드론으로 취득된 음성 정보에서 구조요청 소리 및 방향 감지 딥러닝 모델 개발

Development of a deep-learning model for detecting sound and its direction measurement from speech information acquired by drones

요 약

대형복합재난이 지속적으로 발생하고 있는 현재 사회에서 인명을 구조하기 위한 방법으로 드론 사용이 떠오르고 있다. 재난이나 구조요청 상황에서는 환경소음과 노이즈가 크기 때문에 구조 요청 음성이 어디서 들려오는지 그 방향을 추정하기가 어렵다. 이를 해결하기 위하여 본 논문에서는 소음과 노이즈로부터 음성의 발원방향을 추정하는 딥러닝 모델을 개발한다. 훈련 데이터의 부족 문제를 해결하기 위하여 구조요청 음성 오디오와 환경 소음을 합성하여서 훈련 데이터를 생성한다. 그리고, 해당 데이터를 전처리 한 후, 필요한 특징을 추출하여 설계한 딥러닝 모델을 통해 해당 구조 요청의 발원 방향을 추정한다.

1. 서 론

현재 사회에서는 대형복합재난이 지속해서 발생하고 있다. 대형복합재난은 동시 또는 순차적으로 두 가지 이상의 자연, 사회 재난이 발생하고, 그 영향이 복합화되어 인명, 재산, 기반시설 마비 등 그 피해가 극심하여 국가적 위협이 되어 범부처의 통합적 대응이 필요한 재난이다. 이러한 대형복합재난에서 위험에 처한 인명을 구조하는 방법으로 드론 사용이 떠오르고 있다.

재난현장이나 구조현장에서 드론으로 습득할 수 있는 인명 구조요청 소리는 환경 소음과 노이즈가 클 확률이 높다. 이러한 환경에서 정확한 인명 구조요청 소리를 판별하고 이의 발원 방향을 측정하는 것이 해당 연구의 목적이다. 이를 위해 데이터 전처리와 딥러닝 모델을 개발한다.

본 논문에는 음성 오디오와 환경 소음을 임의로 합성하여 딥러닝 학습에 필요한 데이터 생성, 전처리 및 특징 추출, 학습 모델 구조와 성능 평가 등의 내용을 담고 있다.

2. 관련 연구

2.1 관련 연구 1 : DronAID, A Smart Human Detection Drone for Rescue [2]

사람을 탐지할 수 있는 실시간 자율 무인 기술 시스템인 'DronAID'에 관한 연구로서, 가장 빠른 시점에 생존자의 위치를 정확히 파악하고 구조할 수 있게 하는 것을 목표로 한다. PIR 센서를 탑재해 해당 센서가 탐지가 가능한 반경 안에 사람이 있는 경우, 사람이 방출하는 방사선을 감지하고, 그 위치를 파악할 수가 있다.

2.2 관련 연구 2 : Sound Event Detection, Localization, and Classification for Robotic and Surveillance Applications [2]

Sound Localization에 관한 연구로서, 마이크 배열을 사용하여 음원의 진행방향을 예측하는 것을 목표로 하였다. TDOA(Time difference of arrival)와 GCC(General cross correlation)를 사용하여 소리 방향 예측 알고리즘을 연구하였다.

2.3 관련 연구의 문제점

첫 번째 관련 연구의 경우 희생자를 탐지할 때 사용하는 PIR 센서는 일정한 적외선을 가진 물체가 움직이는 것을 감지하는 것이므로 사람이 움직임이 없는 경우는 감지할 수 없다. 또한, 미세한 정도의 움직임 역시 감지를 하지 못하기 때문에 자연재해 현장에서 희생자가 움직이지 못하면 성능을 기대하기가 어렵다. 더 나아가 센서 자체가 급격한 온도변화와 같은 주변 환경의 영향을 받기 때문에 야외에서 사용할 때는 정확한 탐지가 어렵다.

두 번째 관련 연구의 경우 잡음이 포함되지 않은 음성 정보는 80% 정도의 높은 정확도로 방향을 예측하지만, 바람 소리를 삽입하여 잡음이 추가된 경우는 60% 정도로 그 정확도가 낮아지는 모습을 보였다.

