Documentation, Prototype and Feasibility study

Zappa Roberto, Chettri Nishan

We have three main sides in our prototype and the proposed project too. The Fig.1 flow chart depicts the prototype that we have developed and not exact for the product that we want to develop as we use the materials in the kit to realize a prototype. In the following document we describe more of the project and the feasibility study.

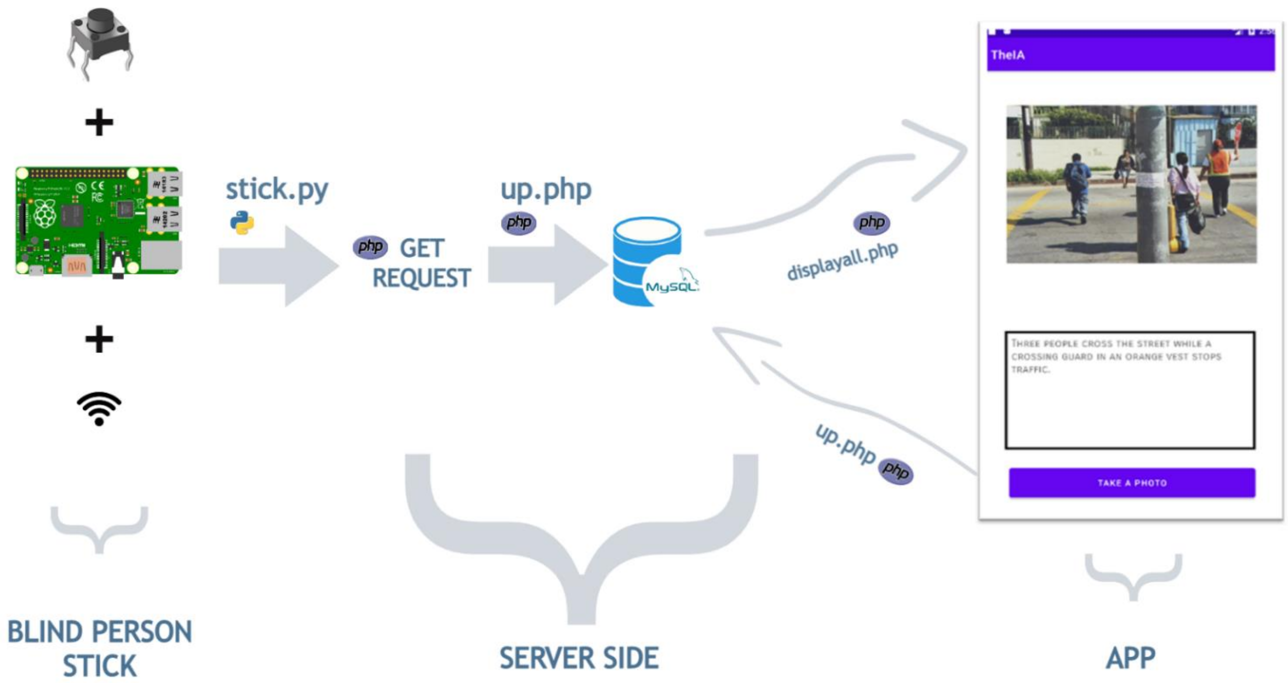


Figure 1.1

1. Blind Person Stick

In the stick of the blind person, we attach a hardware device which has a microcontroller and a BLE (Bluetooth Low Energy) chip and connect the app through it.

Usage:

The Blind person presses the button on the stick, and this is like a trigger that makes the app take a photo (or in our future advancements the eyewear takes a photo and relays it to the app which replies with the description of the image in form of an audio feedback).

Feasibility and Price:

