

### Design and Analysis of Convolutional Neural Network for RF Signal Modulation Classification for In-Orbit Deployment

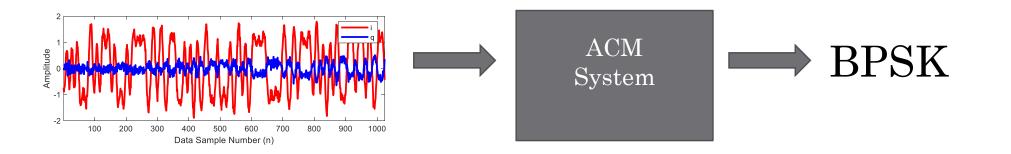
Cognitive Communications for Aerospace Applications (CCAA) Workshop June 21, 2021

**Chris Yakopcic (Prixarc),** Tarek M. Taha (Brisk), Sanjeevi Sirisha Karri (Prixarc), Guru Subramanyam (Prixarc), Aaron D. Smith (NASA Glenn), Janette C. Briones (NASA Glenn)

### **CNN** for Automatic Modulation Classification



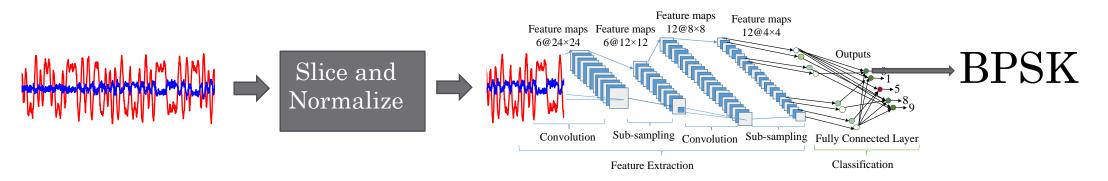
- Automatic signal modulation classification (AMC) is a major research direction of signal recognition.
- AMC is the automatic identification of the modulation format of the transmitted signals by observing the received data samples which are corrupted by the noise and fading channels.
- It is an intermediate operation between the signal detection and the data demodulation
- AMC plays an important role in civilian and military applications such as software-defined radio, cognitive radio, dynamic spectrum management, interference identification and electronic warfare.



### **CNN** for Automatic Modulation Classification



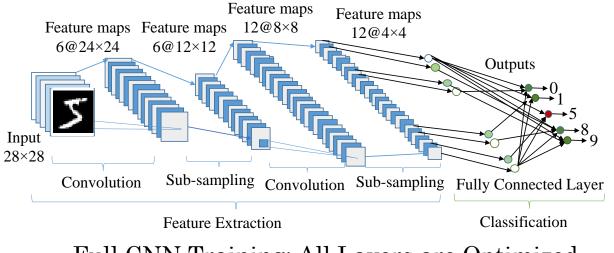
- Deep Learning (DL) has been described as a universal learning approach that is able to solve many types problems in different application domains.
- Our focus is on implementing a DL engine in space that would enable Automatic Modulation Classification (AMC) outside of Earth's atmosphere. Implementation of modulation recognition algorithm would allow for the deployment of real-time, high rate, low-power and useful neural network for RF communications.
- We explored a Convolutional Neural Network (CNN), and a Convolution Neural Network that Implements Transfer Learning (CNN-TL) for the successful classification of different modulation schemes for data transmission.
- The developed software was shown to successfully classify the modulation schemes using the open source Radio ML 2018 dataset.