이러한 문제점들을 바탕으로 본 연구는 사람이 소리를 지르거나 구조요청 신호를 보냈을 때 드론이 이를 인식하고 해당 방향을 추정하는 것을 목표로 진행하였다.

3. 연 구 진 행

3.1 전체 프로세스

본 연구는 <그림 1>과 같이 데이터 확대, 데이터 전

처리와 특징 추출, CNN 모델 학습, 결과 출력의 총 네 가지 과정으로 이루어져 있다.

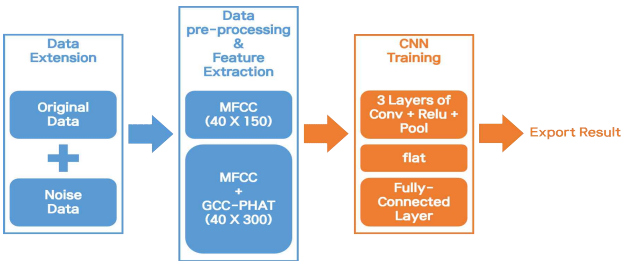

그림 1. 전체 프로세스

3.2 학습 데이터 생성 [3]

3.2.1 음성데이터와 소음 데이터 합성

딥러닝의 경우 학습할 수 있는 데이터가 많을수록 정확도가 올라간다. 데이터의 개수를 많이 확보하기 위해 보유 중인 데이터를 이용해 다른 데이터를 생성하는 과정이 필수적이었다. 제한된 데이터로부터 다양한 경우의 학습 데이터를 생성하기 위하여, LINE ENGINEERING의 Kunihiko Sato가 개발한 파이썬 코드와 쉘 스크립트를 이용하여 스테레오 타입을 가진 음성 오디오와 스테레오 타입의 환경 소음 오디오를 합성하여 학습 데이터를 생성했다. 음성 오디오와 환경 오디오는 모두 wav 파일이며, 이렇게 만들어진 학습 데이터의 SNR(Signal-to-Noise ratio)는 모두 -20으로 고정하였다.

$$SNR_{db} = 20\log_{10}\frac{A_{signal}}{A_{noise}}$$

수식 1. SNR 계산 방법

A_{signal}과 A_{noise}는 각각 음성과 잡음의 '크기' 혹은 '세기'를 나타낸다. '세기'의 정의에는 몇 가지가 있는데, 본 논문에서는 진폭 값의 평균 제곱근(Root Mean Square, RMS)을 각 소리의 세기로 정의한다.

음성 오디오와 환경 소음 오디오, 두 데이터 모두 양자화 bit 수와 샘플링 레이트를 16bit와 44.1kHz로 통일하였다. 이후, 음성 오디오 파일과 환경 소음 오디오 파일을 읽어 들여 byte 객체로 반환하여 오디오 파일 음성 파형의 진폭 값을 취득한다. 취득한 진폭 값의 평균 제곱근(RMS)과 SNR을 이용하여 잡음의 진폭을 임의의 계수만큼 확장 또는 축소한다. 이렇게 변형된 잡음 데이터와 음성 데이터를 합성한다.

두 진폭을 더한 진폭 값이 양자화 bit 수인 16bit의 최댓값을 넘을 수 있으므로 정규화시켜준다.

위 과정으로 각 음성 오디오 데이터와 환경 소음 데이터에 적용해 데이터를 생성할 시, 음성 데이터의 수와 소음 데이터의 수를 곱한 만큼의 데이터를 생성할 수 있다.

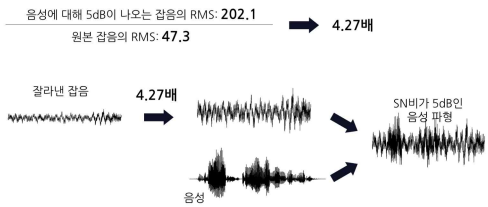

그림 2. 파형 합성 예시

3.2.2 20도별 음성데이터 생성

음성 데이터를 0도, 60도, 120도, 180도만 보유하고 있었기 때문에 더욱 세분화 된 각도 학습을 위해 기존 음성 데이터를 바탕으로 20도 간격으로 음성데이터를 생성한다.

