

# **Himax WE-1 : Tensorflow Lite Micro, Edge Impulse**

1.0.0

2021. 04. 20

SunBeen Moon

# Contents

<b>1. Himax WE-I 보드 소개와 TinyML 소개</b>	<b><a href="#">5</a></b>
1.1. Himax WE-1 보드 소개	<a href="#">5</a>
1.2. TinyML 소개	<a href="#">7</a>
<b>2. 초기 세팅</b>	<b><a href="#">8</a></b>
2.1. Himax WE-1 update Flash Image	<a href="#">8</a>
2.2. Synopsys DesignWare ARC MetaWare Development Toolkit	<a href="#">10</a>
2.3. Check Make version	<a href="#">16</a>
2.4. Arduino IDE	<a href="#">17</a>
2.5. Edge Impulse	<a href="#">18</a>
2.6. Issues	<a href="#">19</a>
<b>3. Tensorflow Lite Micro</b>	<b><a href="#">20</a></b>
3.1. Introduction	<a href="#">20</a>
3.2. Examples	<a href="#">21</a>
3.2.1. 1 – Micro Speech	<a href="#">22</a>
3.2.2. 2 – Magic Wand	<a href="#">23</a>
3.2.3. 3 – Person Detection	<a href="#">24</a>
3.2.4. 4 – Handwriting	<a href="#">25</a>
3.2.5. 5 – Hello World	<a href="#">28</a>
3.3. Issues	<a href="#">29</a>

<b>4. Edge Impulse .....</b>	<b><u>30</u></b>
4.1. Introduction .....	<u>30</u>
4.2. Description of the Edge Impulse Docs .....	<u>34</u>
4.3. Projects' Tutorial .....	<u>35</u>
4.4. Examples .....	<u>60</u>
4.4.1. 1 – Audio(given data : Keyword Spotting) .....	<u>60</u>
4.4.2. 2 – Audio(generated data 1 : '안녕') .....	<u>64</u>
4.4.3. 3 – Audio(generated data 2 : 'sol' vs 'alarm') .....	<u>67</u>
4.4.4. 4 – Image(generated data 1 : 'dog' vs 'cat') .....	<u>71</u>
4.4.5. 5 – Image(generated data 2 : 'mountain' vs 'sea') .....	<u>76</u>
4.4.6. 6 – Accelerator(generated data) .....	<u>79</u>
4.4.7. 7 – Image(MNIST) .....	<u>83</u>
4.5. Issues .....	<u>85</u>
 <b>5. embARC MLI &amp; ARC GNU toolchain .....</b>	 <b><u>86</u></b>
5.1. embARC MLI . .....	<u>86</u>
5.1.1. Introduction .....	<u>86</u>
5.1.2. 설치 방법 .....	<u>86</u>
5.1.3. Example code .....	<u>87</u>
5.2. ARC GNU toolchain .....	<u>89</u>
5.2.1. Introduction .....	<u>89</u>
5.2.2. 설치 방법 .....	<u>89</u>

<b>6. Appendix .....</b>	<b><u>90</u></b>
6.1. Install Oracle Virtual Machine(Regarding the Himax WE-1 board) .....	<u>90</u>

# 1. Himax WE-1 보드 소개와 TinyML 소개

## 1.1 Himax WE-1 보드 소개

Himax WE-1 보드는 Synopsys사에서 end-point AI edge device를 타겟으로 만들어진 보드이다. AI, 5G 그리고 IoT 기술의 융합됨에 따라, 소비자는 (초)저전력 endpoint에서 컴퓨팅과 결합된 on-device AI를 기대한다.

Synopsys DesignWare ARC Processor는 ARM에 이어 임베디드 프로세서 시장 점유율 2위인 제품이다. Synopsys사는 대만에 본사를 두고 있으며 대만, 일본, 한국, 미국 등으로 사업을 확장하고 있다. Synopsys사는 ARC라는 CPU IP를 제공하고 있다. IP의 종류는 사용 목적에 따라, EM, SEM, HS, VPX, EV로 나뉘어진다. 이 중에서 Himax WE-1 보드가 사용하고 있는 IP는 EM Family이며, (초)저전력 임베디드 프로세서를 위해 최적화되었다. Edge device에서 AI 모델이 구동되기 위해서는 빠른 추론 연산이 필요하며, 이 연산을 뒷받침해줄 RISC와 DSP를 지원하고 있다. (Himax WE-1 보드는 EM9D 칩을 사용하고 있다.) Synopsys사에서는 Eclipse-based IDE와 nSIM Instruction-Set Simulator를 지원하고 있다. 또한, embARC.org 오픈 소스 웹사이트를 통해 ARC사의 칩 개발을 위한 기술 지원을 하고 있다. Synopsys embARC MLI를 통해 다양한 AI 모델을 구축할 수 있도록 함수들을 서포트하고 있다. 이와 관련된 내용은 5절에서 기술한다.

ARC Subsystems	Supported Processors	Hardware Accelerators	Integrated Peripherals	Included Software
ARC IoT Communications IP Subsystem	EM11D	✓	SPI, UART(s), GPIO, Digital Front End (DFE), PMU and RTC	DSP library, base communications library, device drivers
ARC Data Fusion IP Subsystem	EM5D, EM7D, EM9D, EM11D	✓	SPI, I2C, I3C, PWM, UART, PDM, ADC I/F, DAC I/F, APB I/F, GPIO	DSP library, audio processing library, peripheral I/O drivers (bare metal), reference designs ARC Sensor & Control
ARC Sensor and Control IP Subsystem	EM4, EM6	✓	SPI, I2C, PWM, UART, ADC I/F, DAC I/F, APB I/F, GPIO	DSP library, peripheral I/O drivers (bare metal)
ARC SoundWave Audio IP Subsystem	AS211SFX, AS221BD	✓	I2S, S/PDIF, Analog Codec I/F, Reset, Clock Management	Multi-core Media Framework, MM MQX Audio Post-processing S/W

Table 1: DesignWare ARC Processor Subsystems

Fig 1. ARC사의 EMxD에서 지원하는 Subsystems

Himax WE-1 보드와 구글 Tensorflow Lite for Microcontroller와 collabo를 진행하여 대표적인 Tensorflow Lite for Microcontroller(이하 TFLM) 예제, Person Detection, Micro Speech, Magic Wand 예제를 지원한다. 또한, Edge Impulse사에서 공식으로 Synopsys사의 Himax WE-1 보드를 지원한다.

Himax WE-1 보드의 주요 스펙은 아래와 같다.

- ARC 32-bit EM9D DSP with FPU
- 400MHz clock frequency
- 2MB Flash
- 2MB SRAM
- Ultra-low power VGA CCM
- SPI/I2C/UART Bridge
- GPIO

등등

이 외 다양한 Himax WE-1 보드의 스펙을 알고자 한다면 아래 요한님께서 정리하여 올려 주신 slack 페이지를 참고 바란다. 간략한 설명은 'WE-1\_Plus\_EVB\_Technical\_Document', 상세한 설명은 'HX6537-A\_DS\_public\_v01\_1\_.pdf'을 참고 바란다.

(slack 링크) <https://nota-workspace.slack.com/archives/C017XEU4VUK/p1612404212069200>

<참고 링크>

(Himax 사 홈페이지) <https://www.himax.com.tw/>

(EM9D 칩 설명 페이지) <https://www.synopsys.com/dw/ipdir.php?ds=arc-em9d-em11d>

(embARC 오픈 소스 웹페이지) <https://embarc.org/>

(Edge Impulse 사의 Himax WE-1 보드 지원) <https://www.edgeimpulse.com/blog/himax-wei-plus>

(DSP 의 NN 알고리즘 처리 원리와 TFLM 동영상) <https://www.youtube.com/watch?v=RhXBRLMjqZA>

## 1.2 TinyML 소개

TinyML은 IoT 기기와 같이 (초)저전력, 제한된 컴퓨팅 능력, 한정된 메모리 환경을 갖는 edge device에서 ML/DL 기능을 실현하는 것을 목표로 하며 이러한 제한된 환경에서 원하는 태스크를 수행하기 위해선 기존과는 차별화된 기술 개발이 요구된다. 현재 전 세계로 TinyML 실현을 위한 다양한 연구, 개발이 진행 중이며 많은 기업들이 관련 솔루션을 출시하고 있다.

(TinyML 유튜브) <https://www.youtube.com/channel/UC9iWqvsWjhowkHWVJquHwkg/videos>

출처 : 요한님

## 2. 초기 세팅

### Small Agenda

2.1 Himax WE-1 Update Flash Image .....	<a href="#">8</a>
2.2 Synopsys DesignWare ARC MetaWare Development Toolkit .....	<a href="#">10</a>
2.3 Check Make Version .....	<a href="#">16</a>
2.4 Arduino IDE .....	<a href="#">17</a>
2.5 Edge Impulse .....	<a href="#">18</a>
2.6 Issues .....	<a href="#">19</a>

### 2.1 Himax WE-1 update Flash Image

[https://github.com/HimaxWiseEyePlus/bsp\\_tflu/tree/master/HIMAX\\_WE1\\_EVB\\_user\\_guide](https://github.com/HimaxWiseEyePlus/bsp_tflu/tree/master/HIMAX_WE1_EVB_user_guide)

위 깃허브 주소에 따라 Himax WE-1 보드의 Flash Image 를 update 해주어야 한다.

아래는 간략히 Flash Image 를 update 하는 과정을 기술한다. (note 가 없을 경우, 깃허브의 순서대로 설치를 진행하면 된다)

(note) Linux version 과 Window version 이 둘 다 있으나, Window version 은 선택, Linux version 은 필수로 실행해야 한다.

	Update Bootloader	Flash Image Update	파일 upload
Linux	O	O	O
Window	X	O	O

Table 1. Flash Image 가 지원되는 목록

#### 1. System Requirements

Window 의 경우 : TeraTerm 을 설치한다.

<https://amsoft.tistory.com/entry/%EA%B0%9C%EB%B0%9C%ED%99%98%EA%B2%BD-%EB%AC%B4%EB%A3%8C-%ED%84%B0%EB%AF%B8%EB%84%90-%ED%85%8C%EB%9D%BC-%ED%85%80-Tera-Term-%EC%84%A4%EC%B9%98-%EB%B0%8F-%EC%82%AC%EC%9A%A9%EB%B2%95>

Linux 의 경우 : Minicom 을 설치한다.

<https://hoho325.tistory.com/29>

(note) Window 에서 Virtual Box 로 Linux 를 설치하는 방법은 Appendix 에 기술되어 있다.



## **2. Check Bootloader Version**

## **3. Update Bootloader Version at Linux Environment**

(note) Linux 환경에서만 설치가 가능하다.

## **4. Flash Image Update**

Window 환경 혹은 Linux 환경에서 Flash 를 update 한다.

## 2.2 Synopsys DesignWare ARC MetaWare

### Development Toolkit

Himax WE-I 보드를 사용하기 위해서는 Synopsys사에서 제공하는 Development Toolkit 을 다운 받아야 한다. Toolkit 을 다운로드 받으면서, Synopsys사의 License 를 받아야 한다. Linux 와 Window 둘 다 세팅 가능하다. 설치되는 toolkit 중 핵심인, 'ccac' compiler 를 통해 원하는 보드에 걸맞게 코드를 변환해 준다.

<https://github.com/tensorflow/tensorflow/blob/master/tensorflow/lite/micro/tools/make/targets/arm/README.md#install-the-synopsys-designware-arc-metaware-development-toolkit>

위 깃허브 링크에서 전체적인 Synopsys DesignWare ARC MetaWare Development Toolkit 을 다운받는 방법이 나와 있다. 위 깃허브를 참고하며 순서대로 프로그램을 다운로드 후, 순서대로 설치하는 방법은 아래와 같다.

#### 1. MetaWare Development Toolkit 선택

<https://eval.synopsys.com/>

위 Synopsys 웹사이트에서 MetaWare Development Toolkit 을 선택한다.

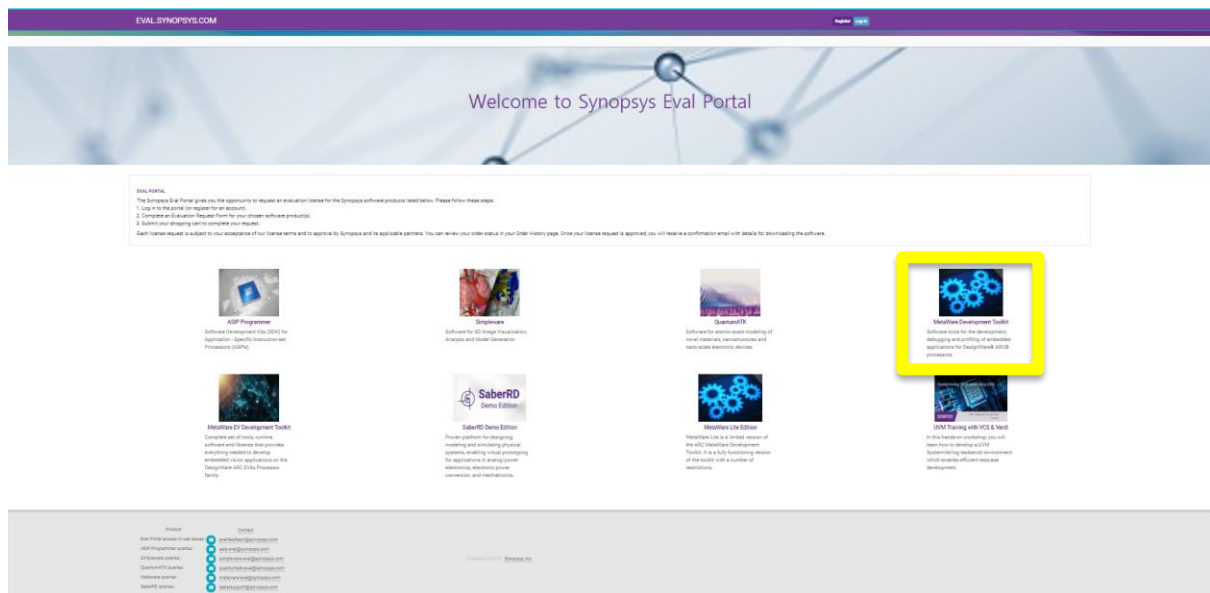


Fig 2. MetaWare Development Kit 선택 창

## 2. 로그인 후, License Request

로그인 후, 아래와 같은 화면이 나오면 Request Trial Now 를 선택한다.

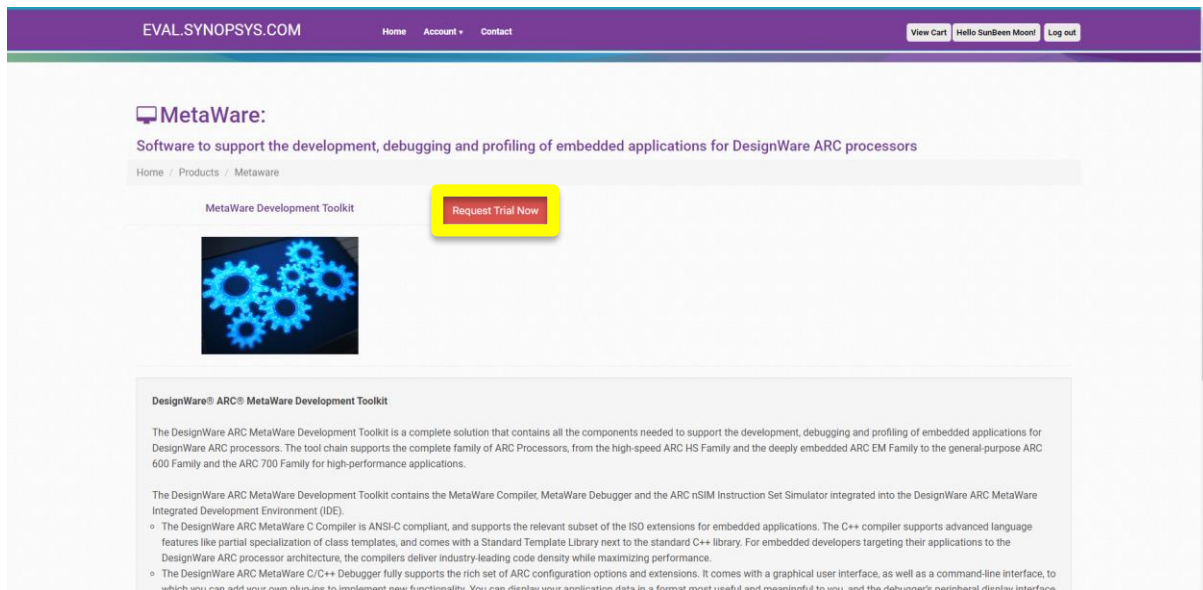


Fig 3. License request 요청 창

아래와 같은 Request Form 을 기입한다.

The screenshot displays the 'MetaWare Evaluation Request Form'. The form is titled 'MetaWare Development Toolkit' and includes a red banner indicating that asterisks (\*) denote mandatory fields. The form contains several input fields: 'Operating System\*' (a dropdown menu), 'License Duration' (a text field with '30 days'), 'License Start Date\*' (a date picker showing '03/02/2021' with a note '(ie : MM/DD/YYYY)'), 'Host ID\*' (a text field with a note 'Enter HostID' and 'Determining the correct hostid for Synopsys Licensing (Ex : D4-81-D7-27-C5-CA)'), 'License Quantity' (a text field with '1'), 'License Type' (a dropdown menu with 'NodeLocked License'), and 'License Use' (a dropdown menu with 'Evaluation Use'). At the bottom, there's a question 'What is the purpose of the evaluation?\*' followed by a text area for the user to provide a brief description of their project.

Fig 4. License Request Form 작성 창

- ✓ Operating System 은 가능하면, 'Linux 64-bit'를 선택한다.

(note) 'Both', Linux 64-bit 와 Window 64-bit 둘 다 선택이 되지 않는다.

(note) embARC MLI 를 제외한 다른 operation 에서는 Linux License 를 사용하므로, Linux 64-bit 를 선택하는 편이 유리하다.

(note) License 는 Window, Linux 둘 다 신청이 가능하다. 한 license 를 신청한다고 다른 license 가 신청이 안 되지 않는다. 즉, Window 64-bit License 와 Linux 64-bit License 를 둘 다 각각 신청하여 사용하여도 된다.

- ✓ Host ID 는 MAC 주소를 입력한다.

(note) <https://usefulinfos.tistory.com/91> 링크를 참고하여 MAC 과 Window 에서 MAC 주소를 확인한다.

(note) Window 의 경우, cmd 창에 ipconfig/all 을 입력한다. '이더넷 어댑터 이더넷'의 물리적 주소를 입력한다.

### 3. View Cart

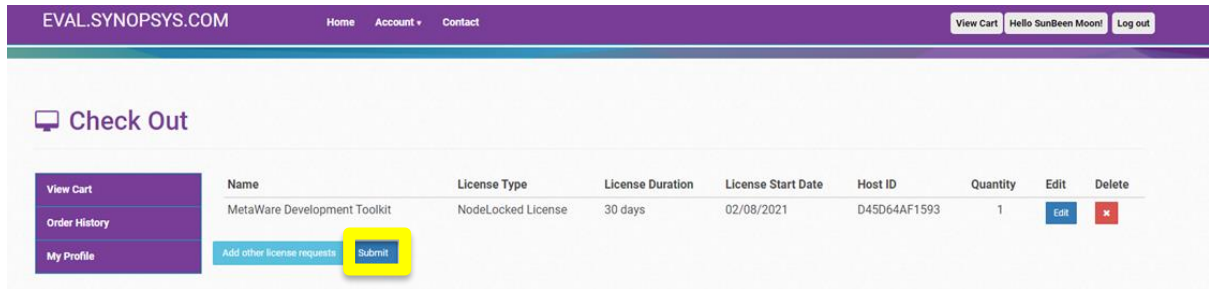


Fig 5. License Request 가 진행된 후 Cart 에 요청된 창

위 그림과 같이 License 가 신청되면, 꼭 'submit' 버튼을 누른다.

