****

FAMS

(Face Attendance Management System 얼굴인식을 통한 출입시스템)

3조: 김주선 우재하 유기쁨 이세연 이재윤 한동준

**[Contents]**

**Chapter I. 소개**

**1.1 프로젝트 소개**

**1.2 주제 선정 배경**

**Chapter 2. 기존 관련 연구 및 기술 동향**

**2.1 기존 얼굴인식 기술**

**2.2 활용 사례**

**Chapter 3. 사용 도구**

**3.1 사용 도구**

**Chapter 4. 얼굴인식 시스템**

**4.1 SW/HW 구조도**

**4.2 얼굴 인식 모델 구축**

**4.2.1 FaceNet**

**4.2.2 MobileFaceNet**

**Chapter 5. Model 성능 비교**

**Chapter 6. 앱 서비스 연동**

**6.1 Flutter를 통한 앱 구현**

**6.2 Firebase를 통한 DB 구현**

**Chapter 7. 기타**

**7.1 기대효과**

**7.2 한계점**

**7.3 프로젝트 진행 일정**

**7.4 상용 기술 소개 및 발전 방향**

**7.5 팀원 소개 및 평가**

**Chapter 8. 참고사항**

**Chapter 0. 참고자료**

**Chapter1. 소개**

* 1. **프로젝트 소개**

사람의 얼굴 모양, 망막 등 얼굴의 특징을 자동화된 장치로 추출해 개인을 구분하거나 인증하는 시스템이며, 얼굴인식은 얼굴 전체보다는 코와 입, 눈썹, 턱 등 얼굴 골격이 변하는 각 부위 50여 곳을 분석해 출입을 통제하는 시스템입니다. 또한 APP을 통해 기업이나 학교에서 출결을 확인 할 수 있게 구현했고, 스마트 폰으로 사진을 찍어 사용자의 얼굴을 학습시킬 수 있도록 구현했습니다.

* 1. **주제선정 배경**

1. 카드를 항상 소지해야 하는 불편함을 해소하기 위해
2. 대리 출석을 방지하기 위해
3. 대규모 강의 시 수업지연 해소를 위해
4. 출석 누락을 방지하기 위해

**Chapter 2. 기존 관련 연구 및 기술 동향**

**2.1 기존 얼굴인식 딥러닝 기술**

얼굴 인식의 경우 얼굴 부위의 특징만 뽑아내면 되기 때문에, 물체를 인식하는 딥러닝 방식을 바로 적용하기 쉬워 다양한 얼굴인식에 관한 논문들이 발표되었다.

**2.2.1 Deep Face**

Deep Face 모델은 Facebook에서 2014년도에 발표한 모델로, 얼굴을 조각 별로 특징을 추출하기 위해서 사전에 학습된 3D 얼굴 모델을 이용하여 특징을 추출하고, affine 변환으로 추출한 데이터를 정렬하였다. 그리고 9개의 Convolutional Layer을 Facebook 기업 내에 있는 자료를 사용하여 학습시켰다. 이때, 얼굴 영역 내의 국소 특징을 효과적으로 추출하기 위해 locally connected convolutional layer을 사용한 것이 특징이다. Deep Face 모델은 LFW Dataset에서 97.35% 정확도를 달성했다. 다만, 이 모델은 weight sharing이 없는 locally connected convolutional layer로 인해 신경망의 파라미터가 1억 2천만 개가 되어 학습시키기가 어렵다는 단점이 있다.

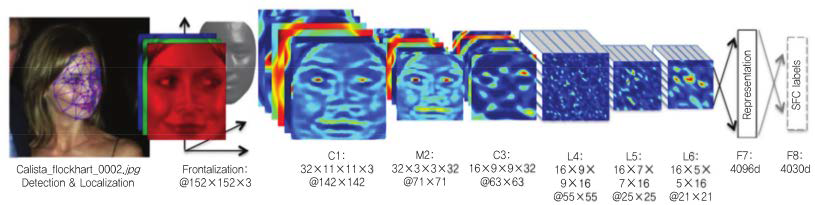
****

그림 1. Deep face network 구조

**2.2.2 VGGFace**

Deep Face 이후로 등장한 구조는 옥스퍼드 대학의 Visual Geometry Group(VGG)에서 제안한 VGG-Face(or Deep FR) 딥 네트워크 구조이다. VGG-Face에서는 인터넷 검색을 통해 직접 만든 대용량의 얼굴 인식을 위한 데이터 셋인 VGG-Face-Dataset을 공개하고, 이 데이터를 이용하여 15개의 Convolutional Layer로 구성된 Deep-Network structure를 학습시켰다. VGG에서는 VGGFace 학습 모델을 제공할 뿐만 아니라, ImageNet video recognition challenge에서의 VGG 구조와 마찬가지로 상대적으로 간단한 3 by 3 Convolutional Filter를 이용하여 학습시킴으로써 VGGFace는 LFW Dataset 에 대해 Deep Face보다 약 1% 정도 개선된 98.95% 성능을 달성하였다. 이 외에도 얼굴인식 성능 개선을 위해 DeepID, DeepID2, DeepID2+, DeepID3과 같은 얼굴인식을 위한 다양한 Deep-Network structure가 제안되었다.

**2.2.3 Face Net**

앞서 설명했다시피 다양한 얼굴인식 Deep-Network-structure에 대한 연구사례가 있었고. 거기에 더불어 Loss function의 재정의가 있었다. 이에 따라 특징을 분별할 수 있는 “거리척도 학습”이 나온다. 대표적으로 Face Net 알고리즘이 있다. Triplet Loss를 정의하여 학습시킨다. 즉, anchor 특징과 동일인에 대한 positive 특징 사이의 거리가 negative특징과 비교했을 때 가까워지도록 하는 것을 의미한다. Deep-Network-Structure는 22개의 Inception-Network-Layer를 사용했고, 500M 장의 영상 데이터 셋으로 학습하였다. 이를 통해 얼굴 영상 입력에 대해 높은 분별력을 갖는 128bytes의 특징 embedding을 수행하였고, LFW데이터셋에 대해 99.6%의 정확도를 보였다. Face Net 연구의 성공으로 고차원의 영상 데이터 맵핑을 위한 angular softmax 손실함수를 정의하여 학습을 수행한 연구가 진행됐다.

