International Journal of Advanced Culture Technology Vol.10 No.2 246-251 (2022)
DOI https://doi.org/10.17703/IJACT.2022.10.2.246

IJACT 22-6-33

Comparison of On-Device AI Software Tools

[1]Hong-Jong Song

[1]Senior Research Engineer, 4th Industrial ICT Team National Radio Research Agency
shjong@korea.kr

Abstract

As the number of data and devices explodes, centralized data processing and AI analysis have limitations due to the load on the network and cloud. On-device AI technology can provide intelligent services without overloading the network and cloud because the device itself performs AI models. Accordingly, the need for on-device AI technology is emerging. Many smartphones are equipped with On-Device AI technology to support the use of related functions. In this paper, we compare software tools that implement On-Device AI.

Keywords: *On-Device, AI, Tool, Machine Learning*

1. INTRODUCTION

From an application point of view, the demand for machine learning systems has increased over the past few years. Machine Learning (ML) has been adopted in a wide range of applications.[1] Cloud provides a suitable machine learning platform because it can easily store large amounts of data, has a low deployment cost and high computing power.[2]

Machine learning is being used in a way that the information collected from smart devices is transmitted to a cloud server for analysis and then received from the device again. In the case of real-time processing in the real world, many problems inevitably arise when processing depends on a cloud server. On-device AI is the way to solve such problems. On-device AI collects, calculates and processes information from smart devices without going through a cloud server.

In this paper, we collect data on on-device AI software tools and compare and analyze the characteristics of each on-device AI software tools based on this.

2. ON-DEVICE AI

On-Device AI is an emerging paradigm that aims to make devices more intelligent, autonomous and proactive by equipping them with machine and deep learning routines for robust decision making and optimal execution in devices' operations.

The advantages of on-device AI are as follows.[3]

- Increased speed and reduced latency. Because on-device AI does not need to communicate back and forth with the cloud, AI functions can perform seamlessly in real-time with no communication or connectivity-related delays.

Manuscript received: May 28, 2022 / revised: June 5, 2022 / accepted: June 9, 2022
Corresponding Author: shjong@korea.kr
Tel: +82-61-338-4653, Fax: +82-61-338-4651
4th Industrial ICT Team National Radio Research Agency Korea

Copyright©2022 by The International Promotion Agency of Culture Technology. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/4.0)

- Improved data security and privacy. Because the neural network is built into the device, there is no need to send sensitive data to the cloud for processing. This means that your personal data is kept on your p.
- Accessibility has been improved. AI and ML capabilities are available completely off-line, so you can now access them anywhere, anytime.
- Lower costs. Businesses can reduce data processing and bandwidth costs by running on-device ML instead of using cloud processing.
- Reduce power consumption and extend battery life. The next-generation AI chip is designed to dramatically reduce power usage and can even work when the device is in sleep mode. This keeps the speed of other functions and preserves battery life.
- Personalized, customized AI models. The AI chip comes with pre-trained data-rich models. With on-device AI, models are fine-tuned based on user input and optimized for individual users.

Table 1. shows the list of on-device AI software tools.[4] It can be seen that most of the companies that provide on-device AI functions are AI chip vendors. This means that the on-device AI function requires an AI chip and data is processed and provided on the AI chip.

