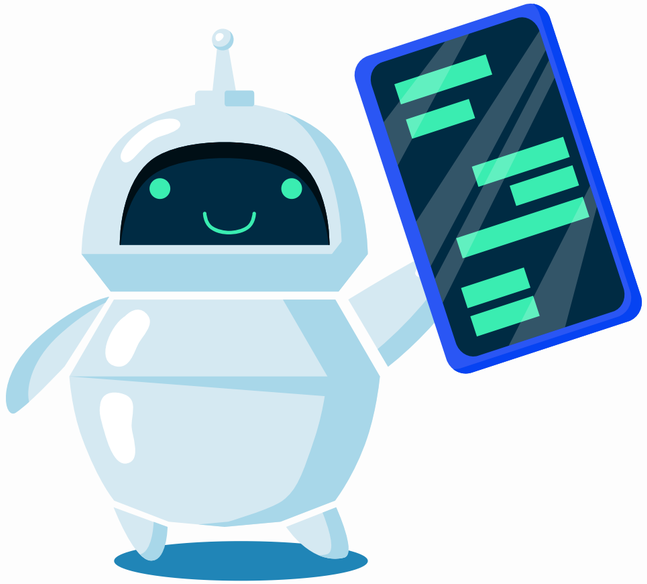
Custom Chatbot



Team member：

109590028吳凱印

109590056溫立賢

Contents

動機

近期由於ChatGPT的熱門，促使其競爭對手如Bard、Bing等相繼問世，也帶動了各式產業與企業對於生成式AI(AIGC)的應用與需求，AIGC 的用途非常廣泛，包括創作、娛樂、教育、研究等。
而我們做這個主題便是受到ChatGPT的啟發，而其開放的API也剛好能讓我們借用其大型語言模型(LLM)做出應用。

LangChain

LangChain是一個軟件開發框架，旨在簡化使用大型語言模型(LLM) 的應用程序的創建。
由於OpenAI的API是無法聯網的，所以如果只使用自己的功能實現聯網搜索並給出回答、總結 PDF 文檔、等功能是無法實現的，所以，我們使用了LangChain這個第三方開源庫。

