

마이크로프로세서 기반의 얼굴 마스크 감지

Face-Mask Detection with Micro processor

Hyunkeun Lim¹ · Sooyoung Ryoo¹ · Hoekyung Jung^{2*}¹Graduate Student, Department of Computer Engineering, Paichai University, Daejeon, 35345 Korea^{2*}Professor, Department of Computer Engineering, Paichai University, Daejeon, 35345 Korea

ABSTRACT

This paper proposes an embedded system that detects mask and face recognition based on a microprocessor instead of Nvidia Jetson Board what is popular development kit. We use a class of efficient models called Mobilenets for mobile and embedded vision applications. MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks. The device used a Maix development board with CNN hardware acceleration function, and the training model used MobileNet_V2 based SSD(Single Shot Multibox Detector) optimized for mobile devices. To make training model, 7553 face data from Kaggle are used. As a result of test dataset, the AUC (Area Under The Curve) value is as high as 0.98.

Keywords : TinyML, Maixduino, Micropython, MobileNet, Mask detection

I. 서 론

본 논문은 마이크로 프로세서 기반의 코로나 온도측정 기기에 딥러닝 기법을 기반하여 얼굴인식과 마스크 착용 여부 판별 시스템을 제안하였다. 기기에 부착된 카메라로부터 입력되는 실시간 영상 데이터로부터 모델 판별 알고리즘을 통해 얼굴인식과 마스크 착용 여부를

판별하여 화면에 검출된 이미지 영역을 표시한다.

사용한 모델은 MobileNet V2[1]를 사용한 SSD(Single shot Multibox Detector)[2]를 적용하여 마이크로 프로세서 기기에서 실시간으로 실행 가능하도록 구현하였다.

II. 본 론

본 논문은 개발환경 구성과 데이터 모델링에 대한 데이터 구축 학습 변환 과정의 내용으로 구성된다.

2.1. 임베디드 기기

사용된 기기로 Maixduino는 Maix Module을 기반으로 AI+IoT 애플리케이션 용 RISC-V64 개발 보드로 표 1과 같다.

Table. 1 Maixduino Specification

CPU	RISC-V DualCore 64bit 400MHz
Interface	WiFi, Bluetooth 4.
Memory	Flash 16MB, RAM 6MB
Support Model	Tiny-Yolo, MobileNet, TensorFlow Lite
Camera	2M pixel
LCD	2.7 inch TFT
Develop Tool	MaixPy, Arduino, OpenMV, PlatformIO

Maix 모듈의 차별화 기능은 모듈의 이중 코어 프로세서에 통합된 CNN(Convolutional Neural Network) 가속기에 있다. 개별 커널기능의 기본 CNN을 빠르게 처리하도록 설계된 신경망 프로세서(Kernel Processing)는 CNN 모델의 개별 계층을 구성하는 나선, 배치 정규화, 활성화 및 풀링 커널 기능을 하드웨어로 구현한다[3].

본 논문에서 사용한 개발 도구로 MaixPy를 사용하였다. MaixPy는 Micro Python 개발환경을 지원하는 실시간 마이크로 컨트롤러 기반 애플리케이션의 개발을 위한 Maix 전용 Python 개발 환경이다.

Received 31 January 2021, Revised 8 February 2021, Accepted 5 March 2021

* Corresponding Author Hoekyung Jung(E-mail:hkjung@pcu.ac.kr, Tel:+82-42-520-5640)
Professor, Department of Computer Engineering, Paichai University, Daejeon, 35345 Korea

Open Access <http://doi.org/10.6109/jkiice.2021.25.3.490>

print ISSN: 2234-4772 online ISSN: 2288-4165

2.2. 모델 개발

마이크로 프로세서에 최적화된 모델로 CNN 계열 중 가볍고 속도가 빠른 MobileNet을 사용한 SSD(Single Shot Multibox Detector) 모델을 사용하였다.

이전 버전인 Mobilenet v1[4]은 기존의 컨벌루션(Convolution) 연산을 대체하여 DC(Depthwise Convolution) 과 PC(Pointwise Convolution) 방법을 사용한다. DC는 동일 채널 내에서 컨벌루션을 하는 방법이고, PC는 채널 간에서 1x1 컨벌루션을 하는 방법이다. MobileNetV1은 Google에서 2017년 개발한 모델로서 기존 네트워크 모델에 비하여 적은 연산량을 가지고도 비슷한 성능을 낸다. 또한 Mobilenet v2는 Mobilenet v1의 속도 저하 원인이었던 PC의 부담을 인식하고 DC의 연산 비중을 늘리는 방법을 사용한다. 사용된 모델의 계층 구조는 표 2와 같다.

