The Korean Journal of Applied Statistics
2021, Vol. 34, No. 6, 979–993

DOI: https://doi.org/10.5351/KJAS.2021.34.6.979
Print ISSN1225-066X / Online ISSN 2383-5818

Light weight architecture for acoustic scene classification

Soyoung Lim[a], Il-Youp Kwak[1,a]

[a]Department of Applied Statistics, Chung-Ang University

Abstract

Acoustic scene classification (ASC) categorizes an audio file based on the environment in which it has been recorded. This has long been studied in the detection and classification of acoustic scenes and events (DCASE). In this study, we considered the problem that ASC faces in real-world applications that the model used should have low-complexity. We compared several models that apply light-weight techniques. First, a base CNN model was proposed using log mel-spectrogram, deltas, and delta-deltas features. Second, depthwise separable convolution, linear bottleneck inverted residual block was applied to the convolutional layer, and Quantization was applied to the models to develop a low-complexity model. The model considering low-complexity was similar or slightly inferior to the performance of the base model, but the model size was significantly reduced from 503 KB to 42.76 KB.

Keywords: acoustic scene classification, light-weight model, deep learning, convolutional neural network

1. 서론

소리는 우리의 일상 환경과 그 속에서 일어나는 사건들에 대한 많은 정보를 담고 있다. 우리는 소리를 통해 우리가 있는 장소를 지각할 수 있다. 예를 들면, 차가 많은 거리에 있다거나 조용한 사무실에 있다는 사실 등을 소리로 짐작할 수 있다. 이러한 정보를 컴퓨터가 자동으로 처리하도록 하는 기술을 음향 장면 분류(acoustic scene classification, ASC)라고 한다. 즉, 음향 장면 분류란 오디오 파일이 녹음된 환경이 어디인지 분류하는 문제이다.

최근 몇 년간 음향 장면 분류는 미국 전기전자기술자협회-오디오 및 음향 신호 처리(Institute of electrical and electronics engineers audio and acoustic signal processing, IEEE AASP)의 관심을 끌고 있다. IEEE ASSP에서 주최하는 detection and classification of acoustic scenes and events (DCASE) 대회가 2013년부터 현재까지 매년 개최되고 있다. DCASE 대회에서는 매년 음향 장면 분류 주제를 다룬다. 대회에 참가하는 팀도 늘어나는 추세이다.

음향 장면 분류 기술은 자율 주행 차량 개발이나 청력이 저하된 사람들을 위한 공간 지각 서비스 등 많은 응용 분야에 사용될 수 있다 (Suh 등, 2018). 또한, AI 음성비서가 주변 상황을 인지할 수 있게 하여 음성비서 서비스의 역할 확대를 기대할 수 있다. 여러 분야에서 큰 잠재력을 가지고 있는 음향 장면 분류 시스템에 대한 연구가 활발히 진행되고 있다.

This research was supported by the National Research Foundation of Korea (NRF) grant funded by Ministry of Science and ICT (2020R1C1C1A01013020).

[1] Corresponding author: Department of Applied Statistics, Chung-Ang University, 84, Heukseok-ro, Dongjak-gu, Seoul 06974, Korea. E-mail: ikwak2@cau.ac.kr

Published 31 December 2021 / journal homepage: http://kjas.or.kr

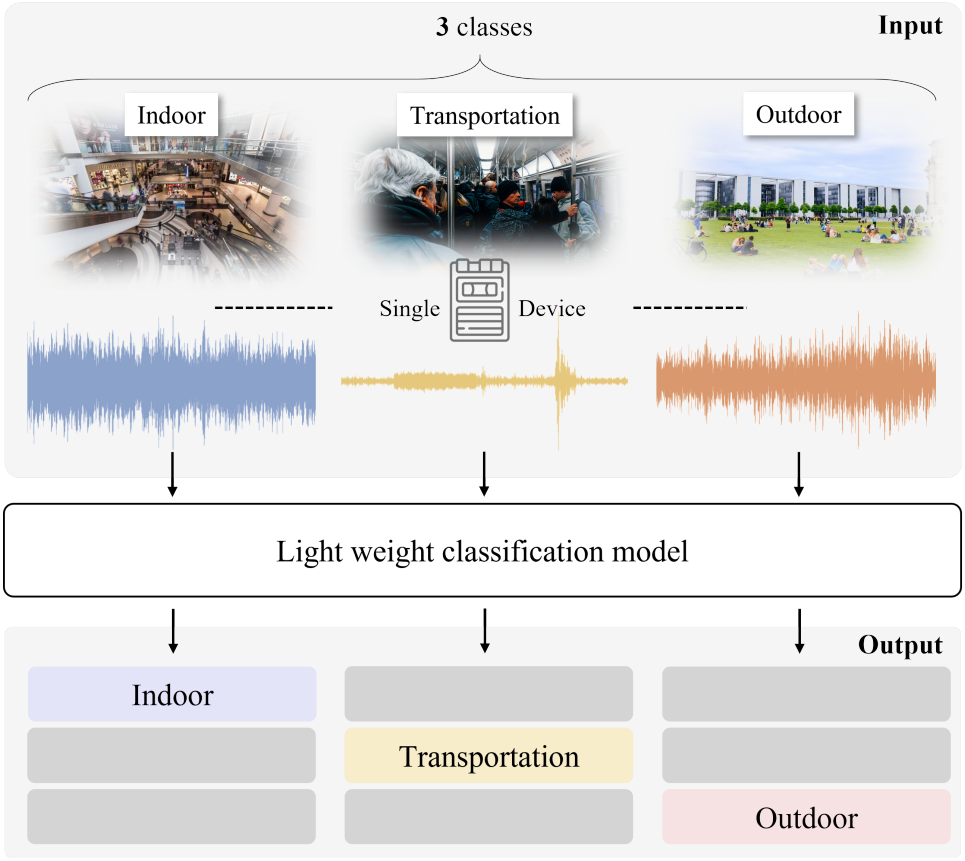


Figure 1: *Overview of the ASC system. Light weight classification model classifies a given environmental audio clip into 3 classes - indoor, transportation, and outdoor.*

최근 딥러닝 기술이 음향 장면 분류 문제에서도 큰 성과를 보였다. 합성곱 신경망(convolutional neural network, CNN) 등 딥러닝 모델과 GPU 기반의 컴퓨팅 기술이 발전하면서 모델의 층이 깊어질수록 뛰어난 모델을 생성할 수 있다고 알려져 있다. 그에 따라 모델 안에 학습해야 하는 파라미터가 늘어나게 되고, 연산량이 늘어나게 되었다. 많은 파라미터 수는 많은 연산량과 학습 시간을 필요로 한다. 학습 시간의 증가는 학습에 요구되는 전산 자원이 필요로 하는 전력을 증대시키기 때문에 에너지 면에서 문제가 될 수 있다. Strubell 등 (2019)은 딥러닝 모델 중 하나인 트랜스포머(transformer) 문제를 학습하여 그 결과물을 얻는 과정에서 쓰이는 전력 생산에 대한 탄소 소비량이 자동차 5대가 구매 시점부터 폐차에 이르기까지 방출하는 탄소 소비량과 같을 정도로 많음을 지적하였다.

