EITN90 Radar and Remote Sensing
Lecture 10: Machine learning approaches to radar signal analysis

Daniel Sjöberg

Department of Electrical and Information Technology

Spring 2019

Learning outcomes of this lecture

In this lecture we will

- Give an introduction to machine learning and simple implementations.
- See how supervised learning has been used in radar target classification problems.
- Consider a case study and live demonstration for gesture recognition.

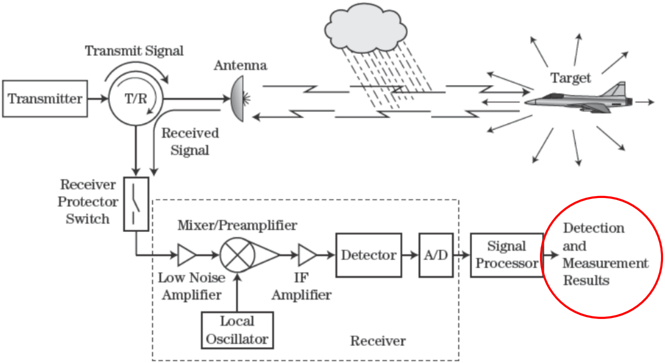


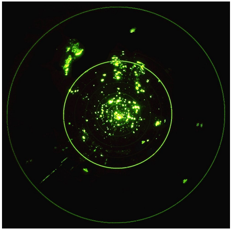
FIGURE 1-1 ■ Major elements of the radar transmission/ reception process.

Outline

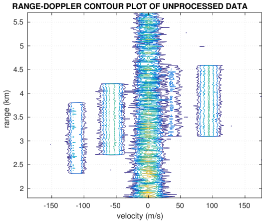
1. **An overview of machine learning problems**
2. **ML in remote sensing**
3. **ML in target recognition**
4. **ML in micro-Doppler analysis**
5. **Case study: gesture recognition with an FMCW radar**
6. **Conclusions**

Outline

Some radar data representations



Classical PPI radar screen



Doppler-range plot

FIGURE 1-32 ■ 1 m resolution SAR image of the Washington, D.C., mall area. (Courtesy of Sandia National Laboratories. With permission.)

SAR-image

Interpretation of the data may require a skilled operator.

Machine Learning

Machine learning

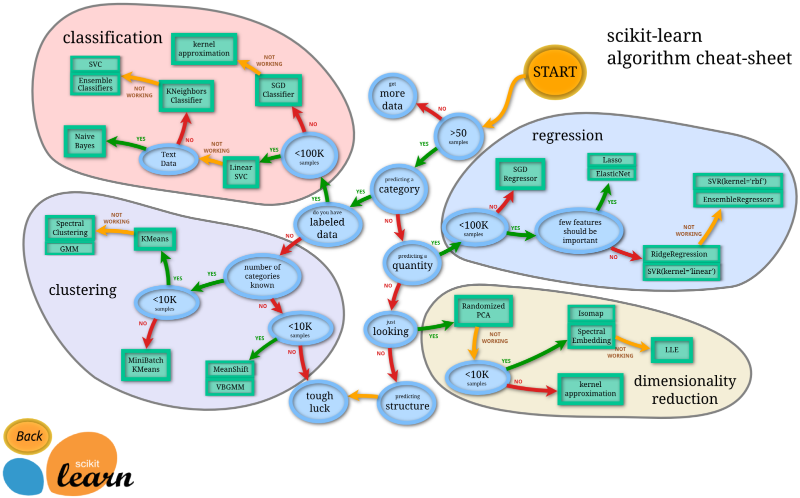
A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E. (*Tom Mitchell, 1997*)

The ML approach focuses on the data rather than how the data is generated. Four main problems are in focus:

- Classification
- Regression
- Clustering
- Dimensionality reduction

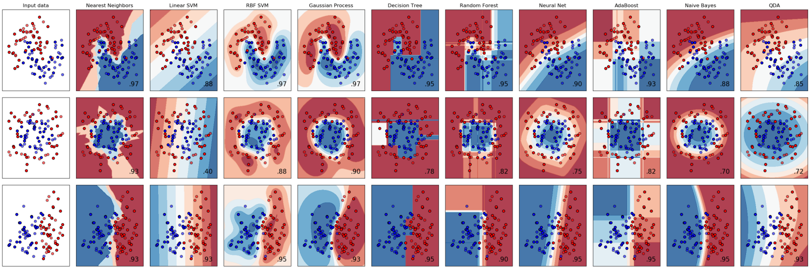
Solving different ML-problems with scikit-learn

https://scikit-learn.org



Different classifiers

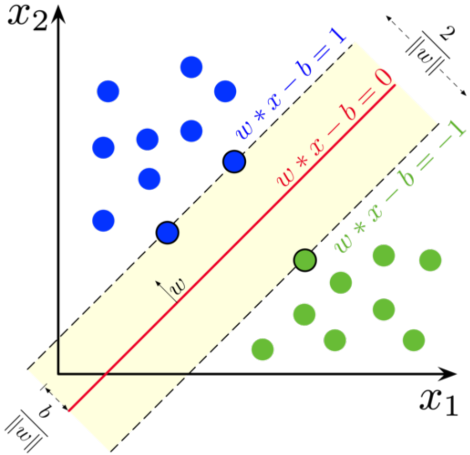
Classification example: training data in 2D, blue or red dots distributed in three different ways.



