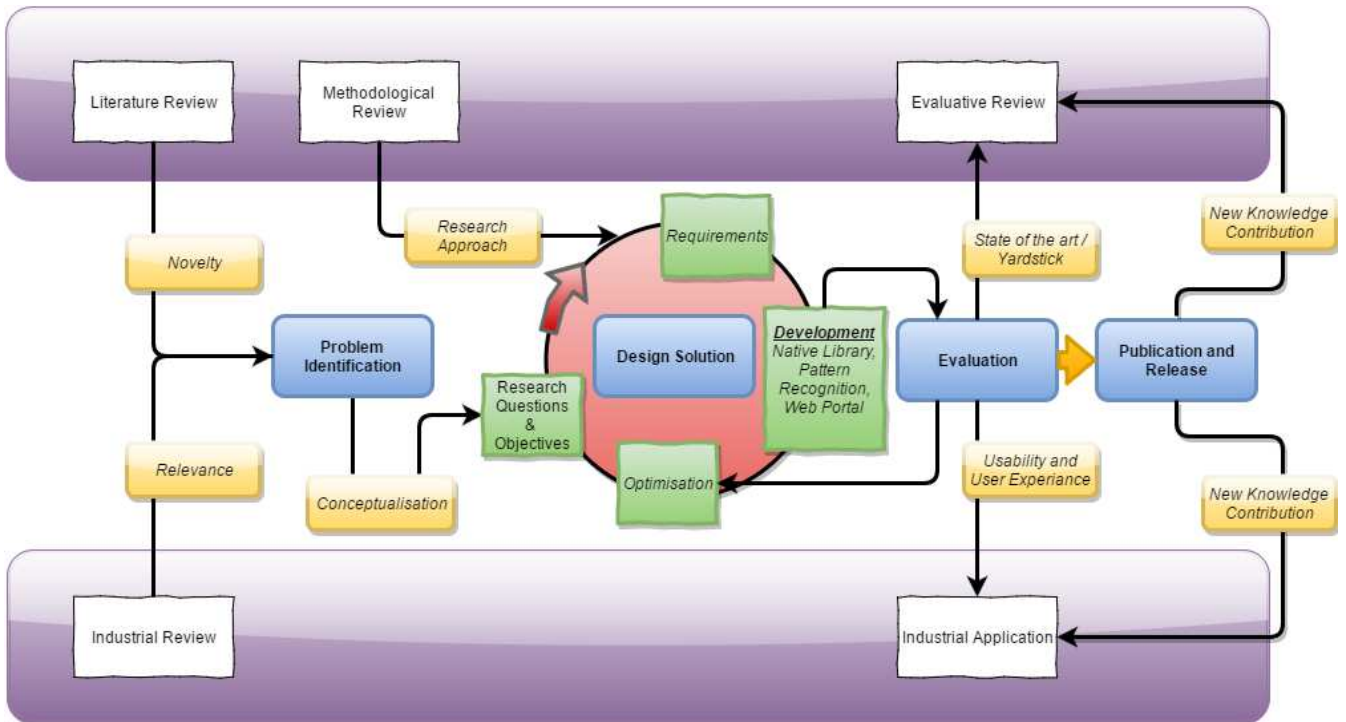


Fig. 5 Research Methodology for the proposed SDK

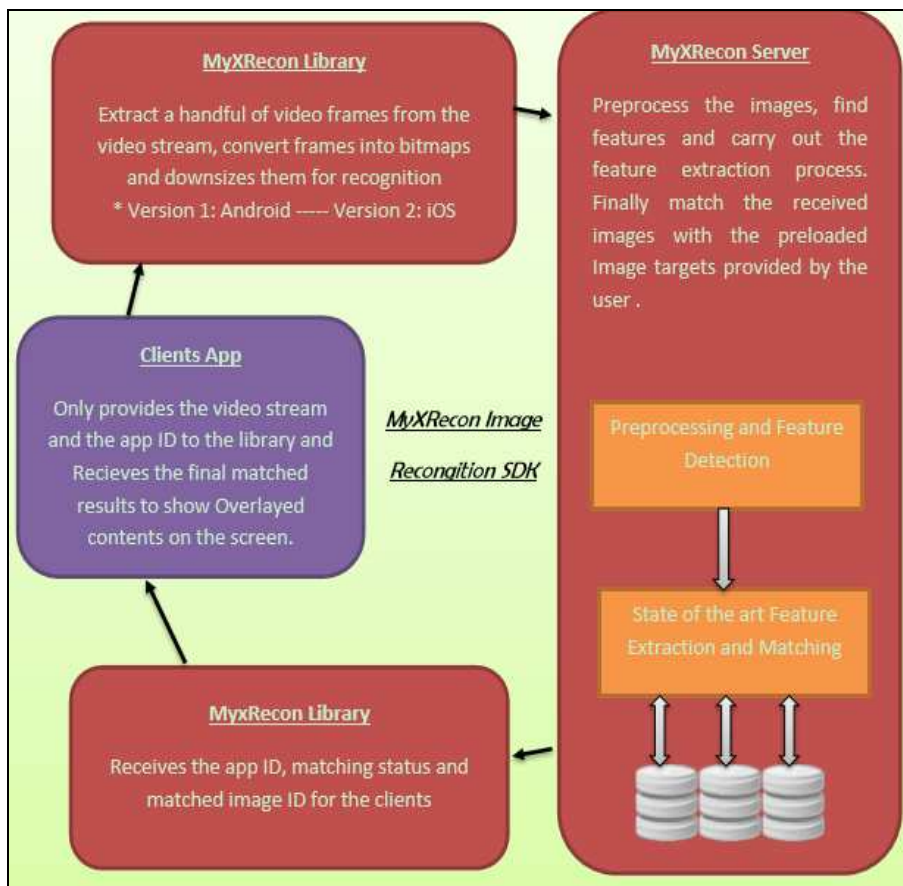


Fig. 4 Development flow for the proposed SDK

If no matching image found with the input image, then “text” column and “FileName” column shows empty or “” shown in Fig. 7.

Time	P	T	Application	Tag	Text
10-09 18:58:36.397	2	:	com.myxla...	Set Real Time Status	"Success"
10-09 18:58:36.398	2	:	com.myxla...	Live Json Result	Status: "Success"
10-09 18:58:36.398	2	:	com.myxla...	Realtime Result	Status: "Success" Message: "Not Found" FileName: ""
10-09 18:58:38.233	2	:	com.myxla...	Set Real Time Status	"Success"
10-09 18:58:38.233	2	:	com.myxla...	Live Json Result	Status: "Success"
10-09 18:58:38.233	2	:	com.myxla...	Realtime Result	Status: "Success" Message: "Not Found" FileName: ""
10-09 18:58:42.219	2	:	com.myxla...	Set Real Time Status	"Success"
10-09 18:58:42.219	2	:	com.myxla...	Live Json Result	Status: "Success"
10-09 18:58:42.219	2	:	com.myxla...	Realtime Result	Status: "Success" Message: "Not Found" FileName: ""
10-09 18:58:44.061	2	:	com.myxla...	Set Real Time Status	"Success"
10-09 18:58:44.062	2	:	com.myxla...	Live Json Result	Status: "Success"
10-09 18:58:44.062	2	:	com.myxla...	Realtime Result	Status: "Success" Message: "Not Found" FileName: ""
10-09 18:58:45.907	2	:	com.myxla...	Set Real Time Status	"Success"
10-09 18:58:45.908	2	:	com.myxla...	Live Json Result	Status: "Success"
10-09 18:58:45.908	2	:	com.myxla...	Realtime Result	Status: "Success" Message: "Not Found" FileName: ""
10-09 18:58:47.748	2	:	com.myxla...	Set Real Time Status	"Success"
10-09 18:58:47.748	2	:	com.myxla...	Live Json Result	Status: "Success"
10-09 18:58:47.748	2	:	com.myxla...	Realtime Result	Status: "Success" Message: "Not Found" FileName: ""

Fig. 7 Example of when no image is matched

If an Image (from the server) is matched to the video feed, “text” column shows - Message: “Found” and “FileName” column shows the specific image name as Fig. 8.

