

연합 학습 기반 분산 FMCW MIMO radar를 활용한 모션 인식 알고리즘 개발

(Federated Learning based Motion Recognition Algorithm using
Distributed FMCW MIMO Radars)

강 종 성, 이 승 호, 이 정 한, 양 윤 지, 박 재 현

Jong-Sung Kang, Seung-Ho Lee, Jeonghan Lee, YunJi Yang, Jaehyun Park

부경대학교 전자공학과

Abstract: In this paper, we implement a distributed FMCW MIMO radar system to obtain Micro Doppler signatures of target motions. In addition, we also develop federated learning based motion recognition algorithm based on the Micro-Doppler radar signature collected by the implemented FMCW MIMO radar system. Through the experiment, we have verified that the proposed federated learning based algorithm can improve the recognition accuracy up to 90%.

Keywords : FMCW MIMO Radar, Federated Learning, Motion recognition using Micro-doppler signature

I. 서 론

최근 딥러닝에 대한 관심도가 높아져 딥러닝에 관한 연구들도 활발하게 이루어졌으며, 이로 인해 육안으로는 구분이 되지 않는 데이터들을 컴퓨터를 이용하여 더 정밀하게 분석하고 컴퓨터가 학습을 할 수 있게 되었다.

하지만 centralized learning 형식의 딥러닝 기술에도 단점이 존재하였는데, 많은 양의 데이터를 server로 업로드할 때 통신량 증가에 따른 통신 비용의 부담이 커지게 되었고, server로 업로드되는 데이터에는 인공지능 스피커의 개인 음성 데이터 혹은 개인의 영상시청 기록 등 개인의 정보가 포함되어 있을 수 있다는 단점이 존재하였다.

이러한 기존 centralized learning 형식의 딥러닝의 단점들을 해결하기 위하여 이번에 본 논문에서 사용한 것이 연합 학습(Federated Learning)이다. [1] 연합 학습은 기존의 딥러닝처럼 centralized learning이 아닌 local model에서 학습 후 시킴으로서 local model parameter를 server로 업로드함으로써 전송하는 데이터의 양을 줄일 수 있고 이용자들의 개인정보가 담긴 데이터가 아닌 local model parameter를 server로 업로드하기에 개인의 정보가 노출되지 않는다. 이러한 연합 학습의 장점들을 이용하여 학습시킬 데이터는 FMCW레이더를 이용한

Micro Doppler Signature(MDS)이다. FMCW레이더는 야간이나 흐릿한 환경과 같이 시야확보가 어려운 상황에서 이용가능하고, 사람의 키, 몸무게, 장소에 상관없이 비슷한 MSD를 얻을 수 있다는 장점이 있어 사용하였다.

본 논문에서는 FMCW 레이더를 이용하여 사람의 움직임 중에서도 걷기, 팔 벌리기, 점프 총 3가지 움직임의 변화를 통해 발생하는 지연시간 얻어지는 데이터 파일을 Matlab을 사용하여 Micro Doppler Signature(MDS)를 얻음으로서 FMCW 레이더 시스템을 구현한다. 또한 FMCW레이더 시스템을 구현함으로써 얻은 MDS들을 연합 학습을 시켜 연합 학습 모델을 구현하였고 기존의 centralized learning과의 차이점도 함께 확인하였다.

II. FMCW 레이더 시스템과 연합 학습 모델



그림 1.
점프

그림 2.
팔 벌리기

그림 3.
걷기

1. FMCW 레이더 시스템

FMCW레이더는 주파수 변조된 신호를 연속적으로 전송하는 방식의 레이더이다. FMCW레이더는 ‘칩(chirp)’이라고 불리는 사인파의 신호를 전송하여 사물을 통해 반사된 여러 개의 칩들을 수신하여 사물로부터 주파수차이에 의한 지연시간을 측정한다. 본 논문에서 사용한 FMCW레이더는 TI사의 awr1642라는 제품으로 mmwave studio라는 프로그램을 함께 제공한다. 이 프로그램에서 한 칩 간의 지연시간과 칩의 개수, 칩이 여러 개 모여서 만들어지는 ‘프레임(frame)’의 개수를 아래의 그림4,5,6과 같이 mmwave studio에서 설정할 수 있다.