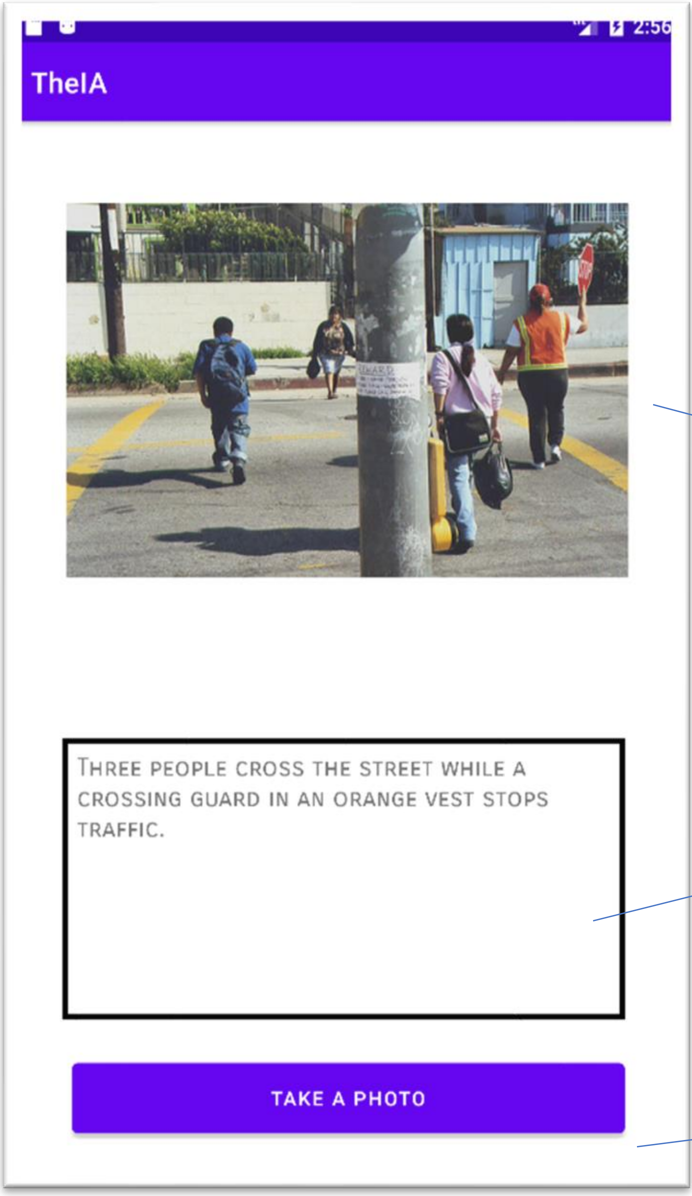
We take in account of power consumption of BLE devices, and we can see for some Flagship BLE chips we can attain long battery life and low power consumption. Also, with the abundance of Bluetooth devices around us, we should not doubt the feasibility and price of Bluetooth devices.

https://www.onsemi.com/company/news-media/blog/iot/bluetooth-low-energy-wireless-protocols-battery-life

Dist = 1.1 km FSK 38.4 kbps 10 Bytes Packet	TX Power	*Average Sleep Time Current:	Average TX Current:	RSSI	Coin Cell:	Transmits per Hour	Transmit Interval (minutes):	**Actual Battery Life (Years):
AXM0F243	6 dBm	2.5uA	27.3 mA	-106 dBm	240mAh	12	5	10.64

2. Android APP (prototype)

The prototype application is written in Java using Android Studio 2020.3.1 Artic fox. There are three sections inside the app:



*This is a replica of the app that we have thought of. Here we just demo what happens when an image is captured and there is no Image captioning model running on the backend.

The first one is a section for displaying photo. We used for demonstration purpose the images from Flickr30k Dataset

This is a section to show the description of the photo. We use recordings of Bixby voice service for text to speech.

Last section is a button used to switch from an image to another.

The application is connected to the stick via a server and PHP API's (as shown in Figure 1.1). The application monitors the state of a database column (true or false) (using a php API "displayall.php") and if the column is true the photo is changed (simulating the capture of image and the audio feedback), and the server is set on false (reset to its initial state using another php API "up.php" and a GET request.

The server or webhosting used for the API's is "000webhost.com" which is a free hosting site. The APIs are available in our Team's GitHub repo (click here).

All the APIs were self-developed using PHP as the main scripting language.

3. Machine Learning Feasibility

Image captioning model on smartphone

Image captioning is the process that generate a textual description of an image using an AI model made by one Convolutional Neural Network and one Recurrent Neural Network. In the first part the CNN extrapolate the features from the images while the RNN does language modelling.[1] The main difficulty in running the image captioning model locally is how to determine the hardware requirements needed for it to work properly. Given the limitations imposed on access and use of hardware external to whatever available in the kits provided to us, it was not feasible for us to implement and test image captioning models. For the feasibility study we're going to examine the hardware used by third party sources to perform image captioning models like what we hypothesized to draw some considerations.

Model - CoCoPIE's image captioning model.

CoCoPIE stands for Compression-Compilation co-design for Performance, Intelligence, and Efficiency. It is a software framework for enabling real-time AI on mainstream end devices. CoCoPIE holds numerous records on mobile AI: the first framework that supports all main kinds of DNNs, from CNNs to RNNs, transformer, language models, and so on; the fastest DNN pruning and acceleration framework, up to 180X faster compared with current DNN pruning on other frameworks such as TensorFlow-Lite; making many representative AI applications able to run in real-time on off-the-shelf mobile devices that have been previously regarded possible only with special hardware support.[2]

Reference Videos(hyperlinks-click on the underlined links below):

- On-mobile real-time image captioning 1 - YouTube
- On-mobile real-time image captioning 2 - YouTube
- On-mobile real-time image captioning 3 - YouTube

Hardware

In the reference videos the smartphone that is used is the Samsung Galaxy S10.