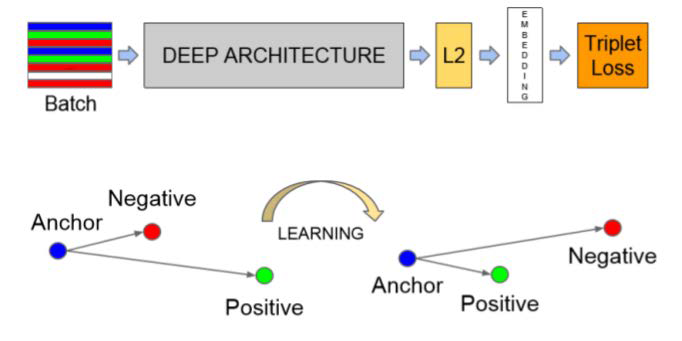
****

그림 2. Triplet Loss 개념도

**2.2.3 MobileFaceNet**

앞서 설명한 모델들은 데이터를 처리하는데 고성능의 그래픽카드와 CPU를 사용했다. 또 이러한 모델들은 수 많은 매개변수를 학습하기 위해 보다 더 높은 성능을 요구한다. 따라서 고성능의 디바이스가 아닌 자동차나 드론, 스마트 폰, Jetson Nano와 같은 환경에서는 위와 같은 모델들은 사용하기에는 너무 많은 연산을 처리하는데 어려움이 있다. 또, 배터리를 사용하게 된다면 자원도 한정되기 때문에 항상 높은 성능을 유지하기 어렵다. MobileFaceNet은 이러한 성능 제한이 있는 환경에서 고효율을 낼 수 있게 설계된 CNN구조를 의미한다. MobileFaceNet이 MobileNetV2과 다른 점은, 저차원 데이터의 정보 손실을 막기 위한 bottlenect layer에서의 공간 확장 인자의 크기를 크게 줄이고, 활성함수를 ReLu대신 PReLu를 사용하고, network초기에 빠른 down sampling 전략을 사용했다는 것이다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Network** | **LFW** | **AgeDB-30** | **Params** | **Speed** |
| **MobileNetV1** | **98.63%** | **88.95%** | **3.2M** | **60ms** |
| **ShuffleNet(1x, g=3)** | **98.70%** | **89.27%** | **0.83M** | **27ms** |
| **MobileNetV2** | **98.58%** | **88.81%** | **2.1M** | **49ms** |
| **MobileNetV2-GDConv** | **98.88%** | **90.67%** | **2.1M** | **50ms** |
| **MobileFaceNet** | **99.28%** | **93.05%** | **0.99M** | **24ms** |

**Table 1. CASIA-Web face Dataset으로 학습한 모델의 성능 비교. 사용한 장치는 모바일 폰의 CPU, Qualcomm Snapdragon 820.**

**2.2 활용 사례**

2.2.1. 북한 ARM CPU 기반 딥러닝 얼굴인식 개발

북한 인공지능 개발 연구원들은 인터넷 접속에 의한 구름연산환경을 사용하지 않고도 가정용이나 휴대용 전자 장치들에서 직접 인공지능기술을 구현하고자 했다. 그 중, 얼굴인식 신경망을 개발했는데, 얼굴 검출에 MTCNN, 얼굴 인식에서 MobileFaceNet기반으로 구현했다고 한다. 이런 신경망 구조 구현과 학습을 위해 Tensorflow를 사용했으며 손실함수는 각도여백손실함수(ArcFace Loss)를 사용했다고 설명했다. 연구원들의 주장으로는, 30만명, 3000만장의 Database에서 얼굴인식 신경망을 학습했다고 한다.

2.2.2. iPhone과 iPad에서 사용되는 얼굴 인식을 이용한 보안 기술(Face ID)

휴대폰의 카메라로, 얼굴에 3만 개의 점을 얼굴에 투사에 사용자의 얼굴 패턴을 추출한 다음, 하드웨어에 데이터를 저장하고 학습한다.

**Chapter 3. 사용 도구 및 서비스 모델**

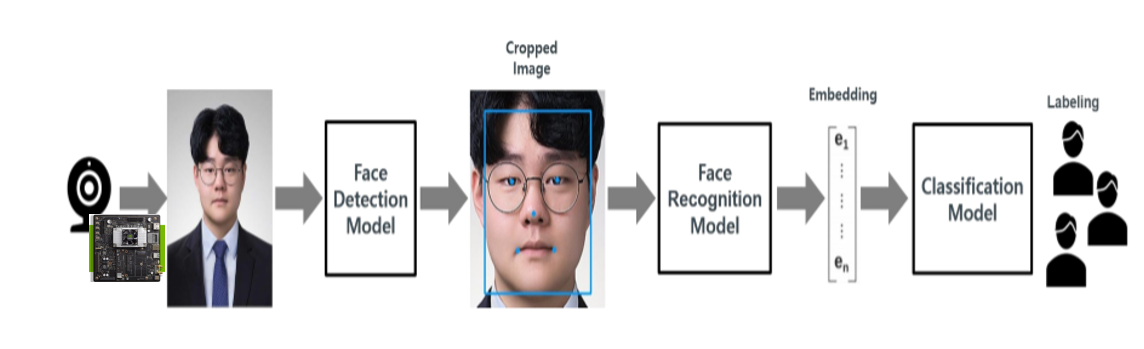
**3.1 사용 도구**

|  |  |
| --- | --- |
| os | Linux Ubuntu 18.04 LTS |
| 프로그래밍 언어 | Python |
| APP | Flutter |
| DB | Firebase |
| 프레임워크, 라이브러리 | Tensorflow, Opencv |
| 딥러닝 모델 | MobileFaceNet |
| 카메라 | Logitech C-920 |