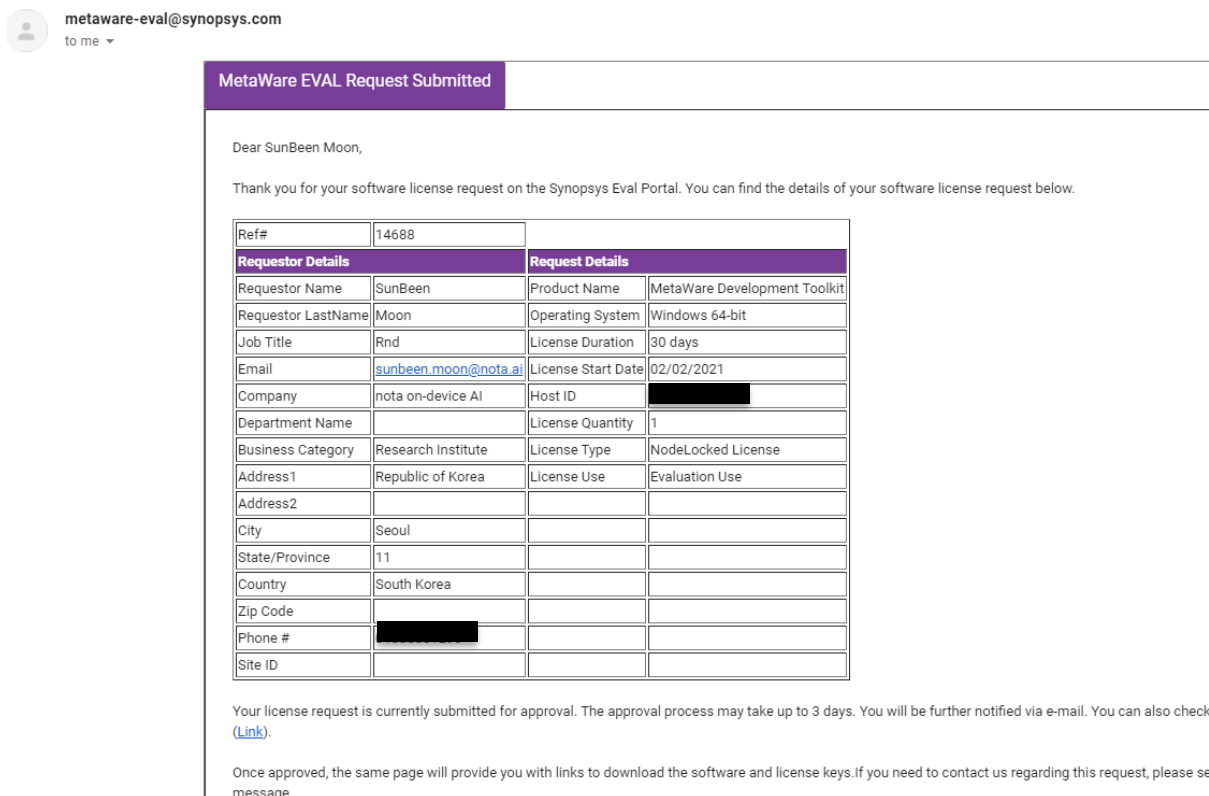


Fig 6. License 가 정상적으로 요청되었음을 확인하는 email

위와 같은 메일이 도착하면 License 신청이 완료된 것이다. License 는 신청한 지, 3 일 이내 나온다. 3 일 이 지나도록 License 확인 메일(Fig 7)이 오지 않을 경우, 위 메일을 회신하여 license 가 나오지 않는 이유를 문의해야 한다.

*(note)* License 유효기간이 신청일로부터 30 일이라 공고하나, License 기한이 따로 없다. License 가 허가된 후, Smart key 를 연결해주면 기한에 관계없이 사용할 수 있다. Software 를 설치하면서, ARC 라는 폴더가 생성된다. ARC 폴더 속 bin 폴더(주요 컴파일러가 담겨있는)는 기한이 없음으로 toolkit 를 계속 사용 가능하다.

#### 4. Download Software

License 가 나오면 아래와 같은 메일 두 개가 도착한다.

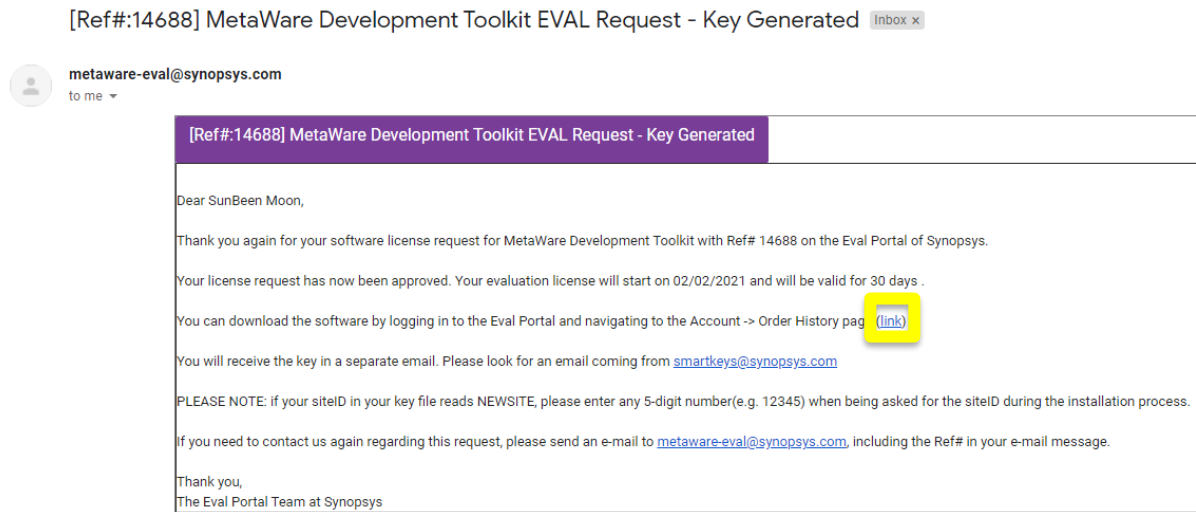


Fig 7. License 가 허가된 이메일

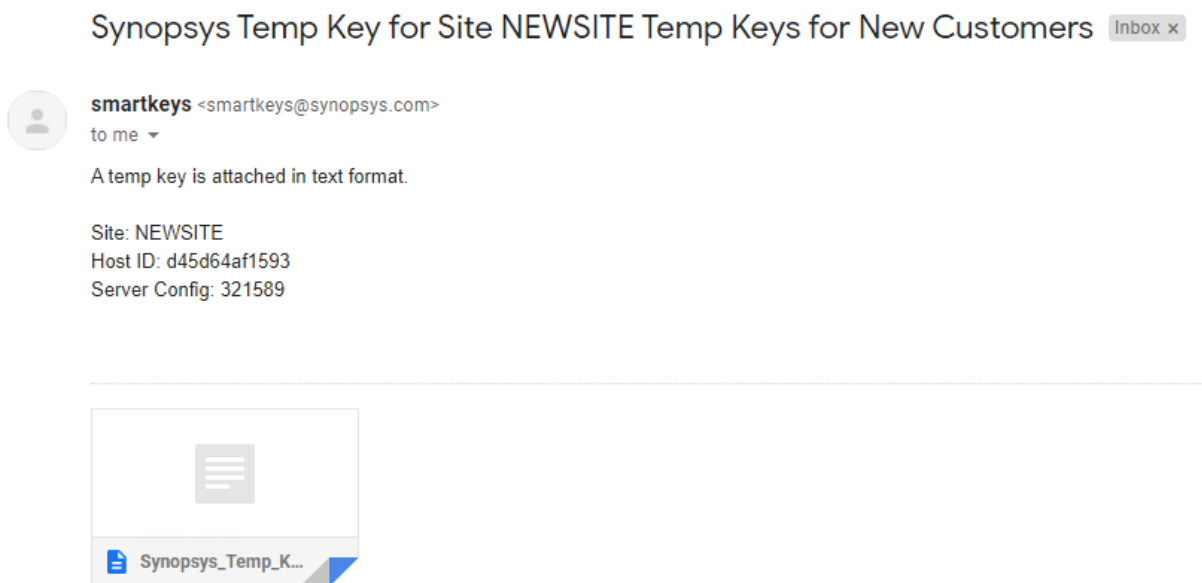


Fig 8. License 가 담긴 첨부파일

Fig 7 메일에서 링크를 타고 Fig 9 과 같은 order history 에 들어간다.

Fig 8 메일에서 Smart Key 를 다운로드 받는다.

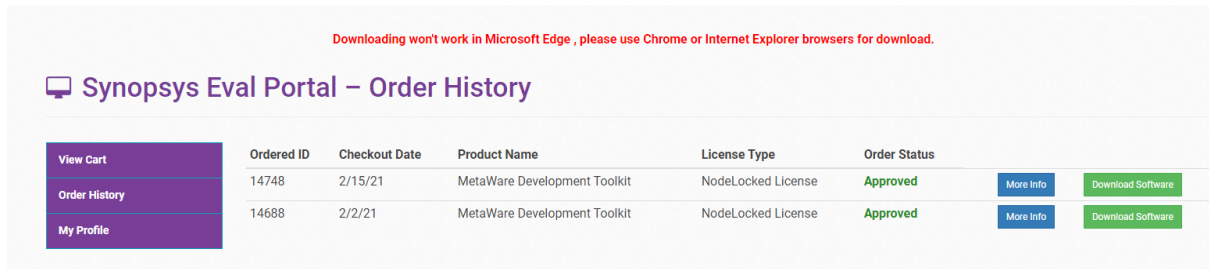


Fig 9. Order History 창

Order History 에 들어가서 Download Software 에서 다운을 받는다.

## 5. Install Software Tools

[https://arc-labs.readthedocs.io/en/latest/getting\\_started/getting\\_started.html#install-software-tools](https://arc-labs.readthedocs.io/en/latest/getting_started/getting_started.html#install-software-tools)

위 사이트를 참고하여 Software 를 설치, SmartKey License 를 Window 와 Linux version 이 다르게 설치한다. 또한, 환경변수를 추가한다.

## 2.3 Check make version

<https://github.com/tensorflow/tensorflow/blob/master/tensorflow/lite/micro/tools/make/targets/arm/README.md#make-tool>

위 깃허브 링크를 따라 make tool 의 version 을 확인한다.

*(note)* Window 의 경우, 아래 링크를 따라 gcc 를 설치하면 된다.

<https://ndb796.tistory.com/364>

*(note)* Linux 의 경우, update 와 upgrade 를 진행하면 된다.



## 2.4 Arduino IDE

[https://github.com/HimaxWiseEyePlus/bsp\\_tflu/tree/master/HIMAX\\_WE1\\_Plus\\_Arduino#--programming-the-himax-we1-evb-with-arduino--](https://github.com/HimaxWiseEyePlus/bsp_tflu/tree/master/HIMAX_WE1_Plus_Arduino#--programming-the-himax-we1-evb-with-arduino--)

아두이노의 경우, 위 깃허브를 참고하면 별 어려움없이 설치를 할 수 있다.

또한, TFLM 에서 제공하는 4 가지 예제를 실행할 수 있다.

Arduino IDE 를 이용한 4 가지 예제는 3 절의 Tensorflow Lite Micro 에 같이 소개된다.

## 2.5 Edge Impulse

### 1. Initial Configuration of Edge Impulse

- <https://docs.edgeimpulse.com/docs/getting-started>

Edge impulse사에서 제공하는 document 페이지이다.

- <https://docs.edgeimpulse.com/docs/himax-we-i-plus>

Edge impulse docs에 적혀 있는 순서대로 진행하면 된다. 아래 동영상을 따라 초기 설정을 하면 문제될 부분이 없다.

### 2. Edge Impulse CLI download or Toolchain

#### Software

- You'll need a build toolchain, either:
  - [ARC GNU Toolchain](#) (free).
  - [DesignWare ARC MetaWare Toolkit](#) (paid):
    - A valid ARC MetaWare license is required to build the firmware.
    - Having a local install of [DesignWare ARC MetaWare Toolkit](#). Make sure `ccac` is in your `PATH`, and that the licenses are in place.
  - Or, building with [Docker desktop](#).
- [Edge Impulse CLI](#) - to flash the firmware.

Fig 10. Edge Impulse를 구동시킬 수 option 목록

Fig 10과 같이 edge impulse를 구동하고 더 나아가 코드 단에서 작업이 가능하도록 환경설치가 가능한 옵션은 위와 같다. 이 중에서 두 가지 필자가 설치해 보았던 두 가지 옵션만 소개한다.

- ✓ Edge Impulse CLI

<https://docs.edgeimpulse.com/docs/cli-installation>

Document 순서를 따라 진행하면 크게 문제될 사항이 없다.

- ✓ Toolchain

<https://github.com/edgeimpulse/firmware-himax-we-i-plus>

git hub 내용을 따라, 진행하면 된다.

(note) 대부분은 기능을 사용하는 데에는 Toolchain까지 설치할 필요는 없으며 Edge Impulse CLI 사용으로 충분하다.

## 2.6 Issues

<Issue 1> Fig 10 의 옵션 중, DesignWare ARC MetaWare Toolkit 과 Edge Impulse CLI 설치한 후 테스트 실행했다. ARC GNU Toolchain 과 Docker 는 아직 테스트 진행하지 않았다.

<Issue 2> K-means Anomaly Detection Edge Impulse CLI 로는 error 가 발생한다. ARC GNU Toolchain 으로 설치해야 한다는 답변을 받았다.

## 3. Tensorflow Lite Micro

### Small Agenda

3.1 Introduction .....	<a href="#">20</a>
3.2 Examples .....	<a href="#">21</a>
3.2.1 Micro Speech .....	<a href="#">22</a>
3.2.2 Magic Wand .....	<a href="#">23</a>
3.2.3 Person Detection .....	<a href="#">24</a>
3.2.4 Handwriting .....	<a href="#">25</a>
3.2.5 Hello World .....	<a href="#">28</a>
3.3 Issues .....	<a href="#">29</a>

### 3.1 Introduction

1.2 절에서 언급한 것과 같이, TinyML 을 지원하기 위해 Tensorflow framework 에서 공식적으로 지원하는 툴이다. TFLM 의 제한적인 computing 과 memory 사용을 효율적으로 지원합니다. 기기내 학습이 지원되지 않으며, 몇몇 연산자들이 지원되지 않는다.

(TFLM 공식 문서) <https://www.tensorflow.org/lite/microcontrollers>

(TFLM 공식 깃허브) <https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/micro>

*(note)* 최신 Tensorflow version 은 Himax WE-1 의 MakeFile 을 지원하지 않음으로, Tensorflow 2.4.0 Version 을 다운로드 받아야 한다. (Himax WE-1 는 Tensorflow 에 2.3.0 부터 등장했다)

(Himax 사에서 TFLM 을 custom 한 깃허브) [https://github.com/HimaxWiseEyePlus/himax\\_tflm](https://github.com/HimaxWiseEyePlus/himax_tflm)

Himax WE-1 보드를 지원하는 TFLM 공식 깃허브 주소가 있고, Himax 보드를 제작한 제작사에서 TFLM 공식 깃허브를 따로 custom 해서 지원한다.

TFLM 공식 깃허브와의 차이점은

1. Himax 제작사 깃허브에는 handwriting 예제가 포함되어 있다.
2. TFLM 공식 깃허브보다 수월하게 build 될 수 있도록 관련 명령어가 간략하다.
3. Himax 제작사 깃허브에는 Hello World 예제가 없다.

첫 번째 차이점은, Himax 사에서 *pytorch→onnx→TF*로 변환하는 과정을 보여주기 위해 추가되었다. 이 외 결과 값, inference time 등에 두드러지는 차이는 없다.

## 3.2 Examples

예제는 아두이노, Tensorflow 공식 깃허브, Himax 제작사의 깃허브 3 가지 버전의 예제가 존재한다. 기초가 되는 모델은 Tensorflow 공식 깃허브 모델이다. 결과 값에 대한 inference time, 반응 속도 등의 차이는 없었다.

세 예제의 차이점은 아래와 같다.

1. Hello World 예제는 아두이노, Tensorflow 공식 깃허브에서 지원하다.
2. Handwriting(MNIST) 예제는 Himax 제작사의 깃허브에서만 지원한다.

✓ 아두이노 코드 예제

[https://github.com/HimaxWiseEyePlus/bsp\\_tflu/tree/master/HIMAX\\_WE1\\_Plus\\_Arduino#--programming-the-himax-we1-evb-with-arduino--](https://github.com/HimaxWiseEyePlus/bsp_tflu/tree/master/HIMAX_WE1_Plus_Arduino#--programming-the-himax-we1-evb-with-arduino--)

✓ Tensorflow 에서 나온 예제

<https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/micro/examples>

✓ Himax WE-1 사의 예제

[https://github.com/HimaxWiseEyePlus/himax\\_tflm](https://github.com/HimaxWiseEyePlus/himax_tflm)

*(note)* 아래 진행되는 예제 모두 빌드하는 데 note 혹은 issue 가 없다. 따라서 깃허브만 첨부한다. 문서 내용대로 진행하면 문제없다.

*(note)* 아두이노의 경우, 추가 document 없이 코드 예제 링크에서 해결할 수 있으므로 뒷 예제에서는 언급하지 않고 결과값만 첨부하였다.

### 3.2.1 1 – Micro Speech

- ✓ TFLM 공식 깃허브

[https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/micro/examples/micro\\_speech#deploy-to-himax-we1-evb](https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/micro/examples/micro_speech#deploy-to-himax-we1-evb)

- ✓ Himax WE-1 사의 TFLM 깃허브

[https://github.com/HimaxWiseEyePlus/himax\\_tflm/tree/master/tensorflow/lite/micro/examples/micro\\_speech](https://github.com/HimaxWiseEyePlus/himax_tflm/tree/master/tensorflow/lite/micro/examples/micro_speech)

(note) 아두이노의 경우, 추가 document 없이 코드 예제 링크에서 해결할 수 있으므로 결과값만 첨부하였다.

- ✓ 결과

	Yes	No	Unknow/Silence
Arduino	Heard yes (209) @50100ms	Heard no (215) @102500ms	Heard silence (204) @1200ms Heard unknown (206) @3200ms
TFLM	Heard yes (212) @133600ms	Heard no (204) @144100ms	Heard silence (220) @800ms Heard unknown (203) @2200ms
Himax	Heard yes (204) @8900ms	Heard no (209) @62600ms	Heard unknown (202) @2200ms Heard unknown (207) @5700ms

Table 2. Micro Speech 결과

#### <Analysis>

감지된 단어(yes, no, unknow, silence) 뒤 숫자는 점수이다. 200 점 이상일 경우에만 일치 항목이 유효한 것으로 간주된다. 뒷 부분의 ms 의 프로그램이 시작된 후의 초이다. 소리를 인식할 때까지 걸리는 시간을 의미하며, 평균적으로 2~3 초 동안 소리를 듣는다. 소리를 듣고 있을 때는, 보드 위 초록 빨강 LED 둘 다 깜박거린다. Yes 로 소리가 인식될 경우 보드 위 초록 LED 가 켜지고, NO 로 인식될 경우 보드 위 빨강 LED 가 켜진다. 테스트 결과, 작은 소리(asmr 정도의 크기)에도 좋은 테스트 결과를 받을 수 있다. Auduino 보다 Tensorflow 공식 깃허브와 Himax 사의 깃허브의 반응속도가 더 빨랐다.

### 3.2.2 2 – Magic Wand

- ✓ TFLM 공식 깃허브

[https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/micro/examples/magic\\_wand#deploy-to-himax-we1-evb](https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/micro/examples/magic_wand#deploy-to-himax-we1-evb)

- ✓ Himax WE-1 사의 TFLM 깃허브

[https://github.com/HimaxWiseEyePlus/himax\\_tflm/tree/master/tensorflow/lite/micro/examples/magic\\_wand](https://github.com/HimaxWiseEyePlus/himax_tflm/tree/master/tensorflow/lite/micro/examples/magic_wand)

(note) 아두이노의 경우, 추가 document 없이 코드 예제 링크에서 해결할 수 있음으로 결과값만 첨부하였다.

- ✓ 결과


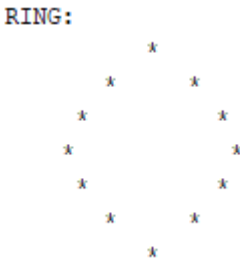
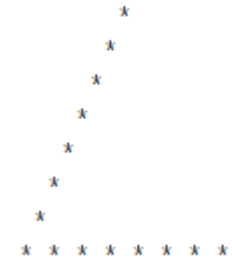






	Wing	Ring	Slope
Arduino			
TFLM			
Himax			

Table 3. Magic Wand 결과

#### <Analysis>

대체적으로 반응 속도가 밀리는 경향이 있다. 프로그램을 실행한 후, 각 행동(wing, ring, slope)를 취하였을 때 느린 반응이 공통적으로 찾을 수 있었다. “원인은 필자가 행동의 정확한 타이밍을

맞추지 못하여 실행하는 점이 문제, 혹은 보드와 host pc 간의 연결 문제 중 한 가지로 추정된다. “ 또한, wing 을 그렸으나 slope 가 그려지는 경우가 있었으며 그 반대로 slope 를 의도하였으나 wing 이 그려지는 경우도 있었다. 3 가지 중 slope > wing > ring 순으로 출력이 원활하게 나왔으며, Arduino 에서 다른 두 가지 경우보다 빠른 반응을 얻을 수 있었다.

### 3.2.3 3 – Person Detection

- ✓ TFLM 공식 깃허브

[https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/micro/examples/person\\_detection#running-on-himax-we1-evb](https://github.com/tensorflow/tensorflow/tree/master/tensorflow/lite/micro/examples/person_detection#running-on-himax-we1-evb)

- ✓ Himax WE-1 사의 TFLM 깃허브

[https://github.com/HimaxWiseEyePlus/himax\\_tflm/tree/master/tensorflow/lite/micro/examples/person\\_detection\\_experimental](https://github.com/HimaxWiseEyePlus/himax_tflm/tree/master/tensorflow/lite/micro/examples/person_detection_experimental)

(note) 아두이노의 경우, 추가 document 없이 코드 예제 링크에서 해결할 수 있음으로 결과값만 첨부하였다.

- ✓ 결과

	Person O	Person X
Arduin o	person score:116 no person score -116 person score:118 no person score -118 person score:118 no person score -118	person score:-36 no person score 36 person score:-27 no person score 27 person score:-34 no person score 34
TFLM	person score:117 no person score -117 person score:116 no person score -116 person score:116 no person score -116	person score:-50 no person score 50 person score:-43 no person score 43 person score:-53 no person score 53
Himax	person score:115 no person score -115 person score:114 no person score -114 person score:116 no person score -116	person score:-51 no person score 51 person score:-60 no person score 60 person score:-50 no person score 50

Table 4. Person Detection 결과

#### <Analysis>

표를 보는 방법은 person score 가 +이며 출력 값이 높으면 사람이 있는 것으로 간주한다. 반대로 no person score 가 +이며 출력 값이 person score 에 비해 높으며 사람이 없는 것으로 간주한다. 사람이 있을 경우, 초록색 LED 가 들어오고 사람이 없다고 판단되며 초록색 LED 등이 꺼진다. Himax 사에서 만든 person detection 의 경우 int8 로 설정되어 있다. 출력 속도가 빠르고 detection 이 잘 된다. 3 가지 경우 모두 detection 이 잘 된다.