본 논문에서는 각 프레임 당 128개의 칩을 가진 8개의 프레임을 전송하도록 mmwave studio에서 설정을 한다. 그리고 위 그림1,23과 같이 움직이는 동작을 수행할 때 반사된 여러 개의 칩들을 수신하여 주파수 차이에 의한 지연시간을 측정하여 FMCW레이더에서 데이터파일이 만들어진다.[2]

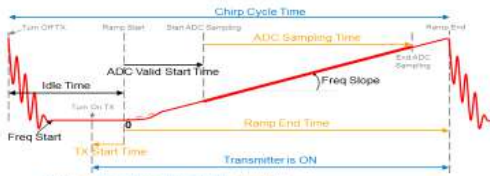


그림 4. FMCW레이더 칩 주기 시간 설명

Profile Id	0	HPF1 Center Freq	179K
Start Freq (GHz)	77.000000	HPF2 Center Freq	350K
Frequency Slope (MHz/us)	23.982	Op-Pwr Backoff TX0 (dB)	0
Idle Time (us)	50.00	Op-Pwr Backoff TX1 (dB)	0
TX Start Time (us)	0.00	Op-Pwr Backoff TX2 (dB)	0
ADC Start Time (us)	6.00	Phase Shifter TX0 (deg)	0.000
ADC Samples	255	Phase Shifter TX1 (deg)	0.000
Sample Rate (ksps)	5000	Phase Shifter TX2 (deg)	0.000
Ramp End Time (us)	60.00	Bandwidth (MHz)	1798.92
RF Gain (dB)	30	Set	Manage Profile
RF Gain Target	30dB		
VCO Select	VCO1	Force VCO Select	
Calib LUT Update		RetainTxCalLUT	RetainRxCalLUT

그림 5. 칩 세부설정

Frame			
Start Chirp TX	0	No of Chirp Loops	128
End Chirp TX	0	Periodicity (ms)	1000.000000
No of Frames	8	Trigger Delay (us)	0.00
Dummy Chirps(End)	0	Active-Ramp Duty Cycle	0.8 %
Trigger Select	SoftwareTrigger	Duty Cycle	1.4 %
☐ Test Source Enable			
Set			

그림 6. 프레임 세부설정

2. 매트랩

앞 선 FMCW레이더에서 만들어지는 데이터파일을 본 논문에서 사용하는 MDS를 얻기 위하여 Matlab으로 데이터파일을 불러온 후 모든 칩에 대하여 FFT를 수행한 후 가장 높은 크기를 가진 칩에서 STFT를 수행한다. FFT는 DFT를 빠르게 수행하는

효율적인 알고리즘이고 STFT는 시간의 길이를 나누어서 푸리에 변환을 하게 된다. DFT는 식(1), STFT는 식(2)와 같이 표현할 수 있다. 식(2)에서 N은 FFT Size로 Window를 얼마나 많은 주파수 밴드로 나누는가이며 $w(n)$ 은 윈도우 함수이다. n 은 윈도우 사이즈로 윈도우 함수에 들어가는 샘플의 양을 의미한다. H는 윈도우가 겹치는 사이즈이다. 본 논문에서는 MDS를 얻기 위하여 $n = 5$, $H = 4$, $N = 2048$ 로 하였다. 모든 칩에 대하여 FFT를 수행을 하면 아래 그림7,9,11과 같은 그래프가 나오게 되고, 그래프에서 볼 수 있듯이 가장 높은 크기를 가진 칩에 대하여 STFT를 수행하고 나면 아래 그림8,10,12와 같은 본 논문에서 사용 할 MDS를 얻을 수 있다.

$$f_j = \sum_{k=0}^{n-1} x_k e^{-\frac{2\pi i}{N} jk} \quad j = 0, \dots, n-1$$

식(1)

$$X(l, k) = \sum_{n=0}^{N-1} w[n] x[n + lH] e^{-\frac{2\pi i kn}{N}}$$

식(2)