### CNN Algorithm



- Learns by extracting features from data samples using trainable convolution kernels (filters)
- Last layer is typical fully connected layer
- Very strong for image recognition and classification

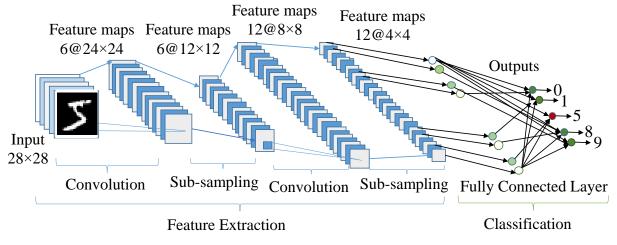


Full CNN Training: All Layers are Optimized

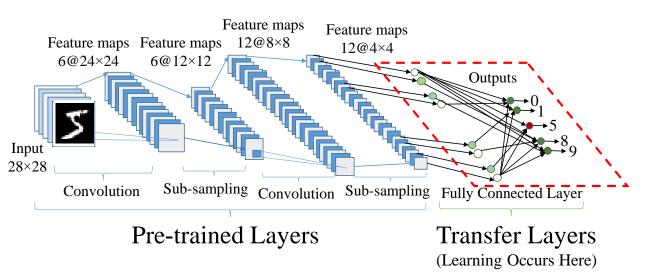
### CNN for Transfer Learning



- Learns by extracting features from data samples using trainable convolution kernels (filters)
- Last layer is typical fully connected layer
- Very strong for image recognition and classification
- Transfer Learning
  - Pre-train the convolution part of the network
  - Train only the fully connected layer with new data
  - Much simpler to implement in hardware



#### Full CNN Training: All Layers are Optimized

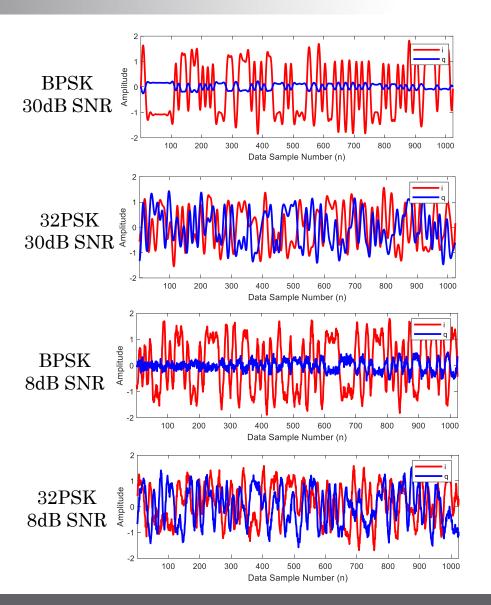


## Radio ML Dataset



- Dataset used in this study
  - RadioML 2018
  - 24 Modulation Classes
  - SNR range: -20 to 30bB
- 4069 samples for each class in each SNR segment
- Sample size 2×1024 for CNN input
  - I channel
  - $\cdot$  Q channel
- Little post processing within data
  - Cut sample lengths
  - Normalize to zero mean and unit variance

Class Number	Mod. Class
1	OOK
2	4ASK
3	8ASK
4	BPSK
5	QPSK
6	8PSK
7	16PSK
8	32PSK
9	16APSK
10	32APSK
11	64APSK
12	128APSK
13	16QAM
14	32QAM
15	64QAM
16	128QAM
17	256QAM
18	AM-SSB-WC
19	AM-SSB-SC
20	AM-DSB-WC
21	AM-DSB-SC
22	$\mathbf{FM}$
23	GMSK
24	OQPSK