Different ML approaches provide different predictions (the red and blue colored backgrounds).

In the following, we focus on support vector machines (SVM) for simplicity of presentation, but there are several alternatives.

Linear Support Vector Machine (SVM)



A linear support vector machine finds the widest linear separation of labeled training data (a hyperplane in higher dimensions).

SVM mathematics

Training data: $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $x_i \in \mathbb{R}^d$ is the observation and $y_i \in \{-1, 1\}$ is the classification. The parameters $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are determined by a convex optimization problem:

Hard margin (training data is separable)

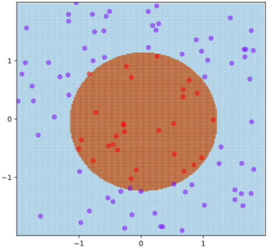
$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} w^{\mathrm{T}} w \\ \text{subject to} \quad & y_i (w^{\mathrm{T}} x_i - b) \geq 1 \end{aligned}$$

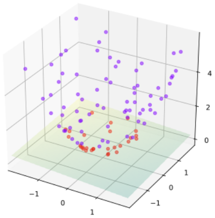
Soft margin (training data not separable)

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} w^{\mathrm{T}} w + C \sum_{i=1}^{n} \zeta_i \\ \text{subject to} \quad & y_i (w^{\mathrm{T}} x_i - b) \geq 1 - \zeta_i \\ & \zeta_i \geq 0, \quad i = 1, 2, \ldots, n \end{aligned}$$

Classification of new data is made using the decision function: $f(x) = \mathrm{sign}(w^{\mathrm{T}} x - b)$. SVMs are considered robust and able to learn from small sets of data.

Kernel trick





When data is not separable by a hyperplane, replace the scalar product $x^{\mathrm{T}}x'$ with a product in a higher dimensional space, like

$$k(x,x') = \varphi(x)^{\mathrm{T}}\varphi(x'), \quad \text{where} \quad \varphi(x) = (a, b, a^2 + b^2)$$

This can allow for finding a separating hyperplane in the higher dimension. A typical kernel is the radial basis function kernel,

$$k(x,x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

Preprocessing

In order to make the data more amenable to ML techniques, some preprocessing is usually required:

- Mean removal
- Scaling
- Normalization
- Binarization
- ...

Unified framework for automatic target recognition (ATR)

- Identify the target set: what will be observed?
- Select the feature set: what is important?
- Observe the feature set: measure accurately
- Test the feature set: train and use a classifier

Unified framework for automatic target recognition (ATR)

- Identify the target set: what will be observed?
  - Broad classes of targets (humans, rabbits, cars, bikes...)
  - Variations within classes (adults, children, minivans, trucks...)
- Select the feature set: what is important?
- Observe the feature set: measure accurately
- Test the feature set: train and use a classifier

Unified framework for automatic target recognition (ATR)

- Identify the target set: what will be observed?
  - Broad classes of targets (humans, rabbits, cars, bikes. . .)
  - Variations within classes (adults, children, minivans, trucks. . .)
- Select the feature set: what is important?
  - Maximize the similarity of objects in the same class while maximizing the dissimilarity of objects in different classes.
- Observe the feature set: measure accurately
- Test the feature set: train and use a classifier

Unified framework for automatic target recognition (ATR)

- Identify the target set: what will be observed?
  - Broad classes of targets (humans, rabbits, cars, bikes...)
  - Variations within classes (adults, children, minivans, trucks...)
- Select the feature set: what is important?
  - Maximize the similarity of objects in the same class while maximizing the dissimilarity of objects in different classes.
- Observe the feature set: measure accurately
  - Processing to increase SNR: averaging, background removal, pulse compression...
- Test the feature set: train and use a classifier

Unified framework for automatic target recognition (ATR)

- Identify the target set: what will be observed?
  - Broad classes of targets (humans, rabbits, cars, bikes...)
  - Variations within classes (adults, children, minivans, trucks...)
- Select the feature set: what is important?
  - Maximize the similarity of objects in the same class while maximizing the dissimilarity of objects in different classes.
- Observe the feature set: measure accurately
  - Processing to increase SNR: averaging, background removal, pulse compression...
- Test the feature set: train and use a classifier
  - Supervised learning: each training observation is given a ground truth from the operator.
  - Unsupervised learning: the training data are clustered into classes, which are then used as ground truth.
  - Reinforced learning: learn from new data through feedback.