### 3.2.4 4 – Handwriting

- ✓ Himax WE-1 사의 TFLM 깃허브

[https://github.com/HimaxWiseEyePlus/himax\\_tflm/tree/master/tensorflow/lite/micro/examples/handwriting](https://github.com/HimaxWiseEyePlus/himax_tflm/tree/master/tensorflow/lite/micro/examples/handwriting)

MNIST 데이터 셋을 활용한 모델이다. Himax사에서 *PyTorch --> ONNX --> TensorFlow --> TensorFlow Lite* 과정을 설명하고 예시를 들기 위해 만들었다. 위 링크의 깃허브의 readme 와 각 폴더를 들어가면 과정을 세세하게 알 수 있다.

- ✓ 결과

number	input	output
0		number [0]:65, [1]:-128, [2]:-82, [3]:-127, [4]:-128, [5]:-124, [6]:-127, [7]:-117, [8]:-128, [9]:-128, result:0
1		number [0]:-118, [1]:43, [2]:-122, [3]:-113, [4]:-113, [5]:-122, [6]:-128, [7]:-103, [8]:-126, [9]:-122, result:1
2		number [0]:-126, [1]:-127, [2]:46, [3]:-127, [4]:-128, [5]:-127, [6]:-128, [7]:-61, [8]:-124, [9]:-122, result:2
3		number [0]:-127, [1]:-127, [2]:-127, [3]:36, [4]:-128, [5]:-127, [6]:-128, [7]:-65, [8]:-128, [9]:-104, result:3







4		number [0]:-128, [1]:-128, [2]:-128, [3]:-126, [4]:52, [5]:-124, [6]:-128, [7]:-102, [8]:-128, [9]:-85, result:4
5		number [0]:-122, [1]:-126, [2]:-127, [3]:-118, [4]:-118, [5]:53, [6]:-118, [7]:-112, [8]:-112, [9]:-126, result:5
6		number [0]:-128, [1]:-128, [2]:-128, [3]:-128, [4]:-128, [5]:-125, [6]:124, [7]:-128, [8]:-128, [9]:-128, result:6
7		number [0]:-124, [1]:-124, [2]:-126, [3]:-109, [4]:-125, [5]:-124, [6]:-128, [7]:81, [8]:-125, [9]:-121, result:7
8		number [0]:-128, [1]:-128, [2]:-128, [3]:-116, [4]:-128, [5]:-108, [6]:-128, [7]:-128, [8]:95, [9]:-128, result:8
9		number [0]:-121, [1]:-127, [2]:-127, [3]:-121, [4]:-128, [5]:-121, [6]:-109, [7]:-124, [8]:-125, [9]:78, result:9

Table 5. Handwriting 결과

**<Analysis>**

Mnist data 를 training 시켜서 Himax WE-1 보드에 올린 결과이다. 0~9 까지 test 데이터 중에서 random 하게 하나를 선택하여서 input 값으로 넣은 후 output 을 보았다. 결과 표는 보는 방법은 result 에 검출된 숫자가 나오며, 검출된 숫자가 아닌 숫자의 경우 -로 표현이 된다.

테스트를 진행하기 위해서 보드에 부착된 카메라를 이용하여 화면에 띄워진 input 데이터를 직접 찍어야 한다.

Mnist data 를 training 시켜서 만드는 모델을 보드에 올리기 위해서 상당 부분을 축소한 부분이 있다. Edge impulse 에서 mnist data 로 training 했을 경우에는 모델의 크기가 큰 부분이 있어 보드위에서 결과의 오류가 많았다. Input data 를 인식하고 script 창에 출력하는 속도가 빠를뿐더러, 카메라로 인식하기 때문에 어느 부분이 인식되냐에 따라 결과 값이 계속해서 변한다. 모델 사이즈를 축소하면서 생기는 문제점이라고 추정된다.

### 3.2.5 5 – Hello World

Hello World 예제는 Tensorflow 공식 깃허브와 Himax 사의 깃허브에서는 지원하지 않는다. Hello World 폴더에 들어가면 Himax WE-1 보드 폴더가 존재하지 않는다.(지원하는 다른 예제의 경우, 예제 폴더 안에 Himax WE-1 보드 폴더가 존재한다). 아두이노 예제만이 존재한다.

✓ 결과

```
x_value: 1.0*2^-1, y_value: 1.0*2^-1
x_value: 1.0*2^-1, y_value: 1.0*2^-1
x_value: 1.0*2^0, y_value: 1.0*2^-1
x_value: 1.0*2^0, y_value: 1.0340478*2^0
x_value: 1.0*2^0, y_value: 1.0*2^-1
x_value: 1.0995567*2^1, y_value: 1.0*2^-1
x_value: 1.0*2^1, y_value: 1.0*2^-1
x_value: 1.0*2^1, y_value: 1.0*2^-2
x_value: 1.0*2^1, y_value: -1.0849024*2^-6
x_value: 1.0*2^1, y_value: -1.0509993*2^-2
x_value: 1.0*2^1, y_value: -1.0849024*2^-1
x_value: 1.0210171*2^2, y_value: -1.0*2^-1
x_value: 1.0995567*2^2, y_value: -1.0*2^-1
x_value: 1.0*2^2, y_value: -1.093378*2^0
x_value: 1.0*2^2, y_value: -1.0*2^-1
x_value: 1.0*2^2, y_value: -1.0*2^-1
```

Fig 11. Hello World 결과

#### <Analysis>

Arduino 의 경우, 보드 위에 모델이 잘 올라간다. 하지만, Arduino prompt 창과 teraterm prompt 창 모두 처음 출력에는 문제가 있다. Reset 버튼을 다시 눌러줘야 정상적으로 출력이 된다.

### 3.3 Issues

<Issue 1> Hello World 가 아두이노 prompt 외에 진행되지 않고 있다. Tensorflow 최신 공식 깃허브에는 Hello World 예제에서 Himax WE-1 보드를 지원하지 않는다.

<Issue 2> Magic wand 반응 속도가 느리다. 또한 모든 모양이 동일한 정확도를 가지고 인식되지 않는다.

<Issue 3> Handwriting 예제에서 카메라가 어느 부분을 인식하는 지 사용자가 파악하기 어려우며, prompt 의 출력속도가 매우 빠름으로 사용자가 인식하기 어렵다.

## 4. Edge Impulse

### Small Agenda

4.1 Introduction .....	<a href="#">30</a>
4.2 Description of the Edge Impulse Docs .....	<a href="#">34</a>
4.3 Projects' Tutorial .....	<a href="#">35</a>
4.4 Examples .....	<a href="#">60</a>
4.4.1 1 – Audio(given data : Keyword Spotting) .....	<a href="#">60</a>
4.4.2 2 – Audio(generated data 1 : '안녕') .....	<a href="#">64</a>
4.4.3 3 – Audio(generated data 2 : 'sol' vs 'alarm') .....	<a href="#">67</a>
4.4.4 4 – Image(generated data 1 : 'dog' vs 'cat') .....	<a href="#">71</a>
4.4.5 5 – Image(generated data 2 : 'mountain' vs 'sea') .....	<a href="#">76</a>
4.4.6 6 – Accelerator(generated data) .....	<a href="#">79</a>
4.4.7 7 – Image(MNIST) .....	<a href="#">83</a>
4.5 Issues .....	<a href="#">85</a>

### 4.1 Introduction

#### ✓ Introduction

Edge impulse 는 edge 기기에 대해서 end-to-end solution 을 진행하고 있는 소프트웨어이다. 센서의 신호를 edge device 에서 구동할 수 있도록, 모델 개발 자동화 툴을 제공하고 있다. 즉, data 수집, training, testing, deployment 까지 모든 과정을 지원한다. 'tinyML Awards 2021, Best Products & Best Innovation'에서 edge impulse EON(Edge Optimized Neural) 컴파일러는 기존 TF lite 의 memory 비효율성을 개선한 점에서 주목받았다. 위 award 에서 edge impulse 을 소개하고 해당 tutorial 을 진행했다. EON 컴파일러에 대한 주요 내용은 한 페이지 뒤에서 기술한다. TFLM 의 개발환경과 사용법을 기술한 도서 '초소형 머신러닝 tinyML'의 제 2 의 저자, Danial Situnayake 가 founding engineer 로 일하고 있다. 또한, edge impulse 사는 다양한 edge device 회사들과 파트너십을 맺고 있으며 그 중 STM 사와 Arm Cortex, Synopsys 사의 Himax 보드 등이 포함된다. 지속적으로 개발자와의 소통을 통해 기술을 발전시키고 있으며, 새로운 solution 이 지속적으로 업데이트되고 있다. 최근에는, NVIDIA Jetson Nano, Raspberry Pi 4 보드에서 linux 환경에서 object detection 을 지원한다.

(edge impulse 홈페이지) <https://www.edgeimpulse.com/>

(edge Impulse Performance metric 설명 페이지) <https://docs.edgeimpulse.com/docs/inference-performance-metrics>

✓ Edge Impulse with Himax WE-1 board

Edge impulse사에서 공식적으로 Himax WE-1 보드를 지원하며, 지속적으로 지원하는 툴을 업데이트하고 있다. 앞서 언급한 대로, 기존 Himax WE-1 보드를 사용하기 위해서는 license를 허가 받아야 했으며 일정 사용기간이 지날 경우 유료화가 진행됐다. 그러나, license가 필요로 하지 않은 오픈소스를 활용한 ARC GNU Toolchain 또한 지원하도록 업데이트되었다.

(edge impulse사에서 소개하는 Himax WE-1 보드 지원) <https://www.edgeimpulse.com/blog/himax-wei-plus>

(edge impulse사의 Himax WE-1 보드 dev guide) <https://docs.edgeimpulse.com/docs/himax-we-i-plus>

(edge impulse사의 Himax WE-1 보드 firmware 지원) <https://github.com/edgeimpulse/firmware-himax-we-i-plus>

*(note)* Himax WE-1 보드 firmware를 설치하지 않아도, edge-impulse-cli을 통해 모델을 구축하고 deploy할 수 있다.

✓ Edge Impulse Forum

깃허브보다는 edge impulse사에서 운영하는 Forum 활성화되어 있다. 깃허브에 질문을 남길 경우, 답변이 달리지 않으나, Forum 질문을 남길 경우 빠른 시간 내에 답변이 달린다. Forum을 통해 update 소식과 bug fix 소식을 빠르게 접할 수 있다.

(edge impulse 포럼) <https://forum.edgeimpulse.com/>

(edge impulse 깃허브) <https://github.com/edgeimpulse>

✓ EON Compiler

TFLM의 비효율적인 메모리 구조를 최적화해 주는 compiler이다. 실제 뒤 example에서 실행한 표를 참조하면 50%가량의 메모리 이득을 얻을 수 있다. 동일한 정확도를 유지하지만, TFLM에 비해 효율적인 메모리 이득을 보인다. EON 컴파일러를 활용하여 TFLM의 interpreter를 거치지 않고, .C++로 컴파일된 코드를 활용한다.

(edge Impulse사의 EON 컴파일러 소개 글) <https://www.edgeimpulse.com/blog/introducing-eon>

✓ Edge Impulse Developer vs Enterprise

	Developer	Enterprise
<b>Model</b>	Free subscription	Enterprise subscription
<b>Team collaboration</b>	X	O
<b>Projects</b>	5	Unlimited
<b>Compute time included</b>	25m	10000m
<b>Support</b>	Forum	Commercial support of service, SDKs and ML
<b>TinyML Impulse Designer</b>	O	O
- <b>Community Blocks</b>	O	O
- <b>Custom Blocks</b>	O	O (Hosted)
<b>Enterprise private blocks</b>	X	O
<b>Customer cloud data buckets</b>	X	O
<b>Data transformation</b>	X	O
<b>Dataset builder</b>	X	O
Royalty free	O	O
Live device data acquisition	O	O
Live classification	O	O
Model validation	O	O
Deployment	O	O

Table 6. Developer 와 Enterprise 의 차이 표

Free Developer 버전과 비용을 지불하는 Enterprise 버전의 사용계약은 위 표와 같다. Bold 하게 표시된 부분이 Free Developer 버전에서 제약이 되는 부분을 표시해 두었다. 이 중에서 설명이 필요한 부분에 대한 내용은 아래와 같으며, 4.1 절 Introduction 부분에서 설명하지 않은 내용은 4.2 절 Tutorial 에서 마저 설명한다.

\*Projects : Developer 의 경우 최대 5 개까지의 project 를 생성할 수 있으며, 초과할 수 없다. 삭제 후 새로 생성해서 사용할 수 있다.

\*Compute time included : Developer 의 경우 최대 25m 까지 compute 가 가능하다. 이 이상으로 초과될 경우, 프로그램이 멈추게 되며 Model 이 만들어 지지 않는다. 큰 데이터를 처리할 수 없으며, training cycle 을 일정 이상으로 올릴 수 없다. 이와 유사한 맥락으로 Learning rate 를 일정 수준 이상으로 증가시킬 수 없다.

\*Support : Developer 의 경우 Forum 을 통해 질문을 올릴 수 있으며 답변은 빠른 편이다. Enterprise 의 목적은 상업적인 모델을 구축하는 것으로 두고 있으므로 보다 전문적인 solution 을 응답 받을 수 있다.

\*Customer cloud data, Data transformation, Dataset builder : Developer 버전에서는 지원이 되지



많은 option 으로 Data 를 처리하는 부분이다. Cloud data 는 데이터를 cloud 로 제공할 수 있는 부분이며, data transformation 은 프로그램에 올라간 data 를 다운로드 받을 수 있는 기능이다. Dataset builder 는 예를 들어 오디오 데이터의 경우, 노이즈를 추가하는 방향으로 data 를 늘릴 수 있다.

(note) Free Developer 버전의 경우 upload 한 data 를 하나하나 각각 다운로드 가능하다. (일괄 다운로드 불가능)

## 4.2 Description of the Edge Impulse Docs

(edge impulse docs 주소) <https://docs.edgeimpulse.com/docs>

Edge Impulse 는 별도의 pdf 파일 혹은 정리된 자료를 사용하지 않고 위 사이트의 documents 를 활용한다. 4.3 절에 이어서 상세히 설명하지만, 뒷 부분에 기술되어 있지 않은 부분은 여기서 기술한다.

(note) 목차 중 Organizations 부분은 Developer 에게만 제공됨으로 확인하지 않아도 된다.

- ✓ Edge impulse 각 block parameter 에 대한 정보를 얻을 수 있는 주소

Edge impulse 는 각 parameter 에 대한 사용법에 대한 상세한 정보를 제공하지 않으므로 정보를 원하는 경우, 각 동영상을 시청하거나 edge impulse documents 목차 중 "tutorial"을 꼼꼼히 확인해 보아야 한다.

(coursera 강의) <https://www.coursera.org/learn/introduction-to-embedded-machine-learning#syllabus>

(tinyML video1) [https://www.youtube.com/watch?v=IRa\\_SH-3MSI&list=PL\\_tws4AXg7as8NoRkYLR2mOTzZoG5yQxw&index=5](https://www.youtube.com/watch?v=IRa_SH-3MSI&list=PL_tws4AXg7as8NoRkYLR2mOTzZoG5yQxw&index=5)

(tinyML video2) <https://www.youtube.com/watch?v=bonLeCwx0GU>

- ✓ 추후 update 할 내용

1. Tutorial 중 Parameter search with Python
2. Tutorial 중 Running your impulse locally
3. Tutorial 중 Processing blocks – Building custom processing blocks

위 세가지 사항은 추후 테스트 진행 후 update 할 내용이다. 3 번 내용에 대한 추가 적인 정보는 아래 링크에 있다.

<https://docs.edgeimpulse.com/reference#ingestion-api>

- ✓ Tips and Tricks

'Increasing model performance'에는 데이터 수가 부족하거나, training dataset 과 test dataset 의 불균형, 각 data 카테고리 별 데이터 수 불균형 문제점에 대해 다루고 있다.

'Data augmentation'은 data 를 처리해서 모델을 실제 환경에 적용했을 때 강점을 보일 수 있는 option 을 기술하고 있다. 이에 대한 내용은 뒤에 한 번 더 언급된다.

## 4.3 Projects' Tutorial

### Small agenda

#### Create New Project

1. Dashboard
2. Devices
3. Data Acquisition
4. Impulse Design
5. Retrain Model
6. Live Classification
7. Model Testing
8. Versioning
9. Deployment

### Create New Project

(edge impulse 시작 페이지) <https://studio.edgeimpulse.com>

로그인을 한 후 project 를 생성한다.

2 번부터는 옆 목차 순서대로 내용을 진행한다. 실제 end-to-end model 을 만드는 순서와는 다를 수 있다. 필자는 보통 device – data acquisition – impulse design – model testing – live classification – (retrain model) – deployment 순서로 진행하며 필요한 경우 dash board 의 option 들을 중간중간 활용한다.

### 1. Dashboard

Dashboard 에서 주목할 사항은 3 가지이다.

#### ✓ Download block output

Download block output			
TITLE	TYPE	SIZE	
Image training data	NPY file	5088 windows	📄
Image training labels	NPY file	5088 windows	📄
Image testing data	NPY file	258 windows	📄
Image testing labels	NPY file	258 windows	📄
Transfer learning model	TensorFlow Lite (float32)	2 MB	📄
Transfer learning model	TensorFlow Lite (int8 quantized)	716 KB	📄
Transfer learning model	TensorFlow Lite (int8 quantized with float32 input and output)	716 KB	📄
Transfer learning model	TensorFlow SavedModel	2 MB	📄

Download block output 은 실제 deep learning 에 전문 지식을 가진 개발자가 활용할 수 있도록, 각 Impulse 단에서 나온 결과를 다운로드 받을 수 있도록 되어 있다. 개발자가 다운로드 받아 새롭게 reproduce 할 수 있다.

(note) 단, Edge Impulse 는 아직 .h5 파일과 같이 이미 processing 된 모델을 업로드 하는 기능은 제공하지 않는다. 즉, Edge Impulse 에서 만들어진 파일을 다운로드 받을 수는 있으나 개발자가 제작한 모델을 업로드하여 결과 값을 받을 수는 없다.

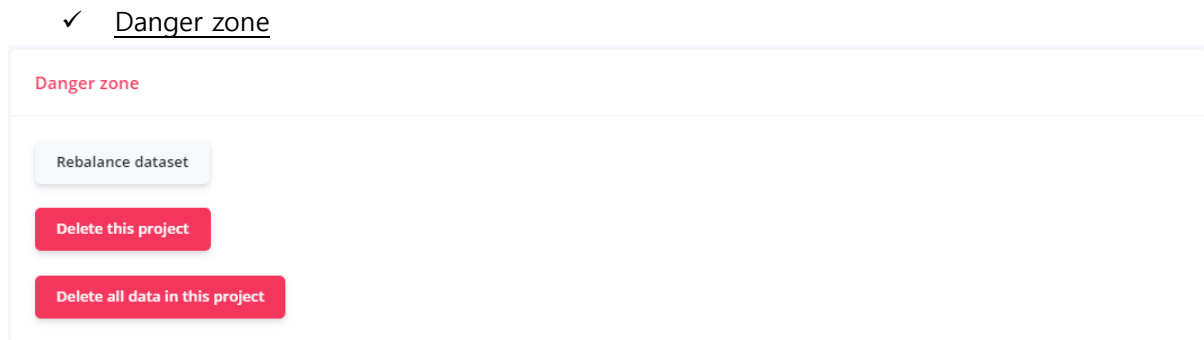


Fig 12. Danger Zone option

Danger Zone 에는 3 가지 option 이 있다.

- Rebalance dataset : training dataset 과 test dataset 의 비율을 조정해 주는 기능이다. 한 쪽으로 dataset 이 쏠릴 경우, model 의 성능이 떨어진다. Edge Impulse 자체적으로 dataset 의 비율을 맞춰주는 block 이다.
- Delete this project : Project 를 삭제하려면 위 block 을 활용하면 된다.
- Delete all data in this project : Project 내 모든 dataset 을 삭제할 수 있는 block 이다.

(note) 필자의 경험상, rebalance dataset 과 delete all data in this project 가 매우 유용하게 사용되었다. Training dataset 과 Test dataset 을 나누기 애매할 경우, 모든 data 를 training 에 업로드한 후, rebalance dataset 을 입력하면 손쉽게 dataset 을 나눌 수 있었다. 또한, impulse design 을 변경하지 않고, 데이터를 추가하거나 새로운 data 를 넣어 결과를 보고 싶은 경우, delete all data in this project 로 모든 dataset 을 삭제한 이후, retrain model 로 손쉽게 모델을 reproduce 할 수 있었다.

#### ✓ Collaborator

Developer 버전이라 하여도, 완성된 project 를 공유할 수 있는 기능이 있다. 함께 개발을 하는 환경을 조성하는 것이 아닌 완성된 project 를 project ID 로 공유하고, Forum 을 통해 아이디어를 나눌 수 있다.

## 2. Devices

연결된 device 를 확인할 수 있는 block 이다. Device 를 통해, data 를 모을 수도 있으며 Live Classification 을 통해 보드 위에 모델을 올리지 않고, Device 로 모델 테스트를 진행할 수 있다. Device 연결 방법은 아래와 같다. Cmd 창에서 진행한다. (Edge Impulse cli 를 활용해야 한다)

- Edge-impulse-daemon --clean

```
C:\Users\NOTA1204>edge-impulse-daemon --clean
Edge Impulse serial daemon v1.12.2
? What is your user name or e-mail address (edgeimpulse.com)? SunBeenMoon
? What is your password? [hidden]
```

Fig 13. Edge-impulse0daemon -clean prompt

위 명령어를 통해 기존 연결되어 있는 연결을 취하하고 새로운 연결을 진행한다.