또한 강력한 컴퓨팅 자원을 바탕으로 다양한 딥러닝 모델이 만들어지지만, 경량 디바이스, 모바일 디바이스 등과 같은 디바이스에서 직접 학습과 추론이 가능할 정도의 수준은 미미하다 (Lee 등, 2019). 위와 같은 이유로 기존의 학습된 모델의 정확도를 유지하면서 더욱 크기가 작고, 연산을 간소화하는 연구인 경량 딥러닝 연구가 활발히 진행되고 있다. Howard 등 (2017)과 Sandler 등 (2018)은 각각 MobileNet v1, MobileNet v2 를 제안하며 핸드폰에서도 활용 가능한 경량화 딥러닝 모형에 대해 연구하였고, Zhang 등 (2018)에서는 MobileNet v2 의 inverted residual block 을 개량한 ANTBlock을 제안하였다. 또한, Jan 등 (2021)은 더욱 늘어나고 있는

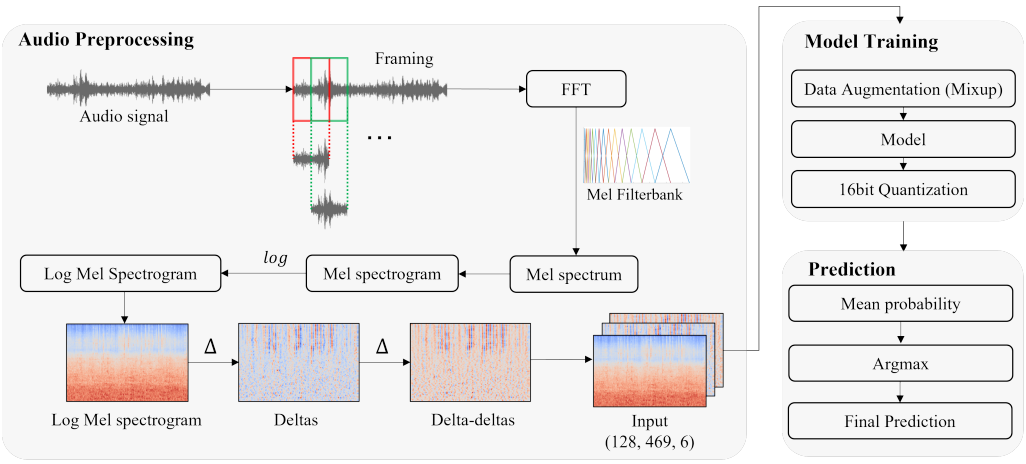


Figure 2: *Overview of proposed system.*

IOT 환경에서 IOT 기기에도 적용할 수 있는데 경량화된 데이터 처리 방향에 대해 연구하였다. 실제로 많은 기업들이 디바이스 자체에서 인공지능이 구현되는 온디바이스(on-device) AI 기술에 관심갖고 있다. 데이터 수집 및 컴퓨팅 파워 여건이 열악한 상황에서 인공지능 기술을 사용할 수 있다는 점에서 주목하고 있다. 온디바이스 AI 기술의 핵심은 경량화이다. 기업에서 온디바이스 AI 기술을 적용하는 사례는 다음과 같다. 세탁기, 건조기에 온디바이스 AI 기술을 적용하여 이용자 개개인의 사용 습관과 패턴을 스스로 학습하고 맞춤형 서비스를 제공한다. 도로 cctv 디바이스에서 직접 교통 혼잡도, 차량 운행량 등의 정보를 실시간으로 분석하여 최적의 교통 신호를 생성하며, 물품 재고 파악, 이상행동 탐지 등 리테일 매장에서 활용할 수 있는 AI 모델이 개발되고 있다.

본 논문에서는 음향 장면 분류 문제에 대한 합성곱 기반 모델을 제안함과 동시에 다양한 딥러닝 경량화 기법을 비교한다. 우리가 다루고자 하는 문제를 Figure 1에 표현하였다. 오디오 데이터 분류 문제를 실제에 적용할 때 일어날 수 있는 문제에 대하여 다뤘다는 점에서 본 연구는 의미가 있다. 즉, 실제 적용에 있어서 한정적인 자원의 문제와 경량 디바이스에서 직접 학습과 추론이 가능하도록 해야하는 경우가 있기 때문에 경량 모델은 연구되어야 하는 분야이다. 본 연구에서는 두 개의 기본 모델을 비교하고 각 모델에 대하여 딥러닝 경량화 기법을 적용하고 이를 비교하였다. 사용한 경량화 기법은 효율적인 합성곱 필터를 이용하여 모델을 경량화 하는 방법과 모델 압축 기술인 Quantization 기법이다. 효율적인 합성곱 필터로 depthwise separable convolution과 bottleneck inverted residual 블록을 이용하였다. 재생산 가능한 연구를 위하여 코드는 github에 공개하였다 (https://github.com/ikwak2/Paper_EfficientASC).

2. 연구 방법

음향 장면 분류를 위한 시스템의 전체 과정은 Figure 2과 같다. 이번 세션에서는 오디오 데이터의 전처리 과정을 설명하고 모델 훈련에 사용된 모델을 설명한다. Log mel-spectrogram, deltas, delta-deltas 피쳐를 입력 피쳐로 사용하였다. 음향 장면 분류 문제를 위하여 합성곱 신경망 기반의 모델을 사용하였다. 여러 가지 딥러닝 경량화 기법을 비교하기 위하여 기본 모델을 정하였다. 딥러닝 경량화 기법으로는 효율적인 합성곱 필터로 제안된 depthwise separable convolution과 bottleneck inverted residual 블록을 적용하였다. 또한 각 모델에 대하여 16bit로 Quantization 하는 모델 압축 기술을 사용하였다.

2.1. 오디오 데이터 전처리 과정

오디오 신호를 합성곱 신경망에 적용하기 위해서는 웨이브폼 형태의 오디오 신호를 이미지로 변환해주어야 한다. 여러가지 피쳐가 사용되고 있지만 DCASE 2019에 참여하였던 Gao와 McDonnell 팀 (McDonnell과 Gao, 2020)이 사용한 피쳐인 log mel-spectrogram, deltas, delta-deltas를 사용하였다. 동일 모델에 오디오 데이터에 많이 사용되고있는 harmonic-percussive source separation (HPSS) 피쳐와 CQT 피쳐를 입력 피쳐로 사용하였지만 log mel-spectrogram, deltas, delta-deltas 피쳐의 성능이 더 좋았기 때문에 이 피쳐를 입력 피쳐로 사용하였다. 일반적으로 오디오 신호의 연속적 특성을 반영하기 위하여 log mel-spectrogram에 시간축에 대해 미분한 deltas, delta-deltas 값을 사용한다. 이후 DCASE 2020 task 1 A에서 우수한 성능을 기록한 Suh 팀 (Suh, 2020) 외의 많은 팀이 deltas, delta-deltas 피쳐를 사용한 것을 확인하였다. 음향 장면 분류 문제에 대하여 log mel-spectrogram, deltas, delta-deltas 피쳐가 잘 작동하는 것을 알 수 있다.

오디오 신호를 log mel-spectrogram으로 만들어주는 방법은 다음과 같다. 먼저 오디오 신호를 1초에 몇 번 샘플링 했는지 나타내는 지표를 sampling rate (sr)이라고 한다. 오디오 신호를 짧은 시간 단위로 잘게 쪼개는 과정을 framing이라고 한다. 이때 프레임을 일정 시간 단위(window size)로 자르되 일정 구간은 겹치도록 한다. 겹치는 정도는 hop length 로 조정되는데, hop length가 window size 의 반이라면 50% 씩 겹치면서 프레임들을 만들어 나가게 된다. 본 논문에서 사용된 오디오 데이터는 48,000Hz의 sampling rate, 입체 음향(binaural)으로 녹음이 된 10초의 오디오 파일이다. 약 42ms의 프레임 간격으로 쪼개지며 약 21ms 겹쳐지는 부분이 있게 framing을 하였다.