Table 1. List of On-Device AI Software Tools

Tool	Company
ML Kit	Google
CoreML	Apple
TensorRT	Nvidia
ACL	Arm
AIMET	Qualcomm

3. ON-DEVICE AI SOFTWARE TOOLS

3.1 ML Kit

Table 2. ML Kit APIs

API	Category	Description
Vision	Barcode scanning	Scan and process barcodes. Supports most standard 1D and 2D formats.
	Face detection	Detect faces and facial landmarks.
	Image labeling	Identify objects, locations, activities, animal species, products, and more. Use a general-purpose base model or tailor to your use case with a custom TensorFlow Lite model.
	Object detection and tracking	Localize and track in real time one or more objects in the live camera feed.
	Text recognition	Recognize and extract text from images.
	Digital Ink Recognition	Recognizes handwritten text and handdrawn shapes on a digital surface, such as a touch screen. Recognizes 300+ languages, emojis and basic shapes.
	Pose detection	Detect the position of the human body in real time.
	Selfie segmentation	Separate the background from users within a scene and focus on what matters.
Natural Language	Language ID	Determine the language of a string of text with only a few words.
	On-device translation	Translate text between 58 languages, entirely on device.
	Smart Reply	Generate reply suggestions in text conversations.
	Entity Extraction	Detect and locate entities (such as addresses, date/time, phone numbers, and more) and take action based on those entities. Works in 15 languages.

Google ML Kit brings Google's machine learning expertise to mobile developers in a powerful, easy-to-use package. ML Kit's processing takes place on the device. This increases speed and unlocks real-time use cases such as processing camera input. It also works off-line and can be used to handle images and text that must remain on the device. ML Kit combines best-in-class machine learning models with advanced processing pipelines and delivers them through an easy-to-use API.[5]

The APIs of ML Kit are composed of vision API and natural language API as follows. The vision API is used for barcode, face, image recognition, object detection and tracking, character recognition, digital ink recognition, and selfie segmentation, while the natural language API is used for language type, on-device translation, smart response, and word entity extraction.

3.2 CoreML

Apple Core ML is optimized for on-device performance of a broad variety of model types by leveraging Apple hardware and minimizing memory footprint and power consumption. Core ML models run strictly on the user's device and do not require a network connection, keeping your app responsive and keeping the user's data private. Core ML supports modern models such as state-of-the-art neural networks designed to understand images, video, sound, and other rich media. Core ML model deployment makes it easy to deploy models to your app using CloudKit. Models from libraries like TensorFlow or PyTorch can be converted to Core ML much easier than before using the Core ML translator. Models bundled with your app can be updated with user data from the device, allowing models to remain relevant to user behavior without compromising privacy.[6]

Table 3. Models in Core ML format

Model	Category	Data	Description
FCRN-Depth Prediction	Depth Estimation	Image	Predict the depth from a single image.
MNIST	Drawing Classification	Image	Classify a single handwritten digit (supports digits 0-9).
Undatable Drawing Classifier	Drawing Classification	Image	Drawing classifier that learns to recognize new drawings based on a K-Nearest Neighbors model (KNN).
MobileNetV2	Image Classification	Image	The MobileNetv2 architecture trained to classify the dominant object in a camera frame or image.
Resnet50	Image Classification	Image	A Residual Neural Network that will classify the dominant object in a camera frame or image.
SqueezeNet	Image Classification	Image	A small Deep Neural Network architecture that classifies the dominant object in a camera frame or image.
DeeplabV3	Image Segmentation	Image	Segment the pixels of a camera frame or image into a predefined set of classes.
YOLOv3	Object Detection	Image	Locate and classify 80 different types of objects present in a camera frame or image.
YOLOv3-Tiny	Object Detection	Image	Locate and classify 80 different types of objects present in a camera frame or image.
PoseNet	Pose Estimation	Image	Estimates up to 17 joint positions for each person in an image.
BERT-SQuAD	Question Answering	Text	Find answers to questions about paragraphs of text.

3.3 TensorRT

NVIDIA TensorRT is a SDK for high-performance deep learning inference, includes a deep learning inference optimizer and runtime that delivers low latency and high throughput for inference applications. TensorRT maximizes throughput with FP16 or INT8 by quantizing models while maintaining accuracy, and fusing nodes in the kernel to optimize usage of GPU memory and bandwidth. It can also choose the best data layer and algorithm based on the target GPU platform, minimizing memory footprint and efficiently reusing memory for tensors. It uses a scalable design to process multiple input streams in parallel, and a dynamically generated kernel to optimize a recurrent neural network over time steps. [7]

The following figure shows the workflow of TensorRT to optimize and infer a trained model.

