ON MOBILE-FRIENDLY APPLICATION FOR SCENE SEGMENTATION

Sichen Wan | MCI Project | Group 17
ECMS, University of Adelaide, South Australia
10th June 2018

"Scene segmentation refers to a group of pixels which have been detected as same object will be masked with different color."

By Ian Reid

ABSTRACT

Our project for Master of Computing & Innovation aims to provide a mobile-friendly application for scene segmentation in Android platform. The application aims to achieve 3 functionalities as planned within 2 milestones. Image segmentation functionality has been featured by tapping interaction whilst the video segmentation functionality in NRT[i] way has been successfully developed in a concise GUI[ii] as planned.

The application will be improved to be more powerful in the future. The application can support both portrait orientation and landscape orientation not only in Android platform but also in IOS platform. In addition, better live streaming video segmentation need to be developed for smooth segmented video in real-time fashion. Voice feedback will be integrated into the application for the visual disable.

TABLE OF CONTENTS

1. INTRODUCTION

The vision of this project is to design and develop an Android mobile-friendly application for scene segmentation. Final deliverable with 3 core functionalities has been achieved as planned.

In a High-Tech Era, technology of computer vision has been developed rapidly during recent decades while the main working platform for computer vision researchers is still the clumsy server. But, obviously, it is not convenient for them to test the functions visually in such working platform.

In order to change this circumstance, a sound solution need to be proposed and provided. We reckon that mobile is more desirable over server because of its portability and accessibility whilst the server has much higher computation power over mobile. Considering merit from both server and mobile, our solution has been proposed. The solution is to develop a mobile-friendly Android application for scene segmentation. This is the main reason why our project aims to deliver a mobile-friendly application for scene segmentation for scene segmentation with the strong support of our university deep learning server.

The current main user is targeted as computer vision researchers. With the help of our application, computer vision researchers can easily test their work like object classification and localization by checking the results of different scene segmentation.

In this final report, our project will be illustrated in 4 aspects. In section 2, 4 main goals with corresponding objectives will be specified in two iterations of milestones. Process, employed tools and system architecture will be shown during design and development in section 3. In section 4, actual deliverable we have achieved will be revealed while the testing report will be shown. In the last section, achieved

work will be summarised before conducting a project reflection. Future work for extension will be proposed for better application in prospect.

2. PROJECT AIMS

As mentioned in Introduction section, the vision of this project is to design and develop an Android mobile-friendly application for scene segmentation. At the beginning, our supervisor, Professor Ian, recommended a scene segmentation website[iii] as reference for us to study.

Figure 001

As shown in Figure 001, the image segmentation functionality and tapping interaction have been achieved in website. Thanks to this guidance, we know well the cornerstone is to replicate the functionalities from this website into Android mobile. Accordingly, we had set up 3 main goals and 1 extra goal towards final deliverable. Each goal with corresponding objectives will be illustrated individually as follow:

2.1 Image Segmentation

Goal 1. IMAGE SEGMENTATION	
Objective No.1	**Application can upload image to the server.**
Objective No.2	**Application can send image segmentation and get server response.**
Objective No.3	**Application can get segmented image shown on image view area.**

First goal is to achieve the image segmentation functionality. This goal, the main goal in milestone 1, is one task for functionality replication. Users can user the application to upload image taken or stored in media to the server at the first step. Then application can send image segmentation request to server for processing the uploaded image. Then server response with image mask and object's property (including name and color) back to the application. Finally, the application can show the segmented image on image view area by overlapping the original image and image mask.

2.2 Tapping Interaction

Goal 2. TAPPING INTERACTION	
Objective No.1	List of objects' name will be shown.
Objective No.2	Object will be shown brighter after users click tag button labelled by object's name.
Objective No.3	Object will be shown brighter after users click the region of the object.

Second goal is to develop the tapping interaction functionality. In order to make the application in mobile-friendly fashion, tapping interaction has been planned as second goal. This goal, one main goal in milestone 2, is another task for functionality replication. We aim to develop tapping interaction for users to enjoy the application. To be more specific, list of name of all different objects can be shown once image segmentation process succeeds. Then users can get more information of specific object in 2 ways. One way is that users can click the tag button labelled by object's name to allow the specific object show brighter. Another way is that users can click the region of the object to allow the specific object show brighter.

2.3 Video Segmentation

Goal 3. VIDEO SEGMENTATION	
Objective No.1	Application can create customised camera.
Objective No.2	Camera automatically takes photos.
Objective No.3	Application displays the segmented video in a NRT fashion.

