ECE 693 – Special Topics:
AI for Radar System Design

Radar Micro-Doppler Classification with Limited Data

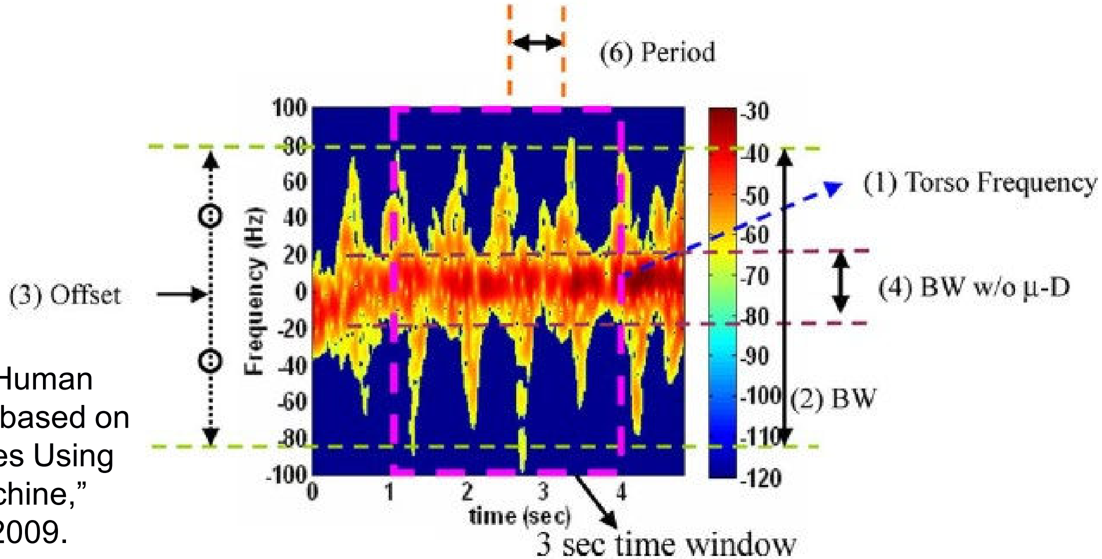
Dr. Sevgi Zubeyde Gurbuz
szgurbuz@ua.edu

Feb. 11, 2022

THE UNIVERSITY OF ALABAMA

Micro-Doppler Features

- Physical features



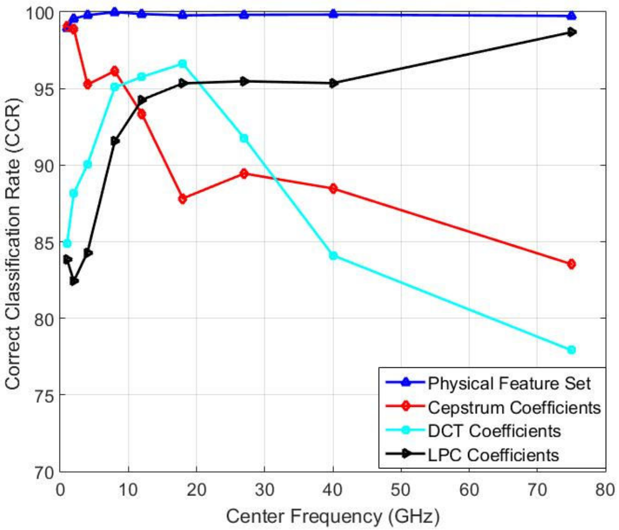
Y. Kim and H. Ling, "Human Activity classification based on micro-Doppler features Using a Support Vector Machine," IEEE Trans. GSRS, 2009.

- Discrete Cosine Coefficients
- Speech Processing Inspired Features
  – Linear Predictive Coding (LPC)
  – Mel-Frequency Cepstral Coefficients (MFCC)

Need to design features that reflect underlying phenomenology of signal

Operational Dependence of Pre-Defined Features

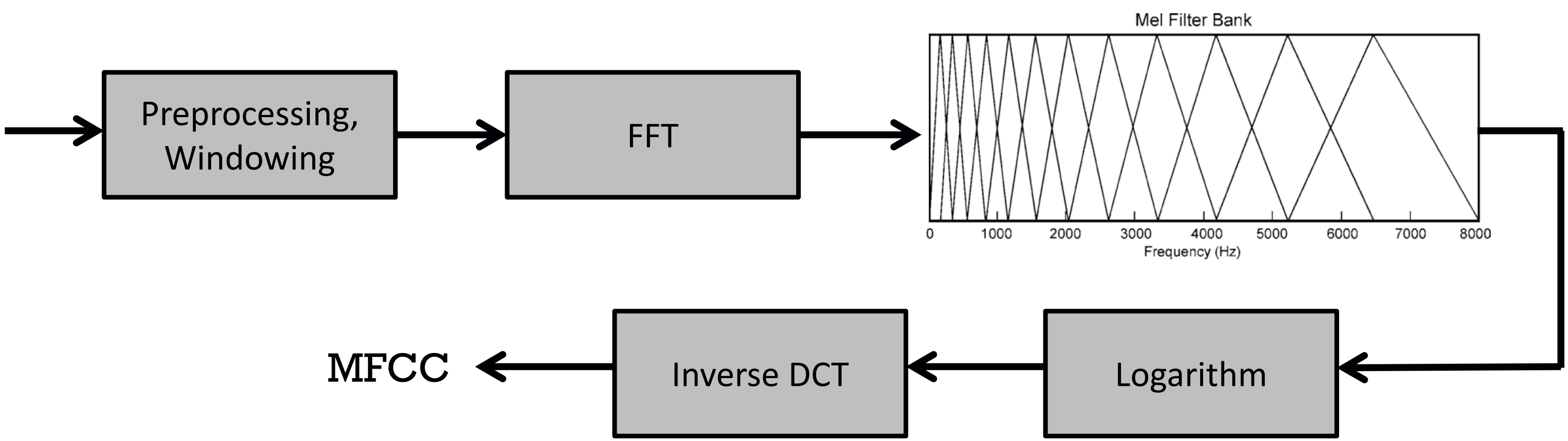
- Factors effecting feature discriminitivity:
  - Radar parameters (e.g. TX frequency)
  - Radar-target geometry (angle)
  - Dwell time
  - SNR
- How can we improve efficacy of features in different scenarios?
  - Adaptive feature selection
  - **Adaptive feature design**



S.Z. Gurbuz, B. Erol, B. Cagliyan, B. Tekeli, "Operational Assessment and Adaptive Selection of Micro-Doppler Features," IET RSN, 2015.

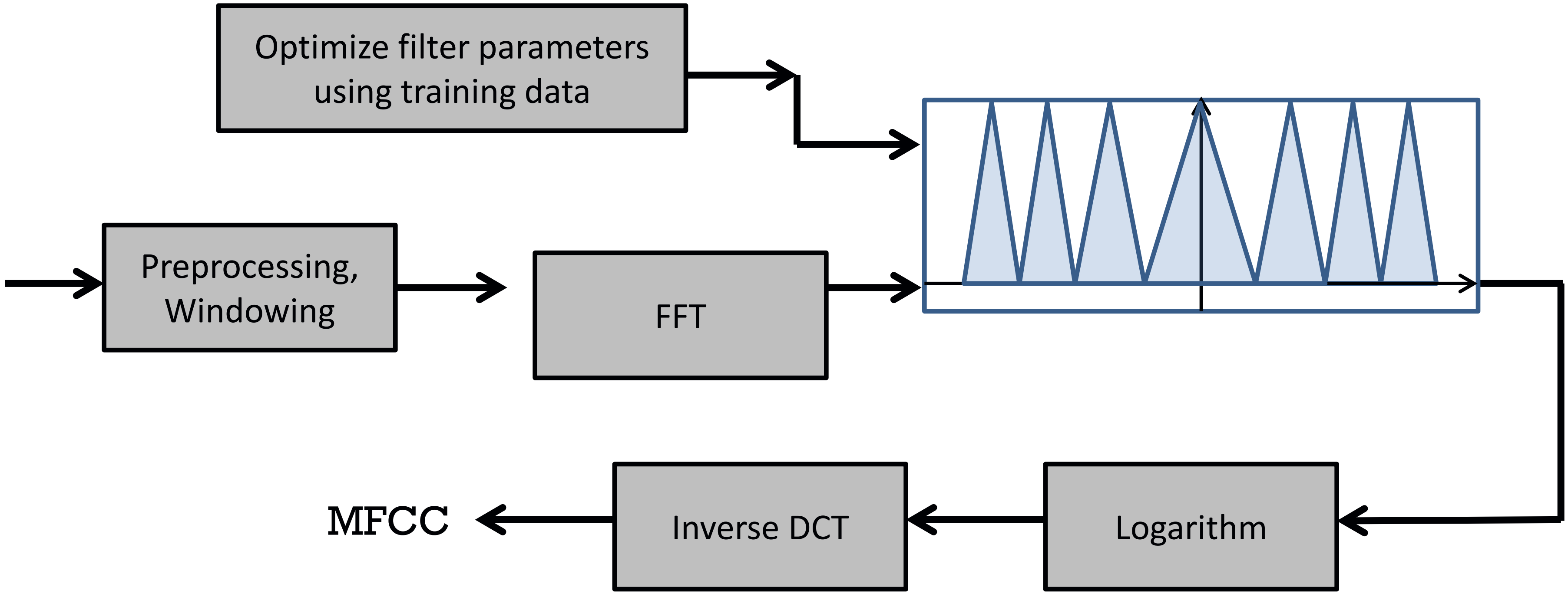
MFCC

- Mel-Frequency Cepstral Coefficients have been popular as features for micro-Doppler classification
  - Problem: mel-frequency scale designed to model human hearing and is irrelevant to physics of radar micro-Doppler



Sevgi Z. Gurbuz (szgurbuz@ua.edu)