잘게 쪼개진 각 프레임에 대하여 Hann window를 적용한 후, 고속 푸리에 변환(fast fourier transform, FFT)을 사용하여 short time fourier transform (STFT) 스펙트로그램 피쳐를 구하게 된다. 이 STFT 피쳐는 시간(time)과 주파수(frequency)의 2차원적 정보를 가지는 데이터가 된다.

사람은 소리를 인식할 때 달팽이관 특성에 따라 1,000Hz 이하의 저주파수(low-frequency) 영역 대를 고주파수(high-frequency)에 대비하여 민감하게 인식한다. 사람이 인지하는 음의 높낮이는 주파수(Hz)와 선형(linear) 관계가 아니라 지수적(exponential) 관계이다. 따라서 주파수(Hz)를 log 함수를 이용하여 비선형 변환을 해주는데 이것이 멜 스케일(mel scale)이다. 주파수(f)를 멜(m)로 바꿔주는 수식, 즉 멜 스케일 함수는 수식 2.1과 같고, 멜(m)을 주파수(f)로 바꿔주는 수식은 2.2이다.

$$m = 2595 log_{10}\left(1 + \frac{f}{700}\right), \tag{2.1}$$

$$f = 700\left(10^{\frac{m}{2595}} - 1\right). \tag{2.2}$$

헤르츠 단위의 주파수 k를 멜 단위 주파수 m에 대응시키는 필터를 만드는 공식은 수식 2.3과 같다. 필터를 그래프로 표현하면 Figure 2의 과정 중 멜 필터뱅크(Mel Filterbank) 그래프와 같다.

$$H_m(k) = \begin{cases} 0 & k < f(m-1), \\ \frac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1)(m), \\ 1 & k = f(m), \\ \frac{f(m+1) - k}{f(m+1) - f(m)} & f(m) < k(m+1), \\ 0 & k > f(m+1). \end{cases} \tag{2.3}$$

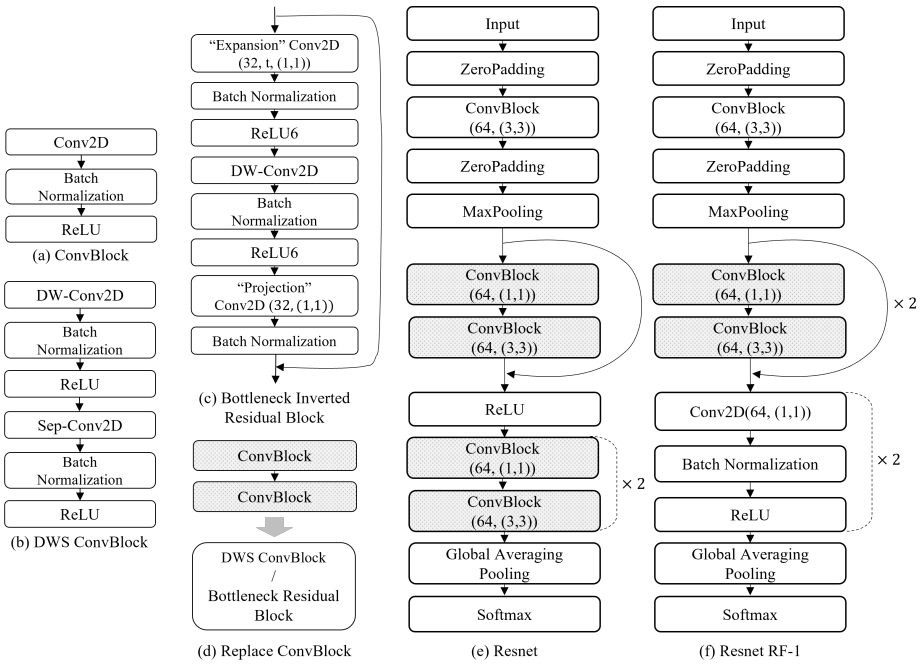


Figure 3: *Model architecture. (a), (b) and (c) describe submodule convblock. (a) is the traditional convblock using 2 dimensional convolution, (b) and (c) are modified efficient version of convblocks. (b) describe depthwise separable convblock and (c) describe bottleneck inverted residual convblock. (d) is proposed Resnet model. (e) is proposed Resnet RF-1 model.*

이 필터의 개수를 정해주는 파라미터가 n_mels이다. n_mels개의 필터는 헤르츠 기준으로 주파수 영역대별로 세밀하게 살피는 필터가 있고 넓게 살피는 필터로 구성되어있다. 우리는 n_mels는 128을 사용하여 128개의 멜-스케일(mel-scale) 필터를 사용하였다. 멜 스케일 필터를 앞에서 생성한 STFT 스펙트로그램 피쳐와 내적하는 필터 뱅크 기법을 적용하면 mel-spectrum을 얻을 수 있다. 이를 시간별로 나열하여 2차원의 이미지로 만든 것이 mel-spectrogram이다. Mel-spectrogram을 로그변환 한 것을 log mel-spectrogram이라고 한다. Log mel-spectrogram을 시간 차원에 대해 한 번 미분한 것이 Deltas피쳐, 두 번 미분한 것이 delta-detlas피쳐이다. 이 세개의 피쳐에 입체 음향 데이터이므로 채널 수가 6이 되고, (주파수, 시간, 채널) 차원의 입력피쳐는 (128,469,6)차원으로 계산되어 사용하였다.

2.2. 기본 모델 구조

딥러닝의 발전으로 최근 음향 장면 분류 대회인 DCASE에 참가한 대부분의 팀들이 신경망을 이용한 모델을 제안하였다. 2019년 대회에서는 146개의 제출물 중 단 5개만이 딥러닝을 사용하지 않았다 (Heittola 등, 2020). 최근 음향 장면 분류 분야에서 좋은 성능을 보이는 모델은 심층 합성곱 신경망 모델이다 (Koutini 등, 2020; Hu 등, 2020; McDonnell과 Gao, 2020). 합성곱 신경망은 이미지 분야에서 좋은 성능을 보이는 딥러닝 중 하나이며, 웨이브 폼 형태의 오디오 신호를 Spectrogram 이미지로 바꾸면 오디오 데이터에 대해서 사용할 수 있다.

기본 모델 구조는 Figure 3의 (e) Resnet과 같다. 모델 크기를 작게 하기 위하여 층을 얕게 쌓았다. 기본

모델에 사용된 합성곱 블록은 Figure 3의 (a) Traditional convblock이다. 이 블록은 합성곱 층(2D Conv), 배치 정규화(Batch normalization), 활성화 함수인 ReLU로 이루어져 있다. 기본 모델에서의 합성곱은 모두 64개의 필터를 사용하였고 (a) Traditional convblock이 (e) Resnet의 음영 표시된 부분에 들어가며, 하나의 잔차 연결(residual connection)로 이루어져 있다 (He 등, 2016). 이때 처음의 합성곱 블럭 이후에 잔차 연결이 있는 블록부터는 커널 크기가 $(1, 1)$인 합성곱 층을 먼저 쌓고 다음에 커널 크기가 $(3, 3)$인 합성곱 층으로 구성하였다. 이는 연산량을 줄여준다. GoogLeNet (Szegedy 등, 2015)에서 사용하였던 bottleneck layer를 사용하였다. 마지막에는 전역 평균값 풀링 층(global averaging pooling)으로 구성하였고 소프트맥스(Softmax) 함수를 이용하여 각 클래스의 확률을 계산해준다.