- Select Com number

```
? Which device do you want to connect to? (Use arrow keys)
> COM3 (FTDI)
COM1 ((□?? ?□??))
```

Fig 14. Choose Com number

연결하고 싶은 com number 를 선택한다.

- Select Project

```
? To which project do you want to connect this device?
SunBeenMoon / catdoll
SunBeenMoon / accelerometer
SunBeenMoon / sound
SunBeenMoon / humanvoice
> SunBeenMoon / dogcat_final
```

Fig 15. Choose project number

원하는 프로젝트를 선택한다.

- Enter Device name

```
? What name do you want to give this device? Himax
[WS] Device "Himax" is now connected to project "dogcat_final"
```

Fig 16. Enter the device name

원하는 Device 이름을 입력한다.


Your devices <span>+ Connect a new device</span>					
These are devices that are connected to the Edge Impulse remote management API, or have posted data to the ingestion SDK.					
NAME	ID	TYPE	SENSORS	REMOTE M...	LAST SEEN
 DT0440X2	DT0440X2	HIMAX_WE_PLUS	Built-in accelerometer, Built-in microph...	<span style="color: green;">●</span>	Today, 15:17:58

Fig 17. Confirm the connected device

초록색 빛으로 device 가 연결되어 있음을 확인할 수 있다.

(note) cmd 창을 종료하면 device 와 연결이 끊긴다.

(note) 기존 연결했던 project 에 다시 연결하고 싶은 경우 'edge-impulse-daemon'만 입력해도 된다.  
(note) edge-impulse-daemon 에 대한 docs 는 <https://docs.edgeimpulse.com/docs/cli-daemon> 에 자세히 나와 있다.

### 3. Data Acquisition

여러가지 방법으로 data 를 모을 수 있으나, device 를 통한 data 수집, mobile phone 을 통한 data 수집, 기존 파일 upload 를 통한 data 수집 3 가지 방법을 소개한다.

#### ✓ Basic Check Point

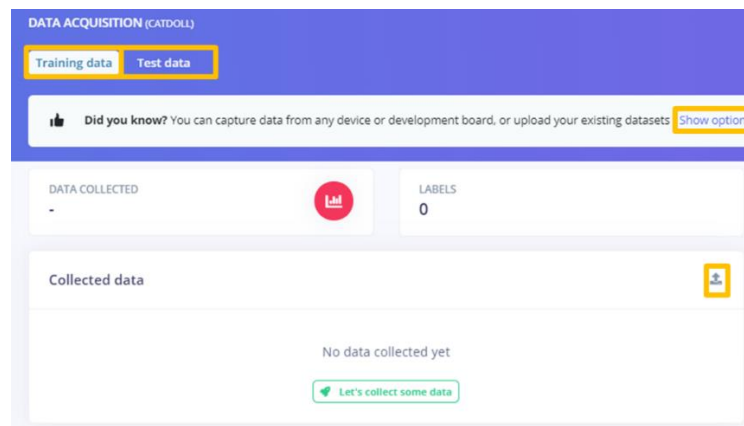


Fig 18. First Window to the Basic Check

맨 위 노랑 색 칸을 보면 Training data 와 Test data 를 구분해서 볼 수 있다. Show option 버튼을 선택하면, 여러가지 option 을 선택하여 data 를 upload 할 수 있다. Option 은 아래와 그림과 같다. 맨 오른쪽 업로드 button 은 기존 컴퓨터에 있는 data 를 업로드 할 수 있는 버튼이다.

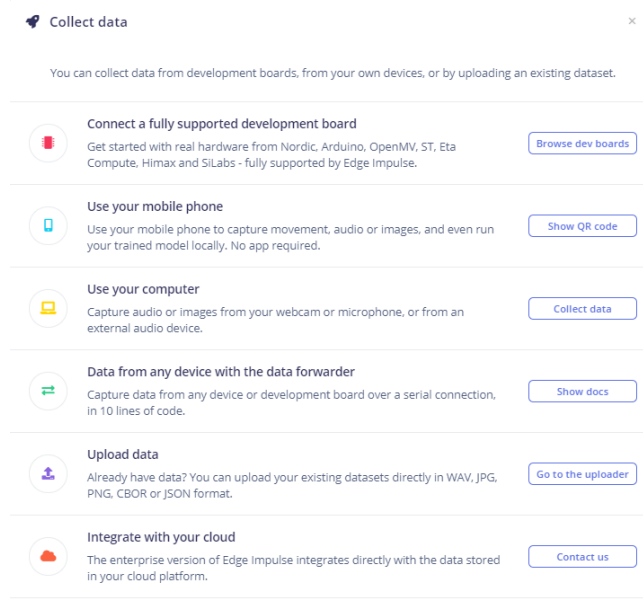


Fig 19. Window for choosing way to collect data

위 옵션에 대한 설명은 아래 절에서 이어서 진행한다.

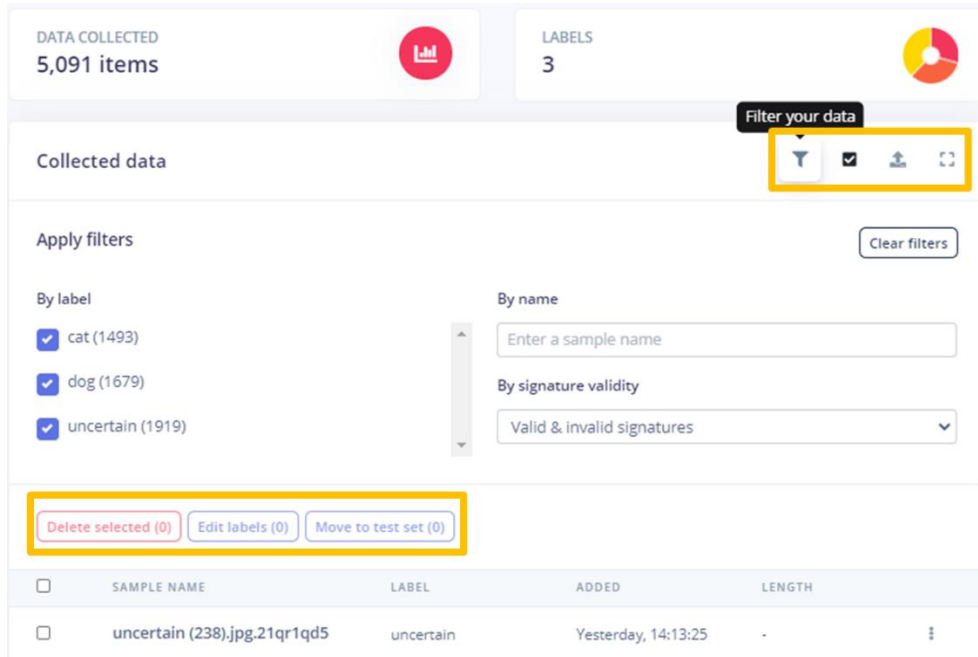


Fig 20. Data Processing Option 1

몇 개의 data 가 모였는 지 알 수 있으며, 원형차트로 데이터의 비율을 확인할 수 있다.

첫 번째 노랑색 박스 버튼을 순서대로 설명하면 다음과 같다. 처음 버튼은 'filter your data'로 label 에 따라 data 를 선택할 수 있다. 그 다음 버튼은 'select multiple items' 여러 데이터를 선택하여 두 번째 노랑 박스의 버튼들을 일괄적으로 처리할 수 있다.

두 번째 노랑색 박스 data 를 삭제할 수 있으며, 라벨을 변경하고 test dataset 혹은 training dataset 으로 옮길 수 있다.

(note) data 의 비율이 유사해야 좋은 모델을 얻을 수 있다.

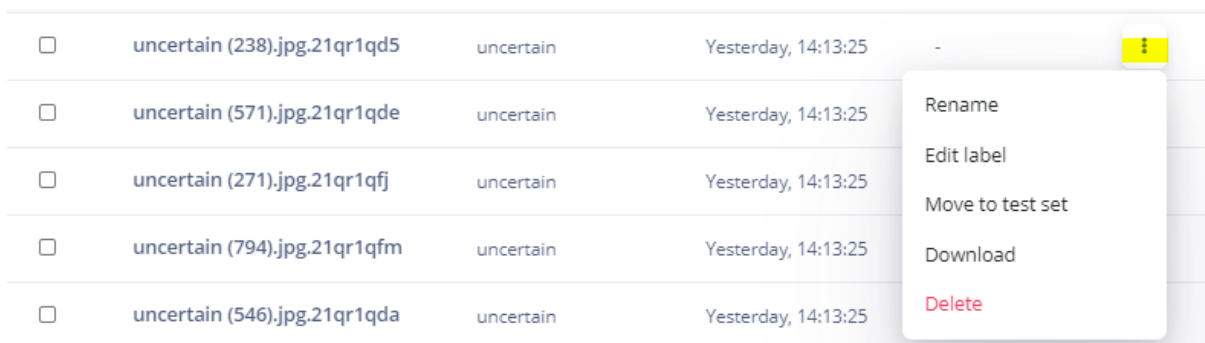


Fig 21. Data Processing Option 2-1

Data 의 맨 오른쪽 버튼을 누르면 다양한 option 들이 나온다. 꼭 필수적으로 확인해야 한다. Data 의 특성에 따라 처리해야하는 경우가 존재하기 때문이다. 뒤 example 에 언급할 예정이다.

(note) sound data 를 업로드 하는 경우 필수적으로 확인해야 한다. Sound data 를 1s 씩 데이터를

처리해야 하기 때문이다.

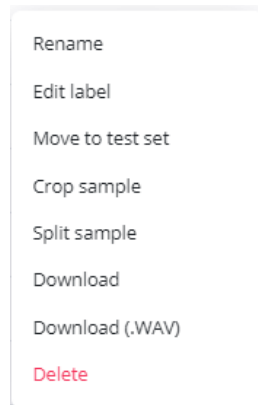


Fig 22. Data Processing Option 2-2

✓ Device 를 통한 data 수집

**Record new data**

Device ⓘ

Himax


Label

Label name

Sensor

Camera (128x96)

Camera feed



Start sampling

Fig 23. Window of Connected Device

노랑색 네모 친 Sensor (Himax board 기준)

- Built-in accelerometer
- Build-in microphone
- Camera(128\*96)
- Camera(320\*240)
- Camera(640\*480)

원하는 data type 에 따라 sensor 를 선택한 후 'Start sampling'을 진행하면 된다.

(note) accelerometer 와 microphone 은 board 에서 인식할 수 있는 Sample length 와 Frequency 가



정해져 있으므로 data 수집이 되지 않을 경우, 경고 메시지를 확인하며 진행하면 된다.

✓ mobile phone 을 통한 data 수집

'show option'을 선택한 이후, 'use your mobile phone'에서 'Show QR code'를 선택한다. QR code 를 인식하면, 해당 웹사이트가 뜬다.

혹은, 핸드폰 인터넷 도메인 창에 'smartphone.edgeimpulse.com'이라 검색한다.

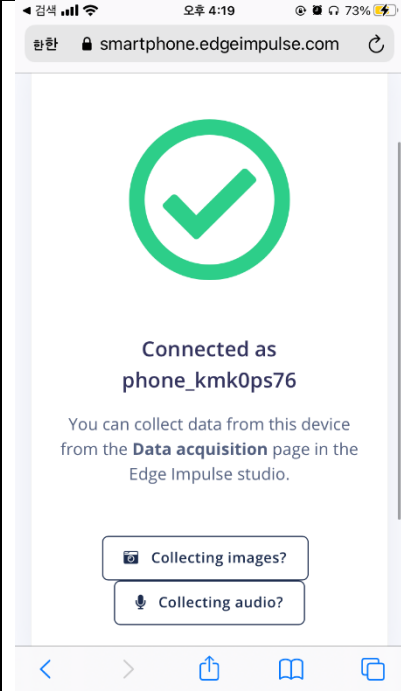
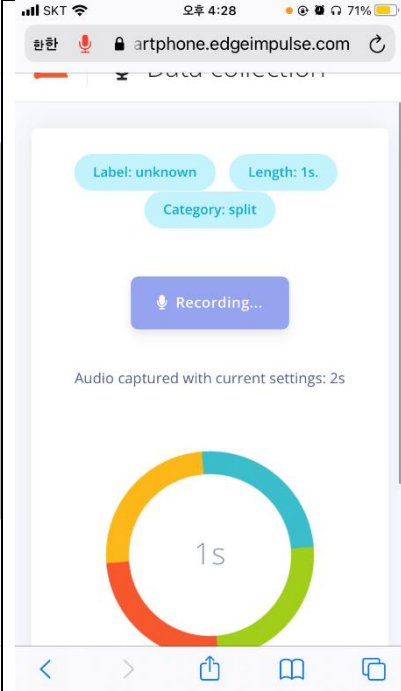
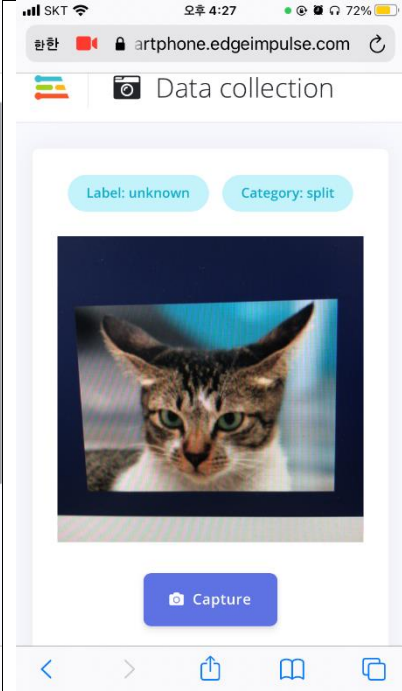
		
웹사이트 처음 페이지	Audio data 수집	Image data 수집

Table 7. Mobile Phone Data Collect Option

웹사이트 처음 페이지에서 image 를 수집할 지, audio 를 수집할 지 선택한 후 데이터를 모으면 된다. Audio data 수집의 경우, Edge Impulse 웹사이트에서와 동일하게 label 을 설정할 수 있으며 sample length 와 category 를 설정할 수 있다. Image data 수집의 경우도 유사하다. Accelerator 로 motion data 수집이 가능하다.

(note) category 의 경우, split : training data 와 test data 를 자율적으로 Edge Impulse 프로그램 상에서 나누어서 upload 한다. Training 과 Test 는 각각 category 로 data 가 upload 된다.

(note) mobile phone 이 꼭 PC 에 연결되어 있지 않아도 진행 가능하다.

(note) Mobile Phone 으로 연결한 웹사이트 맨 아래 쪽을 보면, 'Switch to data collection mode', 'Switch to classification mode' 두 가지 option 을 선택할 수 있다. Live Classification 으로 model test 를 진행 가능하다. 이와 관련된 내용은 Live Classification 에서 설명한다.

✓ 기존 파일 upload 를 통한 data 수집

기존 파일을 upload 하는 방법은 2 가지 이다.

1. Edge Impulse 웹사이트를 통한 upload



Fig 24. Upload Data Option 1 button

노랑색 네모친 button 을 누르면 아래와 같은 창이 뜬다.

**Upload existing data**

You can upload existing data to your project in the [Data Acquisition Format](#) (CBOR, JSON), or as WAV, JPG or PNG files.

Select files

선택된 파일 없음

Upload into category

☒ Automatically split between training and testing ?

☐ Training

☐ Testing

Label

☒ Infer from filename ?

☐ Enter label:

Fig 25. Upload Data Option 1 description

파일 format 이 .CBOR(action, 움직임이 있는 데이터), .JSON, .WAV, .JPG, .PNG 파일 형식 외에 업로드가 되지 않음으로 유의해서 data 를 upload 한다.

2. Edge Impulse CLI 를 통한 upload

- Edge-impulse-uploader --clean

```
C:\Users\NOTA1204\Desktop\dataset\faucet>edge-impulse-uploader --clean
Edge Impulse uploader v1.12.2
? What is your user name or e-mail address (edgeimpulse.com)? SunBeenMoon
? What is your password? [hidden]
Endpoints:
  API:      https://studio.edgeimpulse.com/v1
  Ingestion: https://ingestion.edgeimpulse.com

Upload configuration:
  Label:      Not set, will be inferred from file name
  Category:   training
Requires at least one argument (a .wav, .cbor, .json, .jpg, .jpeg or .png file)
```

Fig 26. Upload Data Option 2

Cmd 창에서 위 명령어를 통해 uploader 연결을 새로 취한다.

- Edge-impulse-uploader -category 옵션 -label 옵션 파일위치/.파일형식

--category 의 옵션은 split, training, testing 으로 나뉜다. Split 은 프로그램에서 알아서 test data 와 train data 를 나눠준다. Training 은 training dataset 으로 testing 은 Testing data 으로 나뉜다.

--label 옵션은 labeling 을 하고 싶으면 내용을 적으면 되고, 파일 이름 자체가 labeling 이 되어 있다면 -label 옵션은 적지 않아도 된다.

```
C:\Users\NOTA1204\Desktop\dataset>edge-impulse-uploader --category training faucet/training/*.cbor
Edge Impulse uploader v1.12.2
Endpoints:
  API:      https://studio.edgeimpulse.com/v1
  Ingestion: https://ingestion.edgeimpulse.com

Upload configuration:
  Label:    Not set, will be inferred from file name
  Category: training

? To which project do you want to upload the data? SunBeenMoon / faucet
[ 1/17] Uploading faucet\training\noise.2uv2e67s.cbor OK (2796 ms)
[ 2/17] Uploading faucet\training\noise.3.uv2m03a.cbor OK (3036 ms)
[ 3/17] Uploading faucet\training\noise.6uv3o3kk.cbor OK (3354 ms)
[ 4/17] Uploading faucet\training\faucet.2uve69m6.cbor OK (3421 ms)
[ 5/17] Uploading faucet\training\noise.12uv593lj.cbor OK (3492 ms)
[ 6/17] Uploading faucet\training\noise.5uv3c56p.cbor OK (3619 ms)
[ 7/17] Uploading faucet\training\faucet.0v1k8j5u.cbor OK (3579 ms)
[ 8/17] Uploading faucet\training\noise.12v1lfgbh.cbor OK (3799 ms)
[ 9/17] Uploading faucet\training\noise.12v1vurlm.cbor OK (3823 ms)
[10/17] Uploading faucet\training\noise.7uv4etlp.cbor OK (3854 ms)
[11/17] Uploading faucet\training\noise.14v1jdiit.cbor OK (4032 ms)
[12/17] Uploading faucet\training\faucet.0uv98b40.cbor OK (4062 ms)
[13/17] Uploading faucet\training\faucet.0v1kv0nt.cbor OK (4121 ms)
[14/17] Uploading faucet\training\faucet.1uv85g0u.cbor OK (4115 ms)
[15/17] Uploading faucet\training\faucet.2v1mlqlu.cbor OK (4173 ms)
[16/17] Uploading faucet\training\faucet.0uv72luj.cbor OK (4239 ms)
[17/17] Uploading faucet\training\noise.13v1livqjs.cbor OK (4218 ms)

Done. Files uploaded successful: 17. Files that failed to upload: 0.
```

Fig 27. Upload Data Option 2 Process

(note) edge-impulse-uploader 관련 docs 는 <https://docs.edgeimpulse.com/docs/cli-uploader> 에 자세히 나와 있다.

(note) data 를 수집할 때, 분류하기 원하는 dataset 뿐만 아니라, uncertain data 를 필수적으로 넣어주어야 한다. 예를 들면, Image classification 의 경우 분류하고자 하는 data 를 각 30 장 씩 넣고 uncertain data 를 50 장(약 1.5 배)가량 넣어주어야 한다. 또 다른 예로는 audio classification 의 경우, 분류하고자 하는 data 의 length 가 10 분이면 모르는 데이터 또한 10 분 이어야 한다.

#### 4. Impulse Design

Model 을 design 할 수 있는 block 이다. 보통 4 가지 단계로 구성된다. 첫 번째 block 은 data type 에 따라 Edge Impulse 에서 지정해 준다. 첫 번째 block 'Images'와 'Time series data'로 구성되어 있다. 네 번째 block 은 고정이다.

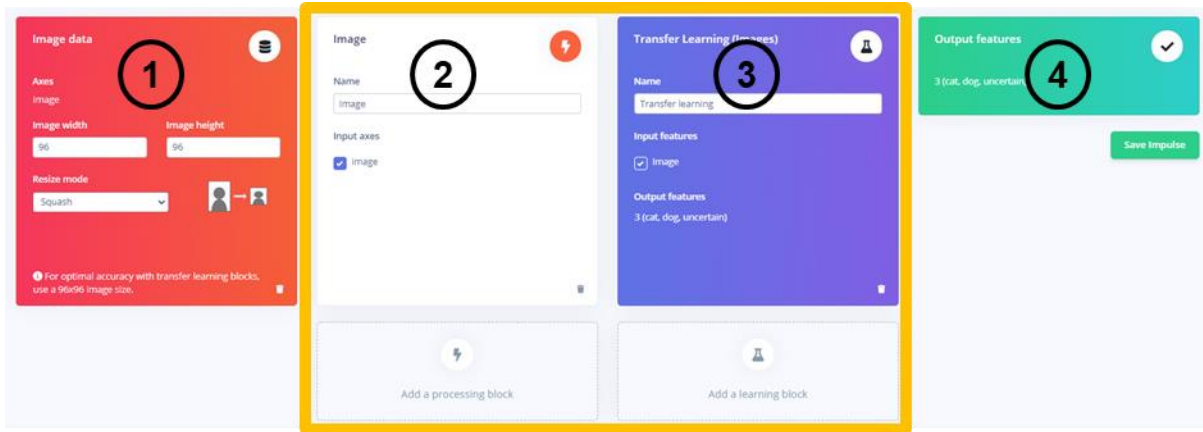


Fig 28. Create Impulse Block

위 사진은 Image model 의 예시이다. 노랑색 네모를 친 부분의 Impulse 를 사용자가 원하는 대로 지정할 수 있다. 각 block 에 대한 설명은 아래와 같다.