Time	Application	Tag	Text
10-09 18:59:14.484	com.myxla...	Live Json Result	Status: "Success"
10-09 18:59:14.484	com.myxla...	Realtime Result	Status: "Success" Message: "Not Found" FileName: ""
10-09 18:59:16.256	com.myxla...	Set Real Time Status	"Success"
10-09 18:59:16.256	com.myxla...	Live Json Result	Status: "Success"
10-09 18:59:16.256	com.myxla...	Realtime Result	Status: "Success" Message: "Found" FileName: "4alifMinyak.jpg"
10-09 18:59:17.548	com.myxla...	Set Real Time Status	"Success"
10-09 18:59:17.548	com.myxla...	Live Json Result	Status: "Success"
10-09 18:59:17.548	com.myxla...	Realtime Result	Status: "Success" Message: "Found" FileName: "4alifMinyak.jpg"
10-09 18:59:19.390	com.myxla...	Set Real Time Status	"Success"
10-09 18:59:19.390	com.myxla...	Live Json Result	Status: "Success"
10-09 18:59:19.390	com.myxla...	Realtime Result	Status: "Success" Message: "Not Found" FileName: ""
10-09 18:59:20.926	com.myxla...	Set Real Time Status	"Success"
10-09 18:59:20.926	com.myxla...	Live Json Result	Status: "Success"
10-09 18:59:20.926	com.myxla...	Realtime Result	Status: "Success" Message: "Found" FileName: "4alifMinyak.jpg"
10-09 18:59:22.769	com.myxla...	Set Real Time Status	"Success"
10-09 18:59:22.770	com.myxla...	Live Json Result	Status: "Success"
10-09 18:59:22.770	com.myxla...	Realtime Result	Status: "Success" Message: "Found" FileName: "4alifMinyak.jpg"
10-09 18:59:24.918	com.myxla...	Set Real Time Status	"Success"
10-09 18:59:24.918	com.myxla...	Live Json Result	Status: "Success"
10-09 18:59:24.918	com.myxla...	Realtime Result	Status: "Success" Message: "Found" FileName: "4alifMinyak.jpg"
10-09 18:59:26.486	com.myxla...	Set Real Time Status	"Success"
10-09 18:59:26.486	com.myxla...	Live Json Result	Status: "Success"
10-09 18:59:26.486	com.myxla...	Realtime Result	Status: "Success" Message: "Found" FileName: "4alifMinyak.jpg"

Fig. 8 Example of when an image is matched

### E. Pattern Recognition

Pattern recognition process needs to be carried out on both android and web portal platform. Feature descriptor is one of the most important components in the pattern recognition process. In this stage, analysis and evaluation between ORB and BRISK descriptor have been carried out to identify the better descriptor for feature selection purpose. BRISK descriptor provided better results compared to ORB because BRISK features are evenly distributed and higher in number. Hence, throughout the works, BRISK descriptor is used to extract the features. When users uploaded target image in the web portal, the target image is rated in the range from zero stars to five stars. The rating process is based on two criteria:

- Number of features in the target image
- Contrast of the target image

The higher the rating of an image target, the stronger is the tracking ability. A rating of zero shows that the target images will be difficult to match during the image recognition process. Hence, users are not recommended to use the zero-star and one-star target image for their system if they want to have accurate image recognition. During the real-time image recognition, the image from the video feed will also be extracted using BRISK descriptor and the feature’s data will send to the server for the matching process. If the video feed image’s features are matched with the server image’s features, then the “FileName” of the image will then send to the phone.

### F. Web Portal

Web portal (<http://myxscan.net/Account/Login>) is developed in order to enable the user and admin to use the SDK. A user is provided to log in initially after signing up into the system. Fig. 9 shows the log in the interface of the image recognition web application.