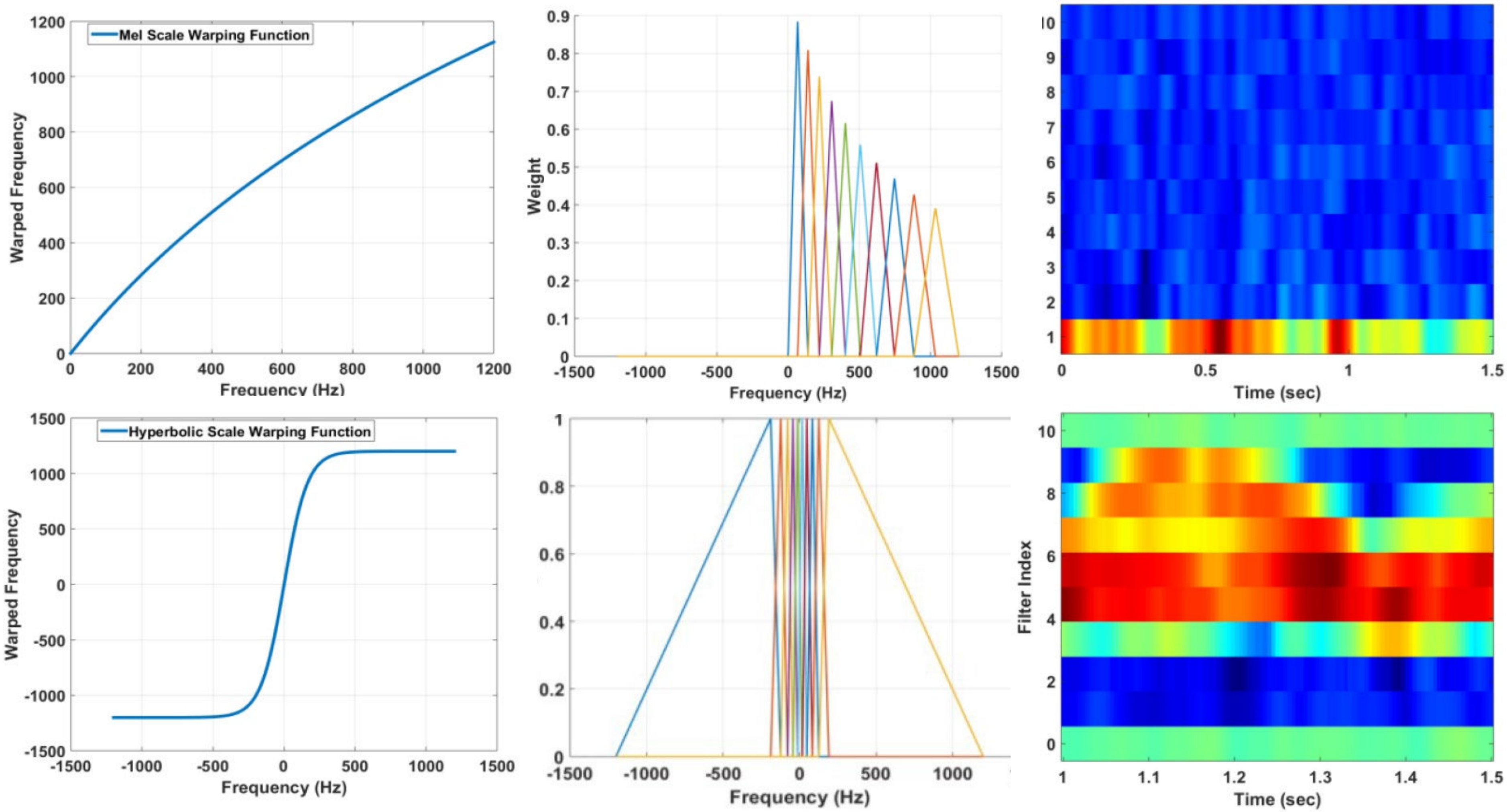
Data-Driven Feature Design



- e.g. if a hyperbolic function is chosen to specify filter spacing, optimize parameters a, b, and c to maximize classification performance $f_{HH} = a\,tanh(f_{Hz} - b)/c$

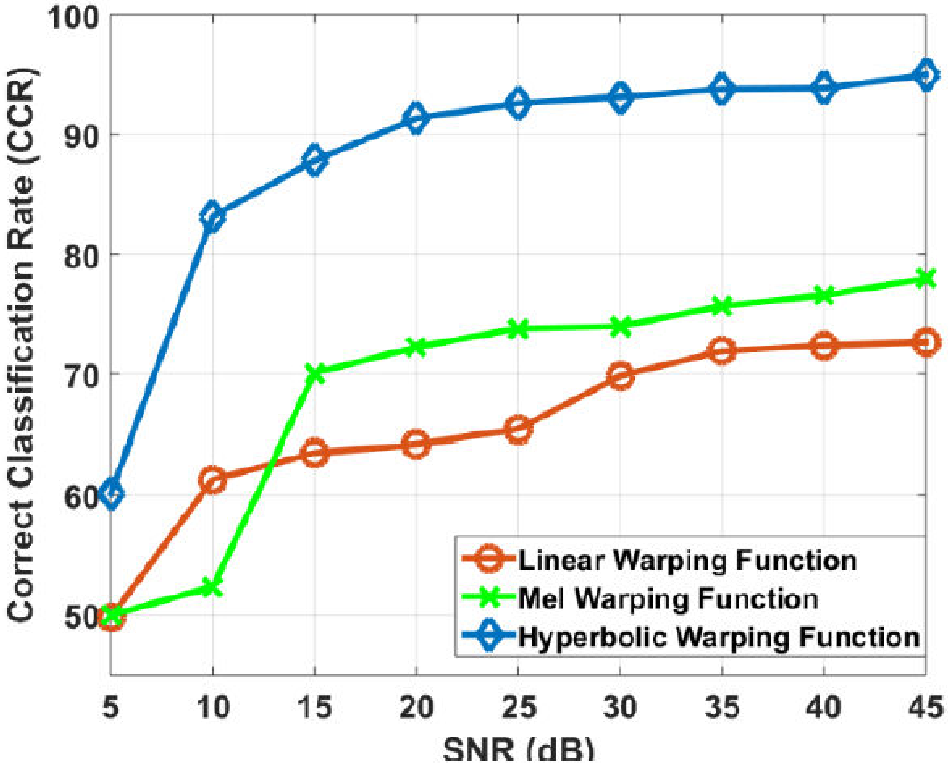
Sevgi Z. Gurbuz (szgurbuz@ua.edu)

Effect of Warping on Filter Bank



Performance Gains

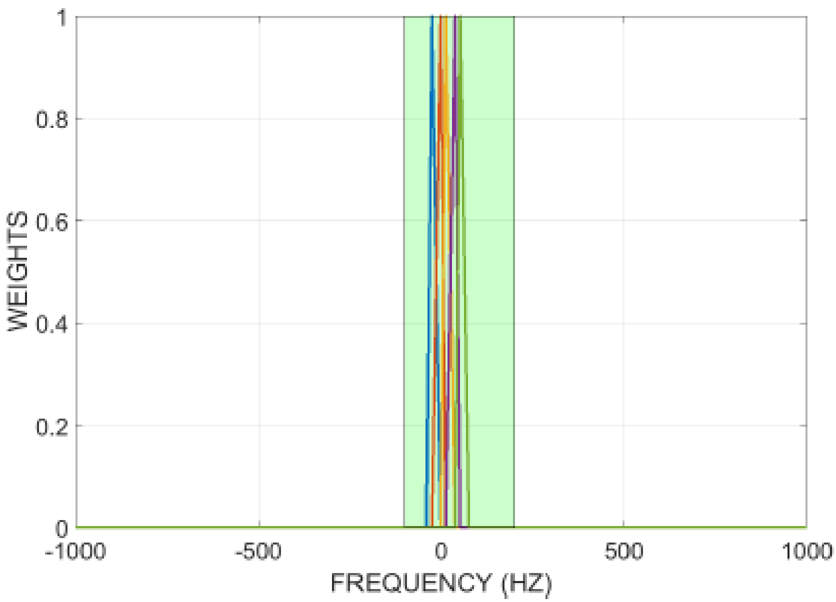
- Significant performance improvement over MFCC for classification of four activities:
  - walking
  - running
  - creeping
  - crawling



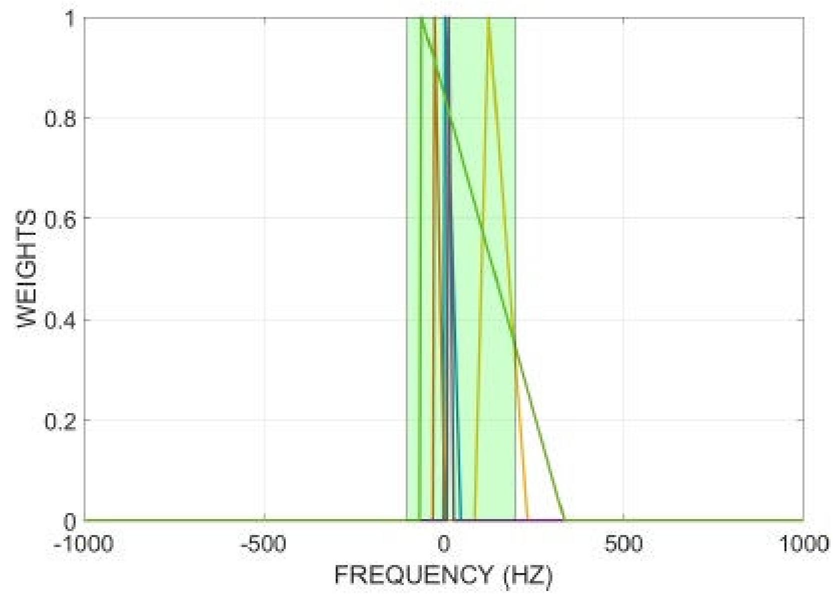
B. Erol and S.Z. Gurbuz, “Hyperbolically warped cepstral coefficients for improved micro-Doppler classification,” IEEE Radar Conference, 2016.