# **CNN** Transfer Learning



• (	CNN used for	r transfer le	earning test			Class Number	Mod. Class
	• 2 convolution la	avers				1	OOK
		C C				2	4ASK
	<ul> <li>2 fully connected</li> </ul>	ed layers				3	8ASK
						4	BPSK
• T	Dataset broken	into two groun	ດຮ			5	QPSK
· 1	Jataset broken		00			6	8PSK
	<ul> <li>Fully train on a</li> </ul>	one set		Transfer Set A-	Ł	7	16PSK
	Transfer learn	the other got				•	
	• Transfer learn	the other set				8	32PSK
						9	16APSK
						10	32APSK
	Layer (type)	Output Shape	Param #			11	64APSK
	======================================	[-1, 32, 1024, 1]	1,632			12	128APSK
	ReLU-2	[-1, 32, 1024, 1]	0			13	16QAM
	MaxPool2d-3	[-1, 32, 512, 1]	0			14	32QAM
	Conv2d-4 ReLU-5	[-1, 64, 512, 1] [-1, 64, 512, 1]	51,264 Ø			15	64QAM
	MaxPool2d-6	[-1, 64, 256, 1]	0			16	128QAM
	Dropout-7 Linear-8	[-1, 16384] [-1, 1000]	0 16,385,000			17	256QAM
	Linear-9	[-1, 24]	24,024			18	AM-SSB-WC
	Tatal mamage 16 461 020			Transfer Set B-	$\mathbf{I}$		
	Total params: 16,461,920 Trainable params: 16,461,	,920				<u>    19                                </u>	AM-SSB-SC
	Non-trainable params: 0					20	AM-DSB-WC
	Input size (MB): 0.01					21	AM-DSB-SC
	Forward/backward pass siz	ze (MB): 1.38				22	$\mathbf{FM}$
	Params size (MB): 62.80					23	GMSK
	Estimated Total Size (MB)	): 64.19				24	OQPSK
							Č Č

.

## **CNN** Transfer Learning



#### • CNN used for transfer learning test

- 2 convolution layers
- 2 fully connected layers
- Dataset broken into two groups
  - Fully train on one set
  - Transfer learn the other set

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 1024, 1]	1,632
ReLU-2	[-1, 32, 1024, 1]	0
MaxPool2d-3	[-1, 32, 512, 1]	0
Conv2d-4	[-1, 64, 512, 1]	51,264
ReLU-5	[-1, 64, 512, 1]	0
MaxPool2d-6	[-1, 64, 256, 1]	0
Dropout-7	[-1, 16384]	0
Linear-8	[-1, 1000]	16,385,000
Linear-9	[-1, 24]	24,024
Total params: 16,461,920 Trainable params: 16,461,920 Non-trainable params: 0		
Input size (MB): 0.01 Forward/backward pass size (MParams size (MB): 62.80 Estimated Total Size (MB): 6		

#### Learn Set A

		оок	4ASK	8ASK	BPSK	OPSK	8PSK	16PSK	32PSK	16APSK	32APSK	64APS	(128AP	•
12	28APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.89	
6	54APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.96	0.04	
3	32APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.94	0.01	0.05	
-	16APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	
	32PSK-	0.00	0.00	0.00	0.00	0.00	0.04	0.09	0.87	0.00	0.00	0.00	0.00	
TR	16PSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.07	0.58	0.36	0.00	0.00	0.00	0.00	
UE	8PSK-	0.00	0.00	0.00	0.00	0.00	0.75	0.07	0.19	0.00	0.00	0.00	0.00	
	QPSK-	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
	BPSK-	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
	8ASK-	0.00	0.02	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
	4ASK-	0.00	0.99	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
	оок	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	

#### PREDICTED

#### Learn Set B, Transfer Learn Set A

	_												
	OOK	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	4ASK -	0.00	0.92	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	8ASK-	0.00	0.23		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	BPSK <sup>-</sup>	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	QPSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.85	0.04	0.07	0.04	0.00	0.00	0.00	0.00
ПE	8PSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.12		0.18	0.12	0.00	0.00	0.00	0.00
TRUE	16PSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.11	0.15	0.63	0.11	0.00	0.00	0.00	0.00
	32PSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.14	0.16	0.20	0.49	0.00	0.00	0.00	0.00
:	16APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
3	32APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
(	54APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.05
12	28APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.18	0.81
		оок	4ASK	8ASK	BPSK	QPSK	8PSK	16PSK	32PSK	16APSK	32APSI	K 64APS	K 128APS

PREDICTED

# **CNN** Transfer Learning



- CNN used for transfer learning test
  - + 2 convolution layers
  - 2 fully connected layers
- Dataset broken into two groups
  - Fully train on one set
  - Transfer learn the other set