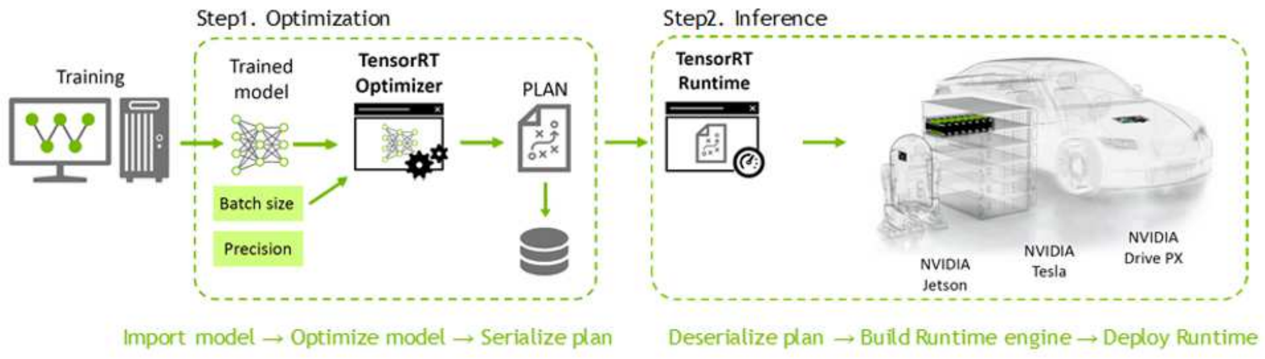

Figure 1. TensorRT Workflow

3.4 Arm Compute Library(ACL)

Arm Compute Library (ACL) is a collection of low-level machine learning features optimized for the Cortex-A CPU and Mali GPU architectures. The library is open source software available under the MIT license. ACL provides more than 100 machine learning functions for CPU and GPU and multi-line algorithms such as GEMM, Winograd, FFT, Direct. It also supports multiple data types such as FP32, FP16, int8, uint8, BFloat16, provides micro-architecture optimizations for key ML primitives, highly configurable build options that enable lightweight binaries, and more.[8]

3.5 AI Model Efficiency Toolkit(AIMET)

Qualcomm AIMET is an open source library for optimizing trained neural network models. It does this by providing advanced model compression and quantization techniques that reduce the model while maintaining operational accuracy. Smaller models provide better runtime performance, lower latency, and lower compute, memory, and power consumption. Developers can integrate AIMET's advanced model compression and quantization algorithms into their PyTorch and TensorFlow model building pipelines for automated post-training optimization and model refinement. Automating these algorithms helps eliminate the need for manually optimized neural networks that can be time consuming, error prone, and difficult to iterate.[9]

The following figure shows the AIMET workflow for generating an optimization model through AIMET compression and quantization of the trained model, and deploying the generated model to a smartphone.

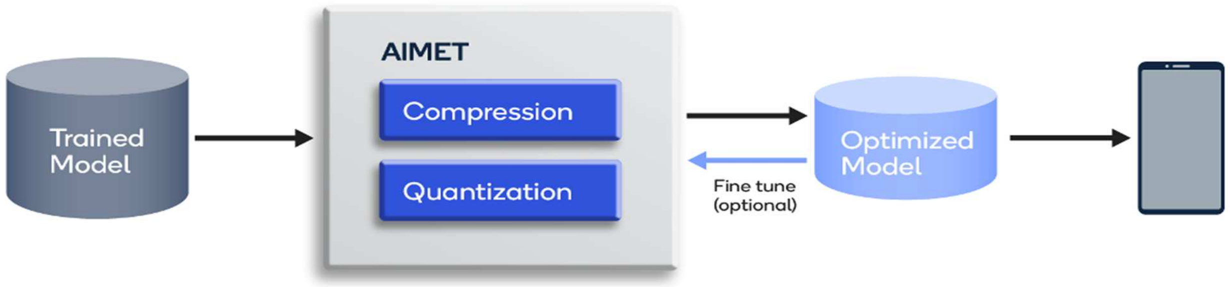

Figure 2. AIMET Workflow

4. COMPARISON OF ON-DEVICE AI SOFTWARE TOOLS

This section compares On-Device AI Software Tools such as ML Kit, Core ML, TensorRT, Arm Compute Library (ACL), and AI Model Efficiency Toolkit (AIMET).