새로운 데ㅋㅋ이터 생성을 위한 기준을 찾기 위해 기존 음성데이터들의 ch1과 ch2의 RMS의 비율을 측정하였다.

각도	ch1 RMS / ch2 RMS 평균
0도	11.8
60도	1.65
120도	0.54
180도	0.12

표 1. 각도별 ch1 RMS / ch2 RMS의 평균

각도별 ch1 RMS / ch2 RMS을 기반으로 20도, 40도, 80도, 100도, 140도, 160도의 ch1 RMS / ch2 RMS의 규칙성을 찾아냈다.

각도	실제 ch1/ch2	예측 ch1/ch2	규칙성
0도	11.8	11.8	X
20도	X	7.0870	이전 각도*0.010169$^{\frac{1}{9}}$
40도	X	4.2565	이전 각도*0.010169$^{\frac{1}{9}}$
60도	1.65	2.5564	이전 각도*0.010169$^{\frac{1}{9}}$
80도	X	1.5354	이전 각도*0.010169$^{\frac{1}{9}}$
100도	X	0.9221	이전 각도*0.010169$^{\frac{1}{9}}$
120도	0.54	0.5538	이전 각도*0.010169$^{\frac{1}{9}}$
140도	X	0.3326	이전 각도*0.010169$^{\frac{1}{9}}$
160도	X	0.1997	이전 각도*0.010169$^{\frac{1}{9}}$
180도	0.12	0.11199	이전 각도*0.010169$^{\frac{1}{9}}$

표 2. 실제 ch1 RMS / ch2 RMS과 예측 비교

실제 ch1 RMS / ch2 RMS와 예측 ch1 RMS / ch2 RMS가 근접하게 일치하므로 예측 비율을 기반으로 20도 간격의 음성데이터를 생성하였다.

3.3 데이터 전처리 및 특징 추출

3.3.1 데이터 전처리

각 채널의 차이점을 통해서 모델을 학습시켜야 하므로 채널 분리가 필수적인 전처리 요소이다. 학습과 테스트에 사용되는 데이터가 스테레오 타입의 wav 파일로 총 2개의 채널을 가지고 있으므로 이를 각각 다른 두 개의 wav 파일로 분리하였다.

채널을 분리한 후, 해당 데이터의 잡음을 제거하기 위해 총 세가지 필터를 사용하였다.

첫 번째로 HighPass 필터를 사용해 높은 주파수의 영역만 남기고 나머지 소음을 모두 제거하는 방식을 시도하였다. 이는 1000Hz를 기준으로 진행하였으며 웬만한 소음들은 이 과정에서 제거되었다. 두 번째로는 Hiss Removal 필터를 사용해 위의 잡음 제거 과정에서 생겨난 부자연스러운 쇳소리를 제거하였으며, 마지막으로 Reduce Hum 필터를 통해 마이크에 의해서 발생하는 치직거리는 잡음까지 제거하며 최대한 완벽하게 사람의 목소리 데이터만 남길 수 있도록 데이터의 전처리를 진행하였다.

3.3.2 특징 추출

위의 전처리 과정을 전부 거쳐 노이즈 제거까지 완료된 데이터는 MFCC(Mel-Frequency Cepstral Coefficient)[4], GCC- PHAT(Generalized Cross Correlation-Phase Transform), STFT(Short-Term Fast Furrier)와 같은 세 개의 특징 추출 단계를 거친다.