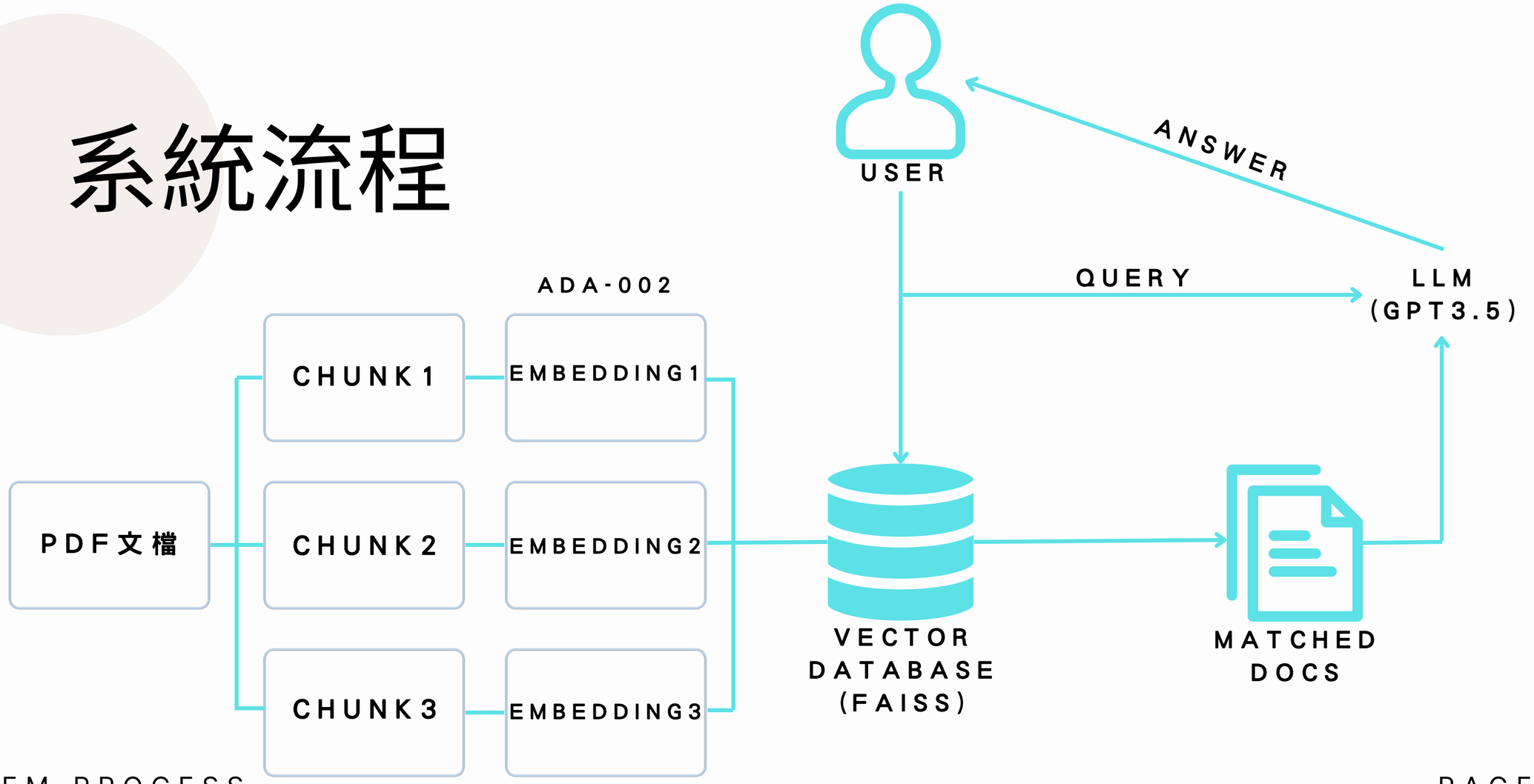
Logo

LangChain

它主要擁有2個能力：
1.可以將LLM模型與外部數據源進行連接
2.允許與LLM模型進行交互
其支持多種模型接口，比如 OpenAI、Hugging Face、Azure AI.....
擁有大量的文檔加載器，比如Email、PDF、Youtube .....
而我們所採用的文檔便是PDF來做為數據輸入

Logo

系統流程


USER
ANSWER
QUERY
LLM
(GPT3.5)
ADA-002
CHUNK1
CHUNK2
CHUNK3
EMBEDDING1
EMBEDDING2
EMBEDDING3
PDF文檔
VECTOR
DATABASE
(FAISS)
MATCHED
DOCS

採用文本

MediaPipe Hands: On-device Real-time Hand Tracking

Fan Zhang Valentin Bazarevsky Andrey Vakunov
Andrei Tkachenka George Sung Chuo-Ling Chang Matthias Grundmann
Google Research
1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA
{zhafang, valik, vakunov, atkach, gsung, chuoling, grundman}@google.com

Abstract

We present a real-time on-device hand tracking solution that predicts a hand skeleton of a human from a single RGB camera for AR/VR applications. Our pipeline consists of two models: 1) a palm detector, that is providing a bounding box of a hand to, 2) a hand landmark model, that is predicting the hand skeleton. It is implemented via MediaPipe[12], a framework for building cross-platform ML solutions. The proposed model and pipeline architecture demonstrate real-time inference speed on mobile GPUs with high prediction quality. MediaPipe Hands is open sourced at https://mediapipe.dev.

1. Introduction

Hand tracking is a vital component to provide a natural way for interaction and communication in AR/VR, and has been an active research topic in the industry [2] [15]. Vision-based hand pose estimation has been studied for many years. A large portion of previous work requires specialized hardware, *e.g.* depth sensors [13][16][17][3][4]. Other solutions are not lightweight enough to run real-time on commodity mobile devices[5] and thus are limited to platforms equipped with powerful processors. In this paper, we propose a novel solution that does not require any additional hardware and performs in real-time on mobile devices. Our main contributions are:

- An efficient two-stage hand tracking pipeline that can track multiple hands in real-time on mobile devices.
- A hand pose estimation model that is capable of predicting 2.5D hand pose with only RGB input.
- And open source hand tracking pipeline as a ready-to-go solution on a variety of platforms, including Android, iOS, Web (Tensorflow.js[7]) and desktop PCs.