(note) 중간에 block 을 2 개 말고 추가로 지정할 수 있으나, 추가하지 않아야 한다. Edge Impulse 에서도 block 을 4 개만 사용하는 것을 추천한다.

✓ 두 번째 block

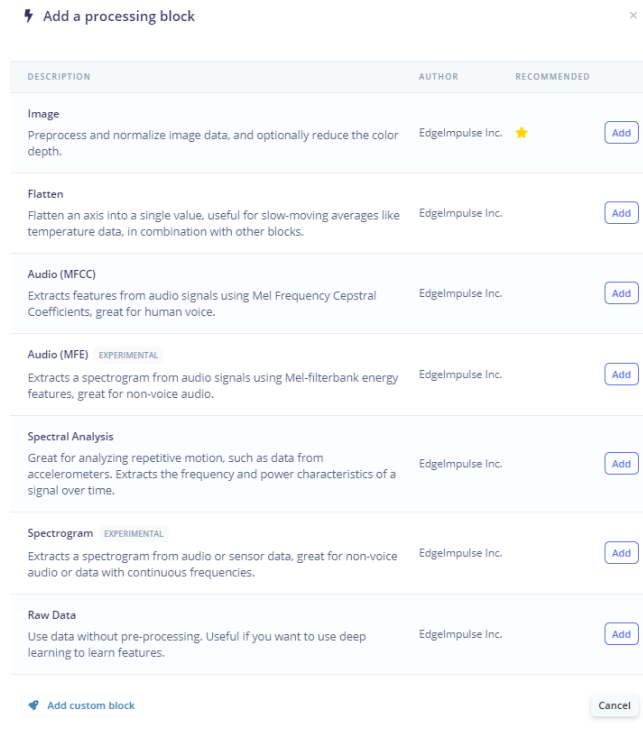


Fig 29. 2<sup>nd</sup> block Options

그림에서 확인할 수 있다시피, Edge Impulse 에서 가장 좋은 Impulse block 을 노랑색 별로 추천해준다. 또한, 밑의 option 에 대한 설명을 읽어보면 추천해준 block 이 가장 적합하다는 것을 알 수 있다. 각 Block 에 대한 설명은 아래에서 간략히 서술하며, 이 외 사용한 block 의 자세한 사항은 4.4 절의 example 에서 기술하기로 한다.

- Image : 이미지 파일 전처리 block
- Flatten : 온도 데이터와 같이 천천히 변하는 항목을 평균 값을 내어 평평하게 만드는 block
- Audio(MFCC) : 사람 목소리에 적합하면 audio 파일 전처리 block
- Audio(MFE) : 사람 목소리 외의 소리에 적합하며 audio 파일 전처리 block
- Spectral Analysis : gesture 와 같은 accelerometer 파일 전처리 block
- Spectrogram : 사람 목소리 외의 소리에 적합하며 audio 파일 혹은 다른 센서 파일 전처리 block
- Raw Data : 따로 전처리 과정을 거치지 않고, 데이터 그대로 사용하는 block
- Add Custom block : 사용자가 직접 코딩을 통해 customize 해서 사용할 수 있는 block 이며 사용법은 추후에 첨부할 예정(<https://docs.edgeimpulse.com/docs/custom-blocks>)

Block 에 속한 세부적인 parameter 에 대한 edge impulse 사의 상세한 설명은 나와 있지 않다. 따라서 example 에서는 필자가 조정한 parameter 에 대한 간략한 기술과 각 parameter 에 대한 정보를 얻을 수 있는 링크를 첨부한다.

(note) 필자가 테스트를 진행해본 결과 Edge Impulse 에서 추천하는 block 이 가장 좋은 model

output 을 보여주었다.

(note) 단, Audio 의 경우 'MFCC'와 'Spectrogram' 모두 좋은 결과를 보였다.

(note) 필자는, 'Flatten' 'Raw Data'외 block 들을 모두 사용해 보았으며 example 에 사용 예를 기록하고 있다.

(note) 필자는, 앞서 언급한 대로 edge impulse 사에서 각 block 속 parameter 에 대한 상세한 설명을 제공하고 있지 않음으로 대부분 초기 설정 값 그대로 사용하였다.

### ✓ 세 번째 block

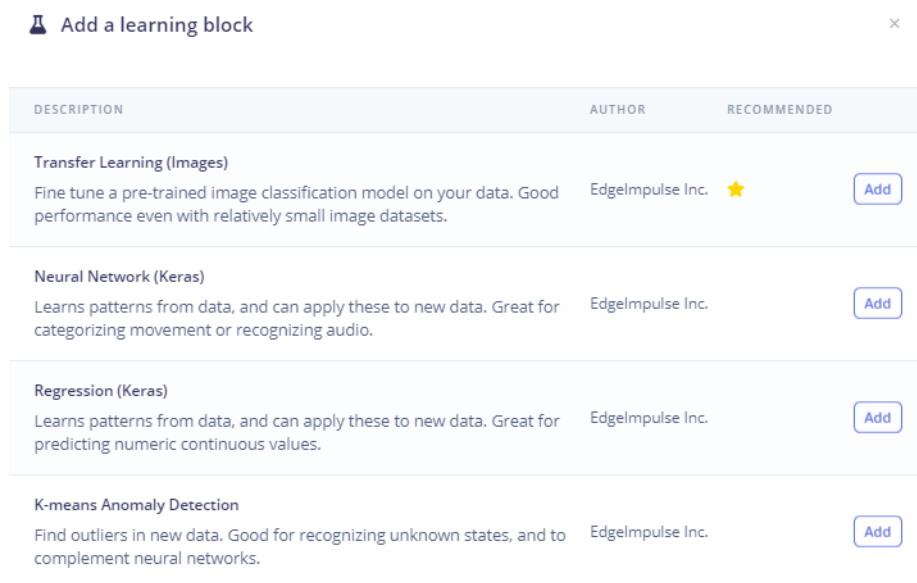


Fig 30. 3<sup>rd</sup> Block Options

두 번째 block 과 동일하게 가장 좋은 impulse 를 노랑색 별을 통해서 추천해준다. 현재, 제공되고 있는 방법은 4 가지 방법이다.

- Transfer Learning(Images)
- Neural Network(Keras)
- Regression(Keras)
- K-means Anomaly Detection

각 block 에 대한 설명은 아래와 같다.

(note) 이 중에서 Regression(Keras)와 K-means Anomaly Detection 는 새롭게 update 된 block 이며, 아직 사용하기에는 error 가 해결되지 않았음으로 추후에 업데이트를 진행할 예정이다.

### 3 Different Editing Methods(K-means Anomaly Detection 제외)

Transfer Learning 과 Neural Network 는 3 가지 방법으로 사용할 수 있다. 아래 그림 예시는 Transfer Learning 사진으로 진행한다.

#### - Visual(simple) mode

Fig 31. Impulse Design Visual mode

초기 setting 되는 모드이다. 딥러닝을 모르는 타 엔지니어 쉽게 사용할 수 있는 api 로 구성되어 있다. 오른쪽 위 노랑색 option 을 눌러 아래 두 가지 방법을 실행시킬 수 있다.

#### - Keras(expert) mode

```

1 import tensorflow as tf
2 from tensorflow.keras import Model
3 from tensorflow.keras.models import Sequential
4 from tensorflow.keras.layers import Dense, InputLayer, Dropout, Conv2D, Flatten, Reshape, MaxPooling2D,
5   BatchNormalization, Conv2D, GlobalMaxPooling2D, Lambda
6 from tensorflow.keras.optimizers import Adam, Adadelta
7 from tensorflow.keras.losses import categorical_crossentropy
8
9 # Load best model from initial training
10 sys.path.append('./resources/libraries')
11 import ei_tensorflow.training
12
13 INPUT_SHAPE = (96, 96, 3)
14
15 base_model = tf.keras.applications.MobileNetV2(
16     input_shape=INPUT_SHAPE, alpha=0.05,
17     weights='./transfer-learning-weights/edgeimpulse/MobileNetV2_0.05_96x96_color_bsize_64_lr_0.05_epoch_574',
18     include_top=True
19 )
20
21 base_model.trainable = False
22
23 model = Sequential()
24 model.add(InputLayer(input_shape=INPUT_SHAPE, name='x_input'))
25 # Don't include the base model's top layers
26 last_layer_index = -3
27 model.add(Model(inputs=base_model.inputs, outputs=base_model.layers[last_layer_index].output))
28 model.add(Dense(8))
29 model.add(Dropout(0.1))
30 model.add(Flatten())
31 model.add(Dense(classes, activation='softmax'))
32
33 model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.0005),

```

Fig 32. Impulse Design Keras(Expert) mode

코드를 사용할 수 있는 사용자가 손쉽게 사용할 수 있도록 python 코드로 제공한다.

- Ipython notebook

Jupyter notebook 형태로 코드가 제공되며, jpyther notebook 을 이용하여 사용자가 customize 해서 모델을 조정할 수 있다.

## Transfer Learning(Images)

Neural Network settings

Training settings

Number of training cycles ?

20

Learning rate ?

0.0005

Data augmentation ?


☐

Minimum confidence rating ?

0.60

Neural network architecture

Input layer (9,216 features)



MobileNetV2 0.35 (final layer: 16 neurons, 0.1 dropout)

Choose a different model

Output layer (3 features)

Fig 33. Transfer Learning prompt

Transfer Learning 의 base 은 'MobileNetV2'로 진행이 된다. 조절 가능한 parameter 은 아래에 기술한다.

- Training Setting
  - Number of training cycles : 에폭 개수를 조정, 최대로 100 개까지 설정할 수 있다.
  - Learning rate : 최소 0.0001 까지 설정할 수 있다.
  - Data augmentation : 데이터를 가공해서 더 높은 accuracy 와 overfitting 을 방지하는 기능( <https://docs.edgeimpulse.com/docs/data-augmentation> )
  - Minimum confidence rating : Threshold 이다. Threshold 밑에 값이 나오면 uncertain 결과 값이 나온다.
- Model Setting
  - MobileNetV2 0.35 / MobileNetV2 0.1 / MobileNetV2 0.05 / MobileNetV2 0.25 : 모델



결과 값 사이즈에 따라 option 이 다르다. 또한, default neuron 개수가 다르다.

- Number of neurons : 마지막 layer 에 위치할 neuron 의 개수를 조절할 수 있다. 각 model 별로 default 인 neuron 개수가 다르다.
- Dropout rate : 0 과 1 사이 값을 선택해야 하며, 마지막 layer 에서 dropout 할 값을 설정할 수 있다.

초기 Number of training cycles 와 learning rate 는 edge impulse 사에서 지정해준다. 대체로 초기 값을 변경하지 않는 편이 더 좋은 결과를 야기했다.

*(note)* Data Augmentation 에 대한 forum 답변이다. 필자는 image 데이터로 실행 시 좋은 결과를 얻지 못하였다.

<https://forum.edgeimpulse.com/t/use-data-augmentation/1619/2>

### Neural Network(Keras)

Neural Network settings

Training settings

Number of training cycles ?

20

Learning rate ?

0.0005

Minimum confidence rating ?

0.60

Neural network architecture

Input layer (9,216 features)

2D conv / pool layer (32 neurons, 3 kernel size, 1 layer)

2D conv / pool layer (16 neurons, 3 kernel size, 1 layer)

Flatten layer

Dropout (rate 0.25)

Add an extra layer

Output layer (3 features)

Fig 34. NN(Keras) Prompt

Neural network 는 Transfer Learning 과 동일하게 Number of training cycles, Learning rate, Minimum confidence rating 을 조정할 수 있다. (Data Augmentation option 은 없다)

사진과 같은 layer 는 edge impulse 가 자체적으로 모델을 제시한다. Image dataset, audio dataset 등 dataset 에 맞추어서 NN 모델을 제시한다. 모델이 layer 는 사용자가 직접 바꾸거나 추가할 수 있다.

추가하거나 변경할 수 있는 option 은 아래와 같다.

LAYER TYPE	
<b>Dense</b> Fully connected layer, the simplest form of a neural network layer. Use this for processed data, such as the output of a spectral analysis DSP block.	<a href="#">Add</a>
<b>1D Convolution / pooling</b> Learn features that take spatial information into account along a single dimension. Use this for raw data, or for DSP blocks that output spatial data, such as the MFCC block.	<a href="#">Add</a>
<b>2D Convolution / pooling</b> Learn features that take spatial information into account along two dimensions. Use this for raw data, or for DSP blocks that output spatial data, such as the MFCC block.	<a href="#">Add</a>
<b>Reshape</b> Turn one-dimensional data from a DSP block into multi-dimensional data. Use this as an input to a convolutional layer. Use this for deep learning on raw data, or to process MFCC output.	<a href="#">Add</a>
<b>Flatten</b> Flatten multi-dimensional data into a single dimension. You need to flatten data from a convolutional layer before returning.	<a href="#">Add</a>
<b>Dropout</b> Reduce the risk of a model overfitting your dataset by randomly cutting a fraction of network connections during training. Can be helpful if your model's training performance is better than its validation performance.	<a href="#">Add</a>

Fig 35. NN(Keras) Options

위 그림처럼 6 가 option 이 있으며, 사용자의 편의에 따라 조정 가능하다.

## K-means Anomaly Detection

Anomaly detection settings

Cluster count

Minimum score before tagging as anomaly ?

Axes [Select all axes](#)

☒ accX RMS ★
 ☐ accX Peak 1 Freq
 ☐ accX Peak 1 Height
 ☐ accX Peak 2 Freq
 ☐ accX Peak 2 Height
 ☐ accX Peak 3 Freq
 ☐ accX Peak 3 Height
 ☐ accX Spectral Power 0.1 - 0.5
 ☐ accX Spectral Power 0.5 - 1.0
 ☐ accX Spectral Power 1.0 - 2.0
 ☐ accX Spectral Power 2.0 - 5.0
 ☒ accY RMS ★
 ☐ accY Peak 1 Freq
 ☐ accY Peak 1 Height
 ☐ accY Peak 2 Freq
 ☐ accY Peak 2 Height
 ☐ accY Peak 3 Freq
 ☐ accY Peak 3 Height
 ☐ accY Spectral Power 0.1 - 0.5
 ☐ accY Spectral Power 0.5 - 1.0
 ☐ accY Spectral Power 1.0 - 2.0
 ☐ accY Spectral Power 2.0 - 5.0

☐ accY Spectral Power 0.1 - 0.5
 ☐ accY Spectral Power 0.5 - 1.0
 ☐ accY Spectral Power 1.0 - 2.0
 ☐ accY Spectral Power 2.0 - 5.0
 ☒ accZ RMS ★
 ☐ accZ Peak 1 Freq
 ☐ accZ Peak 1 Height
 ☐ accZ Peak 2 Freq
 ☐ accZ Peak 2 Height
 ☐ accZ Peak 3 Freq
 ☐ accZ Peak 3 Height
 ☐ accZ Spectral Power 0.1 - 0.5
 ☐ accZ Spectral Power 0.5 - 1.0
 ☐ accZ Spectral Power 1.0 - 2.0
 ☐ accZ Spectral Power 2.0 - 5.0

Fig 36. K-means Prompt

위 사진은 accelerator 를 이용한 gestures 모델 impulse 이다. 조절할 수 있는 parameter 는

- Cluster count
- Minimum score before tagging as anomaly : threshold

Cluster count 와 threshold 값은 data 특성에 따라 edge impulse 에서 제안해 준다. 제안 해 준 값보다 더 큰 수를 조정해본 결과, accuracy 가 심하게 떨어졌다.

(note) 위 그림과 같이, accX RMS, accY RMS 와 accZ RMS 로 나누어지지 않는 경우에는 k-means block 을 사용할 수 없다고 보면 된다.

(note) K-means block 은 높은 정확도를 보이지 않으며, 아직 시행착오 중에 있다. 뒤 example 중에 필자가 시도한 사례를 실었으나 실패하였다.

(note) K-means 에 대한 설명이 부족함으로 설명이 나와 있는 두 링크를 첨부한다.

- <https://www.coursera.org/lecture/introduction-to-embedded-machine-learning/anomaly-detection-nqjHZ>
- <https://www.edgeimpulse.com/blog/embedded-machine-learning-cold-chain>

### Regression

Regression model 은 숫자 데이터에 기반하여 진행하여야 한다. Himax WE-1 보드 자체로는 온도 데이터와 같은 숫자 데이터를 생성할 수 없음으로 필자는 진행하지 않았다. 하지만, forum 을 통해 regression 모델을 이용한 예제 project 를 공유 받았다.

(공유 받은 프로젝트)

[https://studio.edgeimpulse.com/public/17972/latest?\\_ga=2.226218589.631506985.1617671290-771043665.1612167331](https://studio.edgeimpulse.com/public/17972/latest?_ga=2.226218589.631506985.1617671290-771043665.1612167331)

## 5. Retrain Model

Retrain Model 은 Impulse Design 을 변경하지 않는다는 전제하에, data 에 변경이 생겼을 경우 model 을 새롭게 retrain 해 주는 기능이다. 예를 들어 모델을 완성한 후 test 을 진행하였는데, dataset 이 부족하거나, test dataset 에서 training dataset 으로 옮겨서 모델을 retrain 할 수 있다.

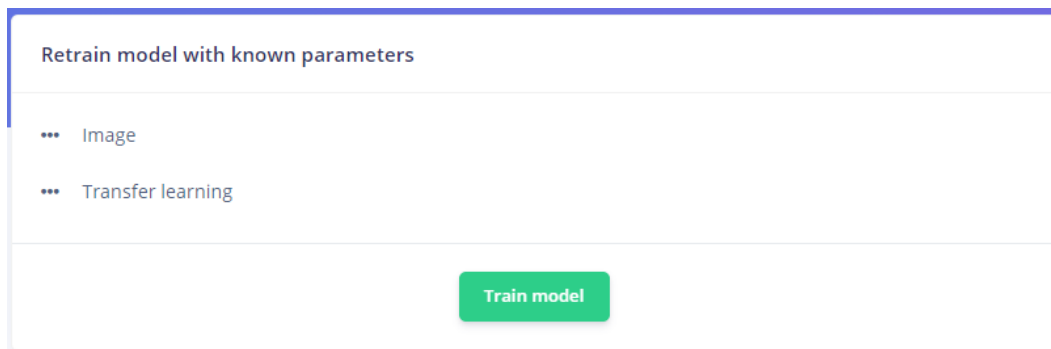


Fig 37. Retrain Model Prompt

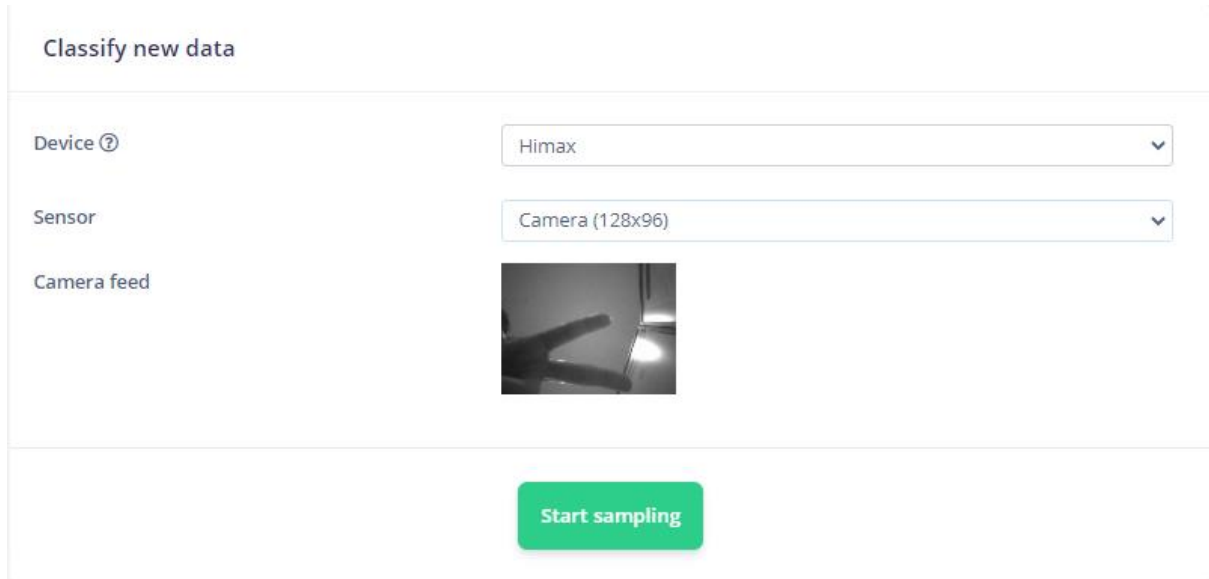
Train model 버튼을 누르면 최적화된 새로운 모델로 피팅이 된다.

## 6. Live Classification

완성된 모델을 직접 device 에 올리지 않고, 웹사이트 혹은 mobile phone 에서 test 를 진행할 수 있다. Device 는 2 절의 Edge Impulse CLI 을 통해 연결이 되어 있어야 하며, mobile phone 은 3 절의 data acquisition 에 적합한 방법으로 연결하는 편이 편하다.

