HARMAN
A SAMSUNG COMPANY

ARTIFICIAL INTELLIGENCE – COC AUDIO STRAUBING

DR. GERALD BAUER

2017-11-06

This presentation contains:

- An overview about research results of several AI systems developed since March 2017 at COC AUDIO Straubing.
- An outlook about further potential applications / investigations related to AI in HARMAN

Introduction

Target and Team

Name: *Dr. Gerald Bauer*
Current Position: *Product Owner in COC Lifestyle Audio*
HARMAN employee: *Since 3 years*
Professional AI Expertise: *More than 10 years*

Thanks to Mr. Gerhard Pfaffinger (Head of COC Audio) I was able to implement and evaluate AI services in the field of audio and automotive in addition to my main task as Product Owner within the COC Audio Lifestyle Group.

The goal was to explore the state-of-the-art of deep learning and machine learning in audio applications and to provide a proof of concept for promising ideas / applications. A strong focus was put on the realization of these services in smart speakers / smart headphones and in automotive applications.

The following slides show the result of my work and the work of my students, which we have successfully completed since March 2017.

Introduction

AI Services – Realized since March 2017 (1)

Gender Recognition based on the speaker‘s voice

Detection of siren sounds in urban environments

Detection of baby scream in home environments

Detection of doorbell sounds in home environments

Environment type recognition based on a single image

Music genre recognition based on sound snipets

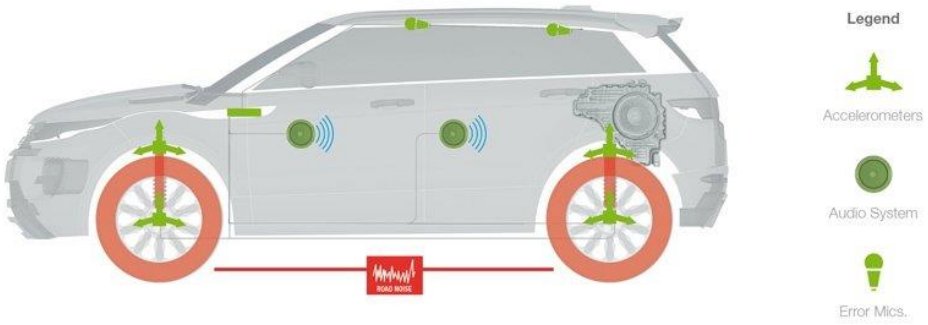
Speaker recognition based on large training data and unknown speaker clustering based on voice samples

*Intelligent noise reduction**

*Classification of music, speech and noise based on sound snipets**

* Student work

AI Services – Realized since March 2017 (II)



Automatic road surface recognition based on car mounted motion sensors

DEEP LEARNING – GENDER RECOGNITION

Overwiew

1. Problem Description
2. Dataset
3. Deep Learning Algorithm
4. Evaluation and Validation
5. Summary

Problem Description

Gender Recognition based on Audio Analysis

Smart speakers are more and more present in our daily environment. Such speakers are able to understand what we are saying and to react properly. Hence it is possible to ask the speaker to provide information like the current weather or to do some work for us like to order products from web shops.

So far there is no speaker on the market which provides the possibility to restrict services to specific users by automatically identifying them by their voice.

Person identification system which are based on voice analysis can be improved by using information about the gender of the person speaking. In this way the search space can be restricted significantly and hence the identification rate can be increased.

Besides, there are much more application scenarios possible. For example, smart speakers can be used in a meeting context to analyze automatically the contribution of female and male speakers and provide information about a balanced speech distribution.

Data Set

LibriSpeech

The analysis presented are based on the LibriSpeech data set which is available for free under: http://www.openslr.org/12/

We use the train-other-500 data set which consists of:

- 564 female speakers
- 602 male speakers

For each speakers several spoken sentences are available.