Filterbank Optimization

- Compare hyperbolic warped with genetic algorithm optimized filter bank

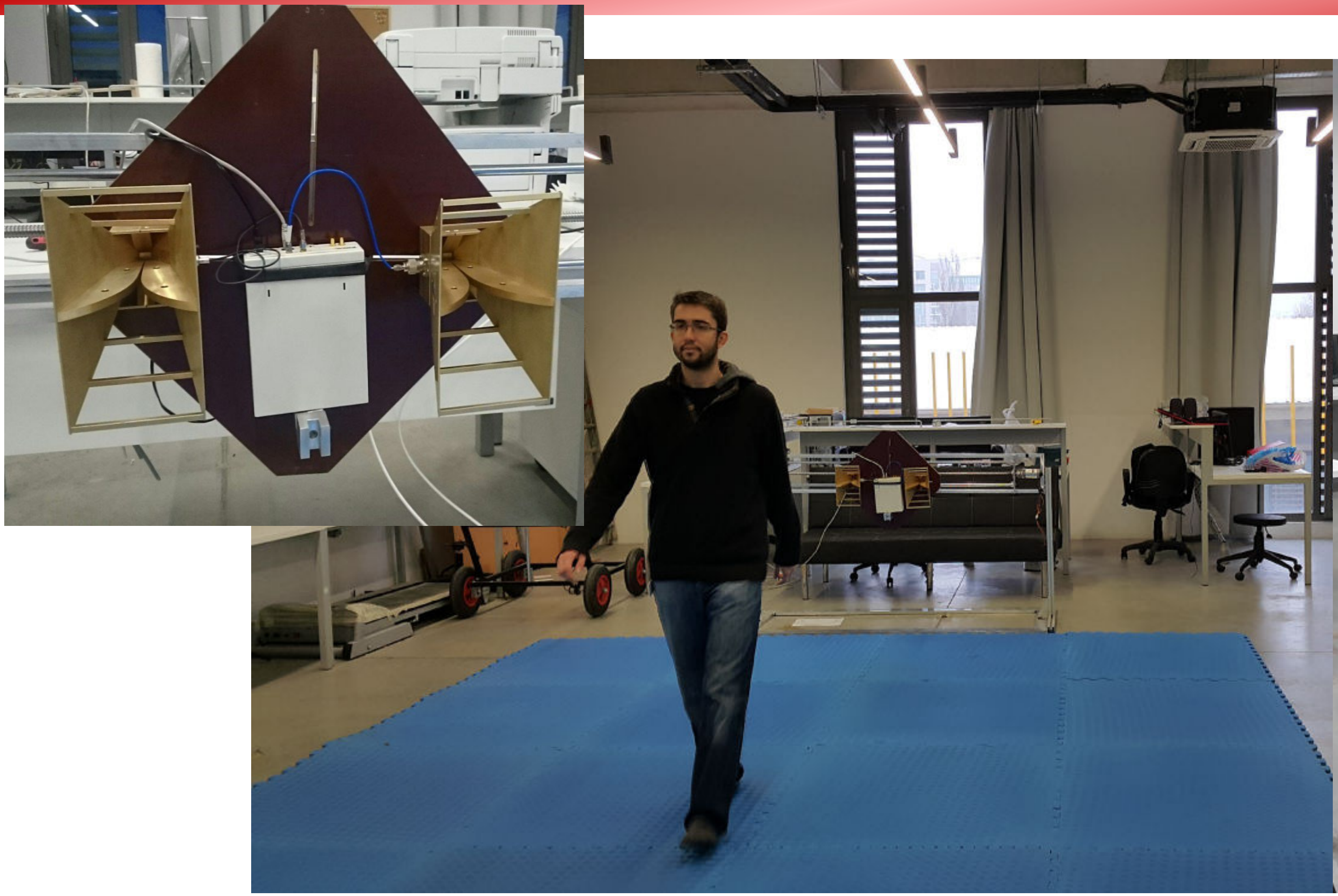


HWCC



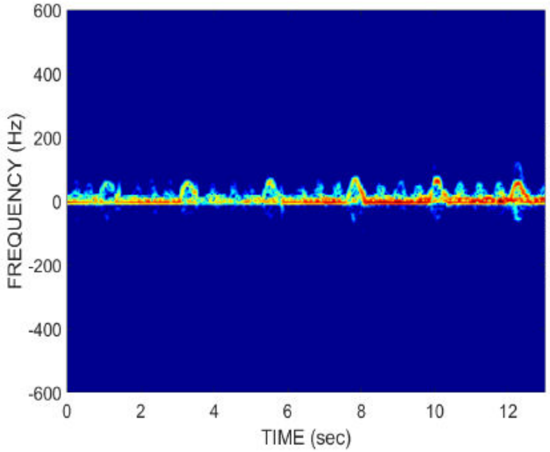
GA-Optimized FWCC

Experimental Dataset: 4 GHz CW

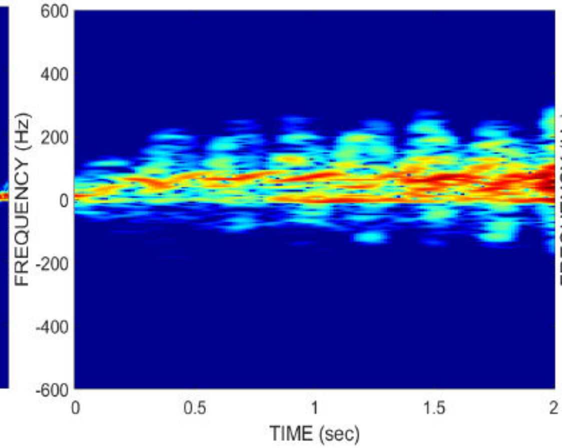


Sevgi Z. Gurbuz (szgurbuz@ua.edu)

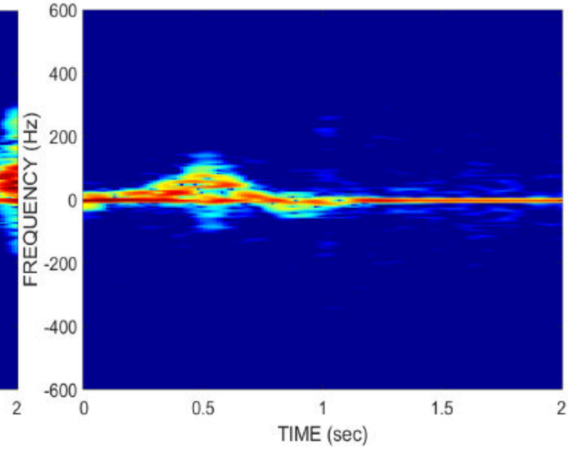
Sample Spectrograms



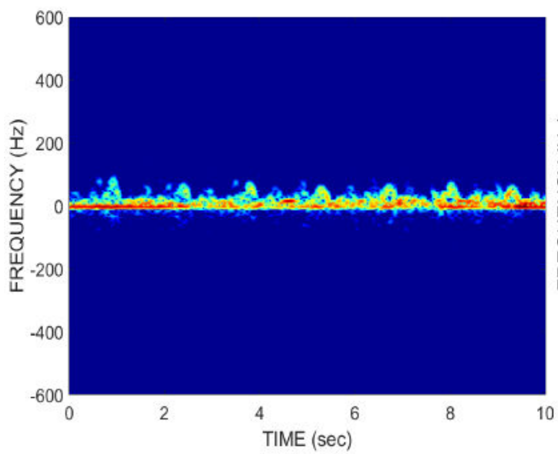
(a) C1: Walker



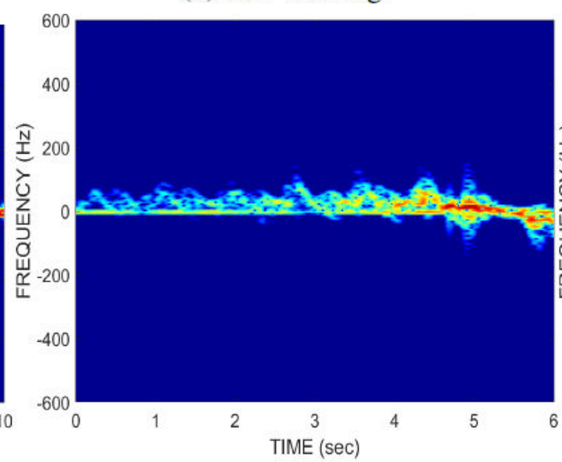
(b) C2: Walking



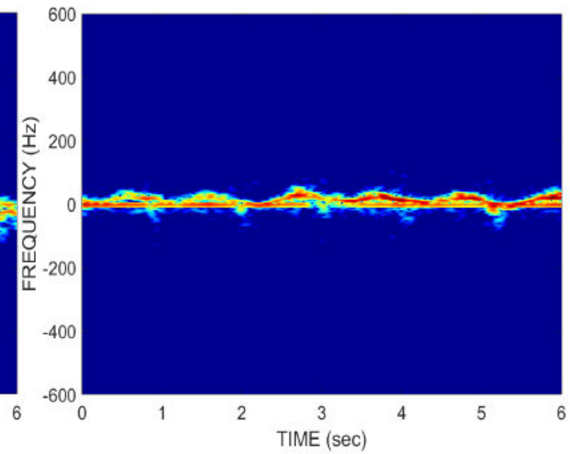
(c) C3: Falling sideways



(d) C4: Using tripod cane

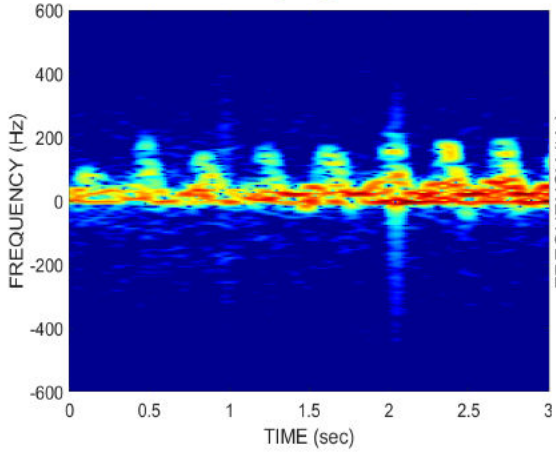


(e) C5: Limping

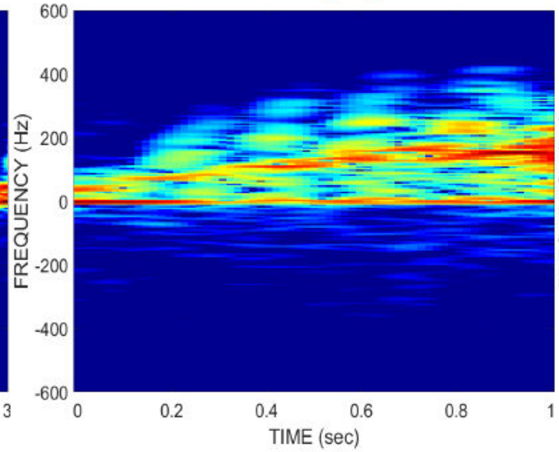


(f) C6: Wheelchair

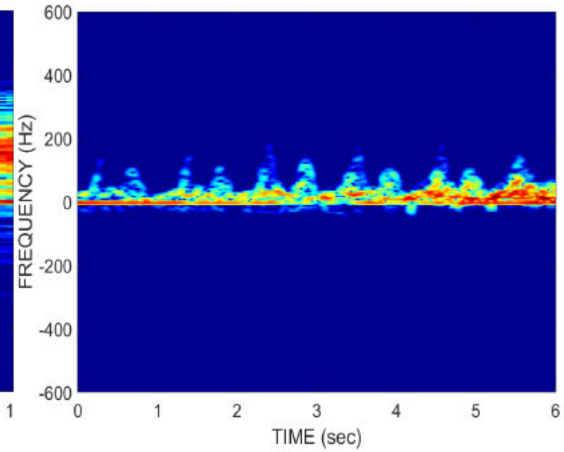
Sample Spectrograms, cont.