Live classification 을 진행하는 경우는 조금 더 자세하게 classification 한 정도를 확인하기 위해서이다. 예를 들어, window size 에 따라(time stamp)에 따라 결과 값이 극명하게 변화하는 audio 의 경우 섬세하게 결과를 확인할 수 있다. 이에 대한 내용은 해당 example 에서 기술하다

✓ Test with device



The screenshot shows a web interface titled "Classify new data". It has three main sections: "Device" with a dropdown menu showing "Himax", "Sensor" with a dropdown menu showing "Camera (128x96)", and "Camera feed" which contains a video player showing a hand gesture. Below these sections is a large green button labeled "Start sampling".

Fig 38. Live Classification Prompt

Sensor 는 (Himax board 기준)

- Built-in accelerometer
- Build-in microphone
- Camera(128\*96)
- Camera(320\*240)
- Camera(640\*480)

있으며 Start sampling 을 통해 당시 device 에서 모으는 data 를 바로 모델에 올려볼 수 있다. 결과는 다음과 같이 나온다.

Classification result

Summary

Name

testing.jpg.21toeilu

Expected outcome

testing

CATEGORY	COUNT
cat	0
dog	0
uncertain	1

Detailed result

☐ Show only unknowns

CAT	DOG	UNCERTAIN
0	0	1.00

Fig 39. Live Classification via Device Output 1

위 모델은 cat, dog, uncertain 을 구분하는 모델임으로 아무것도 없는 천장을 찍었을 경우, uncertain 이 나오는 것을 확인할 수 있다.

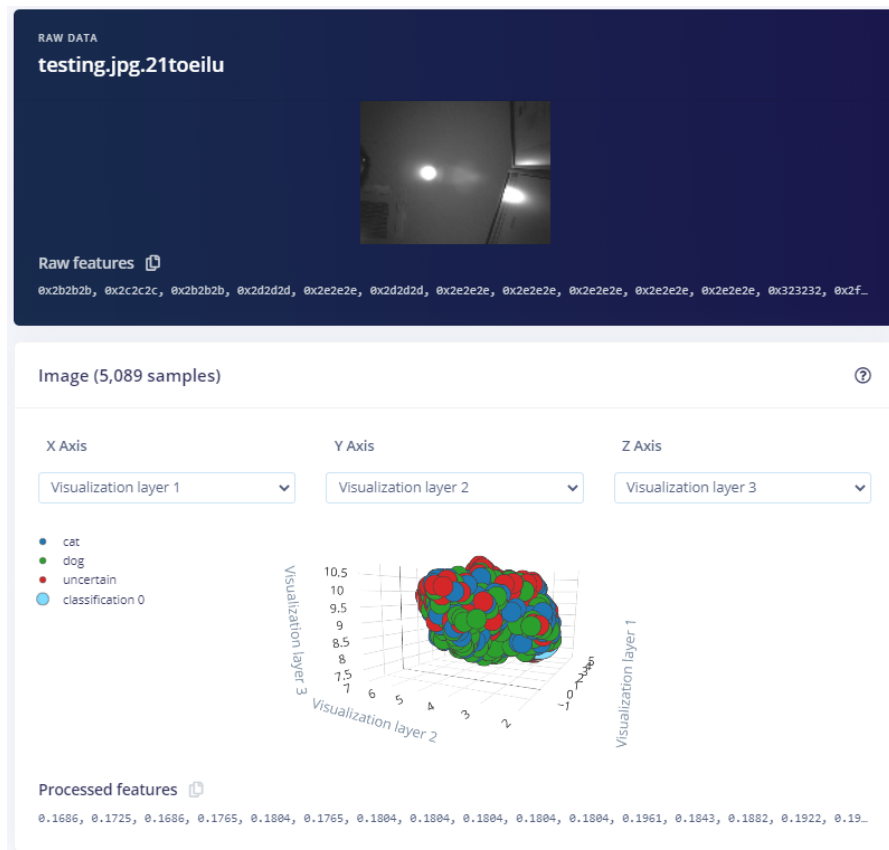


Fig 40. Live Classification via Device Output 2

찍은 raw data 와 분류한 표를 확인할 수 있다.

✓ Test with existing test dataset

Classify existing test sample

uncertain (1106).JPG.21qqstjd (uncertain)

Load sample

Fig 41. Live Classification via Dataset Output

이미 기존에 upload 한 test data 를 load 하여 확인한다. 결과는 test with device 와 동일하게 나타남으로 생각한다.

✓ Test with mobile phone

3 절에서 설명한 QR 코드를 통해 mobile phone 에 연결하거나, mobile phone 검색창에 'smartphone.edgeimpulse.com'을 검색한다. 연결한 웹사이트 맨 아래 쪽에서 'Switch to data collection mode', 'Switch to classification mode' option 중 classification mode 를 선택한다.

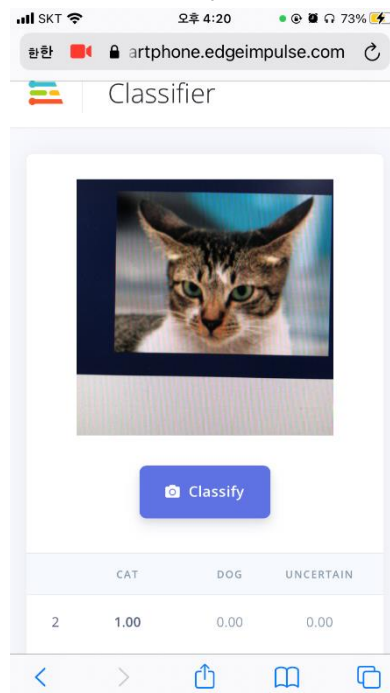


Fig 42. Live Classification via Mobile Phone Output

Mobile phone 에서 직접 모델을 load 하여 결과를 확인할 수 있으며, audio 와 image 두 가지 option 을 제공한다.

(note) Live Classification 에서 얻은 새로운 test data 는 모두 'Data acquisition'의 test data 에 저장된다.

## 7. Model testing

Test data 로 따로 분류해 두었던 unseen data 로 model 을 테스트해 볼 수 있다.

Test data

Classify all

Set the 'expected outcome' for each sample to the desired outcome to automatically score the impulse.

SAMPLE NAME	EXPECTED OUTCOME	LENGTH	ACCURACY	RESULT	
uncertain (1168).JP...	uncertain	-	100%	1 uncertain	
uncertain (1166).JP...	uncertain	-	0%	1 dog	
uncertain (1137).jp...	uncertain	-	100%	1 uncertain	
uncertain (1156).JP...	uncertain	-	NaN%	NaN undefined	
uncertain (1079).jp...	uncertain	-	100%	1 uncertain	

Fig 43. Model Testing Output 1

위와 같이 classification 결과를 확인할 수 있다.

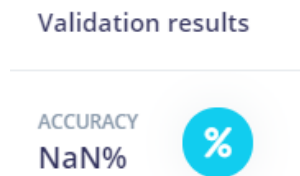


Fig 44. Model Testing Output 2

옆 %을 확인하여 model test 의 결과값을 확인할 수 있다. 하지만 위처럼 NaN%가 오는 경우가 있는데, 이 이유는 test data 중에서 명확하게 분류되지 않은 결과 값을 포함하고 있기 때문이다. 해결방법은 test data 를 삭제하거나, training set 으로 옮겨 새롭게 retrain 하는 방법이 있다.

(note) 위와 같은 방법을 굳이 사용하지 않고, model test 결과를 알고 싶을 경우에는 deployment 에서 EON 결과를 참조하면 된다. 이에 대한 내용은 9 절에 기술한다.

uncertain (1166).JP...	uncertain	-	0%	1 dog	
uncertain (1137).jp...	uncertain	-	100%	1 uncertain	
uncertain (1156).JP...	uncertain	-	NaN%	NaN undef	
uncertain (1079).jp...	uncertain	-	100%	1 uncertain	
uncertain (1072).jp...	uncertain	-	100%	1 uncertain	

Show classification

Move to training set

Rename

Edit expected outcome

Delete

Fig 45. Model Testing Options

노랑색 친 부분을 클릭하여 다양한 option 을 선택할 수 있다. 그 중에서 show classification 을 추천한다. 버튼을 누를 경우, live classification 으로 이동하며 어떤 category 를 모델이 헛갈려 하는 지 알 수 있다.

## 8. Versioning

Project versions					
Version your project to store all data, configuration, intermediate results and final models. You can revert back to earlier versions of your impulse by importing a revision into a new project.					
VERSION	CREATED	USER	DESCRIPTION	PUBLIC	STORAGE LOCATION
1	Yesterday, 17:41:48	SunBeenMoon	Classify Dog vs Cat with Himax WE-1 board	Ⓢ	Edge Impulse hosted Ⓢ

Fig 46. Versioning Option

Version 정보를 저장하여 다음 번에 불러올 수 있다. 노랑색으로 칠한 부분을 누르면 현재 만들어진 model 정보를 저장할 수 있다.

*(note)* 이 전에 기록했던 버전을 불러오면 새로운 프로젝트가 생성된다. 만약 developer 버전의 경우, project 개수 제한이 5 개인 점을 고려해야 한다


## 9. Deployment

### Deploy your impulse


You can deploy your impulse to any device. This makes the model run without an internet connection, minimizes latency, and runs with minimal power consumption. [Read more.](#)

### Create library


Turn your impulse into optimized source code that you can run on any device.




C++ library



Arduino library



Cube.MX CMSIS-PAK



WebAssembly

### Build firmware

Or get a ready-to-go binary for your development board that includes your impulse.



Eta Compute ECM3532 AI Vision



Himax WE-1 Plus



OpenMV library

Fig 47. Deployment board Option



Create Library 와 Build firmware 둘 중 한 가지를 선택하여 build 할 수 있다. 모델에 따라 library 종류와 firmware 종류가 다르며 지속적인 device update 가 진행되고 있다. 예시 사진의 모델은 image classification model 이다.

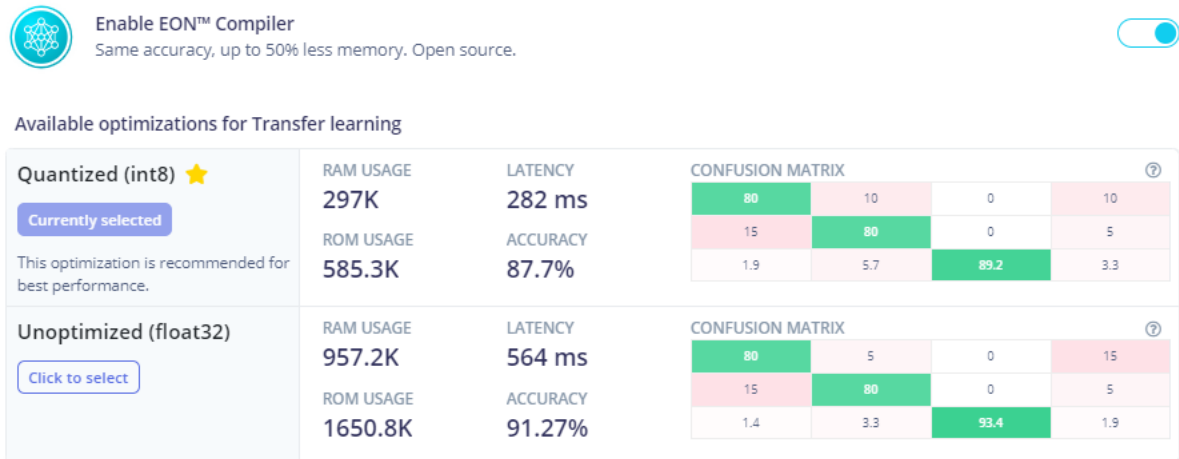


Fig 48. EON compiler

EON compiler 을 켜 상태에서 선택한 device 의 RAM 과 ROM usage, Latency, Accuracy 를 확인할 수 있다.

*(note)* EON 을 켜지 않고 분석을 진행할 수 있는데, RAM 과 ROM 사이즈가 매우 크다.

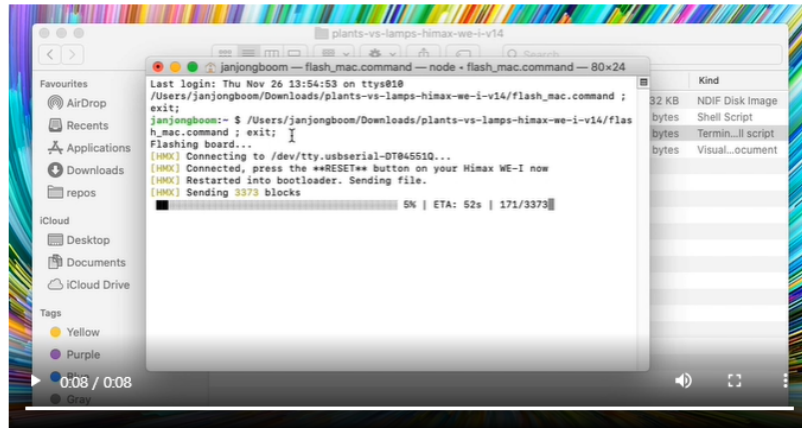
*(note)* Quantized 되지 않는 경우(+EON 을 사용하지 않는 경우) RAM 과 ROM 사이즈가 매우 크기 때문에 'Build'를 누를 경우 error 가 난다. Error 가 나는 경우는 device 가 허용할 수 있는 범위보다 모델의 크기가 크기 때문에 모델을 올릴 수 없어 생기는 오류이다.

*(note)* Accuracy 는 model test 의 정확도이다. 모델 test 에서 NaN%가 나올 경우, deployment 에서 확인할 수 있다.



### Built Himax WE-I firmware

Click on the script for your operating system to flash the binary.



Then, from your terminal or command prompt run: `edge-impulse-run-impulse`

Fig 49. Complete Download Prompt

'Build' 버튼을 누를 경우, 위와 같은 화면이 나오면서 모델이 다운로드 된다. 모델을 device 에 import 하는 방법은 다음과 같다.

- 다운로드한 폴더를 unzip 한다.

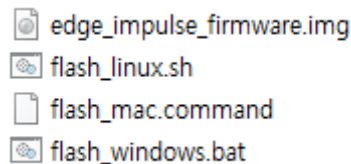


Fig 50. Download Zip Option

- 해당 OS 를 선택하여 device 에 모델을 올린다.

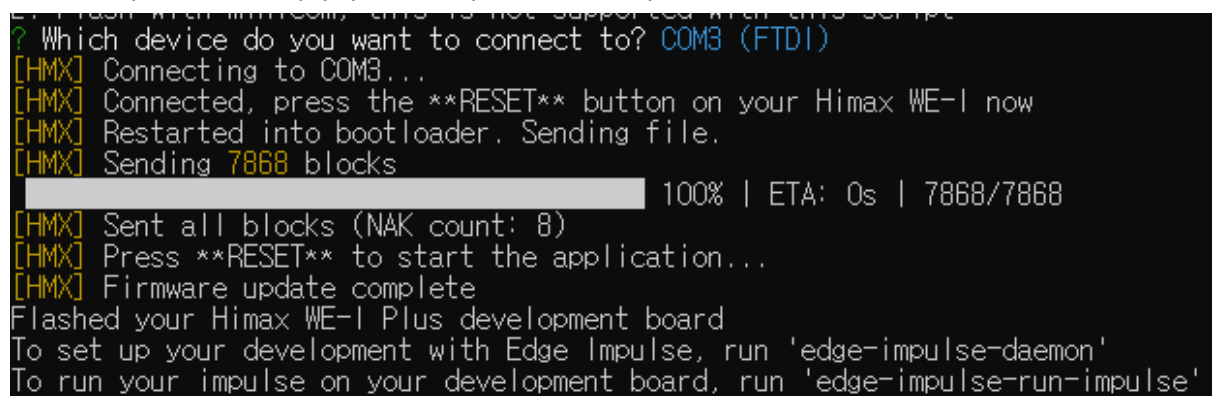


Fig 51. Choose Device to Upload Model

(note) device 가 Edge Impulse 웹 사이트에 연결되어 있거나, 다른 곳에 연결되어 있으면 model 을 올릴 수 없다.

(note) 다운 받은 파일 중에서 .img 파일을 himax-flash-tool 을 이용하여 업로드 할 수 있다. 아래 사이트를 참고하면 된다. <https://docs.edgeimpulse.com/docs/himax-flash-tool>

- Cmd 창에서 edge-impulse-run-impulse 를 눌러 모델을 실행시킨다

```
C:\Users\NOTA1204>edge-impulse-run-impulse
Edge Impulse impulse runner v1.12.2
? Which device do you want to connect to? COM3 (FTDI)
[SER] Connecting to COM3
[SER] Serial is connected, trying to read config...
[SER] Retrieved configuration
[SER] Device is running AT command version 1.6.0
[SER] Started inferencing, press CTRL+C to stop...
LSE
Inferencing settings:
    Image resolution: 96x96
    Frame size: 9216
    No. of classes: 3
HM0360 RevB,C,D Config
Starting inferencing in 2 seconds...
Taking photo...
Predictions (DSP: 1 ms., Classification: 133 ms., Anomaly: 0 ms.):
    cat:          0.0234
    dog:          0.0078
    uncertain:    0.9648
```

Fig 52. Output of Model

## 4.4 Examples

### 4.2.1 1 – Audio(given data : Keyword Spotting)

<Data>

	Training data	Test data
Yes	19m 45s	5m 10s
No	19m 53s	5m 3s
Noise	28m 14s	5m 2s
Unknown	19m 59s	5m 1s

Table 8. Data Setting of Example 1

출처 : <https://docs.edgeimpulse.com/docs/keyword-spotting>

(note) Impulse 를 design 하기 이전에 확인해야할 tutorial

<https://docs.edgeimpulse.com/docs/responding-to-your-voice>

(note) Audio model 을 구상할 때는, 필수적으로 Noise data 와 Unknown data 를 구별하고 싶은 데이터와 동일한 길이로 넣어주어야 한다.

### 1<sup>st</sup> Impulse Design

Time series data – MFCC – Neural Network(Keras) – Output Features

edge impulse 초기 세팅에서 변경하지 않았다.

Number of coefficients	13	Low frequency	300
Frame length	0.002	High frequency	None
Frame stride	32	Coefficient	0.98
Filter number	256	Shift	1
Window size	101		

Epoch = 500

Table 9. MFCC Parameter Setting

(note) MFCC 관련 docs <https://docs.edgeimpulse.com/docs/audio-mfcc>

## Neural Network(Keras)

NN design 은 위와 같으며, edge impulse 초기 세팅과 동일하다.

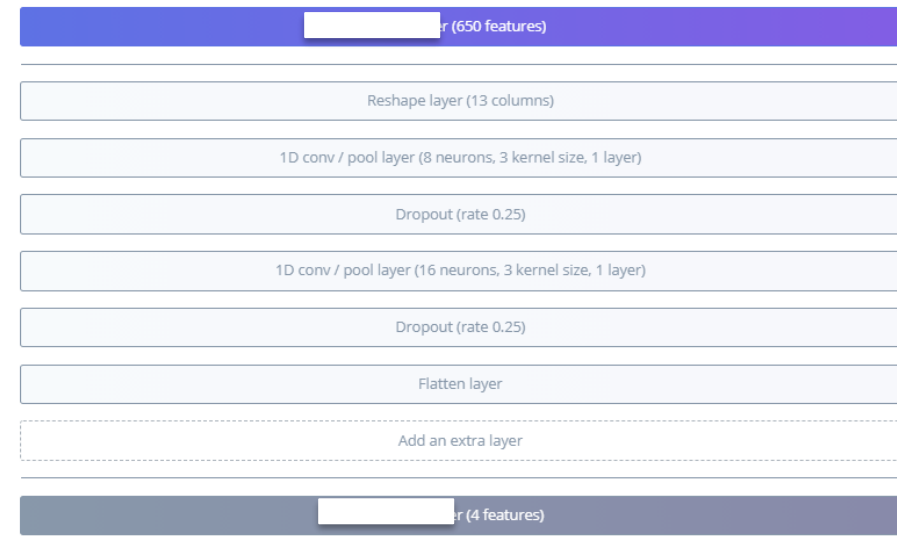


Fig 53. NN Setting of Example 1

### <Result>

		Ver 1	Ver 2	Ver 3	Ver 4
Training cycles		100	100	100	100
Learning rate		0.005	0.001	0.0005	0.0001
Validation		90.1%	90.6%	89.6%	78.7%
Loss		0.29	0.29	0.46	0.58
Model Test		81.99%	82.89%	80.84%	57.73%
EON O	RAM	5.3K	5.3K	5.3K	5.4K
	ROM	36.2K	36.2K	36.2K	36.2K
EON X	RAM	10.6K	10.6K	10.6K	10.6K
	ROM	56.8K	56.8K	56.8K	56.8K

Table 10. Result of example 1 - MFCC

### <Analysis>

Version 1 보다 Version 2 의 성능이 더 좋음을 알 수 있다. Version 1 은 edge impulse 에서 설정한 값 그대로 사용한 결과이며, Version 2 는 Learning rate 를 조금 더 낮추었다. 다른 version 에서는 overfitting 이 발생하였다.

## 2<sup>nd</sup> Impulse Design

Time series data – Spectrogram – Neural Network(Keras) – Output Features

edge impulse 초기 세팅에서 변경하지 않았다.