Table 4. shows the systems and frameworks supported by the on-device AI software tool. It can be seen that most on-device AI software tools support iOS and Android operating systems for smartphones to provide on-device AI functions. Also, most on-device AI software tools support PyTorch and TensorFlow frameworks. This is to use models built with PyTorch and TensorFlow.

Table 4. Supported Systems and Framework of On-Device AI Software Tools

Tool	Supported Systems	Framework
ML Kit	iOS, Android	TensorFlow
CoreML	iOS	PyTorch, TensorFlow, XGboost, scikit-learn, LIBSVM
TensorRT		PyTorch, TensorFlow, ONNX, MATLAB
ACL	Linux, Android, macOS, Bare Metal	
AIMET		PyTorch, TensorFlow

Table 5. Features of On-Device AI SW Tools

Tool	Feature
ML Kit	- Model Deployment : Firebase
CoreML	- Model Deployment - Model Encryption
TensorRT	- Quantization - Layer and Tensor Fusion - Multi-Stream Execution
ACL	- Over 100 machine learning functions for CPU and GPU - Multiple convolution algorithms (GEMM, Winograd, FFT and Direct)
AIMET	- Model Compression - Quantization

Table 5. shows the features of the on-device AI software tool. ML Kit can deploy models to Firebase programmatically using either Python or the Node SDK.

Core ML model deployment makes it easy to deploy models to your app using CloudKit. Core ML's Xcode supports model encryption enabling additional security for your machine learning models.

TensorRT optimizes GPU memory and bandwidth usage by fusing nodes in the kernel, and uses a scalable design to process multiple input streams in parallel.

ACL provides multiple convolution algorithms and many machine learning functions for CPU and GPU.

It can be seen that TensorRT and AIMET support quantization. Quantization maximizes throughput by quantizing the model while maintaining accuracy. AIMET provides model compression and quantization techniques that reduce the model while maintaining operational accuracy.

5. CONCLUSION

In this paper, the characteristics of on-device AI software tools such as ML Kit, CoreML, TensorRT, ACL, and AIMET were investigated.

It can be seen that most on-device AI software tools provide APIs, various learning models, and optimization functions that support various AI functions on-device. Machine learning is being used in a way that the information collected from smart devices is transmitted to a cloud server for analysis, and then received back from the device. For real-time processing in the real world, many problems inevitably arise when processing relies on cloud servers. On-device AI is a way to solve these problems. On-device AI collects, calculates, and processes information from smart devices without going through a cloud server. The convergence of cloud AI and on-device AI is expected to be widely used.

Reviews of on-device AI software tools can help those who develop or plan on-device AI software in the future to decide which features to use in which environments.

REFERENCES

[1] Y. Lee, "Analysis of Automatic Machine Learning Solution Trends of Startups," Vol.8, No.2, International Journal of Advanced Culture Technology, 2020, *https://doi.org/10.17703/IJACT.2020.8.2.297*.

[2] Y. Lee, “Analysis on trends of machine learning-as-a-service,” Vol. 6, No. 4, International Journal of Advanced Culture Technology, 2018, *https://doi.org /10.17703//IJACT2018.6.4.303*.

[3] On-Device Artificial Intelligence: A Game Changer, *https://innodata.com/on-device-artificial-intelligence/*

[4] S. Lee, "Trend of On-Device AI Hardware and Software Technology Development," Weekly Technology Trend, No. 2028, 2021.

[5] ML Kit, *https://developers.google.com/ml-kit.*

[6] CoreML, *https://developer.apple.com/machine-learning/core-ml/.*

[7] TensorRT, *https://developer.nvidia.com/tensorrt.*

[8] Arm Compute Library, *https://www.arm.com/technologies/compute-library*.

[9] AI Model Efficiency Toolkit(AIMET), *https://developer.qualcomm.com/software/ai-model-efficiency-toolkit*.