또한 음향 장면 분류 분야에서는 수용영역(receptive field)의 크기가 작을 때 더 좋은 성능을 보인다는 실험 결과 (Koutini 등, 2019)가 있어서 Figure 3의 (f) Resnet RF-1과 같이 모델을 구성하였다. 모델 (f)의 경우에는 2개의 잔차 블록과 커널 크기가 $(1, 1)$인 합성곱 층을 2개를 쌓았다. 2개의 잔차 블록에는 앞서 사용하였던 convblock (a)가 사용되었고, 커널 크기가 $(1, 1)$과 $(3, 3)$인 블록으로 이루어져있다. 마찬가지로 모두 64개의 필터를 가진 합성곱이다. 전역 평균 풀링 층과 소프트맥스로 구성하였다.

2.3. 경량 모델 구조 및 방법

딥러닝 경량화 기술은 알고리즘 자체를 적은 연산과 효율 구조로 설계하는 경량 알고리즘 연구와 만들어진 모델의 파라미터를 줄이는 기법인 알고리즘 경량화 연구로 나눌 수 있다 (Lee 등, 2019). 경량 알고리즘 연구로는 합성곱 필터를 효율적으로 변경하여 파라미터를 줄이는 방법들이 연구되고 있다. 효율적인 합성곱 필터로 depthwise separable convolution과 bottleneck inverted residual 블록을 사용하였다. 이러한 효율적인 합성곱 필터를 기존 모델 구조에 적용하였다. Figure 3의 (e)와 (f)의 모델 구조에 음영처리된 convblock들을 depthwise separable convolution을 적용한 convblock (b)와 bottleneck inverted residual을 적용한 convblock (c)로 대체하여 경량화 모형들의 적용에 대해 연구하였다.

알고리즘 경량화 연구는 기존 신경망의 불필요한 파라미터를 제거하거나, 파라미터의 표현력을 잃지 않으면서 기존 모델의 크기를 줄이는 연구 분야이다. 모델 압축 기술 중 Quantization (Han 등, 2016)을 사용하였다.

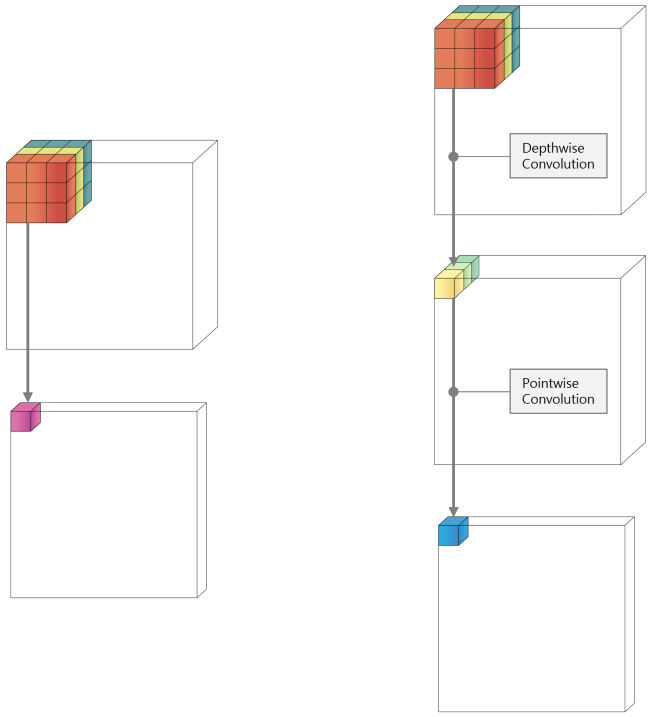
2.3.1. Depthwise separable convolution

Depthwise separable convolution은 Chollet (2017)가 제안한 Xception에서 처음 나온 개념이며, 이후 나온 MobileNet에서도 핵심 아이디어로 사용되었다. Depthwise convolution과 pointwise convolution을 조합하여 사용하는 방식이다.

먼저 depthwise convolution이란 채널마다 따로 필터를 학습하는 합성곱 레이어를 사용하여 각 공간 방향(spatial)의 합성곱을 수행하는 것이다. 일반적인 합성곱은 Figure 4 (a)와 같이 각 채널에 대하여 합성곱 필터를 적용하여 합성곱을 수행한다. 각 정보를 합치게 되어 픽셀당 하나의 채널을 가진 출력 이미지를 만들어내게 된다. 즉, 공간 방향의 연산과 채널 연산을 한 번에 진행한다. 하지만 depthwise convolution 필터의 경우 각 단일 채널에 대해서만 수행되는 필터들을 사용하기 때문에 특정 채널만의 공간 정보만을 얻는 것이 가능해졌다.

Pointwise convolution은 커널 크기가 1×1로 고정된 합성곱 레이어를 말한다. 입력에 대한 공간 방향의 합성곱은 하지 않고, 채널 방향에만 합성곱을 한다. 하나의 필터는 각 입력 채널별로 하나의 가중치만을 가져서 이 가중치는 해당 채널의 모든 영역에 동일하게 적용된다. 이는 입력 채널들에 대한 선형 결합과 같다고 볼 수 있다. 보통 차원 축소를 위해 많이 쓰이고, 연산량을 많이 줄이는 데에 중요한 역할을 한다.

이 두 가지 합성곱을 결합한 depthwise separable convolution은 기존의 합성곱에서 전체 채널 방향과 공간 방향 모두 한꺼번에 고려했던 것과 달리, 채널 방향과 공간 방향을 분리하겠다는 것이다. Figure 4 (b)에 묘사되어있는 것처럼 각 채널에 대한 정보를 먼저 얻고 채널 방향으로 합성곱을 다시 진행한다. Depthwise separable



(a) Standard convolution (b) Depthwise Separable convolution

Figure 4: *Standard convolution and depthwise separable convolution.*

convolution을 사용한 MobileNet v1 (Howard 등, 2017)에서는 depthwise convolution 레이어와 pointwise convolution 레이어 순서로 합성곱 블록을 구성하고, 이 사이에 배치 정규화와 활성 함수 ReLU를 추가하였다.

Figure 3 (b) DWS ConvBlock에 depthwise separable convolution 블록을 묘사하였다. Depthwise convolution과 pointwise convolution으로 구성되어있고 각 합성곱 뒤에는 MobileNet v1에서 처럼 배치정규화, 활성화 함수 ReLU가 있다.

2.3.2. Bottleneck inverted residual convolution

MobileNet v2 (Sandler 등, 2018)는 기존의 MobileNet v1의 아이디어에 추가로 linear bottleneck을 갖는 inverted residual 블록을 제안하였다. 일반적인 잔차 블록은 bottleneck layer 이후 채널을 축소시키지만, inverted residual 블록은 bottleneck layer 이후 채널을 확장시킨다. 저차원 데이터에 합성곱 층을 적용하면 많은 정보를 추출할 수 없기 때문에 저차원으로 압축된 층의 채널을 확장한 후 depthwise separable convolution을 통과시킨다. 그 후에 Figure 5의 마지막 층인 “projection” 층을 거쳐 채널 수를 줄인다. 이때 inverted residual 모듈에 사용된 활성화 함수는 ReLU6이다. 이는 기존의 ReLU에 상한을 두는 것이다. 즉, 6 이상인 값은 6이다. MobileNet v2에서는 inverted residual 구조가 메모리 효율적이라고 설명하고 있다 (Sandler 등, 2018).