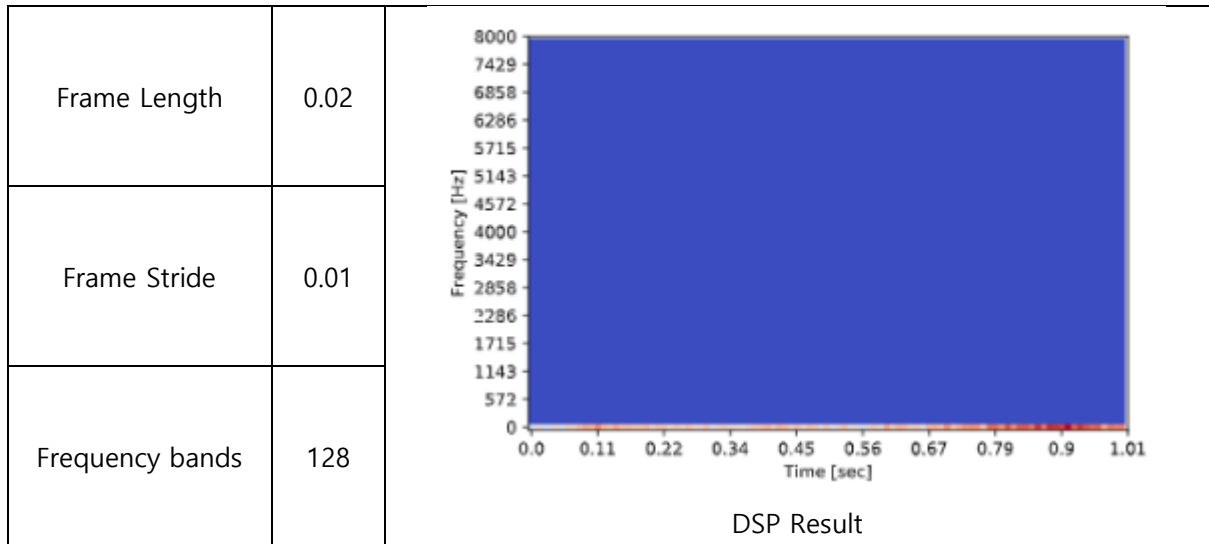


Table 11. Spectrogram Setting Parameter

(note) Spectrogram 관련 docs <https://docs.edgeimpulse.com/docs/spectrogram>

Neural Network Design 은 1<sup>st</sup> Impulse Design 과 동일하다.

### <Result>

		Ver 1	Ver 2	Ver 3	Ver 4
Training cycles		100	100	100	100
Learning rate		0.005	0.001	0.0005	0.0001
Validation		65.7%	46.7%	48.8%	69.0%
Loss		0.94	5.88	1.21	0.80
Model Test		76.07%	71.63%	63.65%	40.95%
EON O	RAM	16.6K	16.6K	16.6K	16.6K
	ROM	38.2K	38.2K	38.2K	38.2K
EON X	RAM	24.1K	24.1K	24.1K	24.1K
	ROM	58.8K	58.8K	58.8K	58.8K

Table 12. Result of Example 1 - Spectrogram

### <Analysis>

Spectrogram 이 MFCC 보다 낮은 성능을 보이고 있다. Spectrogram 은 사람 목소리가 아닌 소리를 구별하는 데 더 강점을 보인다.

### <On the Board Result>

```
Starting inferencing in 2 seconds...
Recording...
Recording done
Predictions (DSP: 185 ms., Classification: 2 ms., Anomaly: 0 ms.):
no:      0.964844
noise:    0.000000
unknown:  0.027344
yes:      0.003906
```

Fig 54. Board Result – No

```
Starting inferencing in 2 seconds...
Recording...
Recording done
Predictions (DSP: 185 ms., Classification: 2 ms., Anomaly: 0 ms.):
no:      0.007813
noise:    0.984375
unknown:  0.003906
yes:      0.003906
```

Fig 55. Board Result - Yes

```
Starting inferencing in 2 seconds...
Recording...
Recording done
Predictions (DSP: 185 ms., Classification: 2 ms., Anomaly: 0 ms.):
no:      0.011719
noise:    0.050781
unknown:  0.054688
yes:      0.882813
```

Fig 56. Board Result - Noise and unknown

1<sup>st</sup> impulse design(MFCC)의 Version 2 model 의 보드 import 화면이다.

## 4.2.2 2 – Audio(generated data 1 : '안녕')

### <Data>

	Training data	Test data
안녕	1m 47s	14s
Noise	1m 34s	26s
Unknown	1m 35s	25s

Table 13. Data Setting of Example 2

출처 : 안녕 – 직접 Himax WE-1 보드를 사용하여 녹음, Noise&Unknown 은 edge impulse docs 에서 제공되는 파일 사용 <https://docs.edgeimpulse.com/docs/keyword-spotting>

(note) 직접 녹음하는 audio model 을 design 하기 전 필수 확인 영상

<https://docs.edgeimpulse.com/docs/responding-to-your-voice>

(note) Audio model 을 구상할 때는, 필수적으로 Noise data 와 Unknown data 를 구별하고 싶은 데이터와 동일한 길이로 넣어주어야 한다.

(note) Himax WE-1 보드를 이용하여, 녹음 시 녹색 불이 들어오며 종료될 시에는 빨강 불빛이 들어온다.

(note) 녹음한 데이터를 1s 로 직접 나누어 주어야 한다.

### 1<sup>st</sup> Impulse design

Time series data – Audio(MFCC) – Neural Network(Keras) – Output features

\*MFCC 와 Neural Network Model 디자인은 4.2.1 절의 audio example 과 동일하다.

### <Result>

	Ver 1	Ver 2	Ver 3	Ver 4
Training cycles	100	100	100	100
Learning rate	0.005	0.001	0.0005	0.0001
Validation	93.1%	91.4%	82.8%	60.3%
Loss	0.20	0.27	0.40	0.84
Model Test	90.63%	79.69%	65.63%	25%



EON O	RAM	5.3K	5.3K	5.3K	5.3K
	ROM	36K	36.1K	36.1K	36.1K
EON X	RAM	10.6K	10.6K	10.6K	10.6K
	ROM	56.6K	56.7K	56.7K	56.7K

Table 14. Result of Example 2 - MFCC

## 2<sup>nd</sup> Impulse design

Time series data – Spectrogram – Neural Network(Keras) – Output features

\*Spectrogram 와 Neural Network Model 디자인은 4.2.1 절의 audio example 과 동일하다.

### <Result>

		Ver 1	Ver 2	Ver 3	Ver 4
Training cycles		100	100	100	100
Learning rate		0.005	0.001	0.0005	0.0001
Validation		84.5%	82.8%	82.8%	79.3%
Loss		1.02	1.01	1.00	0.63
Model Test		78.13%	81.25%	81.25%	46.88%
EON O	RAM	16.6K	16.6K	16.6K	16.6K
	ROM	41.3K	41.3K	41.3K	41.3K
EON X	RAM	24.1K	24.1K	24.1K	24.1K
	ROM	61.9K	61.9K	61.9K	61.9K

Table 15. Result of Example 2 – Spectrogram

### <Analysis>

MFCC 보다 Spectrogram 의 성능이 더 낮게 나왔으나, 앞선 4.2.1 절의 yes 와 no 를 구별하는 모델보다 성능이 우수하다. 구별해야 하는 데이터가 2 개에서 한 개로 줄었기 때문이다.

### <On the Board Result>

```
Predictions (DSP: 186 ms., Classification: 2 ms., Anomaly: 0 ms.):
  Anneung:      0.9961
  noise :       0.0000
  unknown :     0.0000
Starting inferencing in 2 seconds...
```

Fig 57. Board Result - 안녕

1<sup>st</sup> impulse design version 1 model 의 board import 결과 값

### <Live Classification example>

Input : '안녕' 대신 '선빈' 목소리를 1.5 초 동안 보드를 통해 녹음 했다.

CATEGORY	COUNT
Anneung	0
noise	0
unknown	2
uncertain	0

Detailed result

☐ Show only unknowns

TIMESTAMP	ANNEUNG	NOISE	UNKNOWN
0	0	0.05	0.95
500	0.02	0.02	0.96

Fig 58. Live Classification of Example 2

초기 세팅한 window 사이즈에 따라서 50msec 마다 소리를 분석하고 결과 값을 도출하는 것을 확인할 수 있다. Window size 는 impulse 를 생성할 때 첫 번째 block 에서 설정할 수 있다.

학습되지 않은 선빈 목소리를 넣었을 때, unknown 이라고 분류하는 것을 확인할 수 있다.

### 4.2.3 3 – Audio(given data 2 : 'sol' vs 'alarm')

#### <Data>

	Training data	Test data
Alarm	1m 21s	20s
sol	1m 14s	16s
Noise	1m 15s	19s
Unknown	1m 17s	14s

Table 16. Data Setting of Example 3

출처 : 리코더로 '솔' & 아이폰 기본 알람소리 – 직접 Himax WE-1 보드를 사용하여 녹음, Noise&Unknown 은 edge impulse docs 에서 제공되는 파일 사용

(edge impulse 에서 제공하는 파일 링크) <https://docs.edgeimpulse.com/docs/keyword-spotting>

(note) 직접 녹음하는 audio model 을 design 하기 전 필수 확인 영상

<https://docs.edgeimpulse.com/docs/audio-classification>

(note) Audio model 을 구상할 때는, 필수적으로 Noise data 와 Unknown data 를 구별하고 싶은 데이터와 동일한 길이로 넣어주어야 한다.

(note) Himax WE-1 보드를 이용하여, 녹음 시 녹색 불이 들어오며 종료될 시에는 빨강 불빛이 들어온다.

(note) 녹음한 데이터를 1s 로 직접 나누어 주어야 한다.

## 1<sup>st</sup> Impulse design

Time series data – Audio(MFCC) – Neural network(Keras) – Output features

\*MFCC 와 Neural Network Model 디자인은 4.2.1 절의 audio example 과 동일하다.

### <Result>

		Ver 1	Ver 2	Ver 3	Ver 4
Training cycles		100	100	100	100
Learning rate		0.005	0.001	0.0005	0.0001
Validation		85.5%	79.0%	75.8%	43.5%
Loss		0.38	0.50	0.60	1.32
Model Test		73.91%	68.12%	52.17%	1.45%
EON O	RAM	5.3K	5.3K	5.3K	5.3K
	ROM	36.9K	36.9K	36.1K	36.1K
EON X	RAM	10.6K	10.6K	10.6K	10.6K
	ROM	57.5K	57.5K	56.7K	57.5K

Table 17. Result of Example 3 - MFCC

## 2<sup>nd</sup> Impulse design

Time series data – Spectrogram – Neural network(Keras) – Output features

\*Spectrogram 와 Neural Network Model 디자인은 4.2.1 절의 audio example 과 동일하다.

### <Result>

		Ver 1	Ver 2	Ver 3	Ver 4
Training cycles		100	100	100	100
Learning rate		0.005	0.001	0.0005	0.0001
Validation		88.7%	90.3%	90.3%	62.9%
Loss		0.26	0.27	0.28	0.84
Model Test		84.06%	82.61%	85.51%	23.19%
EON O	RAM	16.6K	16.6K	16.6K	16.6K
	ROM	38.2K	38.2K	41.3K	41.3K
EON X	RAM	24.1K	24.1K	24.1K	24.1K
	ROM	58.8K	58.8K	61.9K	61.9K

Table 18. Result of Example 3 – Spectrogram

<Analysis>

MFCC 를 이용하여 디자인한 impulse 가 Spectrogram 을 이용하여 디자인한 impulse 보다 더 좋은 성능을 보였다.

<On the Board Result>

```
Predictions (DSP: 161 ms., Classification: 10 ms., Anomaly: 0 ms.):
alarm:      0.0000
noise :     0.0000
sol:        0.9961
unknown :   0.0000
```

Fig 59. Board Result - 리코더 sol

```
Recording...
Recording done
Predictions (DSP: 161 ms., Classification: 10 ms., Anomaly: 0 ms.):
alarm:      0.9961
noise :     0.0000
sol:        0.0000
unknown :   0.0000
```

Fig 60. Board Result - 아이폰 기본 알람

2<sup>nd</sup> Impulse design 의 Version 2 사용한 결과 값

### <Live Classification example>

Input : 리코더로 학습된 '솔'음 대신 '도'음을 녹음하였다.

CATEGORY	COUNT
alarm	0
noise	2
sol	0
unknown	1
uncertain	0

Detailed result

☐ Show only unknowns

TIMESTAMP	ALARM	NOISE	SOL	UNKNOWN
0	0	0.98	0	0.02
500	0	0.02	0	0.98
1000	0	1.00	0	0

Fig 61. Live Classification Result Example 3

Impulse design 에서 지정한 window 사이즈인 50msec 마다 결과 값을 도출하여서 input 소리가 무엇인지 분별한다.

모델은 주파수를 분석하기 때문에, 도음과 솔음을 구별할 수 있다. 도음을 입력할 시 uncertain 으로 분류하는 것을 확인할 수 있다.

#### 4.2.4 4 – Image(generated data 1 : 'dog' vs 'cat')

##### <Data>

	Training data	Test data
Cat	1493	20
Dog	1679	20
Uncertain	1919	212

Table 19. Data Setting of Example 4

출처 : dog and cat data <https://www.kaggle.com/tongpython/cat-and-dog> / uncertain data : <https://www.kaggle.com/shanmukh05/ml-hackathon> <https://www.kaggle.com/moltean/fruits>

다운로드 받은 데이터를 별도의 가공 없이 사용하 지 않고, 아래 사항을 체크 후 진행하였다.

- 강아지와 고양이가 두 마리 이상인 경우 삭제
- 강아지와 고양이가 둘 다 찍힌 경우 삭제
- 강아지와 고양이가 아주 작게 찍힌 경우 삭제
- 사람과 강아지와 고양이가 같이 등장하나 사람이 주인 경우 삭제
- 이 외 강아지와 고양이와 관련 없는 사진 삭제

제약에 걸리기 때문에, 다운로드 받은 40000 여장의 사진을 edge impulse enterprise version 에서 모두 사용할 수 없다. 또한, input 사진 크기를 squash 하는 방식으로 진행함으로 배경이 화려하거나, 강아지와 고양이가 작게 표현될 경우 좋지 않은 인식 모델이 만들어 진다.

(note) image data 를 처리하는 모델을 만들기 전에 확인해야 하는 영상

<https://docs.edgeimpulse.com/docs/image-classification>

(note) image data 를 이용한 모델을 생성할 경우, classification 하는 input 과 모르는 input 의 비율을 1:1.5 로 넣어주어야 한다.

(note) 모르는 input 의 labeling 을 'uncertain'이라 붙이는 편이 좋다.

(note) 직접 보드에서 image data 를 수집할 경우, 보드에서 깜박이는 불빛에 유의하여야 한다. 보드가 사진을 찍어도 된다는 표시로 카메라 옆 초록 led 를 깜박인다. Led 의 on/off 의 유무에 따라 사진의 밝기가 다르다.




		
원본	Led on	Led off

Table 20. Effect of LED on/off

### 1<sup>st</sup> Impulse design

Image data – Image – Transfer Learning(Images) – Output features

- Image data : 96\*96 (squash)
- Image : Grayscale
- Transfer Learning : MobileNetV2 0.35 (final layer: 16 neurons, 0.1 dropout)

### <Result>

		Ver 1	Ver 2	Ver 3	Ver 4
Training cycles		20	20	20	20
Learning rate		0.0005	0.0001	0.005	0.001
Validation		89.5%	88.8%	89.7%	90.4%
Loss		0.37	0.38	0.46	0.39
Model Test		87.7%	88.49%	88.89%	88.49%
EON O	RAM	297K	297K	297K	297K
	ROM	585.3K	585.3K	585.3K	585.3K
EON X	RAM	440.7K	440.7K	440.7K	440.7K
	ROM	774.4K	774.4K	774.4K	774.4K

Table 21. Result of Example 4 – MobileNet



<Classification with board>







Input	Classification	Result
	Cat : 0.03 Dog : 0.97 Uncertain : 0.00	Dog 정답
	Cat : 1.00 Dog : 0.00 Uncertain : 0.00	Cat 정답
	Cat : 0.11 Dog : 0.89 Uncertain : 0.00	Dog 탈락
	Cat : 0.01 Dog : 0.99 Uncertain : 0.00	Dog 정답
	Cat : 0.15 Dog : 0.23 Uncertain : 0.63	Uncertain 정답
	Cat : 0.00 Dog : 0.55 Uncertain : 0.45	NaN 세모

Table 22. Board Result – Example 4

<Analysis>

1<sup>st</sup> impulse design version3 으로 테스트를 진행했다.

대체적으로 강아지와 고양이를 구별했다. 한 번도 보지 않은 고양이 인형과 강아지 인형에 대해 정답율이 80% 가량이다. 또한, 모르는 데이터에 대해서 uncertain 이라 구별하기도 하지만, 명확하게 구별하지 못하는 경우가 있었다.

- 강아지와 고양이 input 데이터에서 강아지와 고양이 외 뒷 배경이 학습되었기 때문에 구별하지 못했다.
- 이 외 강아지와 고양이 인형을 헛갈려 하는 경우는 적은 데이터와 작은 모델을 사용해야 하는 edge device 에서 수용가능한 범위라고 생각한다.

## 2<sup>nd</sup> Impulse design

Image data – Image – Neural Network(Keras) – Output features

첫 번째 impulse design 과 image data 와 image block 은 동일하며, NN block 의 구성은 아래와 같다.

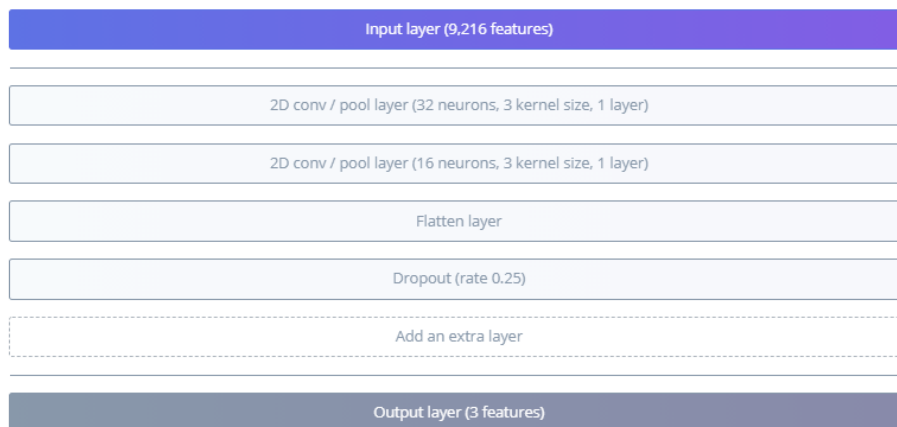


Fig 62. NN Setting of Example 4

### <Result>

	Ver 1	Ver 2	Ver 3	Ver 4
Training cycles	10	10	20	20
Learning rate	0.0005	0.0001	0.0005	0.0001
Validation	63.6%	62.6%	63.6%	64.4%
Loss	0.79	0.83	0.79	0.79
Model Test	51.19%	37.7%	51.19%	42.86%
EON O	RAM	362.5K	362.5K	362.5K

	ROM	60.6K	60.6K	60.6K	60.6K
EON X	RAM	437.9K	437.9K	437.9K	437.9K
	ROM	79.5K	79.5K	79.5K	79.5K

Table 23. Result of Example 4 – NN

#### <Analysis>

Edge impulse 에서 추천하는 NN 모델을 수정없이 사용하였으며, 결과의 정확도가 너무 낮기 때문에 보드에서 테스트를 진행하지 않았다.

- NN 모델의 layer 을 변경하거나 추가하면 더 좋은 결과를 얻을 수 있다.

#### <첨언>

- 데이터셋의 특성마다 다르겠지만, 2 가지 물체를 분류할 경우 2000 장씩 모르는 데이터는 2500 장씩 사용하는 것이 최대치이다.
- Edge impulse 사이트에서 제공하는 tutorial 상에는 분류할 물체를 30 장씩 사용하고 모르는 데이터를 50 장 사용한 모델 디자인을 추천한다. 위 모델의 경우에는 단순하며 카테고리 중 한 가지 물체(ex 물병이라면 한 가지 물병, 다른 종류의 물병은 구별 불가능)에 대해 구별이 가능하다. 즉, 고양이라면 한 마리의 고양이 30 장 input 에 대해서만 분류가 가능하다.
  - 시도해본 사례로는, 강아지 인형 3 가지 종류와 고양이 인형 3 가지 종류를 30 장씩 보드로 찍어 분류 모델을 만든다. 각 한 인형에 대해 분별은 가능하지만 전체 카테고리로 강아지 인형(강아지)로 분류할 수 없다.
- Enterprise 의 경우 제약 사항이 많아 feature 의 수가 많거나, 데이터 수가 많아 학습이 오래 걸릴 경우, 모델 생성이 중단된다. MobileNet 을 이용 뒷 쪽에 데이터에 맞게 조정된 weight 를 붙여나가기 때문에 모델이 길어질 경우 모델 생성이 중단된다.

#### 4.2.5 5 – Image(generated data 1 : 'mountain' vs 'sea')

<Data>

	Training data	Test data
Mountain	2490	22
Sea	2250	22
Uncertain	2980	22

Table 24. Data Setting of Example 5

출처 : <https://www.kaggle.com/shanmukh05/ml-hackathon> mountain 과 sea 외 데이터를 uncertain 데이터로 추가하였다.

(note) 72~73 페이지의 Example 4 의 setting note 를 필수로 확인한 후 진행한다.