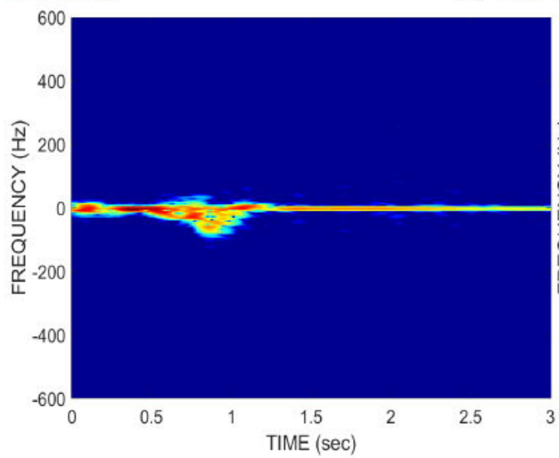
(g) C7: Crawling



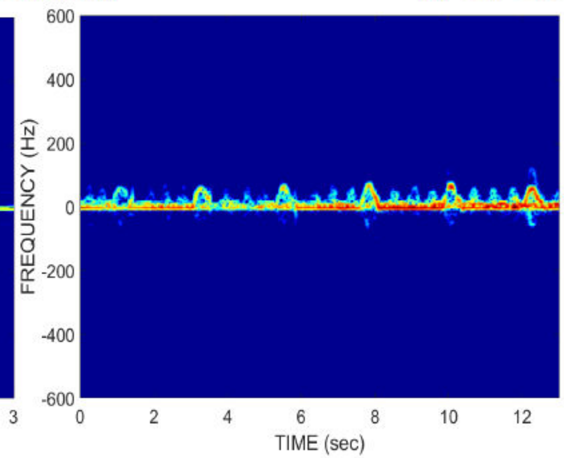
(h) C8: Running



(i) C9: Using crutches



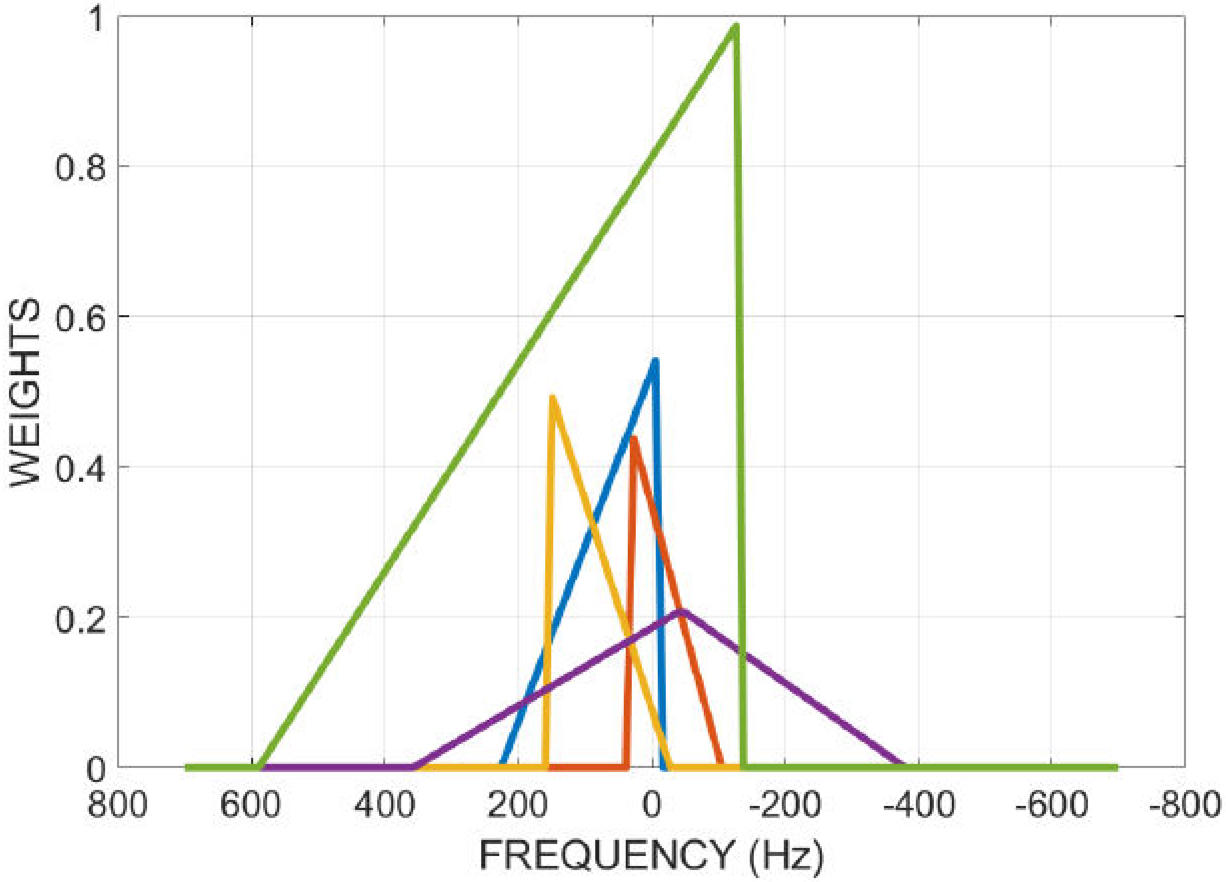
(j) C10: Sitting



(k) C11: Creeping

Sevgi Z. Gurbuz (szgurbuz@ua.edu)

GA-Optimized Filter Bank



Performance of GA-FWCC

94.1%

%	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11
c1	92.9	0	0	0	7.1	0	0	0	0	0	0
c2	0	100	0	0	0	0	0	0	0	0	0
c3	0	0	100	0	0	0	0	0	0	0	0
c4	24.8	0	0	60	7.6	0	7.6	0	0	0	0
c5	0	0	0	17.4	82.6	0	0	0	0	0	0
c6	0	0	0	0	0	100	0	0	0	0	0
c7	0	0	0	0	0	0	100	0	0	0	0
c8	0	0	0	0	0	0	0	100	0	0	0
c9	0	0	0	0	0	0	0	0	100	0	0
c10	0	0	0	0	0	0	0	0	0	100	0
c11	0	0	0	0	0	0	0	0	0	0	100

Sevgi Z. Gurbuz (szgurbuz@ua.edu)

Classification With Pre-Defined Features

- Physical Features

- Transform Based Features
  - Discrete Cosine Transform

- Speech-Processing Features
  - Cepstral Coefficients and Linear Predictive Coding

- A total of 127 features are supplied as input to the multi-class SVM classifier with polynomial kernel.

Multi-Class SVM Performance

Overall Accuracy: 69.7%

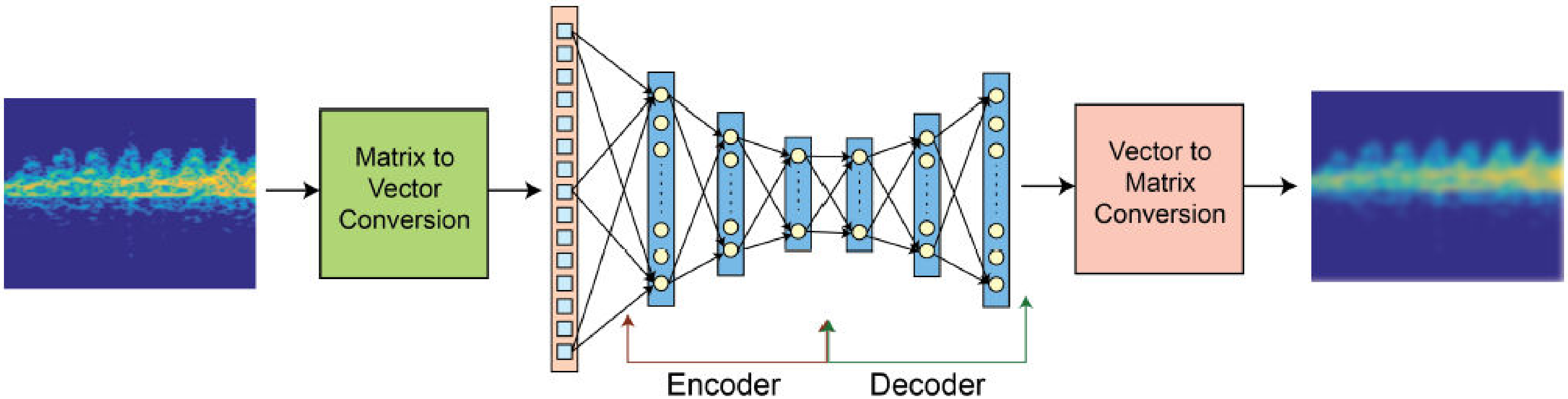
%	Walking	Wheel chair	Limping	Cane	Walker	Falling	Crutches	Creeping	Crawling	Jogging	Sitting	F.C
Walking	**93.2**	0	0	0	0	0	0	1.7	2.8	2.3	0	0
Wheelchair	0	**76.2**	5.3	9.5	5.8	0	1.4	1.3	0.5	0	0	0
Limping	0	9.1	**62.9**	**21.1**	4.1	0	2.6	0.2	0	0	0	0
Cane	0	13	11.9	**61.9**	13.2	0	0	0	0	0	0	0
Walker	0	5.6	5.3	9.9	**77.1**	0	2.1	0	0	0	0	0
Falling	**20.2**	0.6	0.6	0	0	**53.1**	0	0	1.9	0	1.3	**22.3**
Crutches	0.4	2.8	3	1.1	0	0	**90.3**	0	2.4	0	0	0
Creeping	**29.3**	0	0	0	0	0	0	**42.4**	**22.5**	5.8	0	0
Crawling	**29.6**	5.1	6.4	0	0.9	0	0.9	16.1	**29.9**	11.1	0	0
Jogging	4.3	0	0	0	0	0	0	3.1	2.5	**90.1**	0	0
Sitting	6.5	0	0	0	0	5.9	0	0	0	0	**79.9**	7.7
F.C	3.1	0	0	0	0	8.7	0	0	1.5	1.6	4.7	**80.4**

Deep Learning Architectures

- Deep neural networks build upon past research on artificial neural networks (ANNs) by increasing the overall size of the network using many layers of neurons.
  - Autoencoder
  - Convolutional Neural Network
  - Convolutional Autoencoder

Autoencoders

- An autoencoder (AE) is a feed-forward neural network that aims to reconstruct the input at the output under certain constraints.



e

inputs as

- σ denotes a non-linear activation function,
- W denotes weights and
- b denotes the biases of the encoder.

$$e_i = \sigma(Wx_i + b).$$

Autoencoders (cont'd)

- The encoded features are then decoded to reconstruct the given input vector x using

$$z_i = \sigma(\widetilde{W} e_i + \widetilde{b}).$$

- Here, $\widetilde{W}$ and $\tilde{b}$ denotes weights and biases of the decoder. During unsupervised pre-training, the network tries to minimize the reconstruction error $J(\theta) = \frac{1}{N}\sum_{i=1}^{N}(x_i - z_i)$

Autoencoders (cont'd)

- After unsupervised pre-training, the decoder is removed from the network and the remaining encoder components are trained in a supervised manner by adding a softmax classifier with 12 neurons after the encoder.
- Encoder layers have 200-100-50 neurons and decoder layers have 50 100 200 neurons. After unsupervised pre-training the decoder part is removed and, a softmax layer is added at the end of the encoder.
- Hyperbolic tangent activation is used for non-linearity.