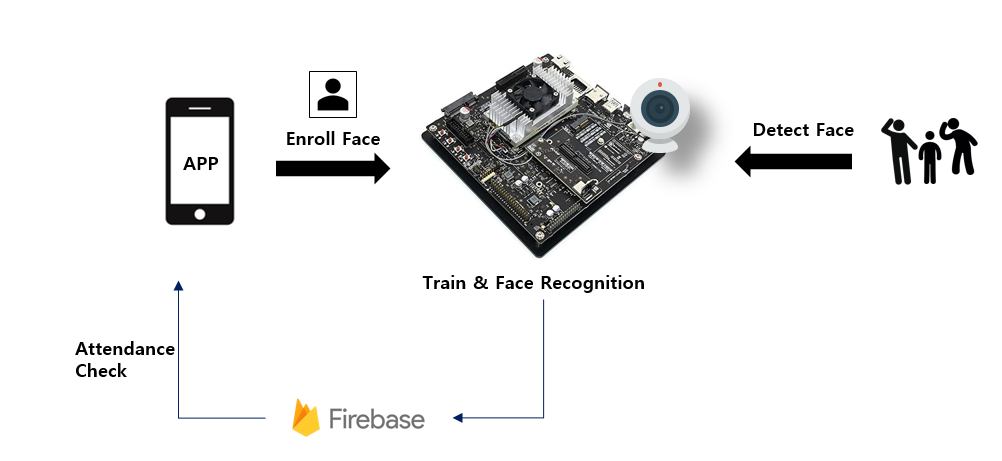
**Chapter 4. 얼굴인식 시스템**

* 1. **SW/HW 구조도**

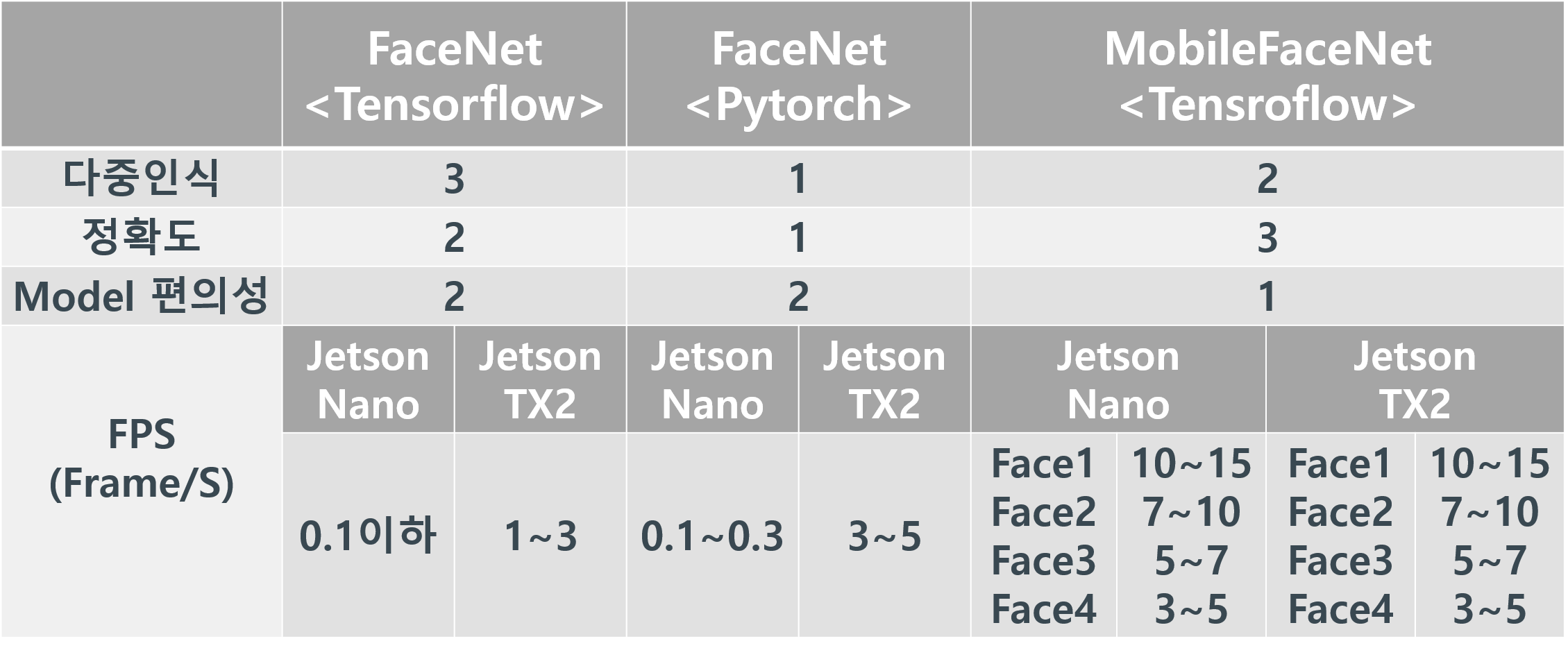
**4.1.1 SW 구조도**



**4.2.2 HW 구조도**



**Chapter 5. Model 성능 비교**

 ****

**-** Face Net과 MobileFaceNet 모델 중 성능은 Face Net이 우수

- Face Net 모델을 연산하기가 Jetson Develop-Kit에서는 쉽지 않음

→ MobileFaceNet이 속도 면에서 월등함

**Chapter 6. 앱 서비스 연동**

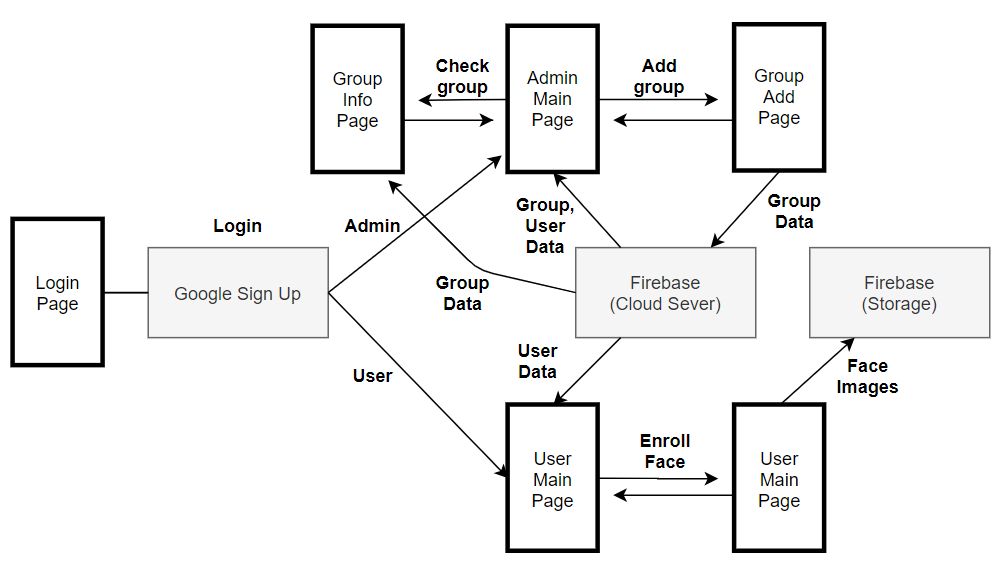
**6.1 Flutter를 통한 앱 구현**

a. flutter

ios, android 개발을 크로스플랫폼 앱 개발의 대표적인 프레임워크는 React, Flutter 두 개가 대표적이다. 우린 클라우드 서버를 활용하여 데이터베이스를 관리하기 위해 Firebase를 사용했고, Firebase와 Flutter 호환성이 React보다 뛰어나서 사용하게 됐다. 