이는 Figure 3 (c)에 묘사되어 있는 것과 같다. 첫 번째 레이어 “expansion” 레이어에서는 채널을 확장해주는데 이때 얼마큼 확장하는지에 대한 파라미터가 t이다. 우리는 t를 2로 하였고 여기서는 필터 개수를 32

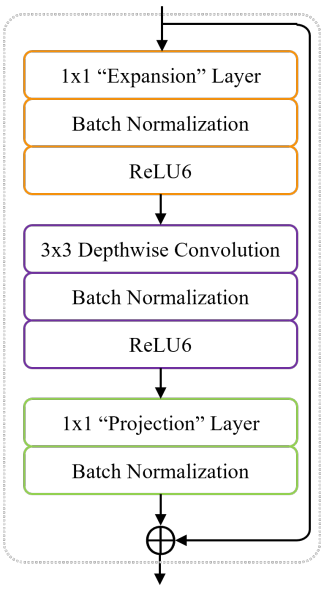


Figure 5: *Bottleneck inverted residual convolution module.*

개로 하였다. 여기서 활성화 함수는 기존 MobileNet v2와 같이 ReLU6를 사용하였다. 다음 층은 depthwise convolution block이고 배치 정규화와 활성화 함수 ReLU6로 구성되어있다. 그다음으론 bottleneck layer인데 이를 “projection” 레이어라고 한다. 이 층에서는 활성 함수를 사용하지 않는다.

2.3.3. 16 bit Quantization

Quantization는 기존의 신경망의 부동소수점 수를 특정 비트 수 만큼 줄이는 방법이다. 만들어진 모델을 16bit로 모든 모델에 대하여 Quantization 하였다. 일반적으로 인공신경망은 활성 노드, 노드 간의 연결, 각 연결과 관련된 가중치 매개변수로 구성된다. 이 중 Quantization 되는 대상은 가중치 매개변수와 활성 노드 연산이다. 신경망을 진행하면 곱셈과 덧셈 연산을 수백만 회 실행해야 하는데, Quantization된 매개변수로 저 비트의 연산을 수행하고, 신경망의 중간 계산 값도 함께 Quantization한다면 모델 사이즈가 줄어든다. Quantization 방법은 DCASE 대회에서 모형 경량화를 위해 많이 사용되고 있어 본 연구의 실험에 사용되었다.

3. 실험

3.1. 데이터 준비

본 논문에 사용된 데이터는 DCASE 2020 Task1 B (TAU urban acoustic scenes 2020 3 class)의 훈련 및 검증 (development) 데이터 셋이다 (Mesaros 등, 2018). 평가 데이터 셋의 경우 정답 라벨이 공개되지 않아서 사용하지 않았다. Tampere University of Technology (TAU)에서 2018년 5월에서 8월까지 수집한 데이터이다. 유럽의 10개의 도시에서 각각 10개 종류의 음향 장면(acoustic scene)이 같은 종류의 기기로 녹음이 되었다. 10개의 음향 장면은 다음과 같이 실내, 실외, 교통수단의 세 가지 주요 클래스로 묶인다.

- 실내: 공항, 쇼핑몰, 지하철역
- 실외: 보행자 거리, 공공 광장, 교통량이 중간인 거리, 도시 공원

Table 1: Dataset description for DCASE 2019 Task 1B

Datasets	Indoor	Outdoor	Transportation	Total
Training set	1,885(29.6%)	2,564(40.4%)	1,905(30%)	6,354
Validation set	819(28.9%)	1,193(42.1%)	819(29%)	2,831
Test set	1,297(31%)	1,604(38.3%)	1,284(30.7%)	4,185
Total	4,001(29.9%)	5,361(40.1%)	4,008(30%)	13,370

- 교통수단: 버스, 트램, 지하철

데이터 셋은 훈련/평가 데이터로 나뉘어 있었고, 이 비율은 각각 70%, 30%이다. 본 논문에서는 훈련 데이터를 훈련/검증 데이터로 70%와 30% 비율로 다시 나누어 훈련하였다. 이때 오디오 녹음이 진행된 도시와 라벨의 비율은 훈련 데이터와 검증 데이터, 테스트 데이터별로 유사하게 구성하였다. 각 데이터의 개수와 데이터 셋의 비율은 Table 1과 같다.

3.2. 데이터 증강 기법(data augmentation)

데이터 증강 기법으로 mixup을 사용하였다. 입력 피쳐인 log mel-spectrogram, deltas, delta-deltas 피쳐를 mixup을 이용하여 증강시켰다. Mixup 증강 기법은 기존의 DCASE 대회에서 많이 쓰인 증강 기법이다 (Koutini 등, 2020; Hu 등, 2020; McDonnell, 2020).

Mixup은 효과적인 데이터 증강 기법이 다 (Zhang 등, 2018). 가중치에 따라 훈련 세트의 서로 다른 샘플을 혼합하는 방식인데, 그 방법은 다음 수식과 같다.

$$X = \lambda X_i + (1 - \lambda)X_j, \tag{3.1}$$

$$y = \lambda y_i + (1 - \lambda)y_j, \tag{3.2}$$

$\lambda \in [0, 1]$ 이며, 이는 파라미터가 α인 베타 분포에서 표본 추출되었다. 즉, $\beta(\alpha, \alpha)$, $\alpha \in (0, \infty)$이다. X_i와 X_j는 서로 다른 데이터 샘플이며, y_i와 y_j는 각 데이터에 해당하는 라벨이다. α는 0.4로 설정하였다. Zhang 등 (2018)에서 α의 값을 0.2와 0.4값을 이용하여 실험을 하였으며, 0.4가 조금 더 좋았다. 또한, 다른 Mixup 을 음성분야에 적용한 논문들에서도 0.4를 많이 사용하고 있어 본 연구에서도 α값으로 0.4를 사용 하였다.

데이터 학습에 있어서 매 Epoch의 2개의 샘플마다 mixup에 의한 표본이 학습된다. 2개의 표본이 mixup에 의해 1개가 된다. 예를 들어, x_1, x_2가 각각 길이가 10인 데이터이고, y_1, y_2 가 Indoor, Outdoor, Transportation의 3개 종류의 라벨이라고 할 때, 아래와 같은 데이터가 있을 수 있다.

$$x_1 = [0, 0, 1, 2, 1, 0, 0, 0, 0, 0], \; y_1 = [0, 0, 1], \tag{3.3}$$

$$x_2 = [0, 1, 1, 1, 1, 1, 1, 0, 0, 0], \; y_2 = [1, 0, 0], \tag{3.4}$$

다음 두 데이터에 대해 mixup 을 적용하면 다음과 같이 새로운 데이터가 생성된다. $\beta(0.4, 0.4)$로부터의 랜덤변수를 추출한다. 임의로 0.1이 추출되었다고 한다면, 새롭게 생성되는 mixup 샘플은 아래의 x_{new}와 y_{new}와 같고, 이런 데이터들이 트레이닝에 사용된다.

$$x_{\text{new}} = 0.1 * x_1 + 0.9 * x_2, \tag{3.5}$$

$$y_{\text{new}} = [0.9, 0, 0.1], \tag{3.6}$$

트레이닝 데이터 수는 동일하게 있으며, 매 Epoch마다 mixup이 랜덤하게 적용된 데이터들이 학습에 사용된다. Mixup이 랜덤하므로, 매 Epoch마다 같은 소스에서 나왔지만 다른 데이터들이 학습되게 된다. 이러한 mixup은 과대적합을 방지해주어 이미지 분야와 음성분야에서 데이터 증강기법으로 자주 사용되고 있다.