Outline

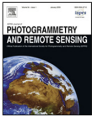
ML in remote sensing

ISPRS Journal of Photogrammetry and Remote Sensing 66 (2011) 247–259

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs



Review article

Support vector machines in remote sensing: A review

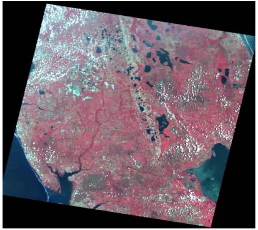
Giorgos Mountrakis *, Jungho Im, Caesar Ogole

Department of Environmental Resources Engineering, SUNY College of Environmental Science and Forestry, 1 Forestry Dr, Syracuse, NY 13210, USA

Review paper summarizing many uses of ML in remote sensing. Results from a few papers are shown in the following, using multispectral data in each pixel to classify the terrain.

MULTISPECTRAL CLASSIFICATION OF LANDSAT IMAGES: A COMPARISON OF SUPPORT VECTOR MACHINE AND NEURAL NETWORK CLASSIFIERS

Nivedita Candade, Research Assistant
Dr. Barnali Dixon, Assistant Professor
University of South Florida, St. Petersburg
140 Seventh Ave South, St. Petersburg, FL 33701.
Phone: 727 553 4863
ncandade@mail.usf.edu
bdixon@stpt.usf.edu



Original image characteristics
Projection: UTM Zone 17N
Spheroid: WGS 84
Datum: WGS 84
Unit: meters
Pixel size: 30m
Systematic Correction
GeoTIFF format

Figure 3. Standard False-Color composite.Combination of bands 2,3 and 4.

Table 2. Description of Landsat TM bands

Band	Color	Wavelength (um)	Applications
Band 1	Blue	0.45-0.52	Separation of soil and vegetation
Band 2	Green	0.52-0.60	Reflection of vegetation
Band 3	Red	0.63-0.69	Chlorophyll absorption
Band 4	Near IR	0.76-0.90	Delineation of water boundaries
Band 5	Mid-IR	1.55-1.75	Vegetative moisture
Band 6	Far IR	10.4-12.5	Hydrothermal mapping
Band 7	Thermal	2.08-2.35	Plant heat stress

Classification results

Table 4. Neural Network, SBP

	citrus	pasture	sod	timber	urban	water	wetland	Total	User's Accuracy (%)
citrus	743	227	0	0	0	0	264	1234	60
pasture	0	100	24	2	0	0	0	126	79
sod	0	0	68	0	0	0	0	68	100
timber	0	0	0	966	0	0	0	966	100
urban	0	0	0	0	15	0	0	15	100
water	0	0	0	0	0	892	0	892	100
wetland	0	1	0	0	0	0	1313	1314	100
Total	743	328	92	968	15	892	1577	4615	
Producer's Accuracy (%)	100	69	74	100	100	100	83		

Overall accuracy= 88.7%

Table 5. SVM- Polynomial kernel, degree 3, c=1000

	citrus	pasture	sod	timber	urban	water	wetland	Total	User's Accuracy (%)
citrus	1211	4	0	18	0	0	1	1234	98
pasture	0	85	36	5	0	0	0	126	67
sod	0	0	68	0	0	0	0	68	100
timber	0	0	0	966	0	0	0	966	100
urban	0	0	0	0	15	0	0	15	100
water	0	0	0	0	0	892	0	892	100
wetland	2	0	0	3	0	0	1309	1314	100
Total	1213	89	104	992	15	892	1310	4615	
Producer's Accuracy (%)	100	96	65	97	100	100	100		

Overall accuracy= 98.48%

Table 6. RBF kernel, nu=0.01

	citrus	pasture	sod	timber	urban	water	wetland	Total	User's Accuracy (%)
citrus	1049	0	0	22	0	0	163	1234	85
pasture	0	116	1	9	0	0	0	126	92
sod	0	0	68	0	0	0	0	68	100
timber	0	0	0	966	0	0	0	966	100
urban	0	0	0	0	15	0	0	15	100
water	0	0	0	0	0	892	0	892	100
wetland	0	0	0	0	0	0	1314	1314	100
Total	1049	116	69	997	15	892	1477	4615	
Producer's Accuracy (%)	100	100	99	97	100	100	89		

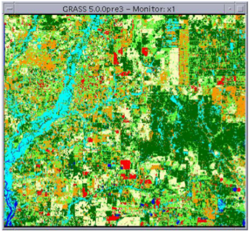
Overall accuracy= 95.77%

Table 7. Linear kernel, c=1000

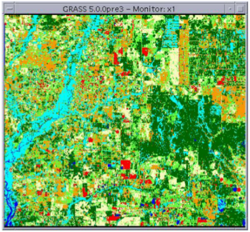
	citrus	pasture	Sod	timber	urban	water	wetland	Total	User's Accuracy (%)
citrus	1054	0	0	28	0	0	152	1234	85
pasture	0	112	6	8	0	0	0	126	89
sod	0	0	68	0	0	0	0	68	100
timber	0	0	0	966	0	0	0	966	100
urban	0	0	0	0	15	0	0	15	100
water	0	0	0	0	0	892	0	892	100
wetland	0	0	0	0	0	0	1314	1314	100
Total	1054	112	74	1002	15	892	1466	4615	
Producer's Accuracy (%)	100	100	92	96	100	100	90		

Overall accuracy=95.79%

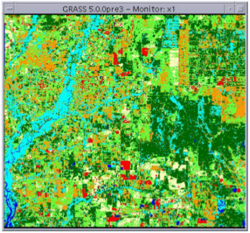
The training data consisted of 350 samples, 50 pixels per class.
The SVMs learn with few number of samples.