Third goal is to achieve the live-streaming video segmentation functionality. This goal, another main goal in milestone 2, is a new challenging task for advanced functionality. As shown above, the application can create customized camera shown as preview on image

view area at the first step. Then camera can automatically take photos in continual way. At the meanwhile, the application can display the segmented frame of video in a NRT fashion.

2.4 System Optimization

Goal 4. SYSTEM OPTIMIZATION	
Objective No.1	**Application can compress the size of image for faster segmentation.**
Objective No.2	**Application can make a new Internet thread to build up a robust Internet connection.**

Last goal is to optimize the application performance. As shown above, one objective is that the application can compress the size of image to speed up the segmentation. The other objective is that a new Internet thread can be made to process Internet connection in a robust way.

3. APPROACHES

3.1 Development Process

We have been using the Extreme programming(XP), one classic Agile software process model in this project. Since this project lasts for 12 weeks with weekly client meetings, it is desirable for us to employ XP software development model to develop our application.

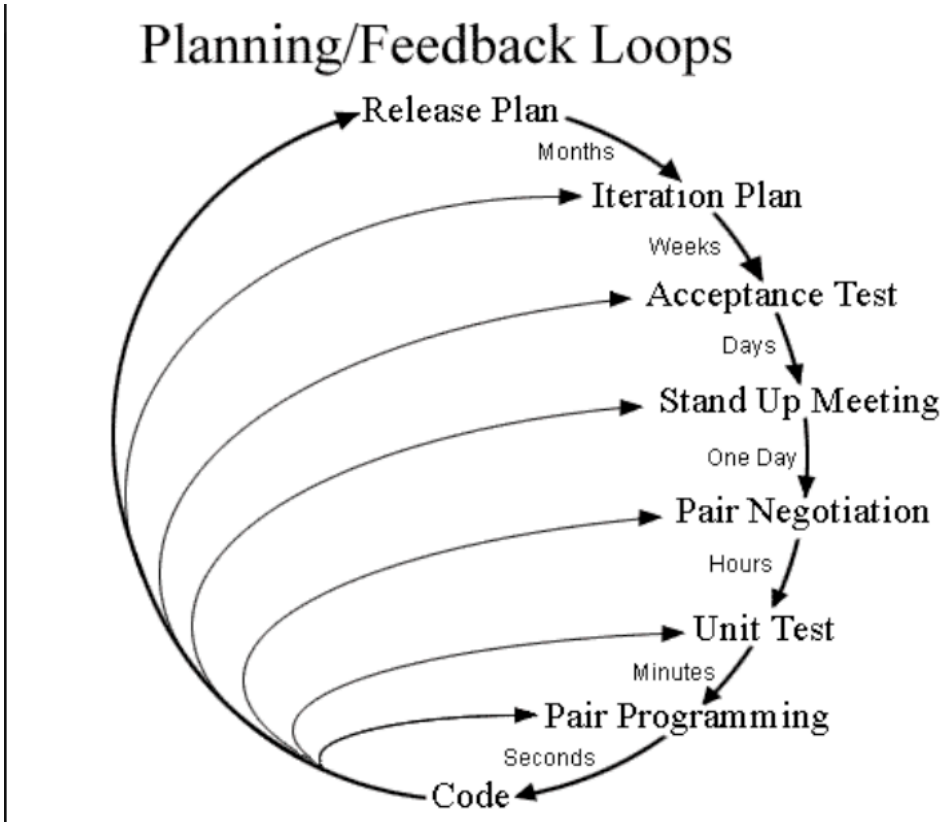

Figure 002

(Source: https://www.wikiwand.com/en/Extreme_programming)

Throughout the whole project, we have been using XP model in different aspects. Thanks to the clear guidance on client requirements in the beginning of semester, our priority is on how to develop high-quality application within fixed time. As shown in Figure 002, from the release plan to code, each task requires cycle of different period.

