Annotating Satellite Images for Machine Learning
Business Case and Draft Plan for Team 10

1 Business Case

1.1 Executive Summary

This project involves modifying and extending the functions of Visual Geometry Group Image Annotator (VIA) and ultimately reform VIA into a tool for annotating images on a per-pixel basis. Our project shall also have two advanced functions e.g. overlaying display and scribble mode.

1.2 Project Overview & Scope

Our project is a web-based tool for generating ground truth data which is used for training machine learning models. We start with annotating satellite images then explore the possibility for video capacity and medical images. In principle, one pixel can only belongs to one instance class. There is a special need for identifying occlusion e.g. road inside and outside shadows. Our project shall allow stacking different label classes on top of input images so that users shall easily identify the overlapping regions (roads in shadow/out of shadow are distinguished by different colors). Our project shall add advanced tool like brush into the toolkit for certain fine-grained annotation problems such as road networks, retina images, etc.

1.3 Market Positioning

Computer vision is one popular research area at present. A simple market survey reveals that there are already many annotators on the market with their own pros and cons. In conclusion, besides unable to generate per-pixel label, VIA meets all basic requirements of our project and adding customized functions to VIA is a more feasible strategy to us as we only got one semester for development and we are lack of front end developing experience. The survey result is listed below.

Name	Advantages	Disadvantages
Deep Extreme Cut (1)	Precise segmentation. Great for image matting	End users are CV experts.
Polygon RNN (2)	Accurately annotate high-resolution objects in images. Interactively annotate using humans-in-the-loop.	End users are CV experts.
Labelme (3)	Easy to use. Relatively light weight	Require installation and registration.
VIA (4)	Extreme portability (less than 200 KB). Low requirement (browsers only). Handy Polygon tools.	Unable to generate per-pixel label

Table 1: Summary of Product Survey

1.4 Business Objective

Our project adapts VIA as the prototype for handling web interaction and polygon tools. VIA is written in HTML, CSS and javascript only. This is the reason why it is extremely portable but also the reason that modifying VIA is a challenge for us. The core goals of our project is generating per-pixel labels and stacking different class labels on top of each other. Beyond that, our tool shall add scribble mode e.g. a brush tool with variable sizes into the toolkit for accelerating the annotating process. This function allows users to swipe through a region which is convenient for identifying road topology. Furthermore, polygonal tools are good for artificial structures, but they are not suitable for organic shapes such as human hearts which are often irregular. A brush with variable sizes is the solution for annotating medical images.

1.5 Benefits and Limitations

1.5.1 Benefits

1. Our tool shall be extremely portable and able to work on most platforms with modern browsers.

2. Web-based only is convenient for collaboration. Experts like ultrasonic clinicians are often busy, not necessarily computer savvy and not in one location. Our tool enables them to annotate images collaboratively when convenient.

3. Overlaying display can be used for identifying occlusion.

4. Scribble mode enables our project for annotating medical images and road topology.

1.5.2 Limitations

1. Due to lack of powerful libraries support, web-based only is an obstacle for developing advanced functions. Importing libraries may also impact our project's portability. There is a chance that we need to rewrite the configuration shells.

2. HTML5 is the latest web protocol but it still has some certain technical restrictions. For instance, it only supports videos in MP4, OGG and WebM formats. This drawback poses a challenge to exploring the capacity of annotating videos. Our function for video annotation cannot be as sophisticated as Video Annotation Tool from Irvine, California (VATIC) which specializes in annotating videos. One promising approach is to treat the video as an image sequence so we can apply our image annotation tool to user-selected frames in the video (Video capacity is for Milestone 2).

3. When saving users' annotations from the canvas as an image, HTML only allows the image be in PNG24/8 format which is not ideal for machine learning.

2 Draft Plan

2.1 Tasks Breakdown & Estimations

I list all the activities that I have done and will do in the chart below and create a Gantt Chart accordingly. For milestone 1, ▨▨▨▨▨▨▨▨▨ in charge of developing generating per-pixel labels, overlaying display and scribble mode. ▨▨▨▨▨▨ is in charge of compositing different label classes into one bitmap. For clarification, his function is independent from the VIA's source code and only accepts outputs of my part as inputs. Each of three modules can be divided into several independent sub-modules. I shall conquer each one of them by the end of week 6.