Metric	Train A (40 Epochs)	Train B (40 Epochs)	Train A Transfer B (40 + 40 Epochs)	Train B Transfer A (40 + 40 Epochs)
Accuracy	91.15%	80.40%	78.32%	83.34%

#### Learn Set A

				o v c v	<b>DDCK</b>			10001	220 CK	464 064	224 064		( 1 2 0 4 0
12	28APSK-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.89
6	64APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.96	0.04
3	32APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.94	0.01	0.05
2	16APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
	32PSK-	0.00	0.00	0.00	0.00	0.00	0.04	0.09	0.87	0.00	0.00	0.00	0.00
TR	16PSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.07	0.58	0.36	0.00	0.00	0.00	0.00
Ы	8PSK-	0.00	0.00	0.00	0.00	0.00	0.75	0.07	0.19	0.00	0.00	0.00	0.00
	QPSK-	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	BPSK-	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	8ASK-	0.00	0.02	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	4ASK-	0.00	0.99	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	оок	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

OOK 4ASK 8ASK BPSK QPSK 8PSK 16PSK 32PSK 16APSK 32APSK 64APSK 128APSK

#### PREDICTED

#### Learn Set B, Transfer Learn Set A

		оок	4ASK	8ASK	BPSK	QPSK	8PSK	16PSK	32PSK	16APSK	32APSk	64APS	K 128APS
12	28APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.18	0.81
(	64APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.05
3	32APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
:	16APSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
	32PSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.14	0.16	0.20	0.49	0.00	0.00	0.00	0.00
TRUE	16PSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.11	0.15	0.63	0.11	0.00	0.00	0.00	0.00
Ы	8PSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.12	0.58	0.18	0.12	0.00	0.00	0.00	0.00
	QPSK <sup>-</sup>	0.00	0.00	0.00	0.00	0.85	0.04	0.07	0.04	0.00	0.00	0.00	0.00
	BPSK -	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	8ASK-	0.00	0.23	0.77	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	4ASK-	0.00	0.92	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	OOK	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

PREDICTED

## **CNN** Optimization



- Deeper CNN
  - More convolution layers
  - More fully connected layers
  - Fewer parameters
  - Experiment is learning all 24 classes and testing using unique untrained data samples

Convolution Layers	Filter Size	FC Layers	Epochs	Parameters	Training Accuracy (%)	Testing Accuracy (%)
$2 \rightarrow 16 \rightarrow 16 \rightarrow 16$	$1  ext{ by } 5$	2048→250→24	40	$521,\!042$	82.0	74.8
2710710710	1 by 3	20407200724	40	519,954	78.9	71.7
$2 \rightarrow 16 \rightarrow 16 \rightarrow 16 \rightarrow 16 \rightarrow 16$	1 by 3	$512 \rightarrow 250 \rightarrow 24$	40	137,522	77.4	75.9
2710710710710710	1 by 5	$512 \rightarrow 128 \rightarrow 24$	40	72,008	71.4	70.28
$2 \rightarrow 16 \rightarrow 16 \rightarrow 16 \rightarrow 16 \rightarrow 16 \rightarrow 16 \rightarrow 16$	1 hr 9	$256 \rightarrow 128 \rightarrow 24$	40	40,024	76.7	75.6
2710710710710710710	1 by 3	2007120724	80	40,024	82.7	81.4
$2 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12$	1 hr 9	192 <b>→</b> 128 <b>→</b> 24	40	30,104	66.3	65.2
	1 by 3	1927120724	80	30,104	75.7	74.8
$2 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12$	1 by 3	$192 \rightarrow 64 \rightarrow 24$	80	16,216	77.8	76.5
	1 by 5	192704724	160	16,216	78.2	77.4
$2 \rightarrow 8 \rightarrow 8 \rightarrow 8 \rightarrow 8 \rightarrow 8 \rightarrow 8$	1 hr 9	128→64→24	80	10,872	68.3	67.4
2707070707070	1 by 3	120704724	160	10,872	72.5	71.7
2 <b>→</b> 8 <b>→</b> 8 <b>→</b> 8 <b>→</b> 8 <b>→</b> 8	1 by 3	128 <b>→</b> 24	160	4,152	54.6	54.06