We have gotten our project as released plan. This plan has lasted for 4 months as fixed life cycle of project. Within this duration, several iteration plans, including pitch, business case, milestone 1 plan, milestone 2 plan, testing plan and final deliverable demonstration, had been carried out one after another every few weeks. In the pitch, we enriched experience on venture pitch for

the product and understandings on motivation, goals, development model, tools and target users in this project. In the business case and draft plan, we grasped requirement of the first iteration milestone, the core functionality. During the development, the acceptance tests had been conducted by us, who act as users in testing part, to verify the functionality we just developed. After frequent testings on new-developed functionality, in the weekly client meeting, we justified the actual result against the planned goal while we clarified the new goal and objectives of each iteration plan by discussing with our client Professor Ian. Since we employed the test-driven development, we had programmed in pair-coding fashion to develop the application step by step. Once some issues were detected as fatal, I had negotiated with teammate for bugs or clashes before we made a great effort to fix them in pair-coding style with frequent unit testings. Each step within the feedback loops is correlated to each other without interruption. Taking the clashes led by different storage paths in Android mobile as example, we had spotted this issue and discussed with our supervisor on this issue in meeting. However, there was no better solution to solve the clashes when users want to access the storage other than built-in media library gallery, like Google Photos or Recently Opened Photos. This issue had perplexed us days until we found a better way to allow users to access chosen photo source. Since we had developed our application in XP, the main functionality can be delivered at first before other secondary issue like this instance can be solved.

This software development model has benefited our project in 3 dimensions:

- Time-saving & Cost-saving: Since XP focuses on the timely delivery of final product, every hour has been accounted for realization of each milestone in the project. Besides, team

saves money because of tidy documentation and effective discussion.

- Constant Feedback: During the project, we had constantly gotten feedback from our supervisor and tutor. In this way, changes on requirement can be modified on time, which results in successful final deliverable.
- Fast Deliverable: Thanks to the regular testings during development stage, functionalities of the application had been developed faster.

3.2 Tools & Languages

- **Java & Android Studio:**
  Since the project aims to develop an Android mobile application, it is manifest that Java is the best option for us to develop such kind of application. The application had been developed purely in Java.
  As the main tool, the integrated development environment (IDE) is of great importance. Since I had software engineering project (SEP) which had been developed in Eclipse in the last semester, we chose to develop the application in Eclipse at the beginning. However, we decided to choose Android Studio instead of Eclipse to develop both functionalities and graphic user interface. The reasons for this choice have been listed as below:

	Android Studio	Eclipse
Gradle Integration	Automatically use Gradle in Quick and integrated manner. No further manual configuration to fit with Gradle.	Mainly use Apache Ant instead of Gradle. It will take times to fit with Gradle if using Gradle.

Code Completion	Auto code completion works excellently with sufficient reminder, which helps programmers to speed up coding.	Auto code completion works. But some auto completed parameters might be too misleading to perplex programmers.
User Interface	Very handly with Drag and Drop functions especially for GUI design.	Too huge to overwhelm programmers. No Drag and Drop functions.
Organization of Project	Work well on multiple projects. In our project, we have created some projects in IDE for different sub-functionality.	Reopen need be done if programmers work on multiple projects, which leads to inconvenience on project organization.
System Stability	Relatively lightweight IDE, which requires lower amount of RAM, no higher CPU. Newly released versions of IDE reduce the bugs into low level.	Heavyweight IDE, which requires higher amount of RAM, higher CPU speed. No many new updated versions of IDE, crashes or unresponsiveness occur occasionally.

- **Github:**

We used Github, the most popular version control repository tool, to manage our code and documentation for our project. As shown in figure 003, project had been organized and managed in order.

Besides using branch for the project, we also used Wiki as shared working platform for documentation and code. We used Project as schedule timetable for us to mark done tasks, monitor the tasks on process and set up new tasks for next goal. We used Issues to report the issues we had been confronted with.