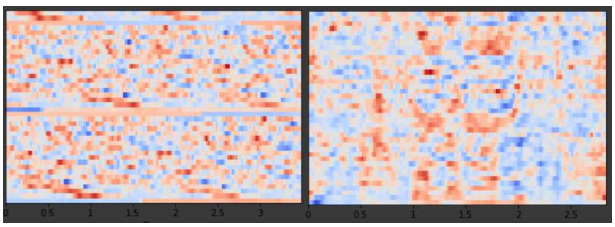

그림 3. 두 채널에서 각각 추출된 MFCC의 스펙트로그램

MFCC는 음정의 높낮이에 영향을 받지 않기 때문에 음성처리에서 대표적으로 사용되는 특징이다. 본 논문에서는 MFCC의 개수를 40개로 고정하여 추출하였으며 한 개의 데이터 당 40 X 150의 일정한 크기를 가진 벡터로 나타내었다. 위의 그림 3은 해당 벡터를 스펙트로그램으로 시각화한 모습이다. 모델을 학습시킬 때는 두 채널의 연관성을 통해 방향을 추정하도록 학습시켜야하기 때문에 이렇게 추출된 두 벡터의 차를 특징 벡터로 사용하였다.

GCC-PHAT은 음원이 서로 다른 두 마이크로 도달하는 시간 차이(Time Difference of Arrival, TDOA)를 구하는 기법의 하나[5][6]다.

$$\hat{G}_{PHAT}(f) = \frac{X_i(f)[X_j(f)]^*}{\left|X_i(f)[X_j(f)]^*\right|}$$

수식 2. GCC-PHAT의 정의

$$\hat{d}_{PHAT}(i,j) = \arg\max_d(\hat{R}_{PHAT}(d))$$

수식 3. TDOA 추정

두 입력 신호 X_i와 X_j에 대해 GCC-PHAT를 구하는 방법은 수식 2와 같다. 이렇게 구해진 GCC-PHAT은 수식 3을 거쳐 최종적으로 음원 도달 시간 차이를 추정할 수 있다. 본 논문에서는 이렇게 추출된 GCC-PHAT의 스펙트럼 성분에 IFFT(Inverse Fast Fourier Transform)의 연산을 수행하였다.

마지막으로 스펙트럼을 짧은 시간마다 반복해서 추출해내는 특징 추출 기법인 STFT를 사용하여 시간에 따른 주파수 성분의 변화와 관련된 특징을 학습시키려고 하였다. 총 window 수를 3000개로 설정해 추출하였으며, 이렇게 추출한 벡터를 MFCC와 동일하게 40 X 150의 크기를 가진 벡터로 나타내어 먼저 추출한 MFCC, GCC-PHAT과 함께 사용하였다.

또한, 더욱 원활한 학습을 위해 모델에 입력으로 사용되는 데이터는 모두 정규화를 거친 후 학습에 사용되었다.

3.4 모델 구조

본 논문에서 제시하는 모델은 CNN(Convolutional Neural Network) 아키텍처를 사용하여 설계되었다. CNN 모델이란 합성 곱 연산을 사용한 신경망 네트워크 모델로, 주로 이미지를 인식하는 데에 쓰이고 있지만 본 연구에서는 MFCC 처리한 음성파일에서 특징을 뽑아내어 전처리하였기 때문에 이를 인식할 수 있는 CNN 모델을 사용하였다. 사용된 CNN 아키텍처는 컨볼루션 계층, 풀링 계층, 활성화 함수 등으로 구성되어있다.

$$C(x,y) = \sum_{a=0}^{k-1}\sum_{b=0}^{k-1} I(x-a, y-b)F(a,b)$$

수식 4. convolution 연산

수식 4는 컨볼루션 계층에서 합성 곱(Convolution) 연산을 나타내는 식이다. 다음 식에서 I는 2차원 입력배열, F는 필터, k는 필터의 크기를 나타낸다.

풀링 계층(Pooling Layer)은 컨볼루션 계층에서 생성된 결과로 중복된 특징값이나, 상호 연관성이 높은 요소들을 제거하고 특징을 강화하기 위해 사용된다. Strides와 Kernel Size, Padding 여부에 따라 데이터의 크기가 바뀌게 된다.

$$f(x) = \max(0, x)$$

수식 5. ReLu 함수

활성화 함수로는 은닉층에서 ReLu를 사용하였다. 기존에 사용되었던 Sigmoid 함수 혹은 tanh 함수보다 CNN에서 데이터를 학습시키고 출력값을 수렴하는 데에 ReLu가 더 높은 성능을 보이기 때문이다.