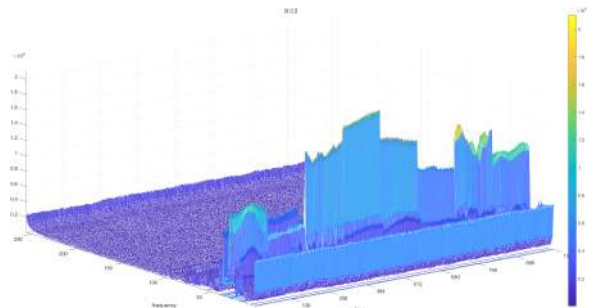


그림 7. 점프 동작의 모든 칩에 대하여 FFT를 수행한 그래프



그림 8. 점프 동작에 대하여 STFT를 한 후 얻은 MDS

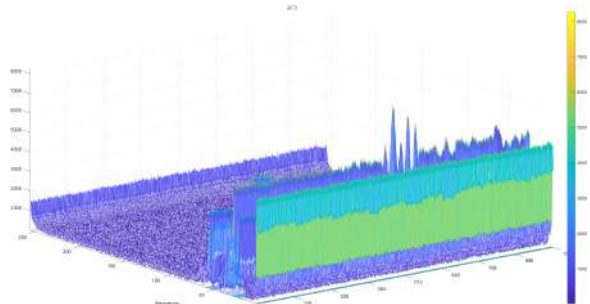


그림 9. 걷기 동작의 모든 칩에 대하여 FFT를 수행한 그래프



그림 10. 걷기 동작에 대하여 STFT를 한 후 얻은 MDS

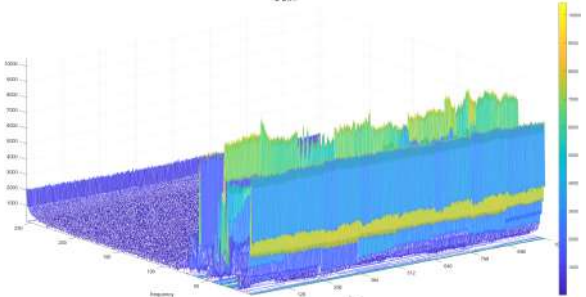


그림 11. 팔 벌리기 동작에 대하여 모든 첩에 대해 FFT를 수행한 그래프



그림 12. 팔 벌리기 동작에 대하여 STFT를 한 후 얻은 MDS

3. 연합 학습

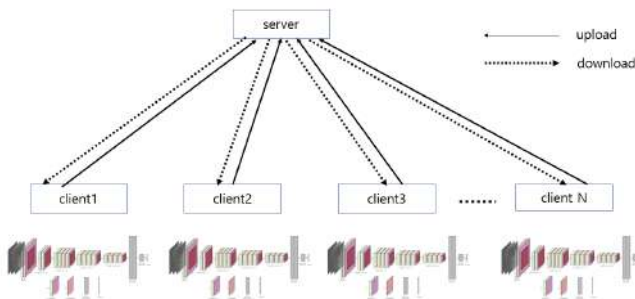


그림 13. FMCW레이더 연합 학습 모델

본 논문에서는 그림13과 같이 다수의 FMCW 레이더시스템에서 각각 다른 동작들을 수행한 후 얻어지는 MDS를 연합 학습을 사용하여 적은 데이터셋으로도 centralized learning보다 더 높은 학습률과 테스트 정확도를 얻을 수 있기에 연합 학습을 사용하였다.

연합 학습은 server에서 각 client들로 데이터를 학습시킬 global model을 배포하고, 각각의 client들은 FMCW 레이더를 이용하여 수집한 MDS데이터로 server에서 배포한 global model을 이용하여 local model을 학습한다. 각 client에서 학습을 반복시키는 횟수를 epoch라고 하며, server에서는 각 client에서 epoch만큼 학습 후 만들어진 local model parame

ter를 취합하여 global model을 업데이트한다. 이 과정을 round라고 한다. 본 논문에서 사용한 epoch은 5, round는 150이다.

본 논문에서는 MDS 데이터셋이 이미지 형식이므로 알고리즘을 이미지 형식을 잘 학습시키기 위하여 cnn 중에서도 vggnet을 사용하였다.[3]

III. 실험 결과

그림1,2,3과 같은 동작을 각각 40번씩 반복을 하였으며 한 동작을 FMCW레이더를 이용하여 인식할때 총 8번의 프레임을 사용하도록 설정하여 매트랩을 통하여 얻어지는 전체 MDS 데이터 개수는 $3 \times 40 \times 8$ 로 총 960개이다.