Autoencoder Performance

Overall Accuracy: 84.6%

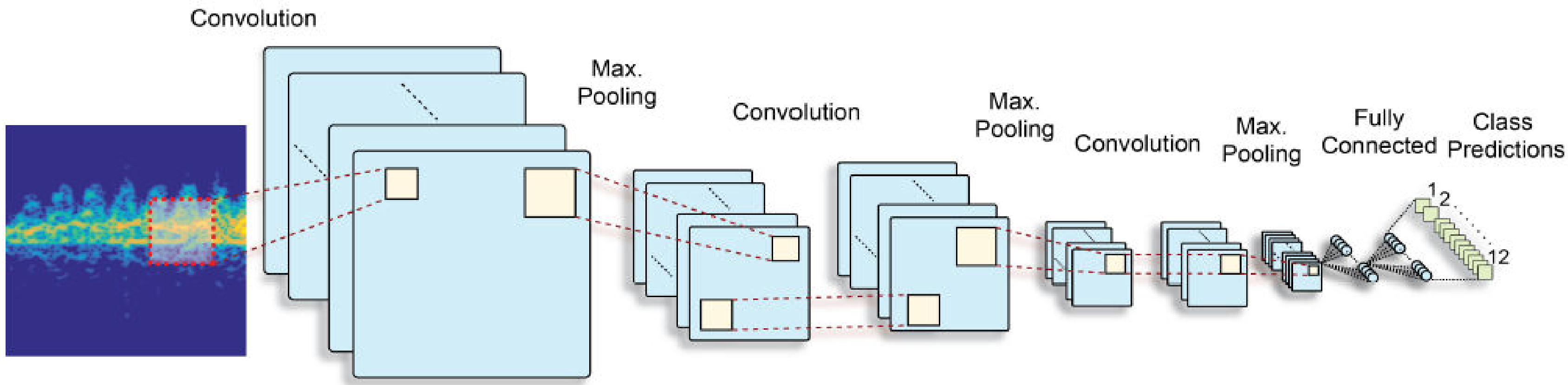
%	Walking	Wheel chair	Limping	Cane	Walker	Falling	Crutches	Creeping	Crawling	Jogging	Sitting	F.C
Walking	**90.9**	0	0	0	0	0	0	4.6	0	4.5	0	0
Wheelchair	0	**93.2**	3.2	0.1	3.5	0	0	0	0	0	0	0
Limping	0	0	**84.6**	6.9	8.3	0	0.2	0	0	0	0	0
Cane	0	0.6	9.4	**69.9**	20.1	0	0	0	0	0	0	0
Walker	0	0	0.5	6.1	**93.4**	0	0	0	0	0	0	0
Falling	0	0	0	0	0	**88.9**	0	0	0	0	0	11.1
Crutches	6.2	0	0.6	0	0	0	**91.9**	0	1.3	0	0	0
Creeping	0	0	0	0	0	0	0	**65.6**	34.4	0	0	0
Crawling	0.4	0	0	0	0	0	0	22.1	**65.1**	12.4	0	0
Jogging	0	0	0	0	0	0	0	0	0	**100**	0	0
Sitting	0	0	0	0	0	20.7	0	0	0	0	**79.3**	0
F.C	0	0	0	0	0	8.1	0	0	0	0	0	**91.9**

Convolutional Neural Networks(CNN)

- CNNs are the current state of the art for image classification.
- Unlike Autoencoders, CNNs can learn locally connected features, which is a fundamental requirement for classification of images.
- CNN architectures generally consist of three elements
  - convolutional layers, pooling layers and fully connected layers

CNN (cont'd)

- The CNN architecture implemented with three convolutional layers comprised of 32 3x3 filters each, and two fully connected layers with 150 neurons/layer.
- Rectified Linear Unit activation is used for non-linearity. The optimization computed using the ADAM algorithm. After each fully connected layer, a dropout operation is applied with a probability of 0.5



CNN Performance

Overall Accuracy: 86.4%

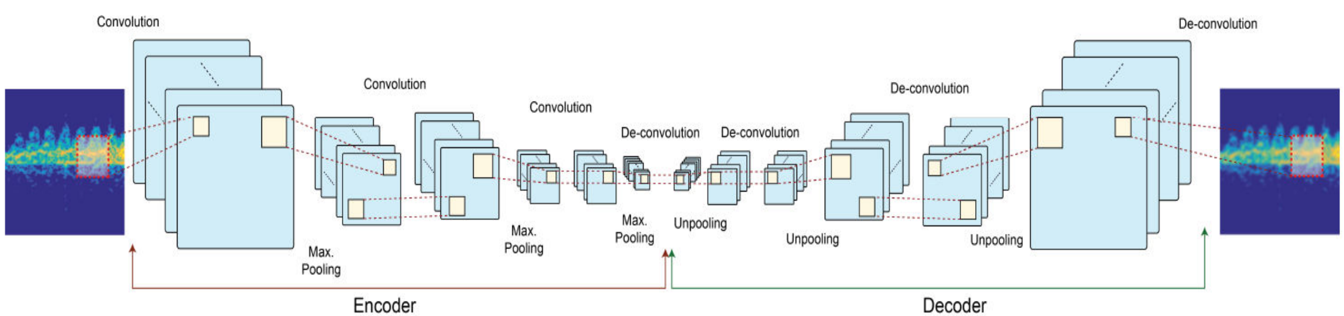
%	Walking	Wheel chair	Limping	Cane	Walker	Falling	Crutches	Creeping	Crawling	Jogging	Sitting	F.C
Walking	**100**	0	0	0	0	0	0	0	0	0	0	0
Wheelchair	0	**100**	0	0	0	0	0	0	0	0	0	0
Limping	0	0	**87.5**	8.4	4.1	0	0	0	0	0	0	0
Cane	0	0	0	**71.4**	28.6	0	0	0	0	0	0	0
Walker	0	0	0	4.2	**95.8**	0	0	0	0	0	0	0
Falling	0	0	0	0	0	**60**	0	0	0	0	0	40
Crutches	0	0	10	0	0	0	**90**	0	0	0	0	0
Creeping	15.4	0	0	0	0	0	7.7	**46.1**	15.4	15.4	0	0
Crawling	0	0	0	0	0	0	0	14.3	**85.7**	0	0	0
Jogging	0	0	0	0	0	0	0	0	0	**100**	0	0
Sitting	0	0	0	0	0	0	0	0	0	0	**100**	0
F.C	0	0	0	0	0	0	0	0	0	0	0	**100**

Convolutional Autoencoder

- Convolutional autoencoders combine the benefits of convolutional filtering in CNN's with unsupervised pre-training of autoencoders.
- Thus, for a given input maxrix P, the encoder computes

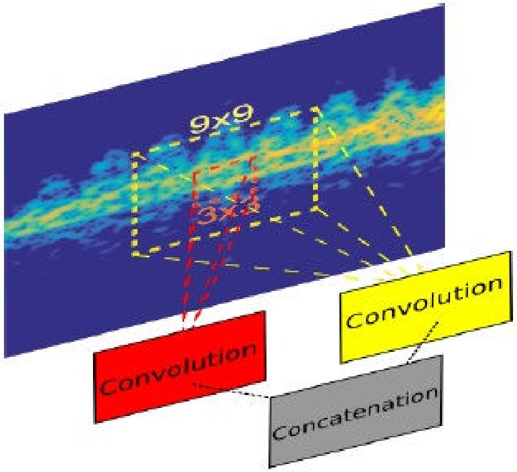
$$e_i = \sigma(P * F^n + b)$$

  - where σ denotes activation function, * represents 2D convolution, F^n is n^{th} 2D convolutional filter and b denotes encoder bias



Convolutional Autoencoder (cont'd)

- In this work, a filter concatenation technique is also applied to capture features of different resolutions from the input. Two convolutional filters of different sizes are used:
  - 9x9 filters capture general features
  - 3x3 filters capture fine details



Convolutional Autoencoder (cont'd)

- Similar to AE the reconstruction can be obtained using

$$z_i = \sigma(e_i * \widetilde{F}^n + \widetilde{b}).$$

- Unsupervised pre-training can be applied to the network, which aims to minimize following equation

$$E(\theta) = \sum_{i=1}^{m} (x_i - z_i)^2$$

- After unsupervised pre-training the decoder part is removed and 2 fully connected layers and a softmax classifier are added at the end of encoder.

Performance of CAE

94.2%

%	c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11
c1	100	0	0	0	0	0	0	0	0	0	0
c2	0	100	0	0	0	0	0	0	0	0	0
c3	0	0	94.5	0	0	0	0	0	0	5.5	0
c4	8.1	0	0	91.9	0	0	0	0	0	0	0
c5	0	0	0	6.9	93.1	0	0	0	0	0	0
c6	0	0	0	0	0	100	0	0	0	0	0
c7	0	0	0	0	0	0	91.5	0	0	0	8.5
c8	0	0	0	0	0	0	0	100	0	0	0
c9	0	0	0	0	0	0	0	0	100	0	0
c10	0	0	0	0	0	0	0	0	0	100	0
c11	0	4.9	0	0	0	0	22.4	0	7.2	0	65.5

Sevgi Z. Gurbuz (szgurbuz@ua.edu)

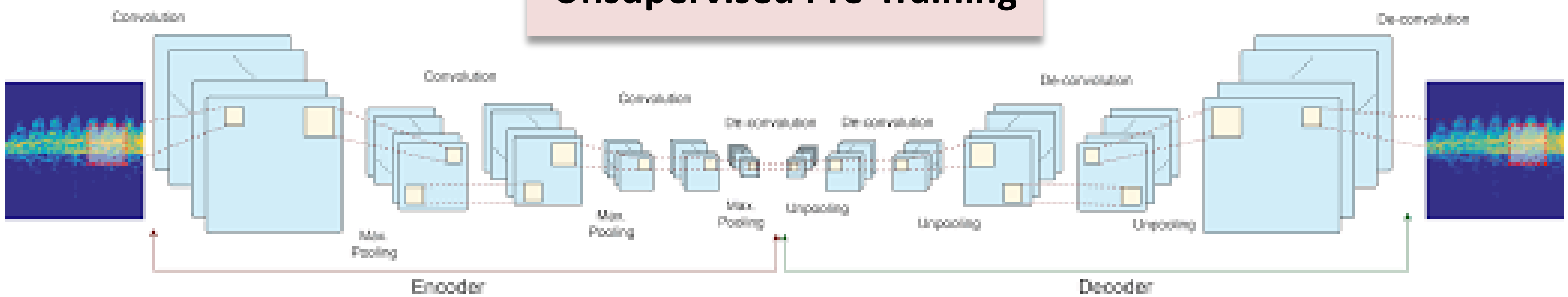
Comparative Results

	Traditional			AE			CNN			CAE		
Gait	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy	Precision	Recall	Accuracy
Walking	0.93	0.5	93.2	0.9	0.93	90.9	1	0.87	**100**	1	0.95	**100**
Wheel chair	0.76	0.68	76.2	0.93	0.99	93.2	1	1	**100**	1	1	**100**
Limping	0.63	0.66	62.9	0.85	0.86	84.6	0.87	0.90	87.5	0.92	1	**92.3**
Cane	0.62	0.6	61.9	0.7	0.84	69.9	0.71	0.85	71.4	0.91	0.92	**90.7**
Walker	0.77	0.76	77.1	0.93	0.75	93.4	0.96	0.74	95.8	1	0.91	**100**
Falling	0.53	0.78	53.1	0.89	0.75	88.9	0.6	1	60	0.89	1	**89.2**
Crutches	0.90	0.92	90.3	0.92	0.99	91.9	0.9	0.92	90	1	0.93	**100**
Creeping	0.42	0.65	42.4	0.65	0.71	**65.6**	0.46	0.76	46.1	0.65	0.89	**65.2**
Crawling	0.3	0.46	29.9	0.65	0.64	65.1	0.86	0.85	85.7	0.91	0.80	**91.7**
Jogging	0.9	0.81	90.1	1	0.86	100	1	0.87	**100**	1	1	**100**
Fastly Sitting	0.8	0.93	80	0.79	1	79.3	1	1	**100**	1	0.91	**100**
Falling from Chair	0.8	0.73	80.4	0.91	0.89	91.9	1	0.71	**100**	1	0.98	**100**
Average	***0.698***	***0.71***	***69.7***	***0.846***	***0.85***	***84.6***	***0.863***	***0.87***	***86.4***	***0.941***	***0.94***	***94.1***