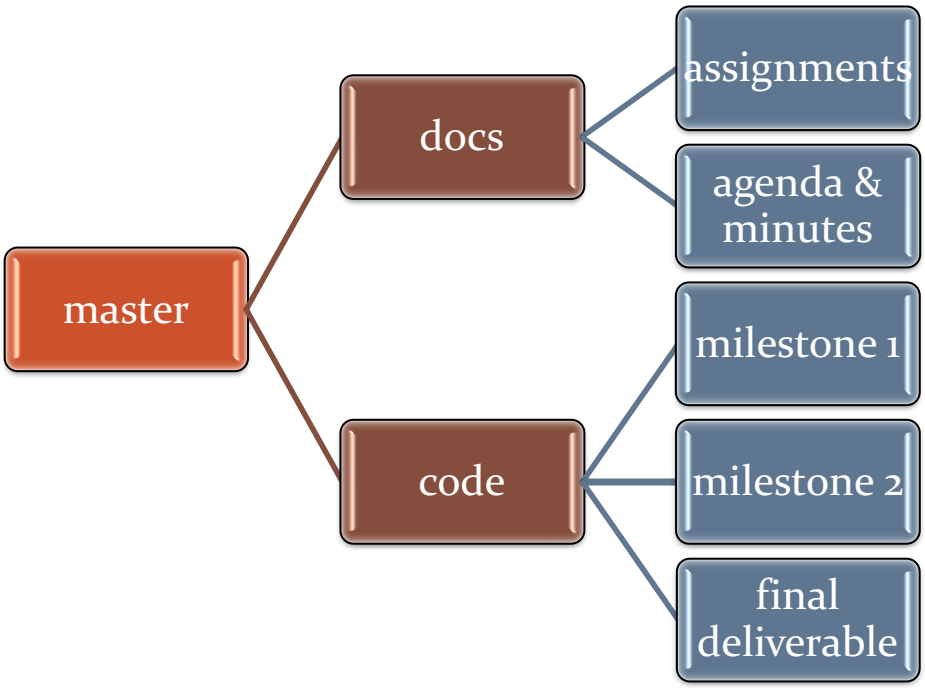

Figure 003

- **Skype:**

Since our supervisor, Professor Ian, travelled to England for work, we need to use Skype to have client meetings with him. Skype is great tool for us to have video communication with the important client. Since it

is very likely for us to meet such situation in the workplace when the client is not in local, we need to use Skype to keep communicating with client in order to get constant feedback on products.

3.3 System Architecture

Figure 004

Before system architecture description, we want to give our sincere gratitude to the Computer Vision team of our university. Thanks to the deep learning server provided by our university, we only need to design and develop an Android application which can provide scene segmentation services for users.

As shown in Figure 004, we used 3 important APIs, Namely, Scene Segmentation API who acts as fundamental method, Okhttp3 API who acts as command messenger and Android Camera API who acts as image capturer.

- **Client-Server Model:**

The system architecture for the application is simple since we employed one of classic models – Client-Server model. As seen in the figure, the Android App acts as client while deep learning Server acts as Server.

- **How Image Segmentation Works:**

The application invokes Okhttp3 API to talk with the server. This API is built as the main communication channel. It means the application sends scene segmentation request to command server to process the uploaded image. And then server invokes Scene Segmentation API to process the uploaded image and get the image mask ready in hand. After successful segmentation process, server sends the response with the image mask and the objects' name and color back to the application. In the end, the application generates the segmented image by overlapping the original image and the image mask.

- **How Video Segmentation Works:**

The application invokes Android Camera API to create customized camera. To be more specific, the application detects and accesses camera at first. Then the application creates a preview class for users to preview what images camera has captured in live-streaming way. Then the application builds a preview layout which is designed to work with the preview class and user interface. And then the application creates listeners for capture. In this case, a recursive method was developed in order to allow camera to take image continually and automatically until users stop the video segmentation. Finally, the application releases the camera for use by other application.

4. RESULTS

4.1 Deliverables

As for the actual result, all goals with corresponding objectives have been successfully achieved as planned.

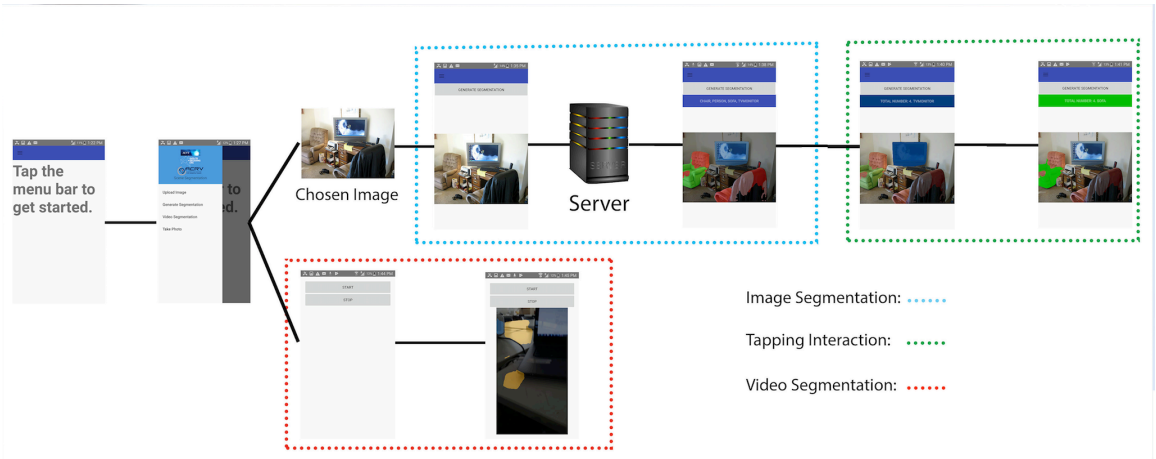

Figure 005

Users can use the application in steps shown in Figure 005. As seen from the left to the right, users can see the simple welcome page after launching the application. Then users can click the menu bar which is located in the left top of the interface. Then users can choose either image segmentation or video segmentation while either image from gallery or image taken now can be used.