Table. 2 Model Layer

Input layer	224x224 feather map, 3 Input Channel, 3x3 Kernel,7x7 Output Pooling
Feature classifier	used MobileNet V2, 19 layer
Classifier	avgpool 1 layer
Dense	1 layer
Output layer	1 layer

모델 개발은 Tensorflow 2.x 프레임워크를 사용하여 학습 및 검증한 후, 모델 성능을 평가하기 위해 ROC-curve와 AUC값을 구하였다.

2.2.1. 학습 데이터 구성

학습에 사용된 데이터는 Kaggle에서 공개된 7553개 [5]의 얼굴 데이터를 사용하여 학습 데이터와 검증 데이터로 구성하였다.

학습 데이터 구조는 각각 2개의 class(mask 착용/미착용 데이터)를 하위 폴더로 갖는다. 학습과 검증은 1:2 비율로 표 3과 같이 랜덤 구성하였다.

Table. 3 Dataset

Face	Without mask	With mask
Training	1930	2165
Verification	3828	3725

2.2.2. 모델 학습

학습에 사용된 하이퍼 파라미터 설정은 표 4와 같다.

Table. 4 Hyperparameter

Hyper Parameter	value
Backbone Name	MobileNetV2, SSD
Epoch	20
Bach Size	32
Optimaizer	Adam
learning Rate	0.0001
Loss function	Binary Crossentropy

학습은 I7-7700 GTX1060 16G PC에서 실행하여 그림 4와 같이 20번 학습하여 18분 소요되었다. 학습 과정은 그림 1과 같다.

```

Epoch 1/20
63/63 [=====] - 52s 829ms/step - loss: 0.0633 - accuracy: 0.9776
Epoch 2/20
63/63 [=====] - 54s 852ms/step - loss: 0.0445 - accuracy: 0.9841
Epoch 3/20
63/63 [=====] - 54s 856ms/step - loss: 0.0463 - accuracy: 0.9821
Epoch 4/20
63/63 [=====] - 53s 847ms/step - loss: 0.0385 - accuracy: 0.9866
Epoch 5/20
63/63 [=====] - 53s 842ms/step - loss: 0.0383 - accuracy: 0.9841
Epoch 6/20
63/63 [=====] - 54s 852ms/step - loss: 0.0378 - accuracy: 0.9876
Epoch 7/20
63/63 [=====] - 55s 866ms/step - loss: 0.0358 - accuracy: 0.9876
Epoch 8/20
63/63 [=====] - 53s 849ms/step - loss: 0.0394 - accuracy: 0.9866
Epoch 9/20
63/63 [=====] - 53s 843ms/step - loss: 0.0417 - accuracy: 0.9866
Epoch 10/20
63/63 [=====] - 53s 846ms/step - loss: 0.0331 - accuracy: 0.9911
Epoch 11/20
63/63 [=====] - 54s 850ms/step - loss: 0.0383 - accuracy: 0.9841
Epoch 12/20
63/63 [=====] - 53s 834ms/step - loss: 0.0336 - accuracy: 0.9876
Epoch 13/20
63/63 [=====] - 54s 864ms/step - loss: 0.0301 - accuracy: 0.9881
    
```

Fig. 1 Model Training

학습 결과 epoch가 증가함에 따라 train Loss는 감소하지만 validation Loss는 상관없이 약 0.1 정도의 값을 갖는다. 이 결과는 그림 2와 같다.



Fig. 2 Loss

정확도는 그림 3과 같이 train과 validation 둘다 0.95 이상의 높은 정확도를 갖는다.

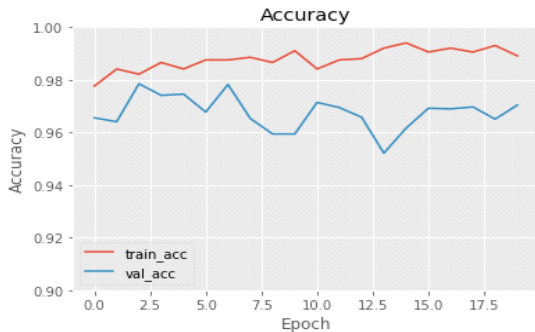


Fig. 3 Accuracy

2.2.3. 모델 평가

Test dataset(with_mask / without_mask)을 활용한 평가는 표 5와 같다.

Table. 5 Evaluation

	with mask	without mask	avg/total
precision	0.69	1.00	0.98
recall	0.99	0.97	0.97
f1-score	0.81	0.98	0.97
support	264	3828	4092

테스트 데이터셋을 활용하여 ROC(Receiver Operating Characteristic) 커브와 아래 면적에 해당하는 AUC(Area Under the Curve) 값을 구한 결과, 그림 4와 같이 AUC값이 약 0.98로 1에 가까운 값을 갖으며 좋은 성능을 보인다.