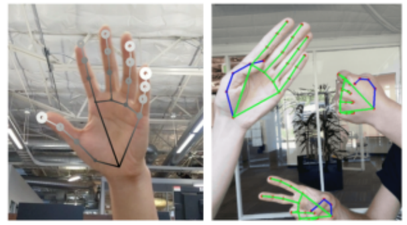


Figure 1: Rendered hand tracking result. (Left): Hand landmarks with relative depth presented in different shades. The lighter and larger the circle, the closer the landmark is towards the camera. (Right): Real-time multi-hand tracking on Pixel 3.

2. Architecture

Our hand tracking solution utilizes an ML pipeline consisting of two models working together:

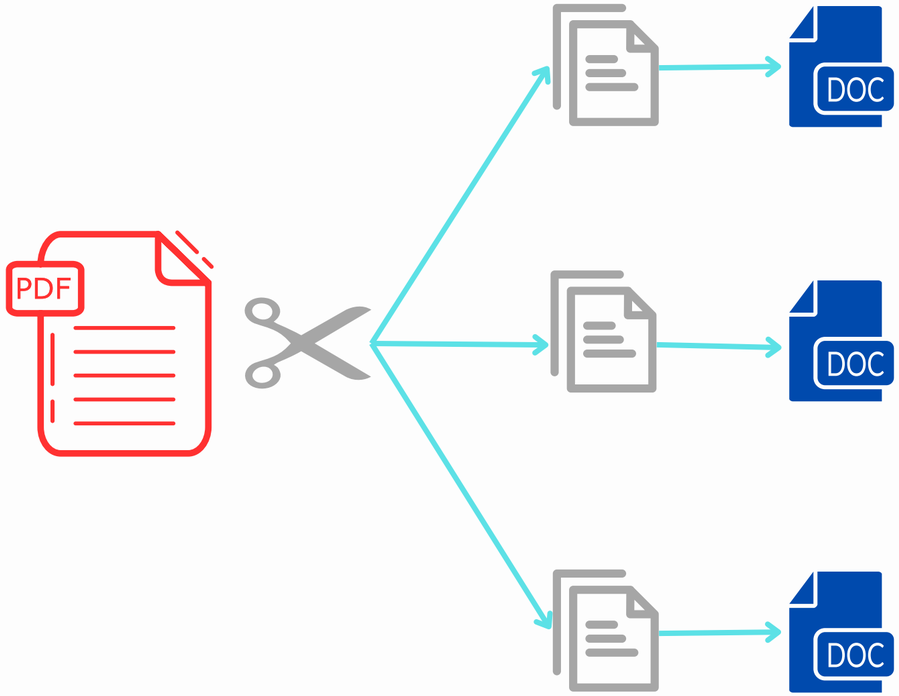
- A palm detector that operates on a full input image and locates palms via an oriented hand bounding box.
- A hand landmark model that operates on the cropped hand bounding box provided by the palm detector and returns high-fidelity 2.5D landmarks.

Providing the accurately cropped palm image to the hand landmark model drastically reduces the need for data augmentation (*e.g.* rotations, translation and scale) and allows the network to dedicate most of its capacity towards landmark localization accuracy. In a real-time tracking scenario, we derive a bounding box from the landmark prediction of the previous frame as input for the current frame, thus avoiding applying the detector on every frame. Instead, the detector is only applied on the first frame or when the hand prediction indicates that the hand is lost.

文本處理

1

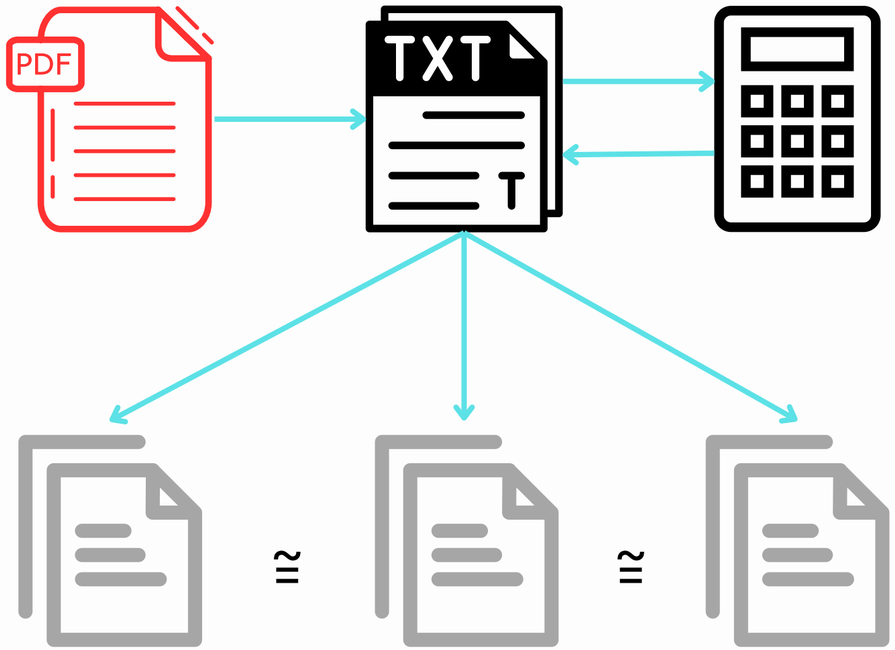
```
[29]  #  Simple  method  -  Split  by  pages
      loader  =  PyPDFLoader("./Mediapipe.pdf")
      pages  =  loader.load_and_split()
```