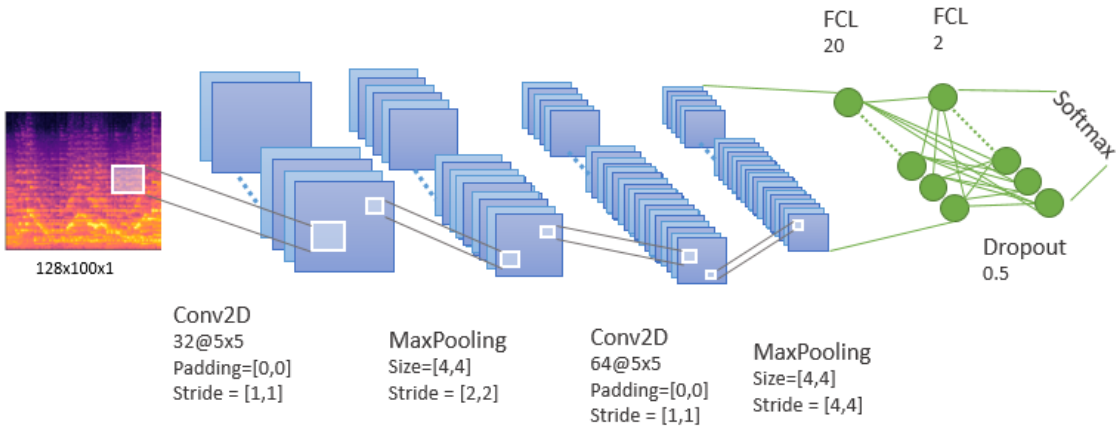
"LibriSpeech is a corpus of approximately 1000 hours of 16kHz read English speech, prepared by Vassil Panayotov with the assistance of Daniel Povey. The data is derived from read audiobooks from the LibriVox project, and has been carefully segmented and aligned."

Deep Learning

A CNN Approach

As shown in related work we use a CNN to differentiate between female and male speakers.

The CNN has the following structure:



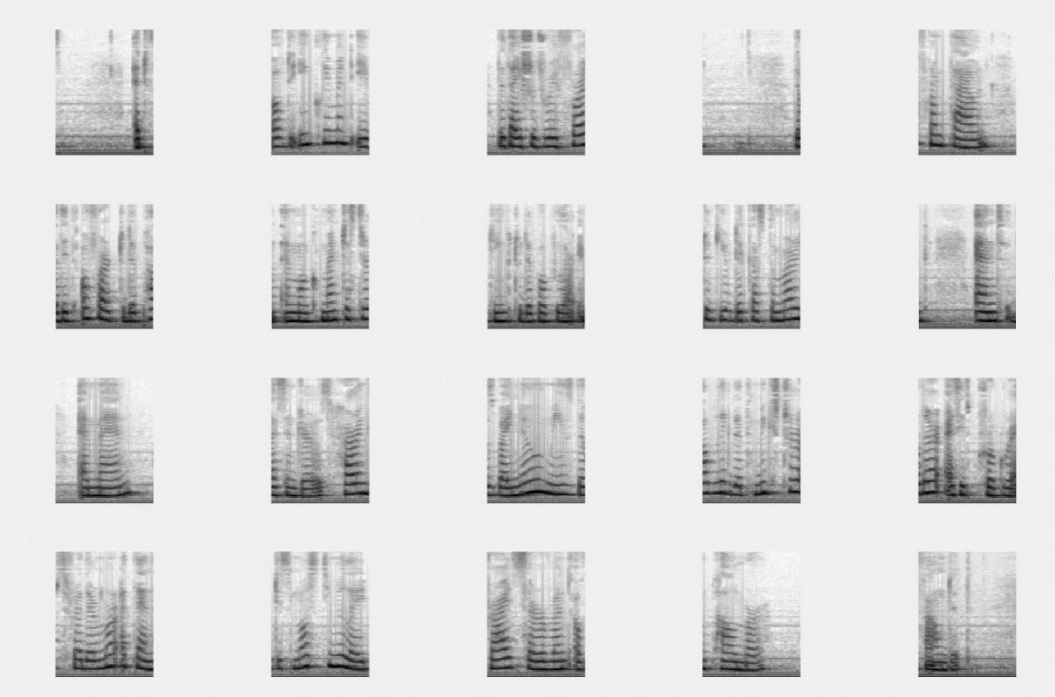
As input we use 128bin Mel-spectrogram images based on 16kHz sampling rate, 1024 FFT size, 160 samples hop size, dynamic range compression and one second snippets of non-overlapping spectrogram pieces.

Deep Learning

A CNN Approach (2)

All in all 840.375 spec images for female speakers and 793.571 spec images for male speakers were computed.

The CNN is trained with 50.000 spectrogram images for each class which were randomly chosen .