또한 프레임워크 결정에 있어 가장 중요한 포인트는 개발 언어라고 생각한다. React는 Javascript기반이고 Flutter는 Dart기반이다. 개발자의 숙련된 언어에 맞춰서 선택하는 것이 효율적이지만 우린 Javascript나 Dart 숙련도가 비슷하여 좀더 직관적인 Dart를 선택하게 되었고 Dart에 맞는 Flutter 프레임워크를 사용하게 됐다

b. firebase  
 클라우드 서버를 사용하기 위해 데이터베이스를 고민하다. AWS와 Firebase둘중 고민하게 됐다.  
본 프로젝트는 개발기간이 짧아 데이터베이스를 빠르게 구축하고 연동하기 위해서 좀더 쉬운 Firebase를 사용하게 되었습니다. 또한 본 서비스는 유저와 간단한 출석 기록만 남기는 형태이므로 대용량 데이터베이스를 사용하기 보다는 NoSQL구조를 사용하기 위해서 사용했습니다.

6.1.2 앱 구조도



**Chapter 7. 기타**

**7.1 기대효과**

본 프로젝트는 얼굴 인식 기술을 이용한 출석 시스템으로 학교뿐만 아니라 회사에서도 출석 또는 출근 확인에 소요되는 불필요한 행위를 줄일 수 있습니다.

첫 째, 카드를 항상 소지해야 하는 불편함을 해소할 수 있습니다.

둘 째, 대리 출석을 방지할 수 있습니다.

셋 째, 대규모 강의 시 수업지연을 해소할 수 있습니다.

넷 째, 출석누락을 방지할 수 있습니다.