3.3. 모델 훈련

손실함수(loss function)로 카테고리컬 크로스엔트로피(categorical crossentropy)를 사용하였고, 아담 옵티마이저(Adam optimizer)를 사용하였다 (Kingma와 Ba, 2015). 초기 학습률(learning rate)은 $10e^3$으로 설정하였고 시그모이달 감소 함수(sigmoidal decay function)를 사용한 학습률 스케줄러(learning rate scheduler)를 이용하여 학습률을 $10e^5$까지 감소하게 하였다. 모든 모델에 대하여 Epoch를 100으로 설정하여 훈련하도록 하였다. 평가 지표로는 정확도(accuracy)를 사용하였다. 매 Epoch마다 학습된 가중치(parameter weights)를 저장하되, 검증 데이터의 정확도가 가장 높게 나올 때마다 저장을 새로 하고 마지막에는 높은 정확도가 나온 가중치를 저장하도록 하였다. 이 가중치를 불러와 평가 데이터에 대해 평가하였다.

4. 결과

결과표를 더욱 명확하게 기술하기 위하여 실험에 사용한 모델 이름을 정리하였다.

- "ResNet" : Figure 3의 (e)에 표현되어있는 기본 모델이다.
- "ResNet RF1" : Figure 3의 (f)에 표현되어있는 기본 모델로, 2개의 잔차 블록과 커널 크기가 (1,1)인 합성곱층 2개로 구성되어 있는 모델이다.
- "DWS" : "DWS"가 모델 이름 앞에 붙으면 Figure 3의 (b) depthwise separable convBlock이 모델의 음영 부분을 대체한 모델을 의미한다.
- "BIR" : "BIR"이 모델 이름 앞에 붙으면 Figure 3의 (c) bottleneck inverted residual ConvBlock이 기존 모델의 음영 부분을 대체한 모델을 의미한다.
- "Q" : "Q"가 모델 이름 앞에 붙으면 모델이 16bit로 Quantization 되었음을 의미한다.
- VGG16 : 커널 사이즈가 (3,3)으로 고정된 16개 층으로 구성된 VGG16 모델 구조를 사용하였다. 3개의 클래스를 구분하는 분류기 모델을 위하여 VGG16의 마지막 부분은 Flatten 레이어와 Dense레이어를 추가하여 모델을 구성하였다. 이때 Keras에서 구현되어있는 모델을 불러와서 사용하였다.
- MobileNetv1 : depthwise seprable convolution이 핵심 아이디어로 사용이 된 MobileNetv1의 구조를 사용하였다. 3개의 클래스를 구분하는 분류기 모델을 위하여 MobileNet v1의 마지막 부분은 Flatten 레이어와 Dense레이어를 추가하여 모델을 구성하였다. 이때 Keras에서 구현되어있는 모델을 불러와서 사용하였다.
- MobileNetv2 : inverted bottleneck residual block을 제안하였던 MobileNetv2의 전체 구조를 사용한 모델이다. 3개의 클래스를 구분하는 분류기 모델을 위하여 MobileNet v2의 마지막 부분은 Flatten 레이어와 Dense 레이어를 추가하여 모델을 구성하였다. 이때 Keras에서 구현되어있는 모델을 불러와서 사용하였다.

Table 2와 Figure 6은 위의 모델에 대한 실험 결과를 보여준다. 각 모델을 10번 훈련시키고 테스트하였다. 각 모델에 대한 검증 데이터에 대한 예측 정확도, 테스트 데이터에 대한 예측 정확도, 각 모델의 크기, 테스트 데이터를 예측할 때 걸리는 시간을 표에 정리하였다. 정확도와 걸리는 시간은 10번의 값의 평균을 기술하였고, 표준편차는 괄호 안에 표기하였다. Table 2의 1, 4번 모델은 depthwise separable convolution이나 bottleneck inverted residual convblock 같은 합성곱 레이어를 사용하지 않은 일반 합성곱 레이어를 사용한 모델이다. 2, 3, 5, 7번 모델의 경우에는 depthwise separable convolution와 bottleneck inverted residual convblock을 사용한 모델이다. 7–12번 모델의 경우 1-6번 모델을 각각 16비트로 Quantization한 모델이다. 또한 본 논문에서 제안한 모델 외에, 일반적으로 많이 쓰이는 CNN 모델인 VGG16과 본 논문에서 사용한 경량화 기법이 제안된 모델인 MobileNetv1과 MobileNetv2를 훈련시키고 테스트하였다. Table 2의 13-15번에 각 모델에 대한 정확도와 모델 사이즈를 기술하였다. 먼저 일반적인 CNN 구조인 VGG16, MobileNetv1, MobileNetv2 의 Test 정확도는

Table 2: Experimental results on light-weight architecture. Each model trained ten times and averaged accuracies and evaluation time are reported with its standard deviation in the parenthesis

	Model	Validation accuracy(%)	Test accuracy(%)	Model size (KB)	Evaluation time (sec)
1	ResNet	92.94 (± 0.24)	94.57 (± 0.47)	503	5.15 (± 0.44)
2	DWS-ResNet	92.48 (± 0.52)	93.46 (± 1.00)	78.51	4.49 (± 0.44)
3	BIR-ResNet	92.34 (± 0.64)	93.83 (± 0.47)	95.64	5.36 (± 0.01)
4	ResNet RF-1	92.96 (± 0.41)	94.94 (± 0.46)	375	5.35 (± 0.01)
5	DWS-ResNet RF-1	92.60 (± 0.40)	93.85 (± 0.56)	92.01	4.41 (± 0.32)
6	BIR-ResNet RF-1	92.32 (± 0.56)	94.08 (± 0.81)	97.01	5.24 (± 0.32)
7	Q-ResNet	92.95 (± 0.24)	94.57 (± 0.49)	255	2.97 (± 0.43)
8	Q-DWS-ResNet	92.44 (± 0.52)	93.48 (± 0.98)	42.76	4.70 (± 0.53)
9	Q-BIR-ResNet	92.30 (± 0.66)	93.80 (± 0.50)	52.07	3.21 (± 0.01)
10	Q-ResNet RF-1	92.93 (± 0.43)	94.93 (± 0.46)	191	2.76 (± 0.52)
11	Q-DWS-ResNet RF-1	92.58 (± 0.42)	93.84 (± 0.58)	49.51	3.30 (± 0.33)
12	Q-BIR-ResNet RF-1	92.32 (± 0.60)	94.08 (± 0.83)	52.51	3.19 (± 0.15)
13	VGG16	89.55 (± 0.64)	91.05 (± 0.64	170,100	15.92 (±0.48)
14	MobileNetv1	89.82 (± 0.52)	89.75 (± 0.71)	238,300	6.99 (± 0.72)
15	MobileNetv2	89.47 (± 0.65)	89.69 (± 0.60)	310,500	8.80 (± 1.72)