#### 1<sup>st</sup> Impulse design

Image data – Image – Transfer Learning(Images) – Output features






- Image data : 96\*96 (squash)
- Image : Grayscale
- Transfer Learning : MobileNetV2 0.35 (final layer: 16 neurons, 0.1 dropout)

<Result>

		Ver 1	Ver 2	Ver 3	Ver 4
Training cycles		20	20	20	20
Learning rate		0.0005	0.0001	0.005	0.001
Validation		88.1%	88.5%	87.4%	88.3%
Loss		0.36	0.36	0.40	0.39
Model Test		74.24%	72.73%	66.67%	71.21%
EON O	RAM	297K	297K	297K	297K
	ROM	585.3K	585.3K	585.3K	585.3K
EON X	RAM	440.7K	440.7K	440.7K	440.7K
	ROM	774.4K	774.4K	774.4K	774.4K

Table 25. Result of Example 5 – MobileNet

<Classification with board>

Input	Classification	Result
	Mountain : 0.99 Sea : 0.00 Uncertain : 0.00	Mountain 정답
	Mountain : 0.15 Sea : 0.85 Uncertain : 0.00	Sea 탈락
	Mountain : 0.00 Sea : 1.00 Uncertain : 0.00	Sea 정답
	Mountain : 0.98 Sea : 0.02 Uncertain : 0.00	Mountain 탈락
	Mountain : 0.34 Sea : 0.00 Uncertain : 0.66	Uncertain 정답


	<p>Mountain : 0.66</p> <p>Sea : 0.01</p> <p>Uncertain : 0.33</p>	<p>Mountain</p> <p>탈락</p>
---	--	---------------------------

Table 26. Board Result – Example 5

### <Analysis>

1st Impulse design 의 version 1 을 사용한 결과표이다.

강아지와 고양이 모델에 비해 feature 수가 많아, 모델 성능이 떨어지는 듯하다. 하지만, 모델이 잘못 예측한 input 사진을 보면 수용가능한 오차라고 생각한다. 예를 들어, 마지막 사진은 glacier 사진이나, 산 비율이 높아 산으로 볼 수 있다.

### 2<sup>nd</sup> Impulse design

Image data – Image – Neural Network(Keras) – Output features

4.2.4 절의 2<sup>nd</sup> Impulse design 과 동일하다.

### <Result>

		Ver 1	Ver 2	Ver 3	Ver 4
Training cycles		10	10	20	20
Learning rate		0.0005	0.0001	0.0005	0.0001
Validation		77.3%	74.4%	fail	fail
Loss		0.57	0.62		
Model Test		48.48%	36.6%		
EON O	RAM	362.5K	362.5K		
	ROM	60.6K	60.6K		
EON X	RAM	437.9K	437.9K		
	ROM	79.5K	79.5K		

Table 27. Result of Example 5 – NN

### <Analysis>

Training cycle 을 높일수록 모델이 만들어지지 않고 fail 됐다.

## 4.2.6 6 – Accelerator(generated data)

### <Data>

	Training data	Test data
Idle	3m	40s
Snake	3m	40s
Updown	3m	40s
Wave	3m	40s

Table 28. Data Setting of Example 6

출처 : Himax WE-1 보드로 데이터 직접 수집

(note) 모델 생성 전에 확인해야 하는 동영상이다.

<https://docs.edgeimpulse.com/docs/continuous-motion-recognition>

(note) 보드를 이용하여 직접 데이터를 수집하면 초록색 led 가 들어오며, 데이터 수집이 종료되면 빨강색 led 가 들어온다.

(note) 필자가 실험 시 mobile phone accelerator 를 이용한 데이터로도 모델 생성이 가능하였다.

(note) document 에 담지 않았으나, edge impulse 에서 제공하는 데이터로 실험했을 경우에도 높은 정확도를 보였다.

<https://docs.edgeimpulse.com/docs/continuous-gestures>

## 1<sup>st</sup> Impulse design

Times series data – Spectral Analysis – Neural Network(Keras) – Output features

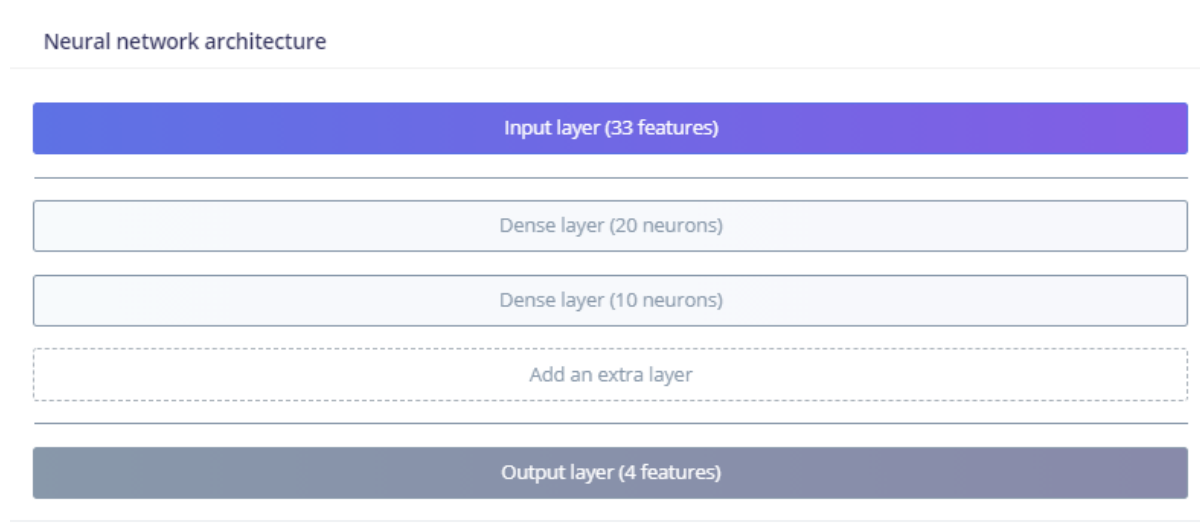


Fig 63. NN of Example 6

### <Result>

		Ver 1	Ver 2	Ver 3	Ver 4
Training cycles		30	20	20	20
Learning rate		0.0005	0.0001	0.005	0.001
Validation		95.9%	94.5%	98.0%	96.6%
Loss		0.10	0.15	0.05	0.11
Model Test		86.45%	86.39%	84.16%	85.02%
EON O	RAM	1.5K	1.5K	1.5K	1.5K
	ROM	15.4K	15.4K	15.4K	15.4K
EON X	RAM	3.8K	3.8K	3.8K	3.8K
	ROM	27.5K	27.5K	27.5K	27.5K

Table 29. Result of Example 6 – Spectrogram



<Classification with board>

```
Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 9 ms., Classification: 0 ms., Anomaly: 0 ms.):
Idle:      0.000000
Snake:     0.000000
Updown:    0.996094
Wave:      0.000000
```

Fig 64. Board Result - Updown

```
Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 9 ms., Classification: 0 ms., Anomaly: 0 ms.):
Idle:      0.000000
Snake:     0.996094
Updown:    0.000000
Wave:      0.003906
```

Fig 65. Board Result - Snake

```
Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 9 ms., Classification: 0 ms., Anomaly: 0 ms.):
Idle:      0.007813
Snake:     0.050781
Updown:    0.000000
Wave:      0.937500
```

Fig 66. Board Result - Wave

```
Starting inferencing in 2 seconds...
Sampling...
Predictions (DSP: 9 ms., Classification: 0 ms., Anomaly: 0 ms.):
Idle:      0.996094
Snake:     0.000000
Updown:    0.000000
Wave:      0.000000
```

Fig 67. Board Result - Idle

1<sup>st</sup> Impulse Design Version 1 로 진행한 결과이다.

## 2<sup>nd</sup> Impulse design

Times series data – Spectral Analysis – K-means Anomaly Detection – Output features

### <Result>

		Ver 1	Ver 2	Ver 3	Ver 4
Cluster count		30	20	20	20
Minimum score before tagging as anomaly		0.30	0.30	0.30	0.01
Model Test		86.45%	86.39%	84.16%	85.02%
EON O	RAM	1.5K	1.5K	1.5K	1.5K
	ROM	15.4K	15.4K	15.4K	15.4K
EON X	RAM	3.8K	3.8K	3.8K	3.8K
	ROM	27.5K	27.5K	27.5K	27.5K

Table 30. Result of Example 6 – K-means

### <Analysis>

Youtube(<https://www.coursera.org/lecture/introduction-to-embedded-machine-learning/anomaly-detection-nqjHZ?isNewUser=true&redirectTo=%2Flecture%2Fintroduction-to-embedded-machine-learning%2Fanomaly-detection-nqjHZ>)를 참고하여서 impulse 를 design 하였다. 모델이 만들어졌으나, 보드에 올리기 위해 build 하는 과정에서 fail 하였다.

## 4.2.7 7 – Image(MNIST)

### <Data>

Training data : 0~9(10 종류) Random 하게 60 개씩

Test data : 0~9(10 종류) Random 하게 15 개씩

Data 출처 : <https://intuitive-robotics.tistory.com/54>

### 1<sup>st</sup> Impulse Design

Image data – Image – Transfer Learning(Images) – Output features

- Image data : 96\*96 (squash)
- Image : Grayscale
- Transfer Learning : MobileNetV2 0.35 (final layer: 16 neurons, 0.1 dropout)

### <Result>

		Ver 1	Ver 2	Ver 3	Ver 4
Training cycles		10	10	20	20
Learning rate		0.0005	0.0001	0.0005	0.0001
Validation		93.3%	73.3%	93.3%	88.3%
Loss		0.33	1.09	0.26	0.66
Model Test		72.00%	9.33%	80.67%	40.00%
EON O	RAM	297K	297K	297K	297K
	ROM	586.2K	586.2K	586.2K	586.2K
EON X	RAM	440.7K	440.7K	440.7K	440.7K
	ROM	775.3K	775.3K	775.3K	775.3K

Table 31. Result of Example 7 - MobileNet

## 2<sup>nd</sup> Impulse design

Image data – Image – Neural Network(Keras) – Output features

첫 번째 impulse design 과 image data 와 image block 은 동일하며, NN block 의 구성은 아래와 같다.

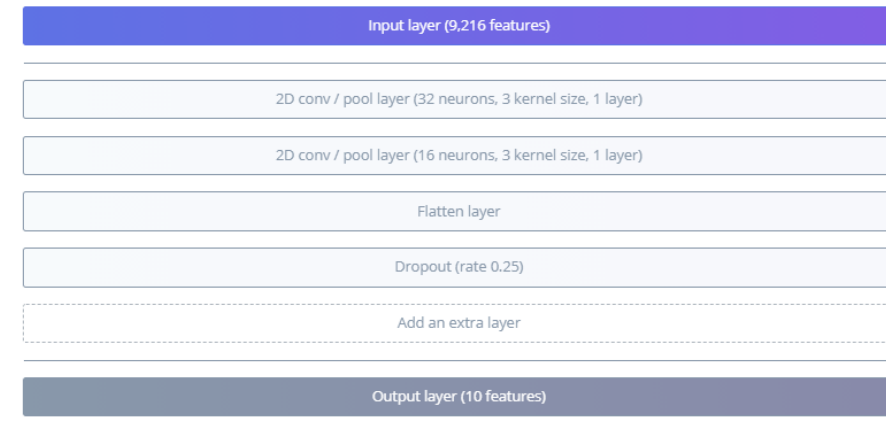


Fig 68. NN of Example 7

### <Result>

		Ver 1	Ver 2	Ver 3	Ver 4
Training cycles		10	10	20	20
Learning rate		0.0005	0.0001	0.0005	0.0001
Validation		85.8%	80.8%	85.8%	86.7%
Loss		0.50	0.63	0.50	0.47
Model Test		76.00%	55.33%	76.00%	70.00%
EON O	RAM	362.5K	362.5K	362.5K	362.5K
	ROM	123.5K	123.5K	123.5K	123.5K
EON X	RAM	437.9K	437.9K	437.9K	437.9K
	ROM	142.5K	142.5K	142.5K	142.5K

Table 32. Result of Example 7 - NN

### <전체적인 Analysis>

TFLM 과 비교하여 inference time 을 비교하고자 하였으나, edge impulse 를 통한 모델의 inference time 은 모두 2ms 이다. 분류하고자 하는 개수가 다른 모델들에 비해 10 가지로 많기 때문에 충분한 데이터를 입력할 수 없었다. 성능이 다른 모델들에 비해 떨어지며, 실제 보드에 import 해보았을 때, TFLM 의 handwriting 예제와 다를 바 없는 성능을 보였다.

## 4.3 Issues

<Issue 1> Regression block 이 update 되면 체크해보고 테스트 진행해야 한다.

<Issue 2> K-means block 의 성능이 떨어지는 부분이 있으므로 체크가 필요하다.

## 5. Example Code

### 5.1 embARC MLI

#### 5.1.1 Introduction

Synopsys사에서 Himax WE-1 보드 속 EM9D와 같은 칩들을 지원하기 위해서 만들어진 library이다. Neural network를 기반으로 모델을 upload할 수 있도록 관련 함수들을 지원한다.

- 자세한 embARC MLI에 대한 설명(ex. 각 함수정의, data type 등)이 첨부된 링크이다.

[https://foss-for-synopsys-dwc-arc-processors.github.io/embarc\\_mli/doc/build/html/index.html](https://foss-for-synopsys-dwc-arc-processors.github.io/embarc_mli/doc/build/html/index.html)

- Synopsys사에서 embARC MLI를 소개한 링크이다.

<https://www.synopsys.com/dw/ipdir.php?ds=machine-learning-inference-library>

#### 5.1.2 설치 방법

단, window에서 gmake 버전 혹은 license를 받아 다운로드한 MetaWare Development tool(eclipse)만 사용 가능하다.

(embARC MLI 공식 깃허브) [https://github.com/foss-for-synopsys-dwc-arc-processors/embarc\\_mli](https://github.com/foss-for-synopsys-dwc-arc-processors/embarc_mli)

그러나, 2021년도 3월 기준 깃허브에 일정 부분 release된 embARC MLI 2.0은 지원이 되지 않는다. 깃허브 자체를 clone하거나, zip 파일로 다운을 받게 될 경우, 지원이 되지 않는 embARC MLI 2.0이 다운된다. 따라서 아래 링크를 따라, embARC MLI 1.1을 다운 받아야 한다.

(embARC MLI 1.1) [https://github.com/foss-for-synopsys-dwc-arc-processors/embarc\\_mli/releases](https://github.com/foss-for-synopsys-dwc-arc-processors/embarc_mli/releases)

version 다운로드를 위 링크에서 다운 받으며 되며, 예제 코드는 맨 위 공식 깃허브를 따라 진행하면 된다.

### 5.1.3 Example Code

Window cmd 에서 진행한 예제 결과를 첨부한다.

#### <Cifar1\_caffe>

build 되는 과정

```
C:\VSmachine\embarc_mli\embarc_mli-Release_1.1\examples\example_cifar10_caffe>gmake run TCF_FILE=../../hw/em9d
.tcf
mdb -run -cl -nsim -tcf=../../hw/em9d.tcf -profile ./bin/example_cifar10_caffe.elf
(mdb:) NSIM_HOME changed to reference 32-bit version: C:\ARC\NSIM\NSIM
HARDCODED INPUT PROCESSING
RGB_2_CHW(w/o IR check):      6830 cycles
ir_conv1.idx(w/o IR check):    1658477 cycles
ir_pool1.idx(w/o IR check):    112054 cycles
ir_conv2.idx(w/o IR check):    1820940 cycles
ir_pool2.idx(w/o IR check):    22761 cycles
ir_conv3.idx(w/o IR check):    478316 cycles
ir_pool3.idx(w/o IR check):    18825 cycles
ir_ip1.idx(w/o IR check):      2754 cycles
ir_prob.idx(w/o IR check):     307 cycles

Summary:
  Layer1: 1777361 cycles
  Layer2: 1843701 cycles
  Layer3: 497141 cycles
  Layer4: 2754 cycles
  Layer5: 307 cycles

  Total: 4121264 cycles

Result Quality: S/N=4383.8      (72.8 db)
```

Fig 69. Progress of Cifar1-caffe

Test 결과

```
C:\VSmachine\embarc_mli\embarc_mli-Release_1.1\examples\example_cifar10_caffe>gmake run TCF_FILE=../../hw/em9d
.tcf RUN_ARGS="small_test_base/tests.idx small_test_base/labels.idx"
mdb -run -cl -nsim -tcf=../../hw/em9d.tcf -profile ./bin/example_cifar10_caffe.elf small_test_base/tests.idx
small_test_base/labels.idx
(mdb:) NSIM_HOME changed to reference 32-bit version: C:\ARC\NSIM\NSIM
ACCURACY CALCULATION on Input IDX testset according to IDX labels set
IDX test file shape: [20,32,32,3,]
Model input shape: [32,32,3,]

  2 of 20 test vectors are processed (1 are correct: 50.000 %)
  4 of 20 test vectors are processed (3 are correct: 75.000 %)
  6 of 20 test vectors are processed (5 are correct: 83.333 %)
  8 of 20 test vectors are processed (7 are correct: 87.500 %)
 10 of 20 test vectors are processed (9 are correct: 90.000 %)
 12 of 20 test vectors are processed (11 are correct: 91.667 %)
 14 of 20 test vectors are processed (13 are correct: 92.857 %)
 16 of 20 test vectors are processed (15 are correct: 93.750 %)
 18 of 20 test vectors are processed (17 are correct: 94.444 %)
 20 of 20 test vectors are processed (19 are correct: 95.000 %)
Final Accuracy: 95.000 % (19 are correct of 20)
FINISHED
```

Fig 70. Output of Cifar1-caffe

## <Face Detect>

```
C:\Vsmachine\embarc_mli\embarc_mli-Release_1.1\examples\example_face_detect> gmake run TCF_FILE=../../hw/em9d.tcf RUN_ARGS=grace_hopper.bmp
mdb -run -cl -nsim -tcf=../../hw/em9d.tcf -profile ./bin/example_face_trigger.elf grace_hopper.bmp
(mdb:) NSIM_HOME changed to reference 32-bit version: C:\ARC\nSIM\nSIM
Detection step #0
  Found a face at ([X:22, Y:17]; [X:58, Y:53])
Detection step #1
Detection step #2
Detection step #3
Detection step #4
  Found a face at ([X:13, Y:11]; [X:55, Y:53])
Detection step #5
Detection step #6
Detection step #7
Detection step #8
```

Figure 71. Progress of Face Detection


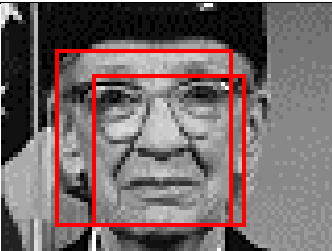
	
input	output

Table 33. Output of Face Detection

## <Keyword Spotter>

```
C:\Vsmachine\embarc_mli\embarc_mli-Release_1.1\examples\example_kws_speech>gmake run TCF_FILE=../../hw/em9d.tcf RUN_ARGS=./test.wav
mdb -run -cl -nsim -tcf=../../hw/em9d.tcf -profile ./bin/example_kws_stream.elf ./test.wav
(mdb:) NSIM_HOME changed to reference 32-bit version: C:\ARC\nSIM\nSIM
0.240  1.215  "on"(99.994%)
1.440  2.415  "stop"(99.997%)
3.600  4.575  "_unknown_"(99.960%)
```

Figure 72. Progress and Output of Keyword Spotter



## 5.2 ARC GNU Toolchain

### 5.2.1 Introduction

Synopsys사에서 개발자들의 위해 칩에 올릴 수 있도록 toolchain 을 제공한다.

- Synopsys에서 소개한 ARC GNU Toolchain 소개 글이다.

[https://www.synopsys.com/dw/ipdir.php?ds=sw\\_jtag\\_gnu](https://www.synopsys.com/dw/ipdir.php?ds=sw_jtag_gnu)

### 5.2.2 설치 방법

<https://github.com/foss-for-synopsys-dwc-arc-processors/toolchain>

위 링크는 ARC GNU Toolchain 의 공식 깃허브이다.

그러나, 일부 기능이 지원이 되지 않는 관계로 어느 정도 pre-built 된 파일 다운을 추천한다.

<https://github.com/foss-for-synopsys-dwc-arc-processors/toolchain/releases/tag/arc-2020.09-release>

위 링크를 따라, GNU Toolchain for ARC Processors, 2020.09 월 것을 다운로드 받는다.

(note) 예제 코드가 없으며, toolchain 을 설치 중 발생한 issue 를 해결 중이다.

## 6. Appendix

### 6.1 Install Oracle Virtual Machine(Regarding the Himax WE-1 board)

1. Oracle Virtual Machine 설치

<https://dog-developers.tistory.com/37>

위 링크를 통해 Oracle Virtual Machine 을 설치한다.

2. Virtual Machine 에 Linux 을 설치한다.

<https://ubuntu.com/download/desktop>

Linux iso 를 다운로드 받는다

<https://www.leafcats.com/151>

위 링크를 통해, Linux 버전을 실행한다.

3. Virtual Machine MAC 주소 변경

Synopsys 사에 License 를 요청할 경우, 본체 컴퓨터의 MAC 주소 그대로 요청한다. 그러나, Virtual Machine 내 MAC 주소와 본체 컴퓨터의 MAC 주소가 일치하지 않기 때문에, License 를 다운 받을 수 없다. 따라서, Virtual Machine 의 MAC 주소를 변경하여야 한다.

<https://ccm.net/faq/40897-virtualbox-how-to-refresh-the-mac-address-of-a-virtual-machine>

4. Virtual Machine 가상 디스크 용량을 늘린다.

굳이 사용하지 않으나, 처음 Linux version 의 Software 를 다운로드 받을 때 eclipse 부터 다양한 파일이 다운로드 되어 용량이 부족한 상황이 발생한다. 따라서, 아래 링크를 따라 디스크의 용량을 늘린 후 software 의 설치가 가능하다.

<https://ndb796.tistory.com/418>

5. 공유폴더를 설정한다.

Virtual Box, host window 사이에 호환이 가능한 공유폴더를 설정하여서 상호 파일을 교환할 수 있도록 한다. 공유폴더를 설정하면, Linux 에서 작업한 파일을 window 상으로 옮겨, teraterm 에서 작동시킬 수 있으며 또는 eclipse 에서 작업을 이어갈 수 있다.

<https://www.manualfactory.net/11052>

나머지 과정은 2.2 의 software download 과정과 동일하게 진행하면 된다.