**7.2 한계점**

1. 얼굴 인식 시 문제점

1.1. spoofing 방지 필요

1.2. 모자나 마스크 등으로 얼굴을 감출 시 인식 한계

1.3. 성능 개선 모델에 따른 인식률 저하

2. APP의 리스트 기능 문제점

2.1. 관리자의 그룹 추가 제거 기능 필요

2.2. 이미 존재하는 그룹에 학생을 관리하는 기능 필요

2.3. cloud fire storage 안의 사진 데이터에서 학생 변경 시 갱신하는 기능 필요

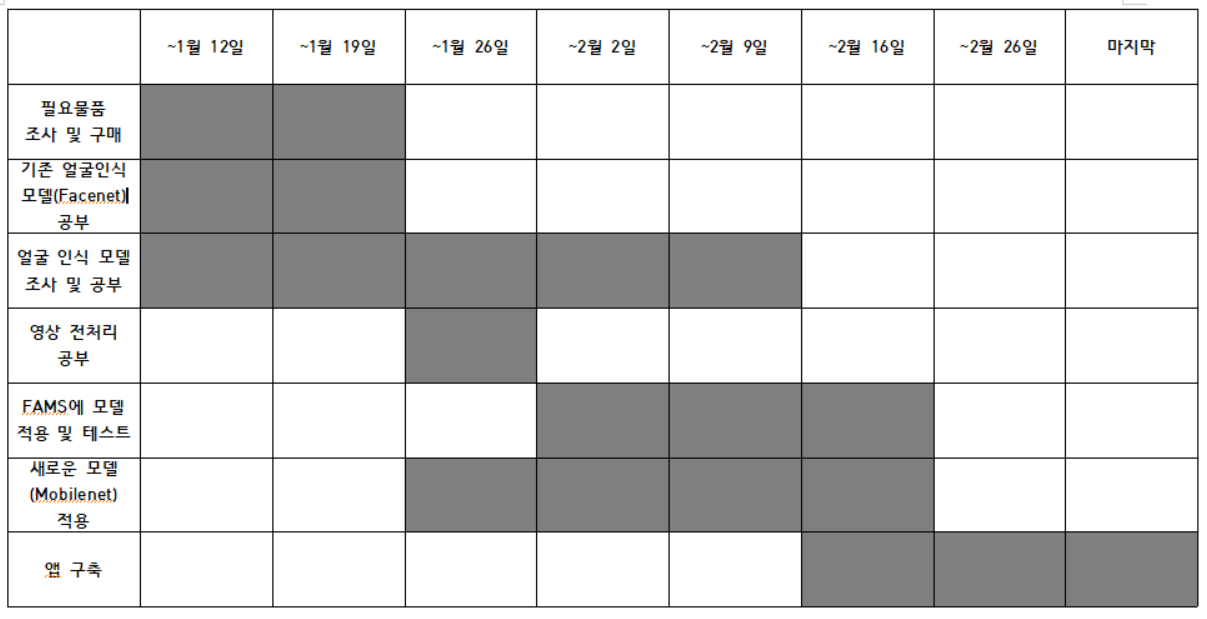
3. 장비의 문제

3.1. Jetson nano, tx2의 성능 vs Desktop의 성능

3.2. 카메라의 각도에 따른 인식 불가

3.3. 학습 시 미리 준비된 모델을 불러올 때, cpu를 하나만 사용하는 문제

**7.3 프로젝트 진행 일정**



**7.4 상용 기술 소개 및 발전 방향**

**7.4.1 상용 기술 소개**

|  |  |  |  |
| --- | --- | --- | --- |
| **상용기술** | **적용기술** | **특징** | **기술수준** |
| **각종 신분증 조회**  **(신용카드, 여권)** | **노화 인식. 얼굴 인식** | **영상 품질 양호**  **(조명, 포즈)** | **상용화 수준**  **(인식율 95%)** |
| **접근제어 시스템**  **(ATM/CD, 출입 통제)** | **얼굴 검출, 얼굴 인식** | **동영상 이용** | **상용화 초기 단계** |
| **공공장소 감시**  **(테러범, 용의자 감시)** | **얼굴 검출, 경보 시스템, 카메라 자동 작동기술** | **동영상 이용,**  **작은 얼굴 영상** | **기초 연구 단계** |
| **범죄자 사진 조회** | **대용량 DB 구축,**  **노화 인식** | **여러 각도의 얼굴 인식. 영상 품질 양호** | **신분 조회**  **기술 수준과 유사** |
| **카지노**  **(사기범, 범죄조직검출)** | **얼굴 검출, 얼굴 인식** | **동영상 이용** | **상용화 초기 단계** |
| **전자간판**  **(타깃 광고)** | **얼굴 검출, 얼굴인식**  **노화인식, 성별 인식** | **동영상 이용** | **상용화 초기 단계** |
| **차량 도난 방지 및**  **졸음 방지** | **얼굴 검출, 얼굴 인식** | **동영상 이용** | **상용화 노력 단계** |
| **스크린 세이버**  **(개인 사용자 인증)** | **얼굴 검출, 얼굴 인식** | **동영상 이용, 사무실**  **환경** | **상용화 초기 단계** |

**7.4.2 발전 방향**

**a. 기술 개선 방향**

Face recognition 기법에는 2-D기법과 3-D Algorism의 두 종류가 있다. 예를 들어 그림은 2D이며, 이 2차원의 이미지만을 통해 사람을 식별하는 알고리즘과, 2-D 이미지를 3차원의 grid로써 표현하면서 얼굴의 특징을 잡아내는 것이 3-D Algorism이다. 3-D Algorism은 옆 모습만 보고도 얼굴을 인식할 수 있으므로 이상적이긴 하지만, 2차원 Algorism에 비하여 계산 소요량이 많아지기 때문에, 실시간으로 높은 정확도를 유지하기엔 많은 처리 속도가 필요하게 된다. 또, 현 단계에서 얼굴 인식 2-D, 3-D Algorism의 공통 challenge는 조명의 강약에 따라 Algorism이 제대로 작동하지 못하는 경우가 생기는 문제와 얼굴의 각도가 맞춰지지 않을 경우에는 판별하는 속도와 정확성이 낮아지는 점, 그리고 얼굴의 일부가 가려졌을 경우에는 Algorism을 적용하기 어렵다는 문제가 있다. 따라서 이미지 전처리 과정에서 조명, 각도, 가려짐에 관한 문제를 GAN, NAN등과 같은 새로운 기술로 해결하려는 노력이 필요하며, Algorism을 효율적으로 구성함으로써 좀더 빠른 처리속도를 구현해 낸다면 real world에서도 높은 처리 속도와 정확도로 상점, 유통, 금융 등 많은 분야에서 상용화 할 수 있을 것이다.

**b. 상용화를 위한 사회적/정책적 개선 방향**

현재 얼굴인식 기술이 상업적으로 이용되지 않는 이유는 소비자 평판과 정부기관에 의한 공적 규칙의 부재 등으로 크게 2가지로 볼 수 있다. 생체인식시스템에 있어 디지털방식으로 생체형판에 서명하는 것은 무결성(integrity)을 보호하기 위해 적절하지만, 그것은 데이터에 어떠한 비밀성도 제공하지 않는다. 만일 생체정보의 프라이버시를 보호하기 위하여 비밀성이 요구된다면 접근제어 또는 암호화가 필요하다. 결국 이러한 개인정보의 노출 이슈는 아직 해결되지 않고 있지만, 접근제어 혹은 암호화 기법에 의한 기술적 보안이 유효한 대응책으로 제시되고 있다. 따라서 이를 보완할 수 있도록 생체정보를 포함한 개인정보에 대한 프라이버시 의식을 고양하여 상호신뢰의 문화를 정착시키고, 이를 바탕으로 한 법제도적 환경을 구축하여 민감한 정보인 생체정보를 두텁게 보호하며, 데이터베이스(DB)를 분산 관리하여 통합에 따른 생체정보유출의 피해를 사전에 예방하는 데 주력해야 할 것이다.

**7.5 팀원 소개 및 평가**

**7.5.1 팀원 소개**

****



우재하

한동준

유기쁨

이세연

이재윤

김주선

**7.5.2 팀원 평가**



우재하

한동준 – 막내이지만 형 같은 모습으로 리더쉽있게 팀을 이끌어가는 모습을 보니 배울점이 매우 많았습니다. 리더의 자질과 팀원들을 이끌어가는게 멋진 친구라고 생각합니다.

유기쁨 – 팀장으로서 번거로운 일들을 많이 맡았는데, 불평없이 책임지고 하는 모습이 인상적이었다. 항상 먼저 프로젝트 진행상황을 확인하고 팀원 간에 자유롭게 의견을 나누는 분위기를 만들어주었다. 또한 팀원들의 사기를 북돋아주어 팀원으로서 의지할 수 있는 리더였다. 개발 경험이 적음에도 스스로 공부하고 일을 찾아서 맡는 등 팀에 많은 도움이 되었다.