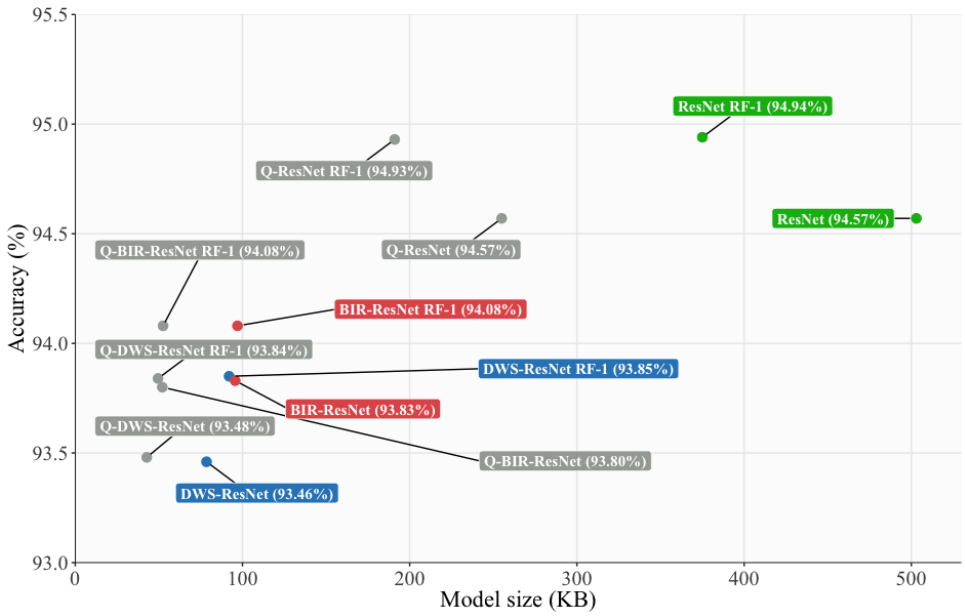


Figure 6: *Graph comparing the model size and performance of proposed models. The green dots are models without any reduction techniques, the blue dots are the models with depthwise separable convolution, the red dots are the models with bottleneck inverted residual, and the grey dots are the models with additional 16bit quantization.*

91.05%, 89.75%, 89.69% 로, 실험에서 사용하고자 하는 ResNet, ResNet RF-1 모형의 94.57%, 94.94% 보다 통계적으로 유의미하게 성능이 안 좋게 나타났다(t-test, 본페로니 교정 적용, p<0.01). VGG16, MobileNet 모

형들이 천만장이 넘는 이미지넷 데이터 학습용으로 제안된 반면, 본 문제의 훈련 데이터는 6,354개로 VGG16, MobileNet 모형을 바로 적용하면 과적합 문제가 발생한 것으로 파악된다. 이는 Koutini 등 (2019) 에서 기존 이미지넷에서 제안된 모형들이 ASC 문제에 바로 적용이 잘 안된다는 연구결과와 일치한다. 시험에서 기본 모형으로 사용한 모형은 ResNet과 ResNet-RF-1 이며, DWS 를 적용하면 통계적으로 유의미하게 성능이 줄어드는 것으로 나타났지만(t-test, 유의수준 0.01%, 본페로니 교정) 약 1.1% 정도의 성능차이로 성능차이는 크지 않은 것으로 나타났다. 반면 모델 크기는 약 3-6배 줄어든 것을 확인할 수 있다. Howard 등 (2017) 에서도 DWS를 적용한 MobileNet 이 모형을 많이 가볍게 하지만 성능은 약간 안좋아지는 것을 볼 수 있다. BIR의 적용에 대해서도 ResNet과 BIR-ResNet 은 통계적으로 유의미한 차이가 보였으나(t-test, 본페로니 교정 적용, $p<0.01$) 정확도는 약 0.5% 정도의 차이로 큰 차이가 없었고, 모형 크기는 크게 줄어드는 것을 확인할 수 있다. Quantization 기법의 적용에 관해서는 6가지의 모든 경우에 대해 통계적으로 유의미한 차이가 발견되지 않았다. Figure 6을 통해 6개의 Quantization 적용된 모형들을 살펴보면 사이즈는 반으로 줄지만 정확도는 거의 비슷함을 확인할 수 있다. 가장 높은 정확도를 가지는 모델은 마지막 Convblock으로 커널이 $(1, 1)$인 합성곱 레이어를 쓴 모델로 정확도는 94.94%이다. 마지막 Convblock을 $(3, 3)$ 커널을 쓴 모델과 비교했을 때 정확도는 0.4% 높아지고 모델 크기도 약 170KB 줄어든 것을 확인할 수 있다. Koutini 등 (2019) 에서 제안한 바와 같이 수용영역(receptive field)을 고려한 디자인이 성능향상에 도움이 됨을 알 수 있다.

Table 2에 Evaluation 데이터 셋에 대하여 예측할 때의 시간을 기술하였다. 실행 시간은 모형 크기 뿐만 아니라 모형 안에서의 연산량과도 관련이 있어 모형 크기가 작다고 실행 시간이 큰 것은 아니지만, VGG16, MobileNetv1, MobileNetv2의 모형 사이즈가 100,000KB으로 큰 모형들은 실행시간이 다소 긴 것으로 나타났다. 하지만, 한 개의 데이터에 대해 예측을 진행한다고 생각할 때 모든 모형들이 데이터 하나당 0.0013초 미만의 시간에 장소를 판단한다. 이는 실제 사용할 때 문제가 없는 속도이다.

Figure 6에서는 모델별로 사이즈-정확도 그래프를 그려 모델끼리 비교하였다. 초록색 점이 경량화 기법을 쓰지 않은 기본 모델이다. 파란색 점이 DWS ConvBlock을 사용한 모델이며, 빨간색 점이 BIR ConvBlock을 쓴 모델이다. 회색점은 각 모델을 16bit로 Quantization한 모델이다. VGG16, MobileNetv1, MobileNetv2의 경우 제안한 모델과의 사이즈 차이가 그래프를 통하지 않아도 확인할 수 있으므로 Figure 6에 포함하지 않았다. 초록색 점 두 개와 바로 왼쪽에 있는 회색 점 두 개를 비교하면 Quantization을 적용하였을때 성능은 비슷하였지만 사이즈는 반으로 줄어드는 것을 확인할 수 있다. 파란색 점, 빨간색 점과 초록색 점을 비교했을 때, 정확도는 비슷하거나 약간 낮아졌지만 사이즈는 3배 이상 줄어들었음을 알 수 있다.

5. 결론

본 논문에서는 음향 장면 분류 문제에 관하여 딥러닝 모델을 제안하고 여러 경량화 기술들을 적용해 비교하였다. 기존 모델의 성능을 유지하면서 모델의 파라미터나 연산량을 줄여 모델을 만드는 기술을 말한다. 우리는 MobileNet v1에서 제안된 depthwise separable convolution과 MobileNet v2에서 제안된 bottleneck inverted residual convolution, 그리고 Quantization이 음향 장면 분류 문제에 있어서도 모델의 사이즈를 줄여주는 데 도움이 될 것이라 기대하였다.

