Visual ID: Textual Description to Image

Business Case and Draft Plan for Team 10

a1829445 - Kelvin, Ting
a1808891 - Pham Gia “Alan” Nghi
a1805827 - Kelin “Colin” Zhu

1 Business Case

1.1 Executive Summary

When crimes are reported, a textual description of the suspect is sent to polices. However, these descriptions contain much information that is challenging for polices to virtualise the suspect, especially in multi-tasking. This project involves generating training data and building a supervised machine-learning algorithm to extract the appearance descriptions of the suspect and ultimately virtualising these descriptions into images. Our project shall also have two advanced functions: appearance extraction from text and converting text into images.

1.2 Project Overview & Scope

To train a supervised machine learning model, data with labels is required. However, these data are not available due to confidentiality. We build the dataset using Tracey [1], a text generator engine creating game dialogue. Training a language model requires a lot of computation power and hardware, which is unavailable in our projects. We tailored a pre-trained model: RoBERTa [2], to extract the appearance description. Lastly, we will generate an image according to the five categorised classes: hair, skin, top cloths, bottom cloths, and accessories. The output image will be generated using a drawing library and will mainly consist of drawing clothing items with the corresponding colour. The machine learning model is hosted in the cloud by Azure Function[3], a cloud service available on-demand that provides all the continually updated infrastructure and resources needed to run our applications. Functions provide serverless computing, which we can build web APIs to fetch the image across platforms.

1.3 Market Positioning

There are many available language models in the market since this is an active research topic. We conducted a market survey revealing their pros and cons. In conclusion, performance, optimisation for appearance description and flexibility are essential to our projects, RoBERTa meets all basic requirements of our project. The survey result is listed below.

Name	Advantages	Disadvantages
OpenAI[4]	Trained by a large amount of corporus.	Not free and customized
BERT[5]	A language that is developed by google, performance fast.	Not taloreid for appearance extraction

Name	Advantages	Disadvantages
RoBERTa[2]	Highly-customized and good performance and optimized for appearance extraction	Need to retrain the whole model

1.4 Business Objective

The goal for our project is as follow: “building a system that can take a textual description of a person's appearance and produce an image that the officers can quickly glance at“ where we named “Visual ID”. Visual ID uses a language transformer model as the text interpreter for handling the Features Extraction functions. It will be written in Python, trained by using custom generative data from a set of categories and labels. Furthermore, our project will utilise the “charactercreator.org" open-source illustration library to generate the output image based on the extracted features. We avoid generating realistic representation due to the concept of uncanny valley in human identification. Instead, we focus on giving a the police officer an idea on how would the suspect described in the text might look like. The core goal of Visual ID is decrease the officer’s work load, as a result, the program will be a web application written in pure HTML, CSS and Javascript where it will fetch the image from an API we setup. The application will allow the police officer to use the program anywhere and on any device where internet connectivity is available.

1.5 Benefits and Limitations

1.5.1 Benefits

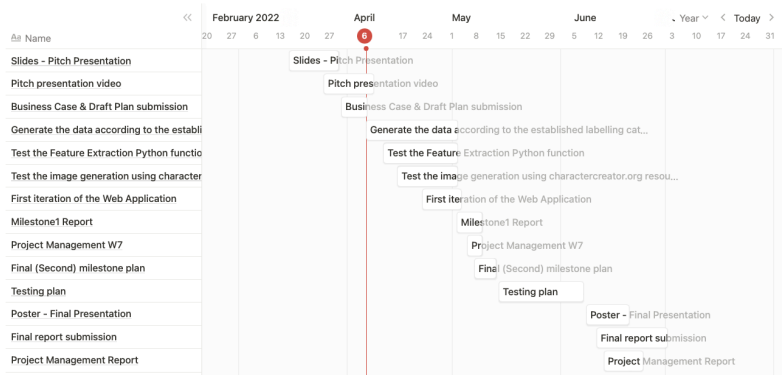
1. Our program can be used anywhere and from any device that support internet connectivity.
2. Unlike most security application where the program usually runs on desktop, our program will run on the web with an external server that compute the receiving text, generate the image and send the image to the officer’s device. Meaning, our program doesn’t rely on the hardware capability, so any device with a modern browser can run.
3. Inside a job, a description sentence is hidden among other information. Our program is able to automatically extract the relevant sentence and run the computation automatically. The process can be automated from the moment the job description is sent.