이세연 – [주제선정, ppt, jetson nano 환경 구축]: 천방지축이지만 엉뚱하면서도 꾸준하고 성실했던 것이 너의 스타일이었고, 네가 있었기에 팀 분위기도 활발하게 돌아갔던 것 같다. 과묵한 나나 자기 주관이 뚜렷한 재윤이는 하지 못했을 역할인 것 같아. 네가 교육 도중 기업에서 입사하라는 권고를 거절했을 때, 그렇게까지 교육에 참여하고 프로젝트에 남아있고 싶어하는 너를 보고, 나는 네가 할 수 있는 부분에서 아낌없이 도와주자고 다짐했었다. 아쉽게도 프로젝트 기간이 짧아서 만족할만한 결과를 만들어내긴 힘들었지만 그래도 네가 있어서 나도 조금 더 프로젝트에 집중할 수 있었던 것 같다. 네가 더 좋은 회사로 갈 수 있었으면 하는 마음도 있었고, 다행히 네가 원하는 회사로 갈 수 있게 된 것 같아서 나도 기쁘다. 어딜 가더라도 네가 소통하는 인재로 통할 것 같으니 걱정은 되지 않네.

이재윤 – 교육 중에도 학교 생활하면서 어려운 AI 분야에 대해 공부하느라 힘들었겠지만, 포기하지 않고 우리와 함께 교육과정 열심히 듣고 매일 강의장도 일찍 오는 부지런한 모습에 한 팀이 되었을 때도 팀에 큰 도움이 될 거라고 생각했는데 역시나 저희 팀이 처음 정해졌을 때 먼저 나서서 팀장도 하겠다고 말하는 모습을 보면서 책임감 뿐만 아니라 리더쉽이 뛰어나다고 생각했습니다. 항상 팀의 분위기를 이끌어나가며 팀장으로서 팀원들이 지치지 않도록 도와주었습니다. AI 쪽 영상 관련 프로젝트를 이번에 같이하면서 인턴도 영상관련 기업에 정해졌는데, 어렵더라도 교육기간 때의 우재하처럼 포기하지 않고 성실하게 열심히 기업가서도 배우면 많이 성정할 것 같습니다.

김주선 - 학교도 다니면서 팀장 역할을 성실히 하는 모습이 책임감이 있다고 생각합니다 또한 하고싶은 것을 이뤄내는 모습은 대단하다고 생각합니다

한동준



우재하 – 동준이형 그 동안 고생 많았어! 형은 항상 자신이 부족하다고 말하지만 누구보다 잘 하고 있고 프로젝트를 진행하면서 상당히 많이 의지할 수 있는 사람이었어. 자신보다 남을 배려할 줄 알고 맡은 역할에 대해서는 책임감을 가지고 일하는 모습 또한 정말 인상 깊었던 것 같아. 앞으로도 계속 연락하면서 지냈으면 좋겠고, 오른 장인 되길^^.

유기쁨 – 개발 경험과 전공지식이 많아 프로젝트 진행할 때 개발 방향을 잡고 설계하는데 주도적인 역할을 했다. 다른 팀원들이 개발 중에 어려움을 겪을 때 먼저 나서서 해결해주고 팀원들이 좌절하지 않게 중심을 잡아주었다. 프로젝트 이외에도 평소 전공에 대한 전문성을 높이려고 노력하는 모습이 보기 좋았다.

이세연 – [앱 개발과 데이터 베이스 연결 및 연동]: 평소 옆자리에 앉아서 많은 것을 알려준 친구. 교육 받을 때도 모르는 코드나, 알고리즘 적인 해결 방법을 몰라 해매고 있을 때 많이 도움 받았네. 평소 자기 자신은 잘 모른다고 자세를 낮추지만 할 때는 또 나도 모르는 코드나 방식대로 자기 자신만의 해결방법대로 길을 해쳐나가지. 그저 강사가 시키는 대로 코드를 쓰거나 전혀 새롭지 않은 방식대로 진행되는 교육에서 너는 자기 자신만의 길을 개척해 나가는 탐험가와 같았다. 그런 점은 그저 정해진 길로만 갔던 내 입장에서 배워야 할 교훈이지 않을까. 프로젝트에서 앱 개발을 거의 혼자서 하면서 정말 어려움이 많았을 텐데 정말 고생 많았다. 그래도 물론, 케익 위에 딸기를 올려놓는 일이었지만, jetson환경에 연결하는 건 내가 어느 정도 도와줄 수 있어서 다행이었다.

이재윤 – 교육기간 중 알고리즘 스터디를 함께 하면서 같이 밤늦게까지 알고리즘 문제를 풀고, 공부하면서 창의성이 뛰어나다고 생각하였는데, 역시나 수업 중에서도 조교님의 방식으로 아닌 자기만의 방식으로 코딩 해나가는 모습을 보면서 정말 잘한다고 생각하였습니다. 이번에 저희 팀에서 App 개발 부분을 유기쁨 팀원과 함께 담당하였는데 어렵고 시간이 매우 오래걸리는 일이지만, 항상 묵묵히 자기가 맡은 부분을 끝까지 성실하게 완성하였고, 다른 팀원들이 물어보면 항상 대답을 잘 말해주었습니다. 인턴 뿐만 아니라 나중에 직장을 가져도 맡은 부분에 대해서 성실하게 잘할 뿐만 아니라 어려운 일을 쉽게 만들 수 있는 그런 사람이라고 생각합니다.

김주선 - 자기가 맡은 파트에 대해서는 충분한 결과물을 만드는 모습을 보고 저 또한 노력해야겠다고 생각했습니다



유기쁨

우재하 – 팀장으로 많이 부족하지만 옆에서 지켜보며 많이 도와준 거 같아 정말 고마웠어. 프로젝트를 진행하는 동안 프로젝트에 대한 방향성을 제시해 주고, 잘 할 수 있다고 격려해줘서 정말 큰 힘이 됐어. 또한, 누나 역할 외에도 다른 팀원 역할까지 잘 도와줘서 배울 점이 많다고 생각해. 앞으로 인턴 3개월도 잘 지내면서 의지 할 수 있는 동료가 될 수 있도록 할게. 프로젝트 정말 고생 많았고, 앞으로도 화이팅!