TASK NAME	START DATE	DAY OF MONTH*	END DATE	DURATION* (WORK DAYS)	DAYS COMPLETE*	DAYS REMAINING*	TEAM MEMBER	PERCENT COMPLETE
Background Research								
Create and Modify Pitch Slides	3/16	16	3/26	11	11	0	C.H Wang	100%
Market Survey	3/9	9	3/14	6	6	0	C.H Wang	100%
Design Team Logo	3/10	10	3/13	4	4	0	C.H Wang	100%
Video Capacity Study	3/16	16	3/22	7	7	0	C.H Wang	100%
Generating Per-Pixel Labels								
Acquire Polygon Regions Info	3/19	19	3/21	3	3	0	C.H Wang	100%
Render Region Info on the Second Canvas	3/23	23	3/27	5	5	0	C.H Wang	100%
Save and Adjust Output for Overlaying	3/23	23	4/3	12	7.2	4.8	C.H Wang	60%
Allow Users Naming Output	3/23	23	3/24	2	2	0	C.H Wang	100%
Overlaying the Results with the Input Image								
Open an Area for Overlaying	3/25	25	3/26	2	2	0	C.H Wang	100%
Upload Input and Masks respectively	3/25	25	4/4	11	8.8	2.2	C.H Wang	80%
Single Layer on Top of Input	3/25	25	4/4	11	6.6	4.4	C.H Wang	60%
Multi-Layer on Top of Input	3/31	31	4/10	11	2.2	8.8	C.H Wang	20%
Save Compositing Result	3/25	25	4/10	17	6.8	10.2	C.H Wang	40%
Developing Scribble Mode								
Brush Tool with Variable Sizes	3/31	31	4/10	11	3.3	7.7	C.H Wang	30%
Add an Eraser	3/31	31	4/10	11	0	11	C.H Wang	0%
Add Scribble Output into Overlaying	3/31	31	4/10	11	0	11	C.H Wang	0%

Table 2: Tasks Breakdown for Milestone 1
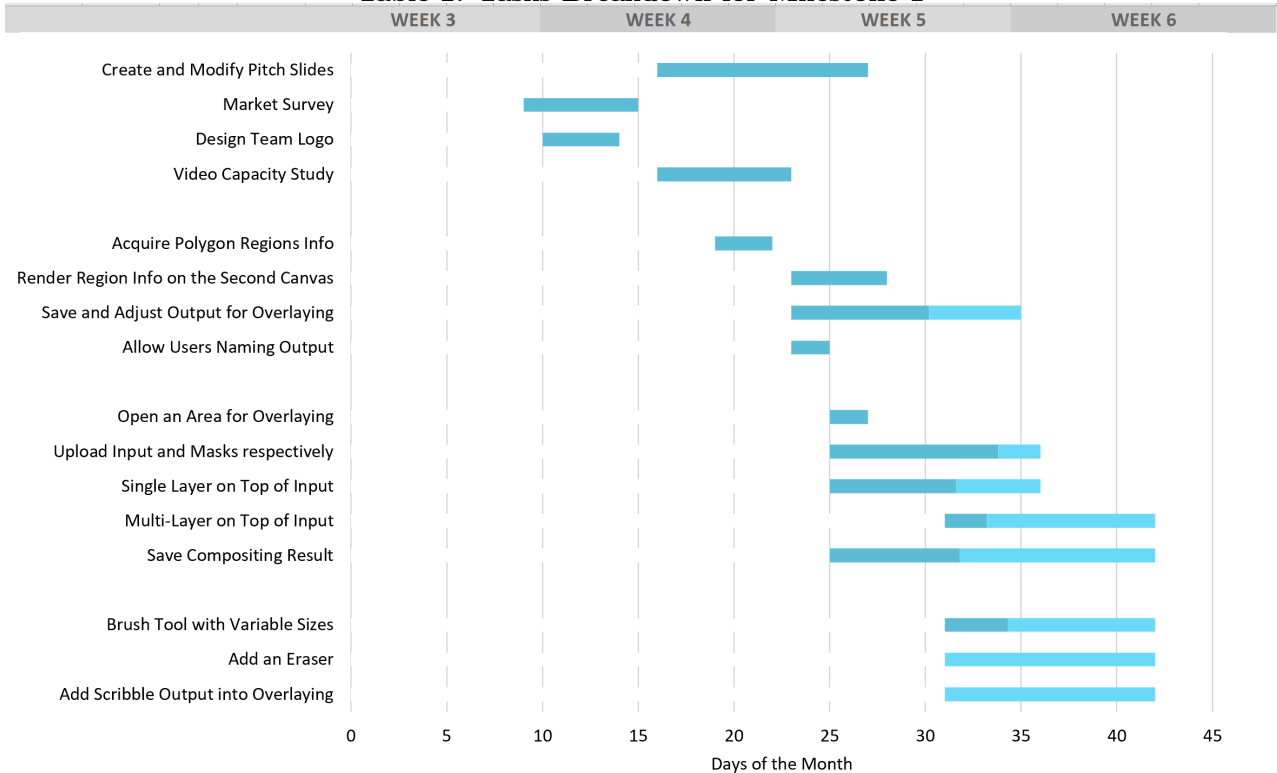
Table 3: Tasks Estimations: Gantt Chart