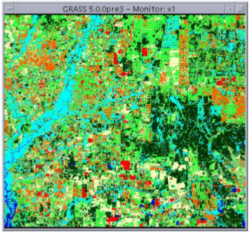
a. Polynomial kernel



b. RBF kernel



c. Linear kernel



d. Neural network

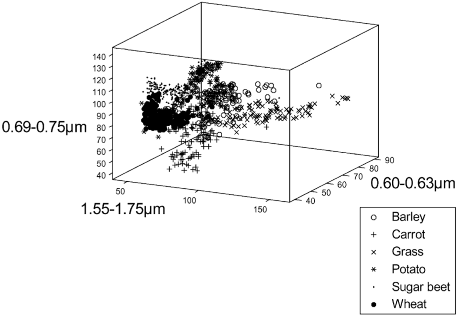
Figure 5. Classified maps using ANN and SVM classifiers

Table 8. Area coverage of the models in %

	citrus	pasture	sod	timber	urban	water	wetland
NN-classifier	13	30.5	13.67	22.81	3.11	0.72	16.12
SVM-poly	15	22	14.83	32.83	3.64	0.74	11
SVM-RBF	15.18	21.59	13.78	30.7	3.5	0.93	14.25
SVM-linear	15.57	27.94	7.36	30.59	3.45	0.93	14.16

A Relative Evaluation of Multiclass Image Classification by Support Vector Machines

Giles M. Foody, *Member, IEEE,* and Ajay Mathur



Data was aquired in 11 spectral bands, only 3 used for processing.

DA	Predicted Classes ↓						
Actual →	Sugarbeet	Wheat	Barley	Carrot	Potato	Grass	Total
Sugarbeet	87	3	0	0	7	0	97
Wheat	3	90	2	1	0	0	96
Barley	0	6	45	0	0	0	51
Carrot	0	1	0	29	3	0	33
Potato	0	2	0	0	23	1	26
Grass	0	0	0	1	2	14	17
Total	90	102	47	31	35	15	320

Overall accuracy = 90.00%

DT	Predicted Classes ↓						
Actual →	Sugarbeet	Wheat	Barley	Carrot	Potato	Grass	Total
Sugarbeet	89	4	1	0	2	1	97
Wheat	8	79	6	1	0	2	96
Barley	3	0	48	0	0	0	51
Carrot	0	0	0	33	0	0	33
Potato	0	2	0	0	23	1	26
Grass	0	0	0	0	0	17	17
Total	100	85	55	34	25	21	320

Overall accuracy = 90.31%

NN	Predicted Classes ↓						
Actual →	Sugarbeet	Wheat	Barley	Carrot	Potato	Grass	Total
Sugarbeet	90	3	1	0	3	0	97
Wheat	3	84	7	1	0	1	96
Barley	0	2	49	0	0	0	51
Carrot	0	2	0	31	0	0	33
Potato	0	2	0	0	23	1	26
Grass	0	0	0	0	0	17	17
Total	93	93	57	32	26	19	320

Overall accuracy = 91.88%

SVM	Predicted Classes ↓						
Actual →	Sugarbeet	Wheat	Barley	Carrot	Potato	Grass	Total
Sugarbeet	89	6	0	0	1	1	97
Wheat	2	88	5	1	0	0	96
Barley	1	1	49	0	0	0	51
Carrot	0	0	0	33	0	0	33
Potato	0	2	0	0	24	0	26
Grass	0	0	0	0	0	17	17
Total	92	97	54	34	25	18	320

Overall accuracy = 93.75%

Fig. 3. Error matrices for the classifications derived from the DA, DT, NN, and SVM classifications trained with the largest training set (containing 100 cases of each class). For clarity, the main diagonal that indicates correct allocations has been highlighted.

Outline

1. An overview of machine learning problems
2. ML in remote sensing
3. **ML in target recognition**
4. ML in micro-Doppler analysis
5. Case study: gesture recognition with an FMCW radar
6. Conclusions

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received August 30, 2016, accepted September 16, 2016, date of publication September 21, 2016, date of current version October 15, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2611492

Automatic Target Recognition in Synthetic Aperture Radar Imagery: A State-of-the-Art Review

KHALID EL-DARYMLI[1], (Member, IEEE), ERIC W. GILL[2], (Senior Member, IEEE), PETER McGUIRE[2,3], DESMOND POWER[3], (Member, IEEE), AND CECILIA MOLONEY[2] (Member, IEEE)

[1]Northern Radar Inc., St. John's, NL A1B 3E4, Canada
[2]Memorial University of Newfoundland, St. John's, NL A1B 3X5, Canada
[3]C-CORE, St. John's, NL A1B 3X5, Canada

Thorough review of target recognition techniques, not only ML.
Too much to go through here in detail.