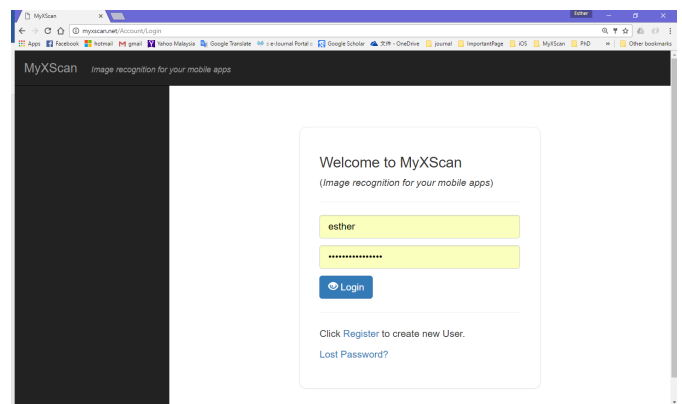


Fig. 9 Log in interface

The user would need to register in the web portal before they start purchase any plan. Fig. 10 shows the user registration interface.

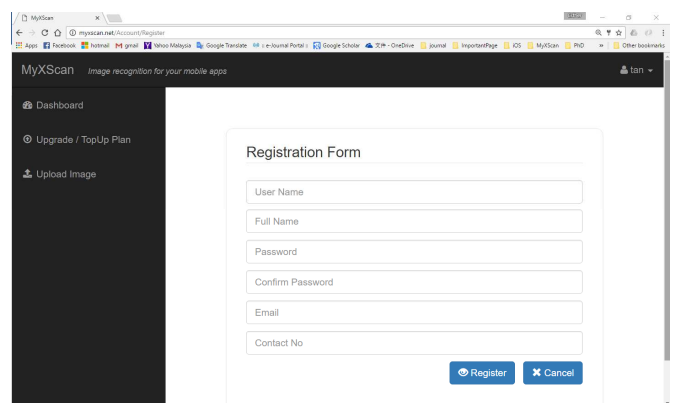


Fig. 10 User registration interface

Various user plans are created to provide scanning functionality based on the number of images and the number of scans for the clients. They are charged based on the number of months. For any new plan, an API key will be provided to the user who does not share it with other people. Generated API key will be required to be embedded inside their mobile application development project to perform the recognition task smoothly. However, to provide

the analytics, in the admin side on the web portal, some extra features are added in order to facilitate monitor each user subscribed plan, details of activities and usage, the number of scans, etc. The pricing for this SDK is shown in Fig. 11.

Plan Name	Number of Images	Number of Scan (/month)	Rate (/month)
FIRST PLAN	50	2500	RM 50.00
SECOND PLAN	200	5000	RM 80.00
THIRD PLAN	500	Unlimited	RM 150.00
FOURTH PLAN	Unlimited	Unlimited	RM 250.00

Select period: 3 months Total: 150

UPGRADE/BUY

Fig. 11 Pricing for proposed SDK

The user can check the information about the plan had bought, and the application had been developed with a unique API Key in Dashboard. Fig. 12 shows the dashboard interface.