## **CNN** Optimization



- Range of CNN designs were evaluated to find tradeoff between number of parameters and accuracy
  - Two bold networks show strong accuracy vs. throughput results

Convolution Layers	Filter Size	FC Layers	Epochs	Parameters	Training Accuracy (%)	Testing Accuracy (%)
$2 \rightarrow 16 \rightarrow 16 \rightarrow 16$	1 by 5	2048→250→24	40	521,042	82.0	74.8
2710710710	1 by 3	20407200724	40	519,954	78.9	71.7
$2 \rightarrow 16 \rightarrow 16 \rightarrow 16 \rightarrow 16 \rightarrow 16$	1 by 3	$512 \rightarrow 250 \rightarrow 24$	40	137,522	77.4	75.9
2710710710710710	1 DY 3	$512 \rightarrow 128 \rightarrow 24$	40	72,008	71.4	70.28
$2 \rightarrow 16 \rightarrow 16 \rightarrow 16 \rightarrow 16 \rightarrow 16 \rightarrow 16 \rightarrow 16$	1 h 9	$256 \rightarrow 128 \rightarrow 24$	40	40,024	76.7	75.6
2710710710710710710	1 by 3	2007120724	80	40,024	82.7	81.4
$2 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12$	1 hr 9	192→128→24	40	30,104	66.3	65.2
	1 by 3	1927120724	80	30,104	75.7	74.8
$2 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12 \rightarrow 12$	1 h 9	192→64→24	80	16,216	77.8	76.5
2712712712712712712	1 by 3	192704724	160	16,216	78.2	77.4
0. \ 0. \ 0. \ 0. \ 0. \ 0. \ 0. \ 0.	1 h 9	199-204-294	80	10,872	68.3	67.4
$2 \rightarrow 8 \rightarrow 8 \rightarrow 8 \rightarrow 8 \rightarrow 8 \rightarrow 8$	1 by 3	$128 \rightarrow 64 \rightarrow 24$	160	10,872	72.5	71.7
$2 \rightarrow 8 \rightarrow 8 \rightarrow 8 \rightarrow 8 \rightarrow 8 \rightarrow 8$	1 by 3	128 <b>→</b> 24	160	4,152	54.6	54.06

# **CNN** Optimization



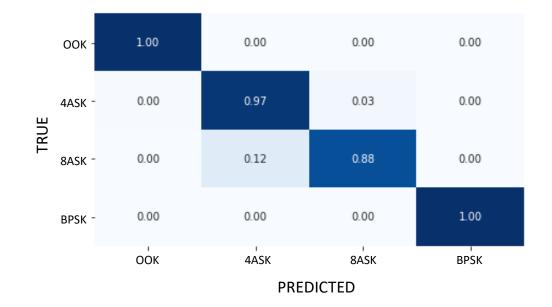
• Deeper CNN	Layer (type)	Output Shape	Param #
<ul> <li>6 convolution layers</li> </ul>	Conv2d-1	[-1, 12, 1024, 1]	 72
U U	ReLU-2	[-1, 12, 1024, 1]	0
<ul> <li>2 fully connected layers</li> </ul>	MaxPool2d-3	[-1, 12, 512, 1]	0
	Conv2d-4	[-1, 12, 512, 1]	432
<ul> <li>More layers and fewer parameters</li> </ul>	ReLU-5	[-1, 12, 512, 1]	0
<i>v</i> 1	MaxPool2d-6	[-1, 12, 256, 1]	0
	Conv2d-7	[-1, 12, 256, 1]	432
	ReLU-8 MaxPool2d-9	[-1, 12, 256, 1] [-1, 12, 128, 1]	0 0
	Conv2d-10	$\begin{bmatrix} -1, 12, 128, 1 \end{bmatrix}$ $\begin{bmatrix} -1, 12, 128, 1 \end{bmatrix}$	432
	ReLU-11	[-1, 12, 128, 1]	432
	MaxPool2d-12	[-1, 12, 64, 1]	0
	Conv2d-13	[-1, 12, 64, 1]	432
	ReLU-14	[-1, 12, 64, 1]	0
	MaxPool2d-15	[-1, 12, 32, 1]	0
	Conv2d-16	[-1, 12, 32, 1]	432
	ReLU-17	[-1, 12, 32, 1]	0
	MaxPool2d-18	[-1, 12, 16, 1]	0
	Dropout-19	[-1, 192]	0
	Linear-20 Linear-21	[-1, 64] [-1, 12]	12,288 768
	======================================		
	Input size (MB): 0.01 Forward/backward pass siz Params size (MB): 0.06 Estimated Total Size (MB)		