한동준 – 같이 일을 하게 되면 시너지가 좋을꺼 같습니다. 개발을 같이 하면서도 능력이 뛰어나고 제가 모르는 부분이나 막혀있을 때 나침반 같은 역할을 해주어서 큰 도움이 된거 같습니다.

이재윤 – 알고리즘 스터디할 때 저희 스터디원들을 잘 이끌어 나갔으며, 항상 주도적으로 공부해야할 것들을 잘 알려주었으며, 실력도 엄청 뛰어났습니다. 그래서 저희 팀에서도 팀장 다음으로써 저희 팀원들을 이끌어나갔으며, 의지가 되었습니다. App개발 뿐만 아니라 저희 얼굴인식 모델 분야까지 전반적으로 저희 프로젝트에 가장 큰 힘이 되었습니다. 저가 만약 회사의 사장이라면 아마 유기쁨 팀원을 꼭 데려가고 싶을 정도로 어떤 직무를 하더라도 엄청난 활약을 할 것이라고 생각합니다.

이세연 – [app개발, 얼굴 인식 모델 조사, ppt]: 알고리즘이나 기타 교육 과목에서 모두 몇 시간은 걸리는 문제를 손쉽게 풀어낼 때, 상당한 엘리트라는 걸 느꼈습니다. 우리 조에서는 없어선 안 될 최고의 인재인 셈이죠. 기쁨 씨가 없었다면 작고 효율 좋은 얼굴인식 모델을 찾기 어려웠을 것 같아요. 앱을 개발하는 역할을 함과 동시에 저에게 기존 모델을 찾아 시험해 볼 수 있게 조언해 주셨으니, 다방면에서 활약하신 거군요. 아마 어딜 가시더라도 엄청나게 활약할 수 있으실 거라 믿어 의심치 않습니다.

김주선 - 5개월동안 짝꿍으로 가까이에서 보고 느낀 점도 많았고 배울 점도 많았습니다 교육을 들으며 많은 도움을 받은 친구였습니다



이세연

우재하 – 안녕하세연! 형 일단 5개월 동안 정말 고생 많았고, 장난도 많이 쳤지만 잘 받아줘서 진짜 고마워! 형은 무엇을 하든 잘 해낼 걸 알기에 별로 걱정이 안돼. 하나에 꽂히면 그걸 성공시킬 때까지 노력하는 모습이 진짜 보기 좋았어. 다른 팀원들에게도 전부 배울 점이 많았다고 했는데, 형한테도 정말 배울게 많은 거 같아. 그냥 입에 발린 말 하는 게 아니고 진짜야!!!! 구미에서 자주자주 봤으면 좋겠어! 몸 조심하고 안녕~!

한동준 – 비전공자 이지만 계속 보면볼수록 하루하루 성장하는 모습이 매우 뛰어났습니다. 어떠한 학문을 공부하고 구조를 이해하고 문제해결을 하는 것이 재능이 있다고 저는 생각합니다.

유기쁨 – 팀원 중 누구보다 열심히 인공지능과 관련된 배경지식을 공부해오고 좋은 의견을 내서 개발 방향을 잡는데 많은 도움이 되었다. 또한 환경 구성이나 개발 중 오류가 발생했을 때 몇시간이고 묵묵히 노력해서 문제를 해결하는 모습이 멋있었다. 전공생이 아님에도 앞으로의 발전이 많이 기대된다.

이재윤 – 처음에는 데이터분석에 관심이 많았는데, 어떻게 하다가 팀의 주제가 영상처리분야에로 정해졌는데, 영상처리 분야에서도 흥미를 가지고 그 누구보다도 열심히 해주어서 정말 고마웠습니다. 그리고 주제에 몰두하여 그 분야에 대해 남들보다 더 열심히 파고들고 공부하는 모습을 보고 배울 점이 많다고 생각하였습니다. 자리가 멀리 떨어져있어서 대화할 일이 교육 초반에는 많지 않았지만, 저도 한가지 일에 몰두하면 그 일을 해결하지 못하면 다른 일에 집중이 잘 안되었는데 그 덕분에 새벽까지 같이 교육장에 남아 프로젝트를 같이 하는 일이 많아졌고, 담당분야도 얼굴인식 모델로 서로에게 많은 도움이 되었을 거라고 생각합니다. 아직 관심 있고 흥미 있는 분야가 많은 것 같아 정말 관심 있는 분야가 하나 정해지면 큰 성과를 이루어 낼 것 같습니다.

김주선 - 관심 있는 한 분야에 몰두하여 완벽히 본인의 것으로 만드는 모습을 보며 제 자신을 뒤돌아보는 계기가 되었습니다



이재윤

우재하 – 처음에는 형이 혼자 다니길래 우리랑 별로 안 친해지고 싶어하는 줄 알고 쉽게 못 친해진 것 같아. 형이나 세연이형은 뭘 해도 잘 해낼 거 같아서 별 걱정이 없다!! 프로젝트 진행하면서 형한테 질문도 많이 했는데 잘 답변해줘서 고맙고, 형 덕분에 프로젝트 잘 마무리 할 수 있었던 것 같아. 잠도 좀 자고 건강도 좀 챙겼으면 좋겠어 사랑해♥

한동준 – 팀원들 중에서 가장 열심히 하고 열정이 있었습니다. 맏형으로써 리더쉽도 있고 문제해결능력이 매우 뛰어났습니다. 비전공자이지만 전공자보다 알고 있는 것들이 많다고 느껴졌습니다. 항상 파이팅 넘치고 프로젝트를 진행할 때 센스가 좋아서 팀원들이 어려움을 겪고 있을 때 큰 도움이 됐습니다. 꼭 나중에 같이 일을 했으면 좋겠습니다.

유기쁨 – 팀원 중에 가장 인공지능 관련 경험이 많아 딥 러닝 모델을 활용하고 코드로 구현할 때 큰 역할을 했다. 또한 밤을 새며 코드를 완성해낼 때 개발에 대한 순수한 열정과 집념이 보였으며, 새로운 것을 배우고 적용할 때 두려움없이 임하는 모습에서 개발자로 배울 점이 많았다.