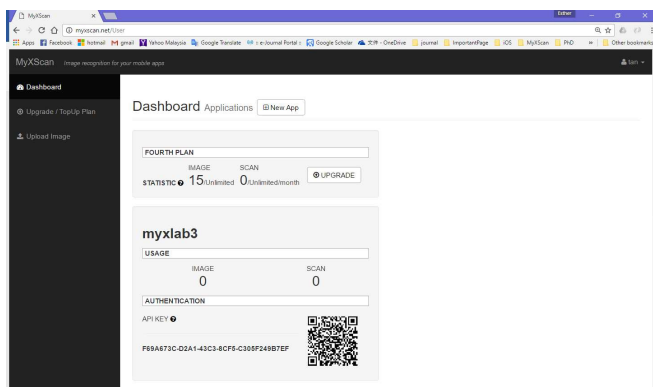


Fig. 12 Dashboard interface

### III. RESULT AND DISCUSSION

The result shows that the cloud-based image target recognition SDK had work efficiently for mobile application (android). The researcher had tested the SDK by uploading 30 images in the web portal (<http://myxscan.net/User/Upload>). Fig.13 shows the 12 images that had been uploaded to the web portal. There are total 16 images with a rating of five stars, 6 images with four stars, 4 images with three stars, 1 image with one star and 3 images with zero star. All the images had been tested by scanning the query images to retrieve the information from the server. Once the query image is matched with the image from the server, a message will send to the mobile application. Fig. 14 shows the client side had successfully retrieved the message from the server and the information of the image are shown on the screen. The response from the server to the client side is very efficient for all the images except for images which have only one star and zero star in the rating system. Images with one star and zero star are failed to retrieve the information from the server because

they have the difficulty to match with the images from the server. Hence, users are not recommended to use any zero-star or one-star images in the system. Users can check the rating of the images in the system and decide whether to remove or maintain the images for the further recognition process.

File ID	File Name	Star
e47dc6e2-1702-4cd5-b7b6-4f680a5a7ae2	1453170650328-885695...	★★★★★
fdbf7fd0-97c5-42de-8e28-dd40df869c11	1453171284532-110486...	★★★★★
585f90ad-d4af-419a-a21f-990154033115	14531713222761792389...	★★★★★
ee9dbe53-9c83-4272-8ec2-b90ba06c2351	16adabiPerencah.jpg	☆☆☆☆☆
560a275a-b028-4854-ab48-2f9a4406c611	21adabiSup.jpg	★★★★★
017f0c04-7187-430a-ba71-4616b41149f6	22adabiPisang.jpg	☆☆☆☆☆
cb8c0703-9410-42c7-836d-1f0297e0e7b2	24heins.jpg	★★★★★
b3febcbf-cde4-4fdf-9c24-ce1d77c0d52f	24heins.jpg	★★★★★
6f5f0a60-4920-4296-b960-e92ccb9c5b12	26maggiSouce.jpg	☆☆☆☆☆
9409ee6f-2d58-4c1c-916b-3496b3278d98	27maggiArab.jpg	★★★★★

Fig. 13 Sample images uploaded in the web portal

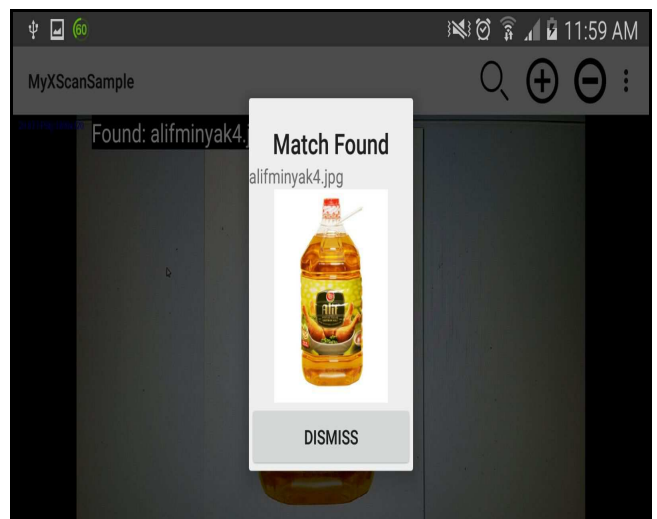


Fig. 14 Successful matching scenarios (image and image name displayed)

Users can view the report on the spot to check on the number of images had been uploaded to the system, and the number of scans had been done by the mobile users as Fig. 15. The report is successfully generated in real time. There are total 30 images been uploaded to the server, and there are total 42 scans been done by the mobile users. Once the mobile users successfully scan and retrieve the information from the server, the number of scans that shows in the report will update immediately.

NAME	API KEY	# IMAGES	# SCAN
MyFirstApp	a0343486-6a9e-4024-84e0-2ff1ef8f47f2	30	42

Row count: 50 Showing 1-1 of 1

Fig. 15 Report for each mobile application.