- **Image Segmentation:**

All works have been shown within the blue dash line, users can upload image or take photo to provide target image to application. Then users can successfully get the segmented image by clicking "GENERATE SEGMENTATION" button.

- **Tapping Interaction:**

All works have been shown within the green dash line, users can see tapped object becomes brighter than other region when they tap detected object or the tag labelled with the detected object's name.

- **Video Segmentation:**

All works have been shown within the red dash line, users can see the live streaming segmented video continually once users click "START"

button. The segmented video can be successfully displayed on the application until users click “STOP” button.

- **System Optimization:**

The specific method to compress the image size has been found and developed in the application. With the help of this method, the speed for image segmentation has been improved. Besides, Internet connection between client and server has been supported by a new Internet thread in order to assure the process of segmentation.

4.2 Tests & Issues

Developed in test-driven development mode, the application has been successfully pass all required tasks show below:

No.	Test Goal	Test Case	Expected Result	Pass/Fai
001	Get the local photo for scene segmentation.	Click “UPLOAD IMAGE” button.	Access to the photo gallery.	Pass
002	Upload photo and Request scene segmentation.	Click “GENERATE SEGMENTATION” button.	Get the segmented image shown in the application.	Pass
003	Download the segmented image to local.	Long press the image.	Segmented image can be saved.	Pass
004	Provide better user interaction and experience.	Tap some place in segmented image or Click the tag button.	Tag button labelled with the name of tapped object will be shown under the image view area.	Pass
005			All objects which are the same as the tapped object will appear while other region in image will disappear.	Pass
006	Provide an advanced scene segmentation functionality for the	Click “VIDEO SEGMENTATION” and Move the camera during video	Built-in camera will be activated.	Pass

007	live thing will can be captured by camera.	segmentation mode. Click the "STOP" button.	The segmented images of original images automatically taken by camera will be continually shown in the image view area.	Pass
008			The application will stop the video segmentation and back to the main menu which contain image segmentation and video segmentation.	Pass

We also met several critical issues during the development. For instance, crashes led by different storage paths in mobile, bugs led by incompatibility between different Android versions and method on customized camera. Among them, customized camera is the most tough task for us. We had spent days to achieve this part since we didn't know how to capture continual pictures automatically. Finally, we developed a recursive method to solve this critical issue.

5. CONCLUSION

5.1 Achieved Work

The project on Android mobile-friendly for Scene Segmentation has been successfully finished with all requirements met. 3 main goals and 1 extra goal have been achieved with acceptable performance of functionalities. All documentations, including pitch, business case, milestones' plans and reports, testing plan and report, agenda and minutes, have successfully done with the pace of development. The code and documentation have been successfully finished in XP software development model. We sincerely acknowledge the guidance and help provided by Professor Ian, Philip, Yasir.

5.2 Future Work

If extension for further development in this project is granted, 4 works will be focused in order to improve the application:

- **Screen Orientation:**

Current version of the application can only show the image properly in portrait orientation. The improvement need to be made once the application can show the image in landscape orientation.

- **Real-Time Video Segmentation:**

Current version of the application can only show the slow video segmentation due to uncontrollable Internet latency and static image segmentation provided by the server. A better live-streaming video segmentation need to be developed to make smoothly segmented video with HFR[iv].

- **Cross-Platform Application:**

Current version of the application is only available in Android platform because of its open source and developer-friendly. The application need to be available in other operating systems like IOS as well in order to be exposed to more users.

- **Voice Feedback:**

Current version of the application only supports plain text information for users as feedback. The application need to develop a voice feedback, which means that the application can read the objects' name and property aloud in front of users. This functionality can be used for the visual disable in the future.

[i] NRT: Abbreviation for nearly real-time.
[ii] GUI: Abbreviation for Graphic User Interface.
[iii] demo.cs.adelaide.edu.au
[iv] HFR: Abbreviation for High Frame Rate.