우리는 음향 장면 분류 시스템을 위하여 log mel-spectrogram, deltas, delta-deltas 피쳐를 사용하였고, 잔차 블록을 얕게 쌓은 기본 모델을 만들었다. 또한 수용영역(receptive field)을 작게 하여 모델을 구성하는 것이 음향 장면 분류 문제에 있어서 효과적이라는 연구 결과에 따라 마지막 레이어를 $(1, 1)$ 커널로 대체한 모델 구조도 기본으로 사용하였다. 이 두 모델은 각각 94.57%, 94.94%의 정확도를 기록하였고, 모델 사이즈는 각각 503KB, 375KB였다. 이러한 기본 모델 구조에 depthwise separable convblock, bottleneck inverted residual convblock을 적용하여 모델을 구성하고 훈련시킨 결과, 93.46%, 93.83%의 정확도를 얻었다. 모델 사이즈는 78.51KB, 95.64KB로 줄었는데 이는 5배 이상 줄어든 것이다. 이에 따라 본 연구에서는 depthwise separable

convBlock과 bottleneck inverted residual convblock이 모델 사이즈는 줄이면서 모델의 성능은 비슷하거나 약간 낮은 성능을 얻을 수 있다는 사실을 확인하였다. 또한 모델 압축 기술인 16bit Quantization를 이용하여 모델 사이즈는 반으로 줄이고 모델 성능은 유지할 수 있음을 알 수 있었다.

향후 연구에서는 다양한 라벨을 가진 음향 장면에 대하여 모델 경량화 기술을 적용하고, ANTNets (Xiong 등, 2019), Pruning (Han 등, 2016), Binarization (Courbariaux 등, 2016) 등의 다양한 경량화 기술을 적용해볼 예정이다. 본 연구에서는 기술적인 문제로 16bit Quantization만 각 모델에 적용하였지만, 향후 연구에서는 8bit Quantization을 추가로 적용한다면 계산량을 더 줄일 수 있을 것이라 기대한다. 본 연구에서는 모델 사이즈가 줄어듬에 따라 모델 예측 시간도 줄어든다는 것을 확인하지 못하였다. 향후 연구에서는 모델 사이즈 이외에도 시간을 줄일 수 있는 경량화 기법을 연구한다면 실제에 적용할 때 유용한 기술이 될 수 있을 것이다. 또한 오디오 데이터 뿐 아니라, 이미지, 정형데이터 등 다양한 데이터에 대한 연구도 진행한다면 다양한 분야에 적용할 수 있을 것이라 기대한다.

References

Chollet F (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, 1251–1258.

Courbariaux M, Hubara I, Soudry D, El-Yaniv R, and Bengio Y (2016). *Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1*, arXiv preprint arXiv:1602.02830.

Han S, Mao H, and Dally WJ (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, *4th International Conference on Learning Representations (ICLR 2016)*.

He K, Zhang X, Ren S, and Sun J (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Heittola T, Mesaros A, and Virtanen T (2020). Acoustic scene classification in dcase 2020 challenge: generalization across devices and low complexity solutions. In *Proceedings of the Fifth Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2020)*, 56–60.

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, and Adam H (2017). *Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications*, arXiv preprint arXiv:1704.04861.

Hu H, Yang CHH, Xia X *et al.* (2020). *Device-Robust Acoustic Scene Classification Based on Two-Stage Categorization and Data Augmentation*, arXiv preprint arXiv:2007.08389.

Jan MA, Zakarya M, Khan M, Mastorakis S, Menon VG, Balasubramanian V, and Rehman AU (2021). An AI-enabled lightweight data fusion and load optimization approach for Internet of Things, *Future Generation Computer Systems*, **122**, 40—51.

Kingma DP and Ba J (2015). Adam: A method for stochastic optimization, *3rd International Conference on Learning Representations (ICLR 2015)*.

Koutini K, Eghbal-Zadeh H, Dorfer M, and Widmer G (2019). The receptive field as a regularizer in deep convolutional neural networks for acoustic scene classification, *27th European signal processing conference (EUSIPCO 2019)*, 1–5.

Koutini K, Henkel F, Eghbal-zadeh H, and Widmer G (2020). CP-JKU submissions to DCASE'20: Low-complexity cross-device acoustic scene classification with rf-regularized CNNs, *DCASE2020 Challenge Technical Report*.

Lee YJ, Moon YH, Park JY, and Min OG (2019). Recent R&D trends for lightweight deep learning, *Electronics and Telecommunications Trends*, **34**, 40—50.

McDonnell M (2020). Low-complexity acoustic scene classification using one-bit-per-weight deep convolutional neural networks, *DCASE2020 Challenge Technical Report*

McDonnell MD and Gao W (2020). Acoustic scene classification using deep residual networks with late fusion of separated high and low frequency paths, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, 141–145.

Mesaros A, Heittola T, and Virtanen T (2018). *A multi-device dataset for urban acoustic scene classification*, arXiv preprint arXiv:1807.09840.

Sandler M, Howard A, Zhu M, Zhmoginov A, and Chen LC (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR 2018)*, 4510–4520.

Strubell E, Ganesh A, McCallum A (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3645–3650

Suh S, Lim W, Jeong Y, Lee T, and Kim HY (2018). Dual CNN structured sound event detection algorithm based on real life acoustic dataset, *The Korean Institute of Broadcast and Media Engineers*, **23**, 855—865.

Suh S, Park S, Jeong Y, and Lee T (2020). Designing acoustic scene classification models with CNN variants, *DCASE2020 Challenge Technical Report*.

Szegedy C, Liu W, Jia Y, *et al.* (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR 2015)*, 1–9.

Xiong Y, Kim HWJ, and Hedau V (2019). *Antnets: Mobile Convolutional Neural Networks for Resource Efficient Image Classification*, arXiv preprint arXiv:1904.03775.

Zhang H, Cisse M, Dauphin YN, and Lopez-Paz D (2018). Mixup: Beyond empirical risk minimization, *6th International Conference on Learning Representations (ICLR 2018)*.

Received June 30, 2021; Revised September 12, 2021; Accepted September 23, 2021

음향 장면 분류를 위한 경량화 모형 연구

임소영[a], 곽일엽[1,a]

[a] 중앙대학교 경영경제대학 응용통계학과

요 약

음향 장면 분류는 오디오 파일이 녹음된 환경이 어디인지 분류하는 문제이다. 이는 음향 장면 분류와 관련한 대회인 DCASE 대회에서 꾸준하게 연구되었던 분야이다. 실제 응용 분야에 음향 장면 분류 문제를 적용할 때, 모델의 복잡도를 고려하여야 한다. 특히 경량 기기에 적용하기 위해서는 경량 딥러닝 모델이 필요하다. 우리는 경량 기술이 적용된 여러 모델을 비교하였다. 먼저 log mel-spectrogram, deltas, delta-deltas 피쳐를 사용한 합성곱 신경망(CNN) 기반의 기본 모델을 제안하였다. 그리고 원래의 합성곱 층을 depthwise separable convolution block, linear bottleneck inverted residual block과 같은 효율적인 합성곱 블록으로 대체하고, 각 모델에 대하여 Quantization를 적용하여 경량 모델을 제안하였다. 경량화 기술을 고려한 모델은 기본 모델에 대비하여 성능이 비슷하거나 조금 낮은 성능을 보였지만, 모델 사이즈는 503KB에서 42.76KB로 작아진 것을 확인하였다.

주요용어: 음향 장면 분류, 경량화 모델, 딥러닝, 합성곱 신경망

이 성과는 2020년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2020R1C1C1A01013020).

[1]교신저자: 서울시 동작구 흑석로 84, 중앙대학교 경영경제대학 응용통계학과. E-mail: ikwak2@cau.ac.kr