Support Vector Machines for SAR Automatic Target Recognition

QUN ZHAO, Member, IEEE
JOSE C. PRINCIPE, Fellow, IEEE
University of Florida

TABLE III
Misclassification Rates and Confuser Rejection Rates (%)

	BMP2	BTR70	T72	Average	Confuser Rejection
Template	11.58	2.04	10.14	9.60	53.47
Perceptron	9.71	0.00	5.84	6.67	27.19
OH	8.69	0.51	3.78	5.42	38.50
SVM	4.94	0.00	7.04	5.13	68.80

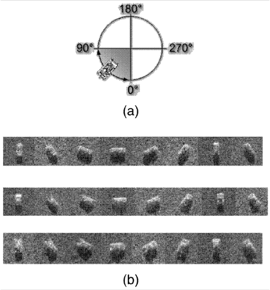


Fig. 3. (a) Illustration of pose. (b) SAR images of target T72, BTR70, BMP2 taken at different aspect angles.

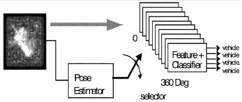


Fig. 4. Classifier topology is depicted. First a pose estimator is applied to image and determines approximate pose of target, then classifier is chosen according to result of pose estimation.

TABLE IV
Confusion Matrices (Counts) of Classifiers and Confuser Rejection when $P_d = 0.9$

	BMP2	BTR70	T72	Rejection
BMP2	483	59	9	36
BTR70	4	188	en 0	4
T72	43	16	427	96
2S1	111	83	38	42
D7	16	4	3	251

(a) Template matching

	BMP2	BTR70	T72	Rejection
BMP2	436	16	41	83
BTR70	0	194	0	2
T72	18	16	502	51
2S1	9	105	100	60
D7	29	68	88	89

(b) perceptron

	BMP2	BTR70	T72	Rejection
BMP2	443	9	42	88
BTR70	0	193	1	2
T72	16	6	519	46
2S1	9	50	117	98
D7	53	9	99	113

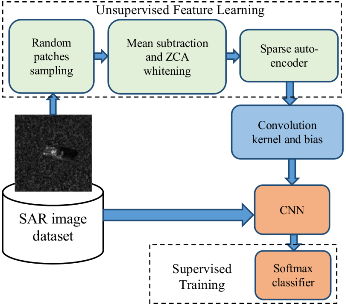
(c) optimal hyperplane

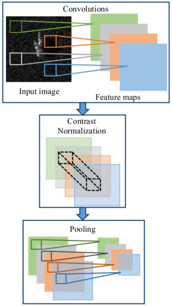
	BMP2	BTR70	T72	Rejection
BMP2	511	14	15	47
BTR70	0	195	0	1
T72	31	10	453	88
2S1	57	24	10	183
D7	53	0	27	145

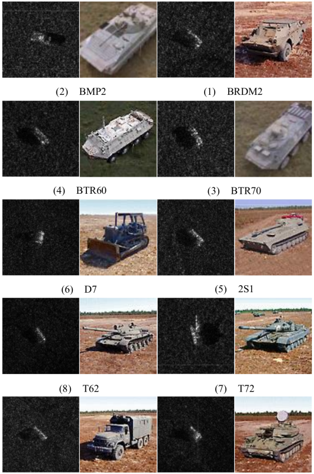
(d) support vector machine

SAR Target Recognition Based on Deep Learning

Sizhe Chen, Haipeng Wang
Key Laboratory for Information Sciences of Electromagnetic Waves (MoE)
Fudan University, Shanghai, China
Email: hpwang@fudan.edu.cn







(2) BMP2 (1) BRDM2
(4) BTR60 (3) BTR70
(6) D7 (5) 2S1
(8) T62 (7) T72
(10) ZIL131 (9) ZSU234

Test set	Classification Results										
	BMP2	*BRDM2*	*BTR60*	*BTR70*	*D7*	*2S1*	*T62*	*T72*	*ZIL131*	*ZSU234*	*Classification accuracy (%)*
BMP2	157	9	2	9	0	4	0	4	6	4	80.5
BRDM2	9	220	6	18	0	3	1	2	15	0	80.2
BTR60	0	11	168	4	4	4	1	2	1	0	86.1
BTR70	3	4	3	181	0	4	0	0	1	0	92.3
D7	0	0	0	0	252	0	8	2	5	7	91.9
2S1	14	9	5	5	0	190	7	22	21	1	69.3
T62	2	1	5	0	4	7	242	3	7	2	88.6
T72	3	3	1	1	0	8	2	168	9	1	85.7
ZIL131	5	6	5	7	1	12	3	9	226	0	82.4
ZSU234	1	1	3	0	4	1	2	7	6	249	90.8
average classification rate: 84.7%											