2.2 Project Management Activities

2.2.1 Collaboration inside Team 10

Based on Dr.Bastian's instruction, the development strategy of our team is divide and conquer. Each of us is responsible for different modules. His module is independent from mine. This mechanism allows we won't affect each other's progress.

1. For the pitch presentation, we rehearsed together and reviewed each other's speech script.

2. As two beginners to the front end development, we exchange our understanding of javascript from time

to time and share useful resources constantly.

3. In order to perform efficient and productive communication, we take turns to take minutes of client meeting then email well-formated version to Dr.Bastian for record.

4. Trello allow us making plan simultaneously online and tracking what we have accomplished. We specifically open up a pin-board for what we did well and what needs to be improved.

5. We use Github for tracking our contributions and version control.

2.2.2 Communication Plan with Supervisors/Clients

Client meeting time is every Friday 11am at IW Room 547. For every client meeting, we:

1. make thorough meeting agenda in advance and send it to Dr.Bastian for review. Then print out the final version and bring it to the meeting.

2. Taking language barrier into consideration, we take detailed notes of feedback from Dr.Bastian and Dr.Ward during the meeting.

3. after the meeting, we submit minutes to Github and email it to Dr.Bastian for record.

4. If we have something unclear or encounter technical issues, we resort to Dr.Bastian for help by email or bring the questions to next client meeting.

2.2.3 Quality Control

Quality control can be divided in three main parts:

1. **Code Quality**: VIA has high quality source code and its logic is well-organized. I choose VIA Version 2.0 as my prototype which contains more than 10,000 lines. Reading and understanding source code is a time-consuming and difficult process. However, I can't extend its code without a comprehensive understanding of how it works. I shall proceed in constant trial and modification.

2. **Function Quality**: Delivering commitments for each milestone on time is my first priority. Therefore with limited resources, in most of case time, I shall be creative and innovative For example, for video annotation, as javascript is unable to acquire the video's low level information like frame rate so I came up with a idea that by ignoring frame rate, I can use timeline for reference and take screenshots 10 times per second. So I can change a video into a sequence of images and apply images annotation tools to it.

3. **Users Experience**: During the 1st iteration, I shall focus on implement basic functions of three modules and gradually improve users experience afterwards.

3 Reference

1. http://www.vision.ee.ethz.ch/ cvlsegmentation/dextr/

2. http://www.cs.toronto.edu/polyrnn/

3. http://labelme.csail.mit.edu/Release3.0/

4. http://www.robots.ox.ac.uk/ vgg/software/via/