Deep Learning

A CNN Approach (2)

The training options are as described in the following:

```
Method: Stochastic gradient descent with momentum
Max Epochs: 15
Initial Learn Rate: 0.0001
Learn Rate Drop Factor: 0.02
Learn Rate Drop Period: 8
MiniBatch Size: 128
```

Evaluation and Validation

The CNN trains as following:

```
Training on single GPU.
Initializing image normalization.
|=========================================================================================|
|     Epoch     |   Iteration   | Time Elapsed |  Mini-batch  |  Mini-batch  | Base Learning|
|               |               |  (seconds)   |     Loss     |   Accuracy   |     Rate     |
|=========================================================================================|
|             1 |             1 |         1.10 |       0.6904 |       52.34% |     1.00e-04 |
|             1 |            50 |        12.61 |       0.6923 |       53.13% |     1.00e-04 |
|             1 |           100 |        24.22 |       0.6890 |       50.78% |     1.00e-04 |
|             1 |           150 |        35.84 |       0.6904 |       54.69% |     1.00e-04 |
|             1 |           200 |        47.45 |       0.6800 |       62.50% |     1.00e-04 |
|             1 |           250 |        59.09 |       0.6611 |       68.75% |     1.00e-04 |
|             1 |           300 |        70.72 |       0.6500 |       67.19% |     1.00e-04 |
|             1 |           350 |        82.35 |       0.6176 |       67.97% |     1.00e-04 |
|             1 |           400 |        93.98 |       0.5729 |       71.09% |     1.00e-04 |
|             1 |           450 |       105.61 |       0.5589 |       71.88% |     1.00e-04 |
|             1 |           500 |       117.24 |       0.4233 |       85.94% |     1.00e-04 |
|             1 |           550 |       128.88 |       0.5032 |       74.22% |     1.00e-04 |
|             1 |           600 |       140.51 |       0.3755 |       80.47% |     1.00e-04 |
|             1 |           650 |       152.14 |       0.3764 |       83.59% |     1.00e-04 |
|             1 |           700 |       163.78 |       0.4754 |       78.13% |     1.00e-04 |
|             1 |           750 |       175.42 |       0.2924 |       85.94% |     1.00e-04 |
|             2 |           800 |       187.21 |       0.3128 |       84.38% |     1.00e-04 |
|             2 |           850 |       198.82 |       0.3369 |       84.38% |     1.00e-04 |
|             2 |           900 |       210.44 |       0.2484 |       89.06% |     1.00e-04 |
|             2 |           950 |       222.12 |       0.3294 |       87.50% |     1.00e-04 |
|             2 |          1000 |       233.75 |       0.3325 |       86.72% |     1.00e-04 |
|             2 |          1050 |       245.40 |       0.3114 |       85.16% |     1.00e-04 |
|             2 |          1100 |       257.04 |       0.3575 |       85.16% |     1.00e-04 |
|             2 |          1150 |       268.68 |       0.3412 |       88.28% |     1.00e-04 |
|             2 |          1200 |       280.32 |       0.2972 |       85.94% |     1.00e-04 |
|             2 |          1250 |       292.00 |       0.2875 |       85.94% |     1.00e-04 |
|             2 |          1300 |       303.70 |       0.3103 |       87.50% |     1.00e-04 |
|             2 |          1350 |       315.35 |       0.2031 |       91.41% |     1.00e-04 |
```

Deep Learning

Evaluation and Validation

...