TRANSFER LEARNING

Approaches for Training under Low Sample Support

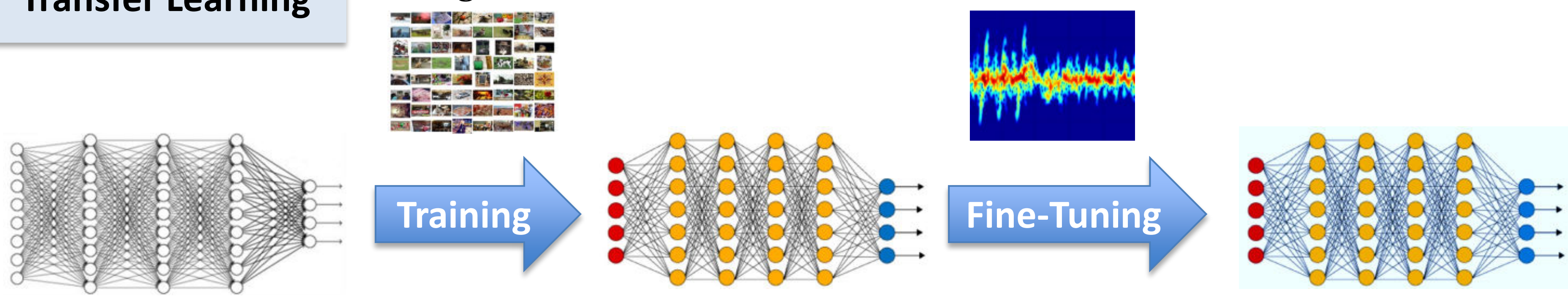
Unsupervised Pre-Training



Transfer Learning

ImageNet

RF Data



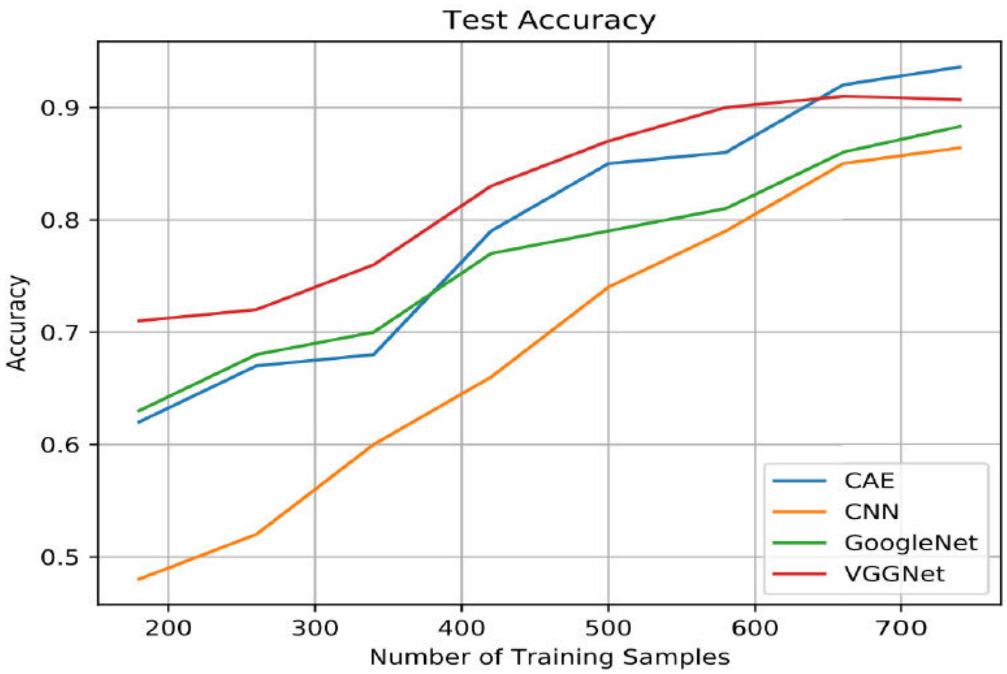
Randomly Initialized DNN

DNN Pre-Trained on Data from Domain A

DNN Fine-Tuned on Data from Domain B

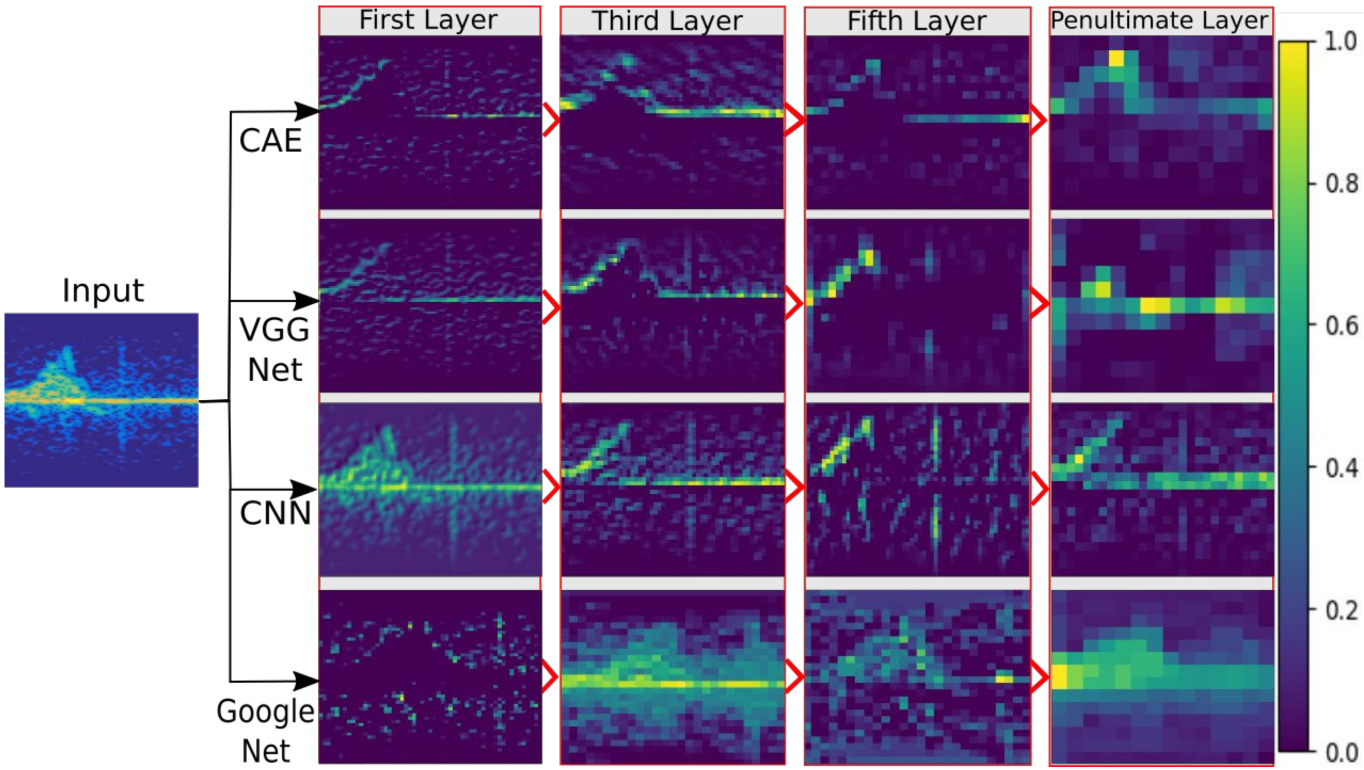
Transfer Learning vs. Convolutional Autoencoders

- Comparison of Initialization Approaches
  - CNN: random
  - Transfer Learning:
    - Data from diff. domain
  - CAE:
    - unsupervised pre-training
- Fine tune with real data afterwards



M.S. Seyfioglu, S.Z. Gurbuz, "Deep Neural Network Initialization Methods for Micro-Doppler Classification With Low Training Sample Support, IEEE Geoscience and Remote Sensing Letters, Dec. 2017.

What Do DNNs Learn?



M.S. Seyfioglu, S.Z. Gurbuz, “Deep Neural Network Initialization Methods for Micro-Doppler Classification With Low Training Sample Support, IEEE Geoscience and Remote Sensing Letters, Dec. 2017.

ONE POSSIBLE SOLUTION:

ACQUIRE DATA FROM MULTIPLE RADARS

How Can We Exploit "Datasets of Opportunity" ?