# Optimized CNN Transfer Learning



Transfer Learning to Add Class

• Step 1: Train CNN to learn 4 modulation types



Layer (type)	Output Shape	Param #
 Conv2d-1	[-1, 12, 1024, 1]	72
ReLU-2	[-1, 12, 1024, 1]	0
MaxPool2d-3	[-1, 12, 512, 1]	0
Conv2d-4	[-1, 12, 512, 1]	432
ReLU-5	[-1, 12, 512, 1]	0
MaxPool2d-6	[-1, 12, 256, 1]	0
Conv2d-7	[-1, 12, 256, 1]	432
ReLU-8	[-1, 12, 256, 1]	0
MaxPool2d-9	[-1, 12, 128, 1]	0
Conv2d-10	[-1, 12, 128, 1]	432
ReLU-11	[-1, 12, 128, 1]	0
MaxPool2d-12	[-1, 12, 64, 1]	0
Conv2d-13	[-1, 12, 64, 1]	432
ReLU-14	[-1, 12, 64, 1]	0
MaxPool2d-15	[-1, 12, 32, 1]	0
Conv2d-16	[-1, 12, 32, 1]	432
ReLU-17	[-1, 12, 32, 1]	0
MaxPool2d-18	[-1, 12, 16, 1]	0
Dropout-19	[-1, 192]	0
Linear-20	[-1, 64]	12,288
Linear-21	[-1, 12]	768
Former Former Former Former Former Former Former Former Former Former Former Former Former Forme Former Former Forme		
Input size (MB): 0.01 Forward/backward pass size Params size (MB): 0.06 Estimated Total Size (MB):		

# Optimized CNN Transfer Learning



#### 1.00 0.00 0.00 0.00 0.00 OOK -0.03 0.97 0.00 0.00 0.00 4ASK -BASK -0.00 0.12 0.88 0.00 0.00 0.00 0.00 1.00 0.00 0.00 BPSK -