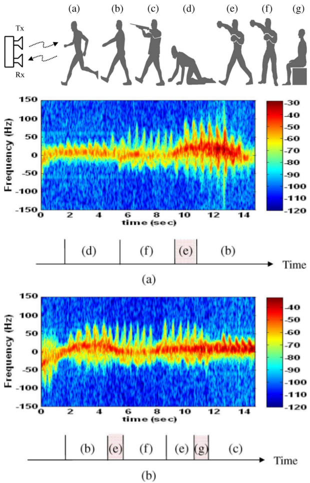
Outline

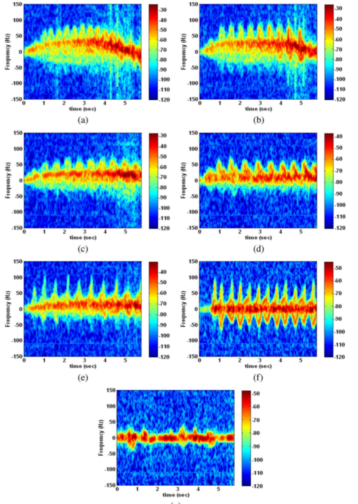
ML in micro-Doppler analysis

- Complex live targets like a waving hand, running horse, flying helicopter etc, present many different velocity components in a Doppler spectrum.
- Recording the Doppler spectrum as a function of time provides a 2D spectrogram $a(f_{\mathrm{d}}, t)$.
- The analysis of $a(f_{\mathrm{d}}, t)$ is called micro-Doppler analysis.
- ML techniques can be used to identify features in the spectrograms and do classification.

Human Activity Classification Based on Micro-Doppler Signatures Using a Support Vector Machine

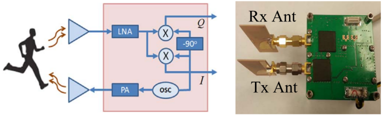
Youngwook Kim, *Member, IEEE*, and Hao Ling, *Fellow, IEEE*





Human Detection and Activity Classification Based on Micro-Doppler Signatures Using Deep Convolutional Neural Networks

Youngwook Kim, *Senior Member, IEEE*, and Taesup Moon, *Member, IEEE*



(a) (b)

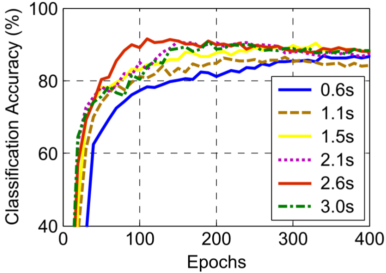


Fig. 4. Outdoor measurements. (a) Human. (b) Dog. (c) Horse. (d) Car.

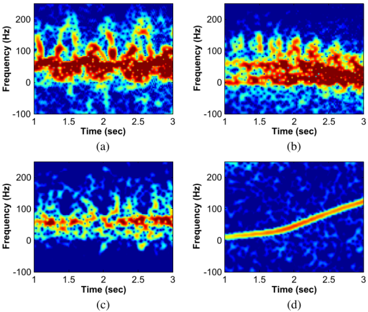


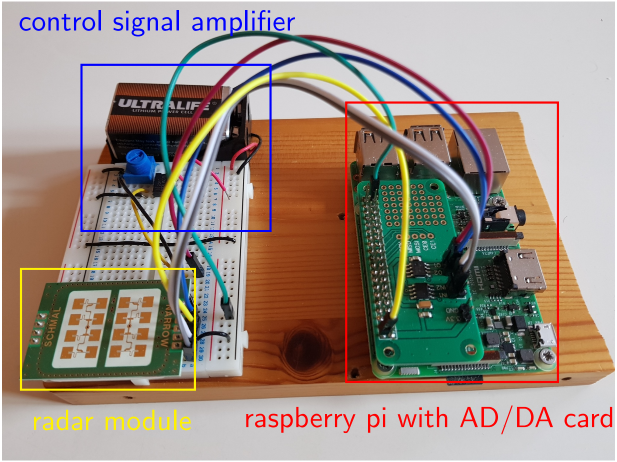
Fig. 5. Sample spectrograms. (a) Human. (b) Dog. (c) Horse. (d) Car.