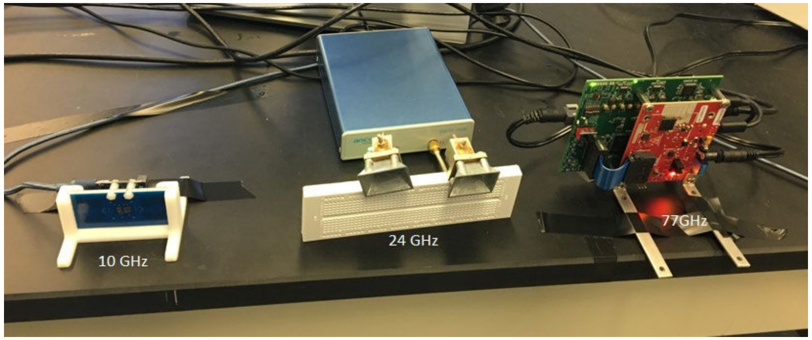
- Different sources of real RF data:
  - In an RF sensor network:
    - Different frequency
    - Different angle
    - But observing the same participant
  - Similar experiments conducted elsewhere
    - Same/different frequency/angle
    - Different participants
  - RF datasets of motion classes, frequency, angle, and participants

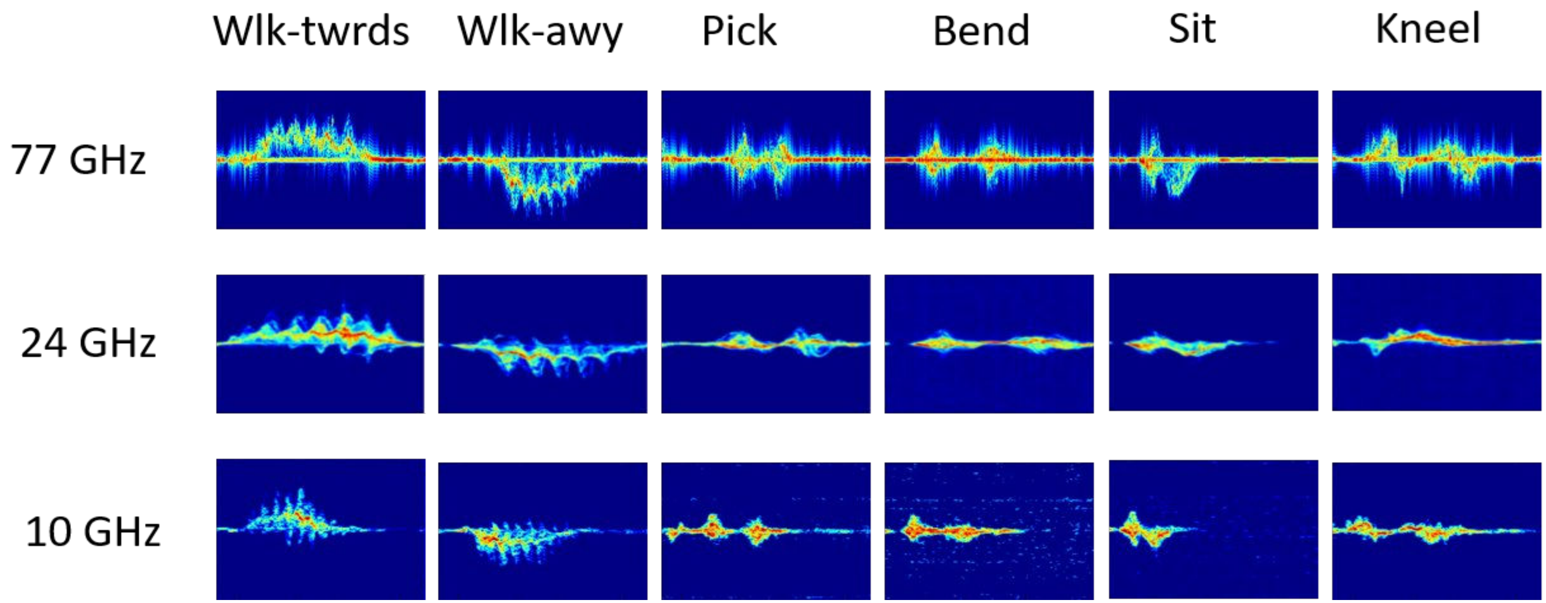
Sevgi Z. Gurbuz (szgurbuz@ua.edu)

Cross-Frequency Training of RF Data

Three RF Sensors:

- **77 GHz** TI IWR 1443
- **24 GHz** Ancortek SDR-KIT
- **<10 GHz** XeThru X4M03





Cross-Frequency Classifciation with Transfer Learning from VGGnet

❑ VGG16 net with top layer modification

- Global average pooling followed by 2 fully connected layers
- Drop out: 0.5
- 77 GHz: batch size 8,Learning rate 2e-4, two Dense layers of size 256, Decay 1e-6, Adam Optimizer
- 24 GHz: batch size 32,Learning rate 1e-4, two Dense layers of size 256, Decay 1e-6, Adam Optimizer
- 10 GHz: batch size 8,Learning rate 2e-4, two Dense layers of size 128, Decay 1e-6, Adam Optimizer

Training	Testing	Accuracy (%)
77 GHz	10 GHz	14.28
	24 GHz	16.66
	77 GHz	89.23
24 GHz	24 GHz	85.57
	10 GHz	15.55
	77 GHz	11.13
Xethru	10 GHz	83.00
	24 GHz	14.21
	77 GHz	9.00

Performance degrades while training and testing with different frequency data

Cross-Frequency Classification with Convolutional Auto-Encoder (CAE)

- **11 different classes:**
  - **60 samples per class for 77 & 10 GHz**
  - **150 samples per class for 24 GHz**

- **CAE:** Total of 5 layers
  - **When decoder removed, 2 dense layers followed by a soft-max layer added**
  - **Number of filters in each layer: 64**
  - **Filter Size: 3x3 & 9x9 filters are concatenated**

Pre-train	Fine Tune	Test	Testing Accuracy
77 GHz	77 GHz	77 GHz	91.5%
		24 GHz	22.5%
		10 GHz	18.9%
	24 GHz	24 GHz	74.4%
	10 GHz	10 GHz	75.5%
24 GHz	24 GHz	24 GHz	91.2%
		77 GHz	28.5%
		10 GHz	40.5%

Pre-train	Fine Tune	Test	Testing Accuracy
24 GHz	77 GHz	77 GHz	83.8%
	10 GHz	10 GHz	81.6%
10 GHz	10 GHz	10 GHz	91.8%
		77 GHz	24.1%
		24 GHz	18.8%
	77 GHz	77 GHz	80.0%
	24 GHz	24 GHz	79.1%

MULTI-MODAL FUSION

Multi-Frequency Radar Network

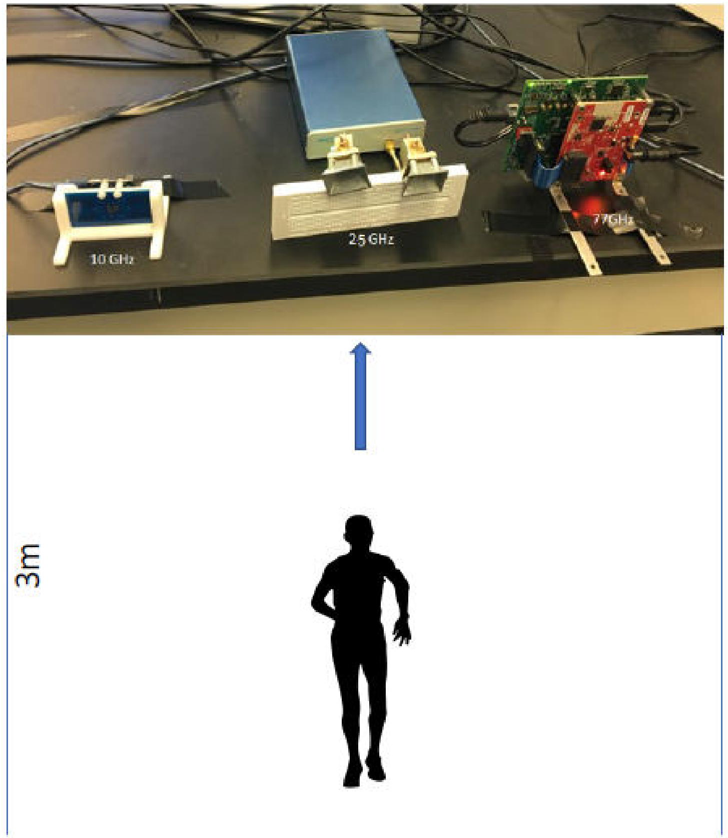
3 Radars operate at different transmit frequencies:

- TI IWR1443 FMCW radar
- Ancortek SDR-Kit 25 GHz FMCW
- XeThru 10GHz UWB impulse radar

11 Human activity classes

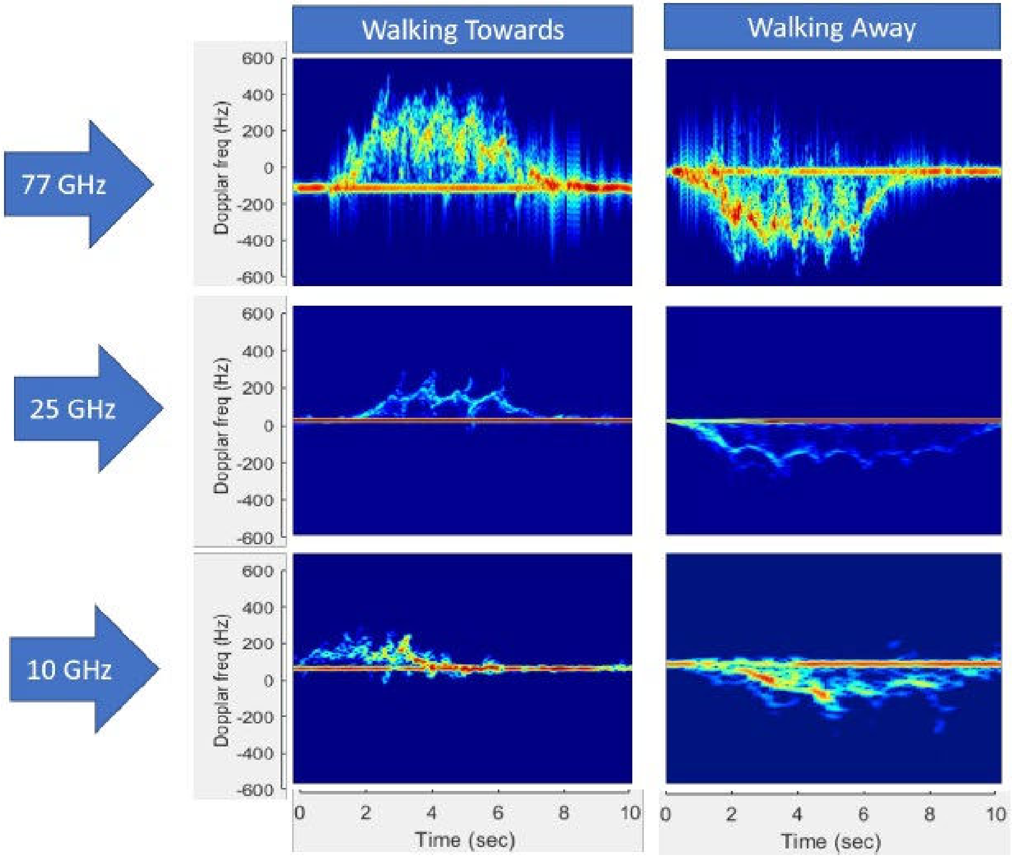
- 6 participants, 10 iterations of each activity
- 660 samples per sensor

Walking Towards	Limping	Kneeling
Walking Away	Short step walking	Crawling
Sitting on a chair	Scissor gait walking	Bending
Picking up an object	Walking on Toes	



Sevgi Z. Gurbuz (szgurbuz@ua.edu)