이세연 – [얼굴인식 및 출석 인증 코드 작성, jetson 환경 구성, ppt]: 문제해결능력이 우리 중에서 가장 좋았던 것 같다. 나였다면 뭐 하나만 해도 몇 시간은 걸리는 문제들을 금방금방 해결해 나가는 걸 보면 이게 사람인가 싶기도 했고, 프로젝트 주제에 대해서도 네가 생각한 것 보다 다른 조원들이 생각한 주제를 더 심도 있게 들어 보려고 노력하는 게 보이더라. 주제 선정 후, jetson nano나 tx2 같은 필요한 도구나 장치들을 먼저 밤낮없이 다뤄보고 나나 재하한테 넘겨줘서 좀 더 시간을 절약할 수 있었던 것 같아. 코드를 짜는 것도 네가 도와주지 않았더라면 아마 많이 힘들었을 것 같다. 프로젝트를 좀 더 일찍 시작했었더라면 분명 좀 더 많은 역할을 도맡아서 하려 했겠지. 다음에 같이 또 뭔가를 같이 만들 날이 왔으면 좋겠다. 그 때는 나도 네가 하는 걸 미리 알아채고 도와줄 수 있을 것 같다.

김주선 - 앞서나가서 조원들의 부족한 부분을 도와주고 프로젝트 관련하여 많은 경험을 공유해주는 모습을 보며 저 또한 많은 것을 배웠습니다



김주선

우재하 – 안녕 누나! 5개월 동안 정말 고생 많았어. 밝은 성격으로 항상 웃게 해줘서 고마워!!! 앞으로 누나 하는 일 전부 잘됐으면 좋겠고 건강했으면 좋겠어!

한동준 – 처음 같이 면접볼때는 의욕이 넘치고 정말 열심히 할려는 모습을 보았습니다. 프로젝트가 진행 중 어려움이 닥치자 관심이 떨어지는 모습이 아쉬웠습니다. 하지만 본인이 좋아하는 일을 찾으면 매우 열정 있고 잘할 꺼 같은 느낌을 받았습니다.

이세연 – [ppt]:주선아, 맨 처음 우리 조별 회의와 피피티 작성에서 도와줬었는데 사건 사고가 많아서 후반에는 거의 볼 수가 없었구나. 너 역시 AI에 관심이 많았던 것 같았는데 끝까지 함께할 수 없다니 아쉽다. 분명 통계학과인 너의 역할 역시 중요했을 텐데 말이야. 다음에 우리가 또 함께 뭔가를 만들 날이 온다면, 그땐 너의 활약을 볼 수 있겠지. 나는 그때까지 네가 어떤 불행이 닥쳐도 이겨내고 너만의 길을 개척해 나갔으면 좋겠다.

이재윤 – 처음에는 의욕도 앞서고 열심히 할려는 모습을 보았습니다. 끝까지 같이 함께 했으면 좋았을 텐데 큰 아쉬움이 많이 남았습니다.

유기쁨 – 프로젝트 중후반부터 참여율이 적어 아쉬웠다.

**Chapter 8. 참고사항**

**[jetson nano, tx2 환경에서의 설치]**

[tensorflow version=1.15]

nvidia 포럼

https://devtalk.nvidia.com/default/topic/1048776/jetson-nano/official-tensorflow-for-jetson-nano-/

[pytorch version=1.2.0 with torchvision version=0.4.0]

nvidia 포럼

https://devtalk.nvidia.com/default/topic/1049071/jetson-nano/pytorch-for-jetson-nano-version-1-4-0-now-available/

주의할 점: pytorch 버전에 맞는 torchvision 버전을 무조건 맞춰줘야 함

[opencv 4.1 with cuda]

영상 참조

https://www.youtube.com/watch?v=tFGZjVUR\_Ck&t=257s

github 주소

https://github.com/jetsonhacksnano

-installSwapfile

Jetson nano는 최소 swap 메모리 6기가 이상으로 설정

-buildOpenCV

buildOpenCV.sh 파일에서,

22line에 NUM\_JOBS=$(nproc)를 원하는 cpu 코어만큼 설정 가능.

Jetson nano는 총 코어가 4개 이므로 많아도 2 코어 이상 쓰지 않도록 설정.

만약 코어 4개를 전부 사용하게 되면 Jetson nano가 버거워함.

영상에서는 1개를 추천하고 있음. tx2에서는 3~4개로 실행 가능.

ex) NUM\_JOBS=1

[onnxruntime] - Option(=필수 아님)

desktop에서는 onnxruntime이 pip로 깔리지만 jetson환경에서는 깔리지 않는다.

https://github.com/microsoft/onnxruntime

build.md 참조

jetson 환경에서 설치하는 가이드 라인 사이트

https://github.com/microsoft/onnxruntime/issues/2684#issuecomment-568548387

주의할 점:

tx2와 nano의 arch number가 다르다는 점.

build.sh를 실행할 때, cmake의 c++ compiler library 버전이 맞지 않아 error가 뜰 수 있다.

그 때는 맞는 cmake 라이브러리를 설치해주자.

**Chapter 0. 참고자료**

* Sheng Chen, ‘MobileFaceNets: Efficient CNNs for Accurate Real-Time Face Verification on Mobile Devices’, <https://arxiv.org/ftp/arxiv/papers/1804/1804.07573.pdf>, arXiv, 2018.
* 익명, ‘[Paper] MobileNetV2: Inverted Residuals and Linear Bottlenecks’, <https://seing.tistory.com/58>, 2020.02.27[검색].
* melonicedlatte, ‘MobileNet이란? 쉬운 개념 설명’, <http://melonicedlatte.com/machinelearning/2019/11/01/212800.html>, 2019.01[검색].
* 강진규,’북한,ARM CPU 기반 딥러닝 얼굴인식기술 개발’, <https://www.nkeconomy.com/news/articleView.html?idxno=1976>, 2019.09.25[검색].
* 김형일, 문진영, 박종열, ’딥러닝 기반 고성능 얼굴인식 기술 동향’, <https://ettrends.etri.re.kr/ettrends/172/0905172005/33-4_43-53.pdf>, ETRI, 2018.

Project github

<https://github.com/haspberry/hustar_project>

App github

<https://github.com/dongjuny/fams>