Outline

Gesture recognition

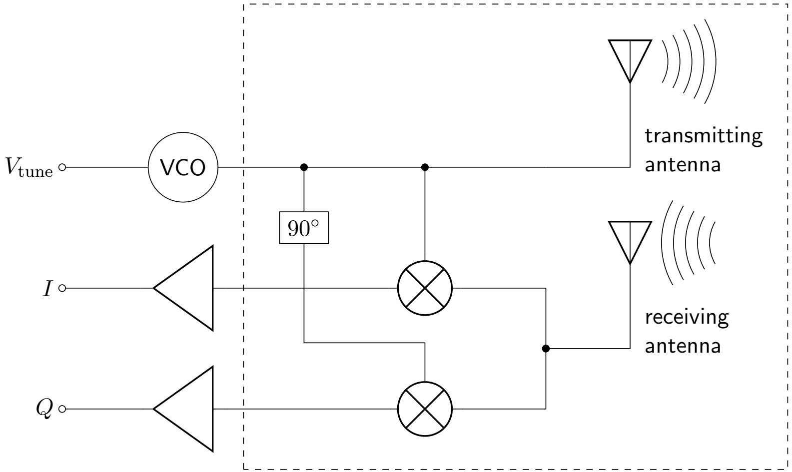
In 2015, Google released a 60 GHz radar platform (Soli) for gesture recognition. Here, we will see that similar functionality can be achieved using the 24 GHz radar system from the lab.

Demonstration



The same radar equipment as used in the lab last Friday has been augmented with a simple gesture recognition program.

Principal operation of the radar system



FMCW operation, stepped frequency.

Signal model

After down-conversion, we form the analytical signal from I and Q:

$$s(t) = I(t) + \mathrm{j}Q(t) = \underbrace{\left(A\mathrm{e}^{\mathrm{j}(-4\pi(f_0+f_\mathrm{d})R/c+\phi)}\right)}_{=A'}\mathrm{e}^{\mathrm{j}(2\pi f_\mathrm{d} t - 4\pi \Delta f(t) R/c)}$$

Sample the up-chirp and down-chirp signal at Δt and take the Fourier transform:

$$s_n = \begin{cases} A'\mathrm{e}^{\mathrm{j}n2\pi(f_\mathrm{d}\Delta t - 2BR/(Nc))} & \text{up-chirp} \\ A'\mathrm{e}^{\mathrm{j}n2\pi(f_\mathrm{d}\Delta t + 2BR/(Nc))} & \text{down-chirp} \end{cases} \quad \Longrightarrow \quad \hat{s}_n = \mathrm{FFT}[s_n]$$

Combine M signals into one data frame:

$$\text{one frame} = \left\{ \begin{pmatrix} \hat{s}_0^{(0)} & \hat{s}_0^{(0)} & \cdots & \hat{s}_N^{(0)} \\ \hat{s}_0^{(1)} & \hat{s}_0^{(1)} & \cdots & \hat{s}_N^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{s}_0^{(M)} & \hat{s}_0^{(M)} & \cdots & \hat{s}_N^{(M)} \end{pmatrix}_{\text{up-chirp}}, \begin{pmatrix} \hat{s}_0^{(0)} & \hat{s}_0^{(0)} & \cdots & \hat{s}_N^{(0)} \\ \hat{s}_0^{(1)} & \hat{s}_0^{(1)} & \cdots & \hat{s}_N^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{s}_0^{(M)} & \hat{s}_0^{(M)} & \cdots & \hat{s}_N^{(M)} \end{pmatrix}_{\text{down-chirp}} \right\}$$

Only the amplitudes of these $2M \times N$ complex numbers are used, and each frame is normalized with its peak value.

Choosing system parameters

- Carrier frequency $f_0 = 24\,\mathrm{GHz}$.
- Bandwidth $B = 425\,\mathrm{MHz}$.
- Sampling time $\Delta t = 0.34\,\mathrm{ms}$.
- Doppler shift $f_\mathrm{d} = \frac{2v}{c} f_0 = \frac{v}{1\,\mathrm{m/s}} \frac{2\cdot 24\cdot 10^9}{3\cdot 10^8}\,\mathrm{Hz} = \frac{v}{1\,\mathrm{m/s}} 160\,\mathrm{Hz}$.
- Sampled received signal $s_n = A' \mathrm{e}^{\mathrm{j}n2\pi(\mp \frac{2BR}{Nc} + f_\mathrm{d}\Delta t)}$ (up- and down-chirp).
- Phase shift relative to 2π at full scale $n = N$:

$$\begin{aligned} \text{Range:} \quad & N\frac{2BR}{Nc} = \frac{2\cdot 425\cdot 10^6 \cdot 0.5}{3\cdot 10^8} = 1.4 \\ \text{Doppler:} \quad & N f_\mathrm{d} \Delta t = N\cdot 160\cdot 0.34\cdot 10^{-3} = 0.054N \end{aligned}$$

  The choice $N = 32$ makes range and Doppler almost equally balanced with more than 2π phase change.
- Dwell time $T_\mathrm{d} = N\Delta t = 32\cdot 0.34\cdot 10^{-3}\,\mathrm{s} = 0.011\,\mathrm{s}$.
- Collecting $M = 16$ dwells in one frame gives aquisition time $T_\mathrm{f} = 2MT_\mathrm{d} = 2\cdot 16\cdot 0.011 = 0.35\,\mathrm{s}$, in the order of the time of a typical gesture.