Chipset: Qualcomm Snapdragon 855

CPU: Qualcomm Kryo 485 Octa-core

GPU: Qualcomm Adreno 640

AI Engine: Hexagon 690

Qualcomm Snapdragon 855 is the hight end SoC of late 2018 made by Qualcomm using TMSC 7nm process.

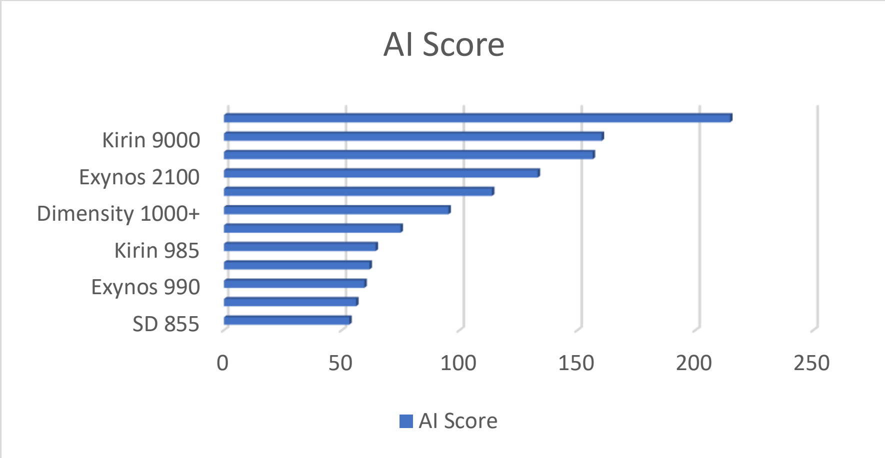


Figure 1 – Partial list of high end Soc since 2018 [3]

In Figure 1, We can see some of most famous flagship SoC made since 2018. As you can see every hight-end Soc younger than the Snapdragon 855 is powerful enough to do the inference of the model in real time.

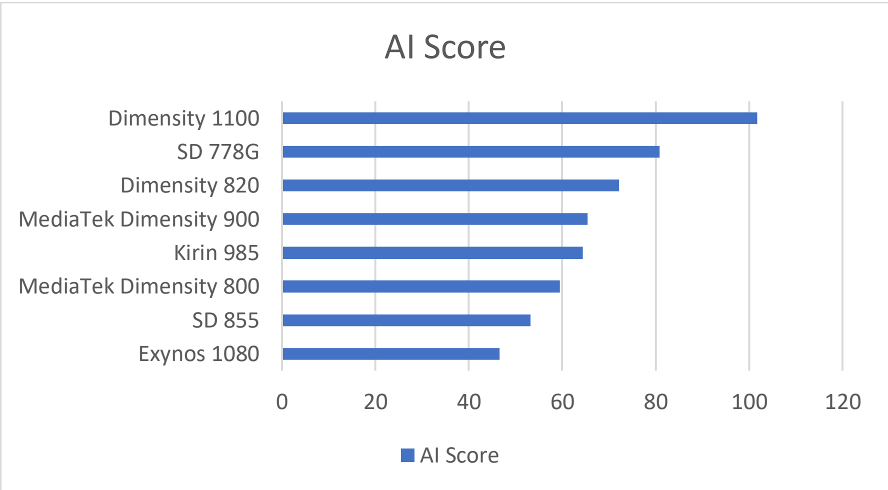


Figure 2 – Partial list of Mid-range Soc since 2018 [3]

Even in the Mid-range market nowadays there are SoC more powerful than SD855 as shown in Figure 2.

Takeaway from ML Feasibility study:

Since in reference video there is no waiting time between capture of image and receiving the description, we can assume that even SoC that are not as powerful as SD855 could do the inference on the device using of course more time. Otherwise, if the smartphone does not have the minimum requirements, it could be possible to use a third-party cloud service like the Amazon Web Services (AWS) to do the inference. In this scenario using an internet connection the smartphone will send to the servers the photo and receive the description of it.

References Cited

1. Image Captioning in Deep Learning (URL: https://towardsdatascience.com/image-captioning-in-deep-learning-9cd23fb4d8d2)
2. CoCoPIE (URL: https://arxiv.org/pdf/2003.06700.pdf)
3. AI-Benchmark (URL: https://ai-benchmark.com/ranking_processors.html)