0.00

8ASK

PREDICTED

1.00

BPSK

0.00

QPSK

• Transfer Learning to Add Class

to a new unlearned class

• Step 2: Test with 4 learned modulations in addition

Layer (type)	Output Shape	
Conv2d-1	[-1, 12, 1024, 1]	72
ReLU-2	[-1, 12, 1024, 1]	0
MaxPool2d-3	[-1, 12, 512, 1]	0
Conv2d-4	[-1, 12, 512, 1]	432
ReLU-5 MaxPool2d-6	[-1, 12, 512, 1]	0
	[-1, 12, 256, 1]	0
Conv2d-7	[-1, 12, 256, 1]	432
ReLU-8 MaxPool2d-9	[-1, 12, 256, 1]	0 0
Conv2d-10	[-1, 12, 128, 1]	
ReLU-11	[-1, 12, 128, 1]	432 Ø
MaxPool2d-12	[-1, 12, 128, 1]	0
Conv2d-13	[-1, 12, 64, 1]	
ReLU-14	[-1, 12, 64, 1]	432 Ø
MaxPool2d-15	[-1, 12, 64, 1]	0
Conv2d-16	[-1, 12, 32, 1] [-1, 12, 32, 1]	432
ReLU-17	[-1, 12, 32, 1] [-1, 12, 32, 1]	432
MaxPool2d-18	[-1, 12, 52, 1] [-1, 12, 16, 1]	0
Dropout-19	[-1, 12, 10, 1]	0
Linear-20	[-1, 64]	12,288
Linear-21	[-1, 04]	768
Total params: 15,288 Trainable params: 15,288 Non-trainable params: 0		
Input size (MB): 0.01 Forward/backward pass size Params size (MB): 0.06 Estimated Total Size (MB):		

QPSK -

0.00

OOK

0.00

4ASK

# Optimized CNN Transfer Learning



<ul> <li>Transfer Learning to Add Class</li> </ul>						Layer (type)	Output Shape	Param #
<ul> <li>Step 3: Use transfer learning to train only the fully connected layers</li> </ul>						e fully Conv2d-1 ReLU-2 MaxPool2d-3	[-1, 12, 1024, 1] [-1, 12, 1024, 1] [-1, 12, 512, 1]	72 0 0
• Step 4: Test if the network is able to learn all 5 modulations						MaxPool2d-6 Conv2d-7 ReLU-8	[-1, 12, 512, 1] [-1, 12, 512, 1] [-1, 12, 256, 1] [-1, 12, 256, 1] [-1, 12, 256, 1]	432 0 432 0
						MaxPool2d-9 Conv2d-10 ReLU-11 MaxPool2d-12 Conv2d-13 ReLU-14	$\left[ \begin{array}{cccc} -1, \ 12, \ 128, \ 1  ight] \\ \left[ \begin{array}{cccc} -1, \ 12, \ 128, \ 1  ight] \\ \left[ \begin{array}{cccc} -1, \ 12, \ 128, \ 1  ight] \\ \left[ \begin{array}{cccc} -1, \ 12, \ 64, \ 1  ight] \\ \left[ \begin{array}{ccccc} -1, \ 12, \ 64, \ 1  ight] \\ \left[ \begin{array}{ccccccc} -1, \ 12, \ 64, \ 1  ight] \end{array}  ight]$	0 432 0 432 0
оок -	1.00	0.00	0.00	0.00	0.00	MaxPool2d-15 Conv2d-16 ReLU-17 MaxPool2d-18	$\begin{bmatrix} -1, 12, 32, 1 \end{bmatrix}$ $\begin{bmatrix} -1, 12, 16, 1 \end{bmatrix}$	0 432 0
4ASK -	0.00	0.95	0.05	0.00	0.00	Training Here	[-1, 1 <u>9</u> 2] [-1, 64] [-1, 12]	0 12,288 768
TRUE BASK -	0.00	0.07	0.93	0.00	0.00	Total params: 15,288 Trainable params: 15,288 Non-trainable params: 0		
BPSK -	0.00	0.00	0.00	0.97	0.03	Input size (MB): 0.01 Forward/backward pass size	e (MB): 0.46	
QPSK -	0.00	0.00	0.00	0.02	0.98	Params size (MB): 0.06 Estimated Total Size (MB):	: 0.53	
	ООК	4ASK	8ASK PREDICTED	BPSK	QPSK			

## **Conclusion and Future Work**



- Summary
  - CNN for AMC
  - Low power deployment of signal modulation classification
    - Through CNN optimization
  - Transfer learning makes system adaptable
    - $\cdot$  Also reduces complexity of training if deployed on custom hardware
- Future Work
  - Hardware Survey
    - Best options for low SWaP deployment
    - Algorithm refinement
      - Optimize throughput and classification accuracy
    - Dataset improvement
      - $\cdot$  Generate custom dataset using SDR for real world examination