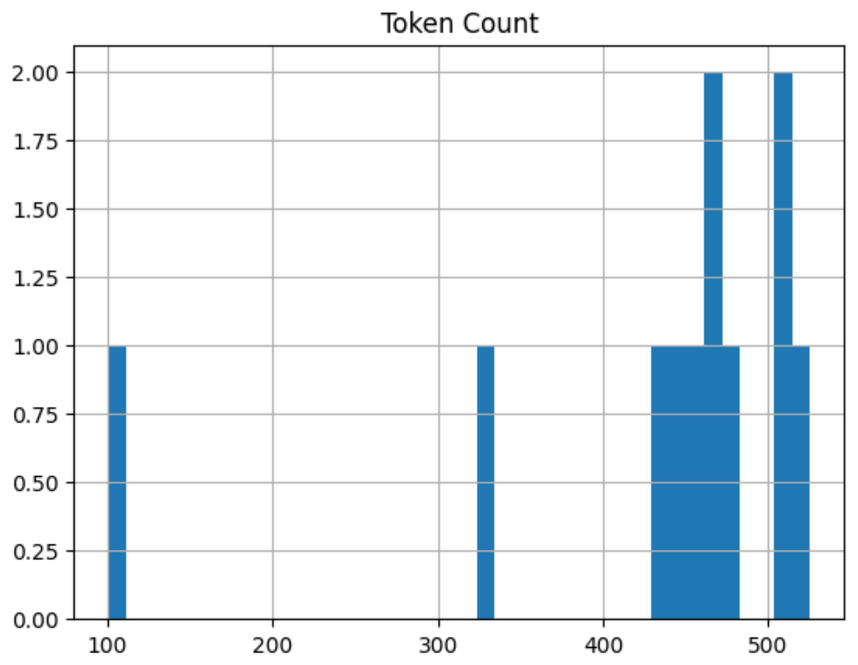
文本處理

2

```
[30]  # Another method - Split by chunk

      import textract
      doc = textract.process("./Mediapipe.pdf")

      with open('Mediapipe.txt', 'w') as f:
            f.write(doc.decode('utf-8'))

      with open('Mediapipe.txt', 'r') as f:
            text = f.read()

      tokenizer = GPT2TokenizerFast.from_pretrained("gpt2")

      def count_tokens(text: str) -> int:
            return len(tokenizer.encode(text))

      text_splitter = RecursiveCharacterTextSplitter(
            chunk_size = 512,
            chunk_overlap  = 24,
            length_function = count_tokens,
      )

      chunks = text_splitter.create_documents([text])
```