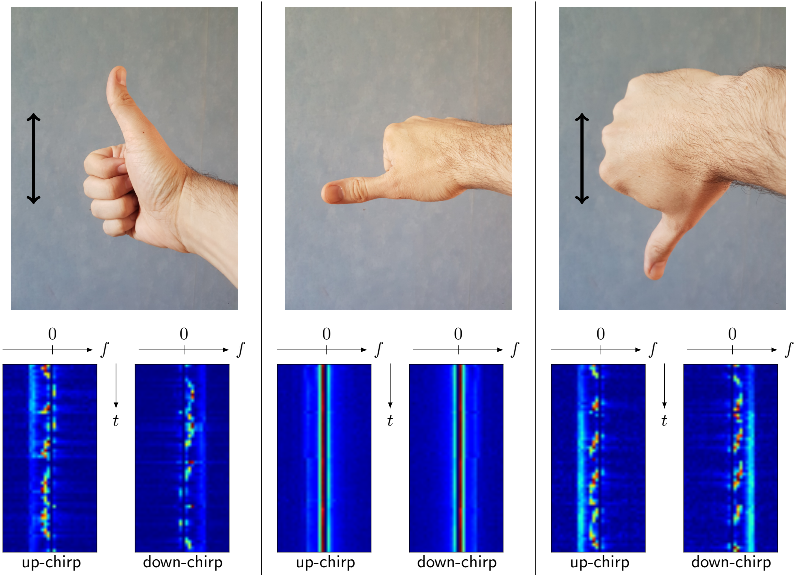
Data processing

Each raw data frame goes through the following processing:

- Background subtraction.
- Estimation and subtraction of a ramp signal.
- Fourier transform (FFT).
- Taking the amplitude and normalizing by the maximum amplitude in the frame.

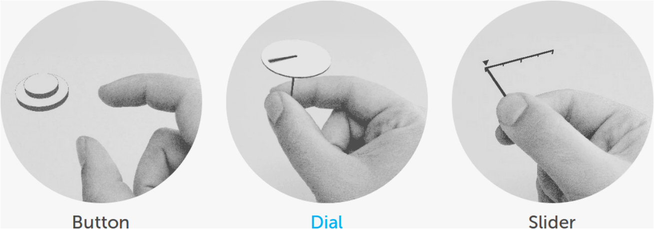
A number of frames are used for training the SVM, which is subsequently used to classify gestures performed.

Gestures


0
f
t
up-chirp
down-chirp
0
f
t
up-chirp
down-chirp
0
f
t
up-chirp
down-chirp

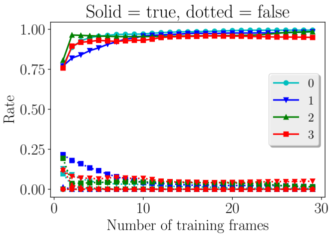
Additional gestures evaluated

- Static: a hand at different altitudes above the sensor (separated roughly by the resolution $c/(2B) = 35\,\mathrm{cm}$).
- Circle: clockwise or counter-clockwise rotation of index finger in non-radial motion (should be difficult to distinguish).
- Soli gestures:

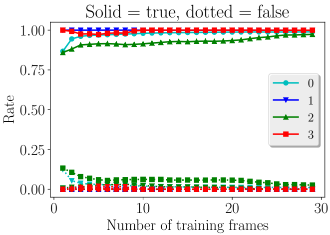


Each gesture was repeated until 100 frames had been recorded. For $1 \leq n \leq 29$, n frames were chosen for training and 70 frames for evaluation. The selection was random, and the outcome averaged over 100 selections.

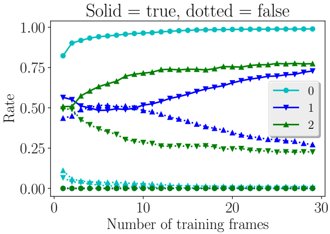
Results for different sets of gestures



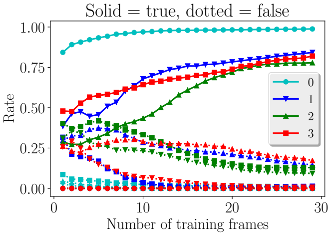
Thumb



Static



Circle



Soli

Discussion of the gesture recognition system

- The system can distinguish between movements with significantly different Doppler spectrograms.
- The system performs well for thumb up/mid/down, and static poses separated by the resolution.
- The system performs less well for non-radial motion, and micro-motions like the Soli gestures.
- With one single sensor, the gestures need to be adapted to the sensor; with several sensors, more relaxed requirements for the gestures are expected.

Outline

Conclusions

- Some ML approaches to radar signal analysis have been reviewed, with particular emphasis to supervised learning using SVMs.
- Terrain classification in SAR images has been performed based on multispectral data in each pixel.
- Target classification in SAR images may require an initial pose estimation. Open data is scarce.
- Gesture classification in Doppler spectrogram is a field of current research and implementation, expected to emerge in consumer products.