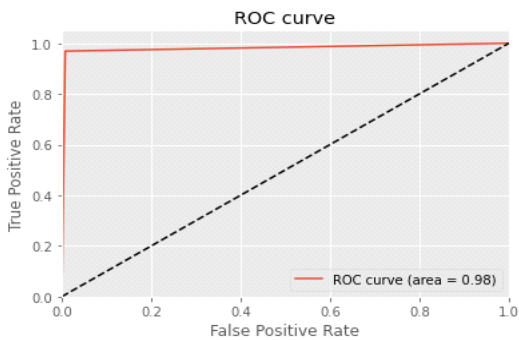


Fig. 4 ROC Curve

2.3. 모델 적용

학습 모델 파일은 3차례의 변환 과정을 통해 Kendryte Model로 전환된다. Keras로 컴파일된 모델 파일은 h5와

PB 확장자를 가지며 TensorLite 버전으로 변경 후 Kendryte Compiler 를 사용하여 최종 kmodel 확장자로 변경하고 Maix 개발 보드에 Import 시킨다. 얼굴과 마스크 검출 결과를 그림 5와 6에 보인다.

III. 결론

학습한 Mask-Detection 모델의 AUC 값이 약 0.98로 높은 값을 보인다. Loss & Accuracy 그래프에 근거하여, Epoch을 7~8 정도로 설정하고 학습할 경우 오버피팅(overfitting) 방지에 더 효율적일 것으로 기대된다.

본 논문은 코로나 출입관리 시스템에 적용할 목적으로 마스크 착용 감시 장치를 개발하였다. 최근 코로나로 인한 출입통제 시스템에 관한 연구가 진행되고 있다[6]. 기존 현재 얼굴 인식 임베디드 기기로 Jetson 개발보드가 가장 많이 사용되지만[7] 발열 문제와 모바일 디바이스보다 상대적으로 높은 가격 그리고 부팅 시간과 상주 메모리가 필요한 OS 기반이라는 단점들이 있다. 임베디드 기기에서의 딥러닝 최적화 연구가 진행되고 있다[8,9]. 본 논문에서 적용한 개발 보드로 저가이면서 CNN 하드웨어 가속 기능 그리고 아두이노와 같은 Form을 사용하는 Maix 개발보드를 사용하였다.

향후 본 기기를 확장하여 센서들을 부착하고 개발된 모델에 센서 데이터를 확장하여 출입관리에 필요한 확장된 모델링을 개발할 예정이다.

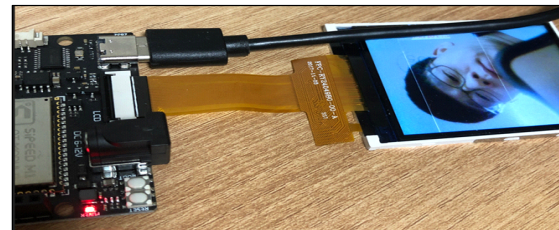


Fig. 5 Face Detection

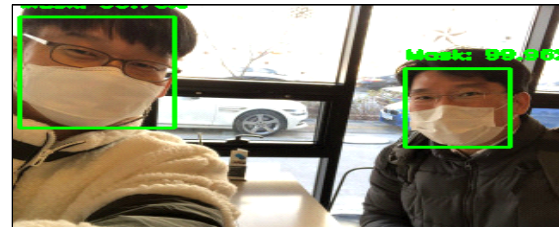


Fig. 6 Mask Detection

ACKNOWLEDGEMENT

This work was supported by the Startup Growth Technology Development Program funded by the Ministry of SMEs and Startups(MSS, Korea).

REFERENCES

- [1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "Ann-Arbor-SSD: Single Shot MultiBox Detector," *University of Michigan*, 2018.
- [2] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Google Inc.*, 2018.
- [3] Use a Low-Cost Module and MicroPython to Quickly Build AI-Based Vision and Hearing Devices. [Internet]. Available: <https://www.digikey.kr/en/articles/use-a-low-cost-module-micropython-ai-based-vision-hearing-devices>.
- [4] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *Google Inc.*, 2017.
- [5] Face Mask Detection Dataset, [Internet]. Available: <https://www.kaggle.com/omkargurav/face-mask-dataset>.
- [6] S. H. Lee, "Deep learning based face mask recognition for access control," *Journal of the Korea Academia-Industrial cooperation Society*, vol. 21, no. 8, pp. 395-400, 2020.
- [7] H. B. Yoo, M. S. Park, and S. H. Kim, "REAL TIME FACE DETECTION METHOD USING TENSORRT AND SSD," *KIPS Transactions on Software and Data Engineering*, vol. 9, no. 10, pp. 323-328, Oct. 2020.
- [8] M. S. Pak and S. H. Kim, "Face Detection using Deep Learning in Embedded Platform," *Proceedings of the Korea Information Processing Society Conference*, pp. 827-829, 2018.
- [9] H. K. Kim, J. Y. Kim, and H. K. Jung, "Convolutional Neural Network Based Image Processing System," *Journal of Information and Communication Convergence Engineering*, vol. 16, no. 3, pp. 160-165, Sep. 2018.