본 연구에서는 컨볼루션 계층 3개와 풀링 계층를 쌓았고 은닉층의 활성화 함수로 ReLu를 사용하였다. 또한

결과값을 fully-connected 레이어로 연결하여서 이를 20° 간격으로 분류하였다.

4. 성능 평가

4.1 60° 간격으로 구분

먼저 60° 간격으로 구분한 결과이다. 설계된 모델에 전처리 된 데이터를 입력값으로 넣고 이를 학습시켜 4개의 클래스로 구분한다. 사용된 데이터는 모두 8948개이며, Train:Validation:Test는 각각 6:2:2의 비율로 나누어 사용하였다. 설계된 모델에 전처리 되어진 데이터를 입력값으로 주어지게 되면 이를 설계된 CNN 모델의 과정을 거쳐 4개의 클래스로 구분하게 된다.

표 3은 MFCC로 전처리를 진행한 데이터와 MFCC와 GCC기법을 사용하여 전처리를 진행한 데이터를 입력값으로 학습을 진행한 결과이다.

실험 결과, MFCC 하나의 특징값을 사용하였을 때 보다, GCC까지 특징값으로 사용하였을 때 모델이 더욱더 신뢰성 있는 정보를 제공함을 볼 수 있다.

	MFCC	MFCC+GCC
Accuracy	70%	76%

표 3. 60° 모델 학습 결과

4.2 20° 간격으로 구분

다음은, 더 정확하고, 구체적인 발원 방향을 파악할 수 있도록 20° 간격으로 구분한 결과이다. 사용된 데이터는 각도별로 300개씩, 총 3000개를 6:2:2의 비율로 나누어 사용하였다.

	MFCC+GCC	MFCC+GCC +STFT
Accuracy	43.5%	69%

표 4. 20° 모델 학습 결과

표 4는 MFCC와 GCC기법으로 전처리를 진행한 데이터와 STFT 기법을 추가하여 전처리를 진행한 데이터를 입력값으로 모델에 주어 학습을 진행한 결과이다. STFT 기법을 추가로 전처리한 데이터가 69%의 정확도를 보이며 기존 연구 보다 조금 향상되었음을 보여 준다.

5. 결론

본 논문에서는 드론에서 획득한 음성정보로부터 음성의 발원방향을 추정하는 딥러닝 모델을 제안하였다. 또한, 부족한 학습데이터를 위해 학습 데이터 생성 방법을 제안하였다. 제안한 방법은 60° 간격으로 구분할때는 76%의 정확도를, 20° 간격으로 구분할 때는 69%의 정확도를 보여준다. 이후에는 이러한 성능을 더욱 향상시킬 수 있도록 모델을 설계할 계획이다.

본 연구에서 진행한 모델을 이용한다면, 재난 상황에 드론을 이용하여 구조요청이 어느 방향에서 생성되었는지 빠르게 파악하여 인명 구조에 도움을 줄 수 있을 것으로 파악한다.

6. 참고문헌

[1] Rameesha Tariq , Maham Rahim , Nimra Aslam , Narmeen Bawany , Ummay Faseeha , DronAID : A Smart Human Detection Drone for Rescue. In IEEE, 2018

[2] Nguyen Van Quang, Sound Event Detection, Localization, and Classification for Robotic and Surveillance Applications. PhD thesis, UST, 2015

[3] “*딥러닝 음성 인식에 필요한 훈련 데이터를 직접 만들어보자*”, LINE Engineering 블로그, 2018년 10월 01일 수정, 2019년 4월 14일 접속, https://engineering.linecorp.com/ko/blog/voice-waveform-arbitrary-signal-to-noise-ratio-python/

[4] Sahidullah, Md and Saha Goutam. *Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition*, 2012.

[5] 권오현, 장준혁. *심화 신경망을 사용한 다채널 마이크 구조에서의 음원 방향 추정*, 한국통신학회 하계종합학술발표회, 2017.

[6] Knapp, C. H. and Carter, G. C. *The generalized correlation method for estimation of time delay*, IEEE Transactions on Acoustics, Speech and Signal Processing ASSP-24, 1976.