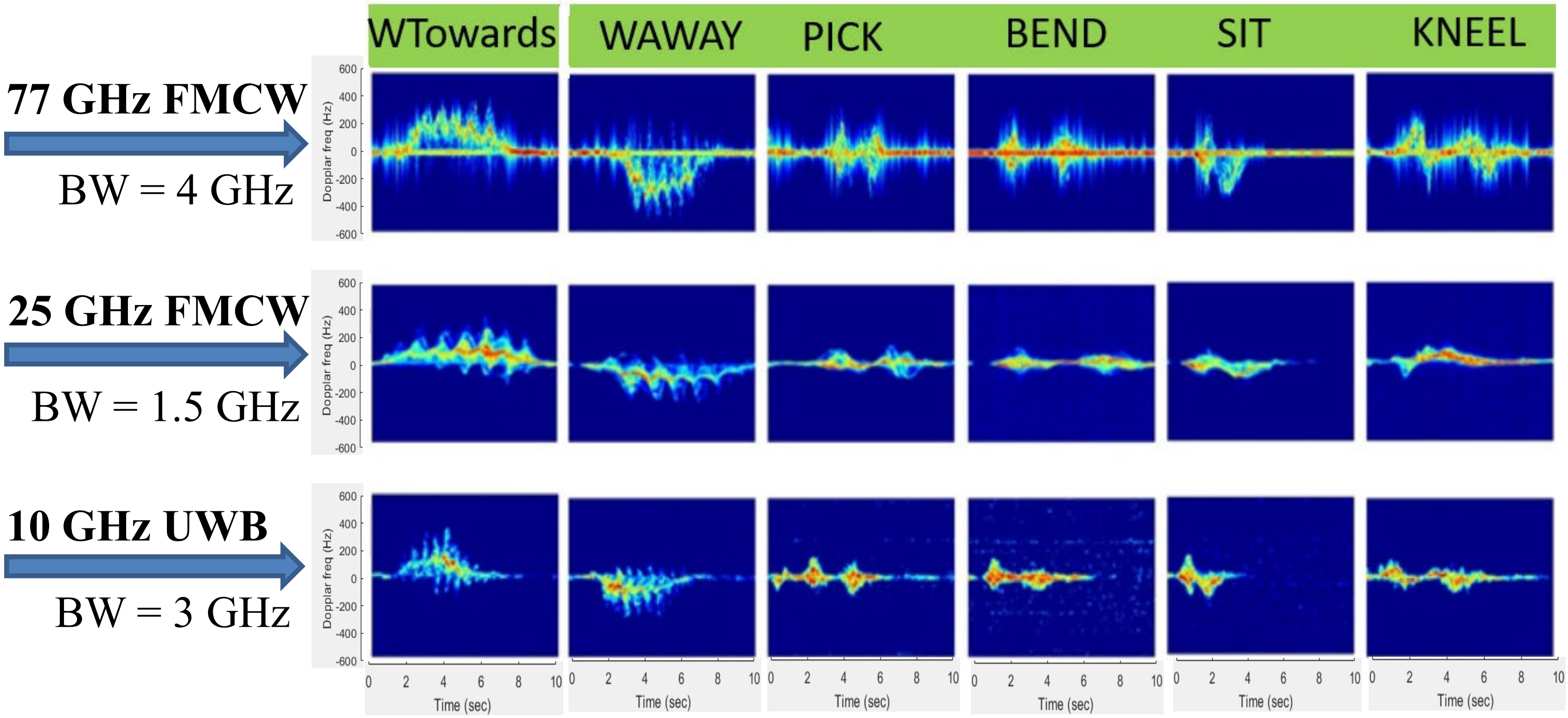
Difference in Target and Clutter Signatures



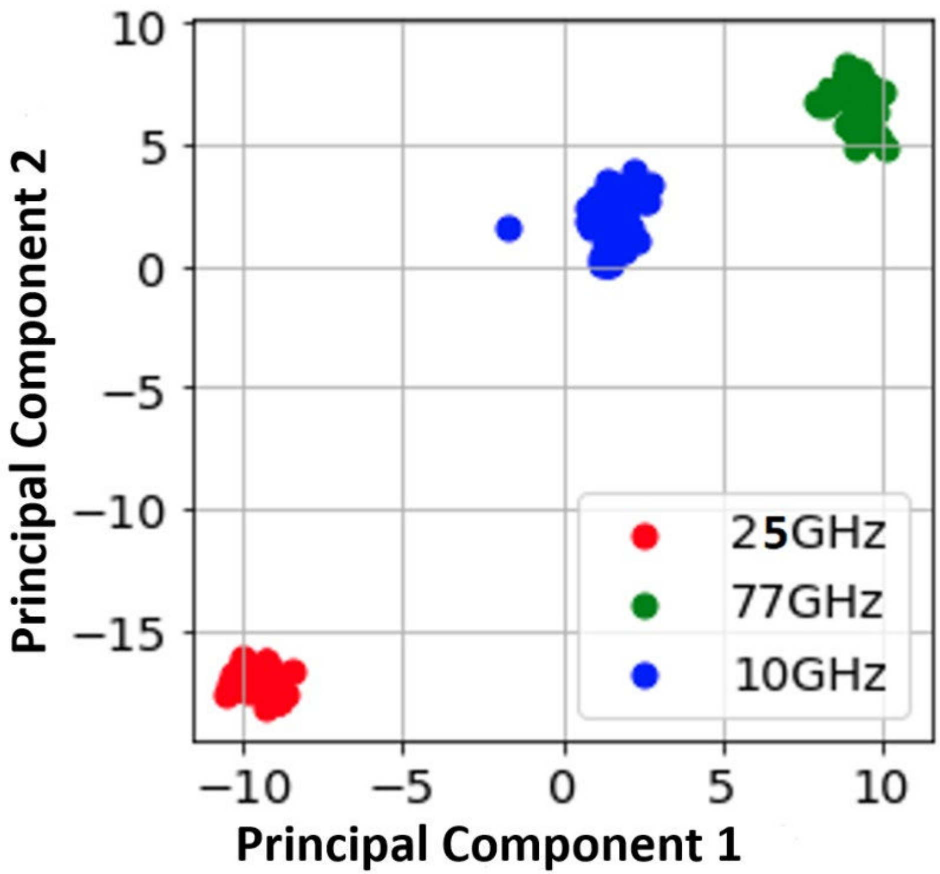
- ❖ Removing Ground clutter for 77GHz degrades classification performance for fine grain motion recognitions by 6% [1].
- ❖ For 25 and 10 GHz, presence of ground clutters weaken the signal strength.

[1] S. Z. Gurbuz et al., "American Sign Language Recognition Using RF Sensing," in IEEE Sensors Journal, vol. 21, no. 3 Feb.1, 2021,

Observations of mD Signatures



Visualization of Feature Space with t-SNE



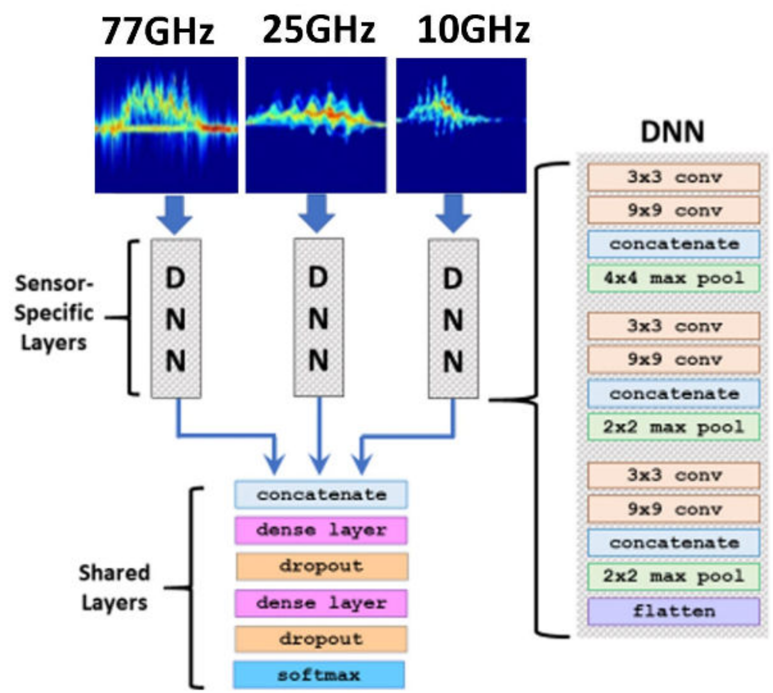
Cross-Frequency Fusion

Modality Tuning:

1. Freeze the shared layers and train the sensor specific layers.

2. After some number of epochs, unfreeze sensor specific layers and train the entire network end-to-end.

L. Castrejon, Y. Aytar, C. Vondrick, H. Pirsiavash, and A. Torralba, ´ “Learning aligned cross-modal representations from weakly aligned data,” in 2016 IEEE CVPR, 2016, pp. 2940–2949.



Training Type	Network	Accuracy
No Modality Tuning	Cross Modal DNN	93.00%
With Modality Tuning	Cross Modal DNN	98.46%

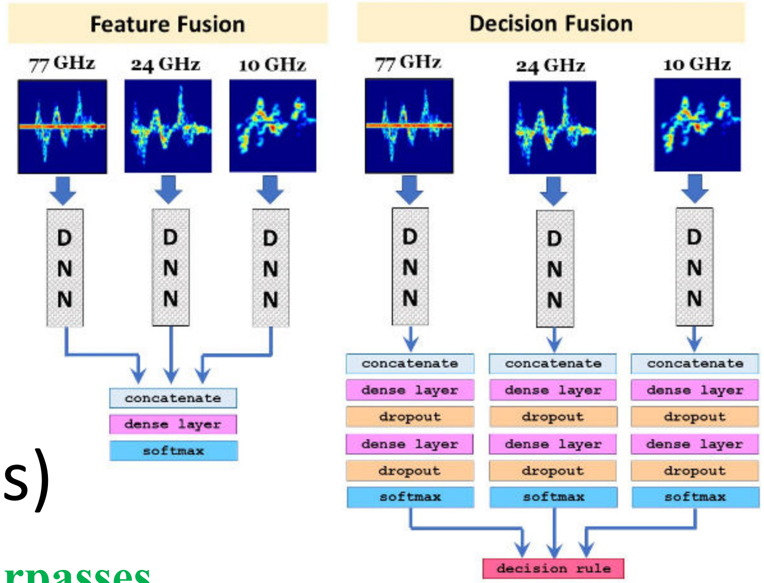
Comparison with Other Types of Fusion

- More challenging multi-frequency (77/25/10 GHz) dataset:

  → Recognition of **20 ASL Signs**

  ➢ 50 samples / class

  ➢ 5 fluent signers: 2 deaf + 3 Child-of-Deaf Adults (CODAs)



Fusion Type	Accuracy
Multi-Frequency Fusion DNN	95.53%
w/No Modality Tuning	83.67%
Feature Level Fusion	79.59%
Decision Fusion	75.00%

Surpasses alternative types of fusion