Hi there 👋

I received the M.Sc. degree in 2021 from Yonsei University, Seoul, Korea. Currently interested in designing MLOps architecture with IaC especially AWS CDK.
My CV

Synapsego: VLM, SLM serving

`#Airflow` `#Triton` `#FastAPI` `#vLLM` `#MLOps` `#AI` `#VLM` `#SLM`

- **Synapsego: Automated Visual Synopsis Creation**
  - Audio transcription model serving with Ray Serve
  - VLM multimodal inference serving and SLM serving with vllm + FastAPI
  - Vision models serving with Triton BLS
  - Pipelining with Airflow DAGs

Shot Up : Personalized AI Agent

`#AWS` `#CDK` `#Serverless` `#IOS` `#AI` `#Agent`

- **shot up: AI assistant for screenshots**
  - App Store Link
  - AWS architecture.
  - Analyzing with AI Agent
  - Semantic search.
  - AWS CDK(IaC) for deploying AWS resources.

Aug : Location-based AR SNS

`#AWS` `#CDK` `#Serverless` `#Event-driven` `#IOS` `#AR`

- **aug: spacial social**
  - App Store Link
  - AWS serverless event-driven architecture.
  - Video streaming & image content distribution.
  - AWS CDK(IaC) for deploying AWS resources.

MLOps

`#AWS` `#CDK` `#MLOps` `#AI` `#CCTV`

- **MLOps for AI Surveillance Camera**
  - Architecture Diagram
  - Dataset pipeline using AWS Fathom(co-developing service with SKT)
  - Design & implement train, inference, conversion, deploy pipeline
  - Video streaming & image content delievery
  - Semantic image search
  - AWS CDK(IaC) for deploying AWS resources.

- **Radio tower anomaly detection with drone images**
    - Design serverless architecture for batch inference pipeline.
    - AWS CDK(IaC) for deploying AWS resources.

Libuv Game Server

`#linux` `#libuv` `#C++` `#C#` `#Unity`

- **LibuvGameServer**
    - On Ubuntu 18.04, using libuv for TCP connection.
    - Based on libuv's event loop, Network IO is single threaded.
    - **[Bug]** Segmentation Fault after continuous Disconnection and Connection.
- **LibuvGameServerClients**
    - Dummy Client is based on C# .NET framework
    - Client is based on Unity
    - Demo avaiable here

Realtime Pose Estimation with Unity 3D Avatar

`#mediapipe` `#blazepose` `#Unity` `#3D` `#C#` `#Android`

- **BlazePoseWithUnity**
    - Only has Demo and brief description of project due to confidential rights

Face Detection & Recognition

`#Face Recognition` `#Android` `#Firebase ML Kit` `#TensorFlow 2.0` `#TensorFlow Lite` `#Java` `#Python`

- **JHFace**
    - Face recognition training & testing framework implemented with TensorFlow 2, Keras
    - Supported backbones: **MobileNet, MobileNetV2, InceptionResNetV2, InceptionV3, ResNet50, ResNet50V2, ResNet101V2, MnasNetA1, MnasNetB1, MnaseNetSmall, NASNetLarge, NASNetMobile, Xception, MobileNetV3Large, MobileNetV3Small, EfficientNetLite0 ~ Lite6, EfficientNetB0 ~ B7**
    - Supported losses: **ArcFace**, **CosFace**

- **Android-FaceRecognition**
    - Is runtime face identification on Android device. I used IJB-C dataset for testing labels.

- **FaceBird**
    - Is game applicaiton which utilize ML Kit for controlling the bird with rotating Face

Gaze Tracking (Eye Tracking)

`#Gaze Tracking` `#Android` `#Firebase ML Kit` `#TensorFlow Lite` `#PerCom 2021` `#GAZEL` `#Java` `#Python`

- **GAZEL**
  - Is a **Personalized Runtime Mobile Gaze Tracker**.
  - This work is official implementation of GAZEL framework which is published in PerCom 2021(GAZEL: Runtime Gaze Tracking for Smartphones) .
- **GazeBird**
  - Is game application which utilize GAZEL for controlling the bird.
- **MLKitGazeDataCollectingButton**
  - Is gaze data collecting application for making gaze estimation model required for GAZEL.

TensorFlow Lite Python

`#TensorFlow Lite` `#Python` `#Interpreter`

- Works on TensorFlow Lite Python Interpreter.
  Based on: https://www.tensorflow.org/lite/examples, converted Android Java(Kotlin) code to Python
  - TFLitePoseEstimation
  - TFLiteDetection
  - TFLiteClassification
  - TFLiteSegmentation

Power Management

`#Mobile` `#PM` `#flask` `#svm` `#dark mode`

▶ Works on optimizing Mobile device battery

▶ Works used to log iBeacons

Visualization on Web-browser

`#HighCharts.js` `#Go.js` `#flask` `#csv` `#Web browser`

▶ Visualizing Charts