1.5.2 Limitations

1. Since the dataset of such police job description and illustration library are limited, our program might not be able to detect certain phrases or appearance. We can add more data later on but it can only be done after which might need to go through many iterations before we can finally settle on a MVP.
2. Most security technology infrastructure is outdated meaning integrating our program to existing tools can prove to be difficult.
3. Our program relies on internet connectivity so in a case where the police officer has slow access to one, the program might not be able to work fast enough to provide an instant feedback.

2 Draft Plan

2.1 Tasks Breakdown & Estimations



The tasks and roles assigned to all members in the team ranges from GitHub commits to Progress Journal. Kelvin Ting will be in charge of the Feature Extraction Machine Learning model, Colin Zhu will be in charge of the generative data and Alan Pham will be in charge of the image generation and web application. We will focus on the MVP and finish it in somewhere between May and June. We will initiate a test and refine process in early June and start finalising at late June.

Milestone 1	Activities	Projected Outputs
Minimum Viable Product	Generate the data according to the established labelling	Clean Generated Data that can be used for training
	Test the Feature Extraction Python function	The algorithm should be able to extract the relevant sentence and appearance features appropriated to the subject
	Test the image generation using charactercreator.org	Generate a correct visual representation from the extract features
	First iteration of the Web Application	Be able to run on the web and fetch the image from an API

2.2 Project Management Activities

2.2.1 Collaboration inside Team 15

As each one of us has different specialty, we avoid interfering each other works but still maintain constant communication. We have setup a weekly meeting with Clients and between ourselves to keep up with the progress. We take shift for writing Agenda and Meeting Minutes depend on the main subject of the meeting. We also keep a Progress Journal series where we document every major progress (everything is documented on Github and Notion). For the pitch presentation, we assign our speaking role according to our specialty. Alan is in charge of writing the base script and setting up the slide design. Kelvin and Colin comes in to edit the script to match with their way of presenting. This collaboration is made possible by using pitch.com where we can make our pitch slides together in realtime. Furthermore, we decided to do a proper video recording and editing to enhance the overall professionalism so we ask Colin to book a room and Alan to provide the microphone and camera.

Here are the overall management key points in our team:

1. For the pitch presentation, we use pitch.com to collab with each other. We rehearsed together and reviewed each other's speech script.
2. We use Notion as a knowledge hub to store information
3. With Thomas's advice, our client and supervisor, we use Github as a version control software.
4. We take turns to write Agenda, Meeting Minutes and Progress Journal

2.2.2 Communication Plan with Supervisors/Clients

Client meeting time is every Monday 10am on Zoom. For every client meeting, we go through the Agenda and Progress Journal on what we have done so far. We talk about the current challenge that we face and brainstorm a solution.

2.2.3 Quality Control

Quality control can be divided in three main parts:

1. **Design Quality**: We want our program to be easy-to-use but also visually pleasing since the last thing that any officer want is visual cluttering. Therefore the design must be clear, no

excessive animation, straight to the point with minimal buttons required and responsive across devices.

2. **Backend Quality**: The code of the project must be well documented and commented to allow any further development in the future.

3 Reference

1. http://tracery.io
2. https://ai.facebook.com/blog/roberta-an-optimized-method-for-pretraining-self-supervised-nlp-systems/
3. https://docs.microsoft.com/en-us/azure/azure-functions/
4. https://openai.com
5. https://medium.com/axinc-ai/bert-a-machine-learning-model-for-efficient-natural-language-processing-aef3081c24e8