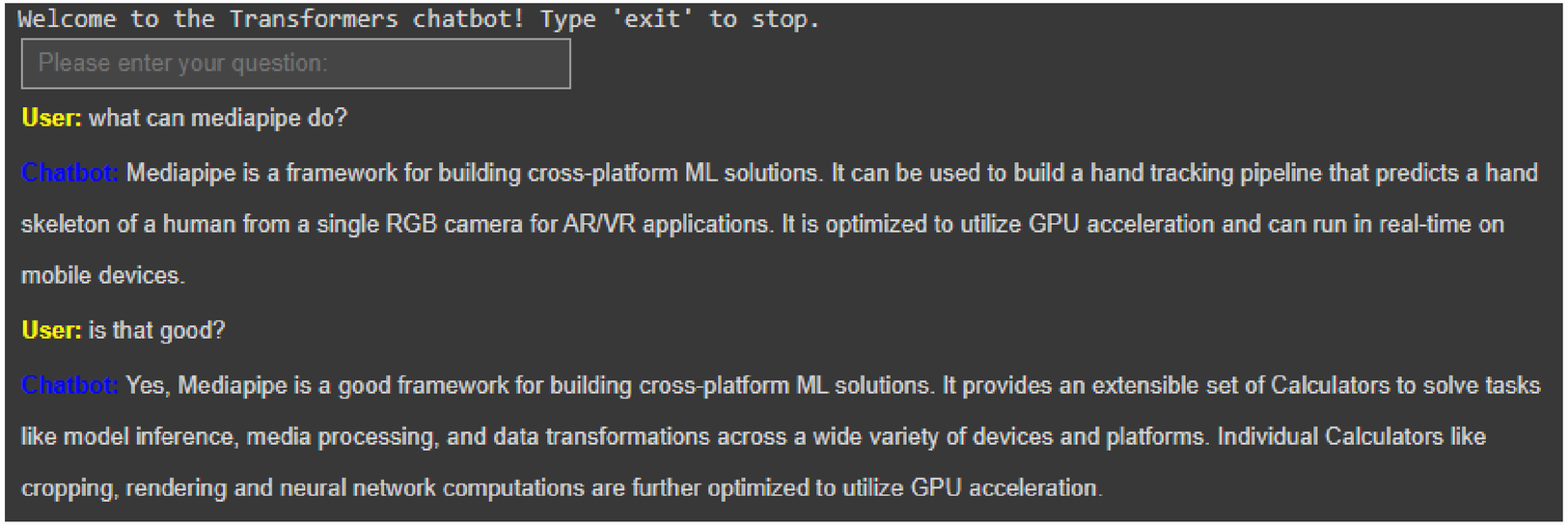


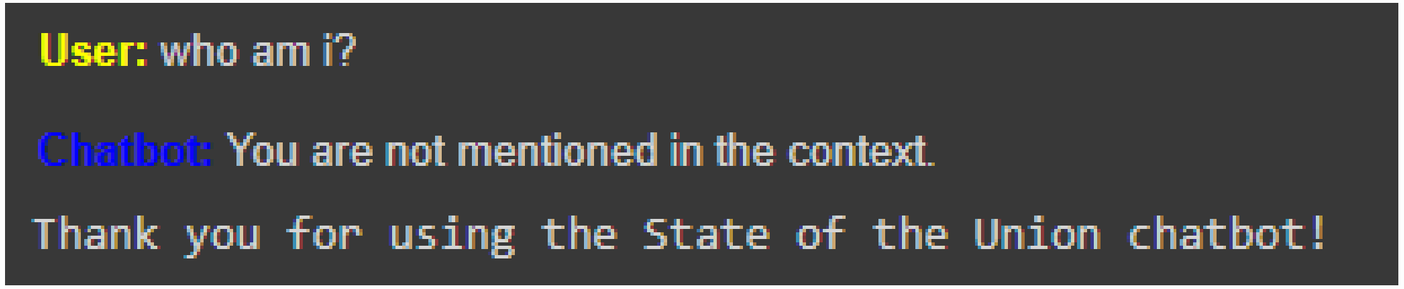
文本處理&建立數據庫




```
[32]  #  Get  embedding  model
      embeddings  =  OpenAIEmbeddings()

      #  Create  vector  database
      db  =  FAISS.from_documents(chunks,  embeddings)
```

Output


Welcome to the Transformers chatbot! Type 'exit' to stop.
Please enter your question:
User: what can mediapipe do?
Chatbot: Mediapipe is a framework for building cross-platform ML solutions. It can be used to build a hand tracking pipeline that predicts a hand skeleton of a human from a single RGB camera for AR/VR applications. It is optimized to utilize GPU acceleration and can run in real-time on mobile devices.
User: is that good?
Chatbot: Yes, Mediapipe is a good framework for building cross-platform ML solutions. It provides an extensible set of Calculators to solve tasks like model inference, media processing, and data transformations across a wide variety of devices and platforms. Individual Calculators like cropping, rendering and neural network computations are further optimized to utilize GPU acceleration.


User: who am i?
Chatbot: You are not mentioned in the context.
Thank you for using the State of the Union chatbot!

Thanks!