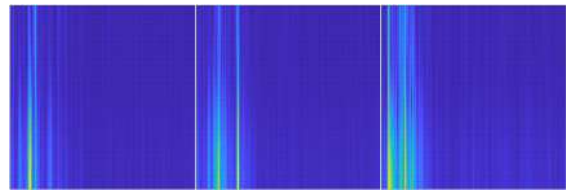


그림 14. 점프 그림 15.팔 벌리기 그림 16. 걷기

위의 그림과 같이 각 동작마다 320개의 MDS를 매트랩에서 STFT를 수행하여 획득하였지만 본 논문에서 local model에서 사용한 알고리즘인 vggnet에 학습을 시키기에는 MDS의 크기가 너무 크기에 모든 MDS의 크기를 224×224 로 변경을 하였다.

각 동작마다 320개의 MDS중에서 270개는 훈련 데이터셋으로 나머지 50개는 테스트 데이터셋으로 분류하여 테스트 데이터셋과 비교하였을 때의 정확도와 loss값을 그래프로 나타내어 확인해보았다.

또한, 앞선 2절에서 설명하였듯이, 연합 학습과 딥러닝과의 적은 데이터셋의 양을 이용하여 학습률의 차이를 설명하고자 연합 학습과 딥러닝과의 학습률과도 비교하였다. 기존 딥러닝에서의 학습 모델 또한 연합 학습에서의 알고리즘과 동일하게 vggnet으로 하였으며 연합 학습 배경과 비슷하게 하기 위하여 전이학습으로 하였으며 round 또한 150으로 동일하게 하였다.

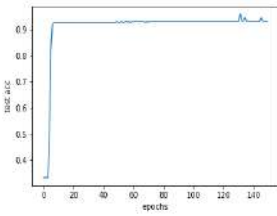


그림 17. 딥러닝 테스트
정확도 그래프

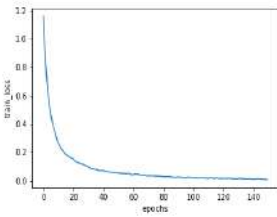


그림 18. 딥러닝 훈련 손실
그래프

위 그림에서 알 수 있듯이 기존 딥러닝에서 테스트 데이터셋의 정확도는 첫 epoch가 실행될 때 정확도가 33%이며 5번째 epoch부터 80%로 증가하며 epoch가 증가할수록 정확도가 상승하는 것을 확인할 수 있다. 150번째 epoch에서 테스트의 정확도는 93%이다. 훈련 손실 그래프에서는 epoch가 80번째가 될 때 거의 0.01이 됨을 알 수 있다.

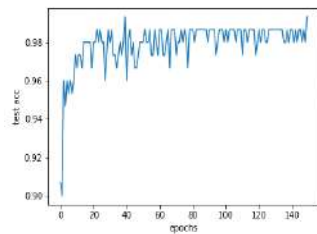


그림 19.연합 학습 테스트
정확도 그래프

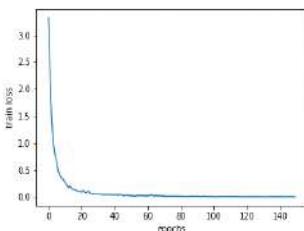


그림 20. 연합 학습 훈련 손실
그래프

하지만 연합 학습에서의 테스트 데이터 셋의 정

확도는 첫 epoch부터 90%이며 epoch가 증가할수록 테스트의 정확도는 100%에 육박하였다. 훈련 데이터의 손실 그래프에서도 5번째 epoch에서 이미 학습의 손실률이 0.05로 손실이 거의 없는 것을 확인할 수 있다. 이를 통하여 MDS를 학습시키는 성능이 epoch가 적고 데이터의 양이 더 적은 상황에서는 연합 학습이 더 우수함을 확인할 수 있었다.

참 고 문 헌

- [1] H. Brendan, McMahan Eider, Moore Daniel, Ramage Seth, Hampson, Blaise Aguera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data" in arXiv:1602.05629v3 [cs.LG] 28 Feb 2017
- [2]<https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-1-range-estimation?keyMatch=FMCW%20RADAR&tisearch=search-support>
- [3]<https://towardsdatascience.com/preserving-data-privacy-in-deep-learning-part-1-a04894f78029>