#### IV. CONCLUSION

A cloud-based image recognition SDK is developed to scan the product using a smartphone. This work shows the detailed research methodology and the development flow of the proposed SDK. Based on the study of the current state of the art image recognition algorithms, this research demonstrates an efficient and robust image recognition services which is capable of detecting noise free, partially blurred, dark and occluded images with lower price compared to Vuforia SDK. The contribution of this work can be applied to both; the scientific body of knowledge in the form of improved algorithms and research framework as well as towards the development of industrial applications in various domains. The improvement of the proposed SDK will be further demonstrated and validated using execution time for matching, robustness, used memory space and user statistics compared with state of the art image target applications and SDK's in the future work.

#### ACKNOWLEDGMENT

This research is supported by PPRN-2015-008.

#### REFERENCES

- [1] V. Veronice, Y. Yelfiarita, and D. Darnetti, "Analysis of Characteristics Extension Workers to Utilization of Information and Communication Technology," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 5, pp. 303-305, 2015.
- [2] V. Chang, *Delivery and Adoption of Cloud Computing Services in Contemporary Organizations*: IGI Global, 2015.
- [3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, et al., "Above the clouds: A berkeley view of cloud computing," *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS 28, 2009 (2009)*, 2009.
- [4] A. Othman, T. S. M. T. Wook, and S. M. Arif, "Quantization Selection of Colour Histogram Bins to Categorize the Colour Appearance of Landscape Paintings for Image Retrieval," *International Journal on Advanced Science, Engineering and Information Technology*, vol. 6, 2016.
- [5] E. Baccarelli, N. Cordeschi, A. Mei, M. Panella, M. Shojafar, and J. Stefa, "Energy-efficient dynamic traffic offloading and reconfiguration of networked data centers for big data stream mobile computing: review, challenges, and a case study," *IEEE Network*, vol. 30, pp. 54-61, 2016.
- [6] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless communications and mobile computing*, vol. 13, pp. 1587-1611, 2013.
- [7] L. Kovács, "Shape retrieval and recognition on mobile devices," in *International Workshop on Computational Intelligence for Multimedia Understanding*, 2011, pp. 90-101.
- [8] S. Umair, U. Muneer, M. N. Zahoor, and A. W. Malik, "Mobile Cloud Computing Future Trends and Opportunities," *Managing and Processing Big Data in Cloud Computing*, p. 105, 2016.
- [9] W. Zhao, Y. Sun, and L. Dai, "Improving computer basis teaching through mobile communication and cloud computing technology," in *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, 2010, pp. V1-452-V1-454.
- [10] C. Yin, B. David, and R. Chalon, "Use your mobile computing devices to learn-Contextual mobile learning system design and case studies," in *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on*, 2009, pp. 440-444.
- [11] E. Vartiainen and K. Väänänen-Vainio-Mattila, "User experience of mobile photo sharing in the cloud," in *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, 2010, p. 4.
- [12] H. Flores, S. Narayana Srirama, and C. Paniagua, "Towards mobile cloud applications: Offloading resource-intensive tasks to hybrid clouds," *International Journal of Pervasive Computing and Communications*, vol. 8, pp. 344-367, 2012.
- [13] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, 1999, pp. 1150-1157.
- [14] G. Yu and J.-M. Morel, "ASIFT: an algorithm for fully affine invariant comparison," *Image Processing On Line*, vol. 2011, 2011.
- [15] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, 2004, pp. II-506-II-513 Vol. 2.
- [16] X. Yang, Q. Liu, C. Liao, K.-T. Cheng, and A. Girgensohn, "Large-scale EMM identification based on geometry-constrained visual word correspondence voting," in *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, 2011, p. 35.
- [17] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, 2006, pp. 2161-2168.
- [18] D. Bekele, M. Teutsch, and T. Schuchert, "Evaluation of Binary Keypoint Descriptors," in *20th IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 3652-3656.
- [19] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *ECCV'10 Proceedings of the 11th European Conference on Computer Vision: Part IV*, 2010, pp. 778-792.
- [20] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2564-2571.
- [21] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2548-2555.