14	10200	2377.48	0.1948	92.97%	2.00e-06
14	10250	2389.06	0.1753	93.75%	2.00e-06
14	10300	2400.66	0.2164	92.19%	2.00e-06
14	10350	2412.25	0.2701	88.28%	2.00e-06
14	10400	2423.89	0.1971	90.63%	2.00e-06
14	10450	2435.49	0.1395	96.09%	2.00e-06
14	10500	2447.09	0.1931	92.19%	2.00e-06
14	10550	2458.70	0.2041	92.19%	2.00e-06
14	10600	2470.31	0.2218	92.19%	2.00e-06
14	10650	2481.91	0.1501	96.88%	2.00e-06
14	10700	2493.52	0.2057	88.28%	2.00e-06
14	10750	2505.14	0.2357	92.19%	2.00e-06
14	10800	2516.75	0.3098	89.06%	2.00e-06
14	10850	2528.37	0.1848	92.97%	2.00e-06
14	10900	2540.00	0.2200	91.41%	2.00e-06
15	10950	2551.77	0.2380	90.63%	2.00e-06
15	11000	2563.35	0.2060	92.19%	2.00e-06
15	11050	2574.94	0.1322	95.31%	2.00e-06
15	11100	2586.53	0.3047	89.84%	2.00e-06
15	11150	2598.14	0.1317	93.75%	2.00e-06
15	11200	2609.74	0.1658	92.97%	2.00e-06
15	11250	2621.34	0.2731	91.41%	2.00e-06
15	11300	2632.94	0.1648	92.97%	2.00e-06
15	11350	2644.55	0.2477	90.63%	2.00e-06
15	11400	2656.16	0.2315	93.75%	2.00e-06
15	11450	2667.77	0.2105	91.41%	2.00e-06
15	11500	2679.37	0.1218	94.53%	2.00e-06
15	11550	2690.98	0.1819	93.75%	2.00e-06
15	11600	2702.60	0.2146	90.63%	2.00e-06
15	11650	2714.22	0.2586	89.06%	2.00e-06
15	11700	2725.91	0.0873	97.66%	2.00e-06
15	11715	2729.39	0.2202	93.75%	2.00e-06

Elapsed time is 3063.499548 seconds.

Evaluation (1)

The CNN is evaluated with remaining spec images from all speakers. The data was excluded from the training phase.

Female: 743.571
Male: 790.375

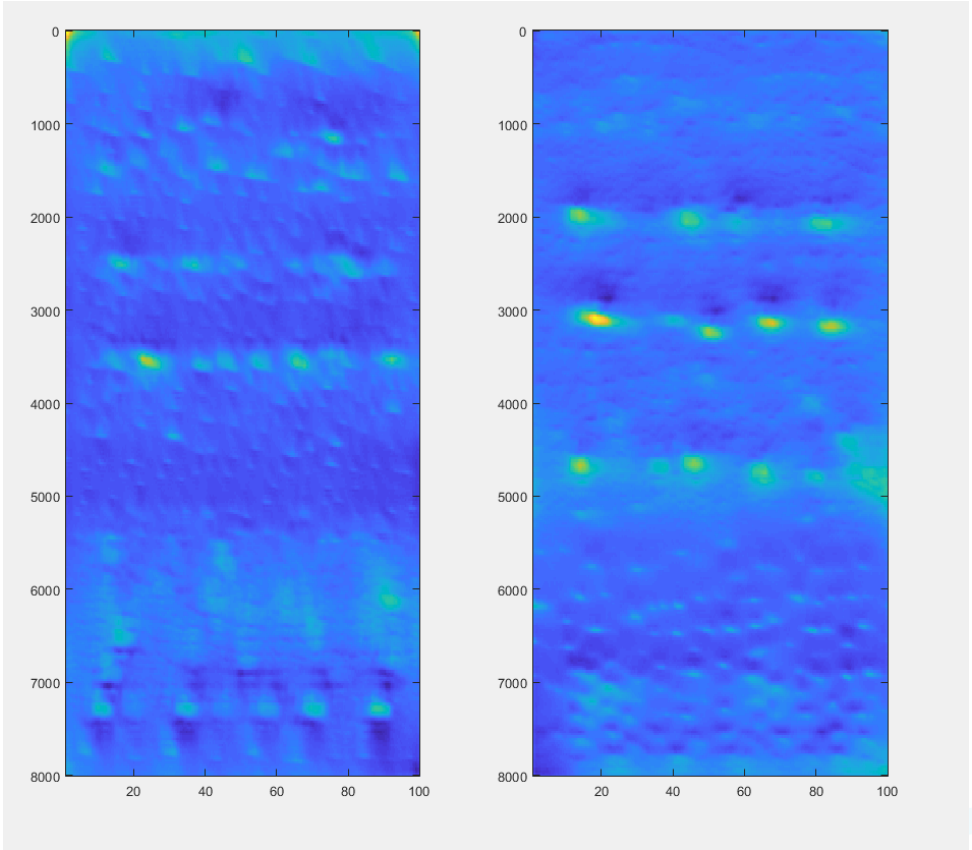
The CNN reached an accuracy of **93.03%**.

Real / Predicted	Pred. Female	Pred Male
Real Female	691.324	52.247
Real Male	54.625	735.750

The net structure (Matlab data file) has a size of 887kB.

Evaluation (2)

Deep Dream Image

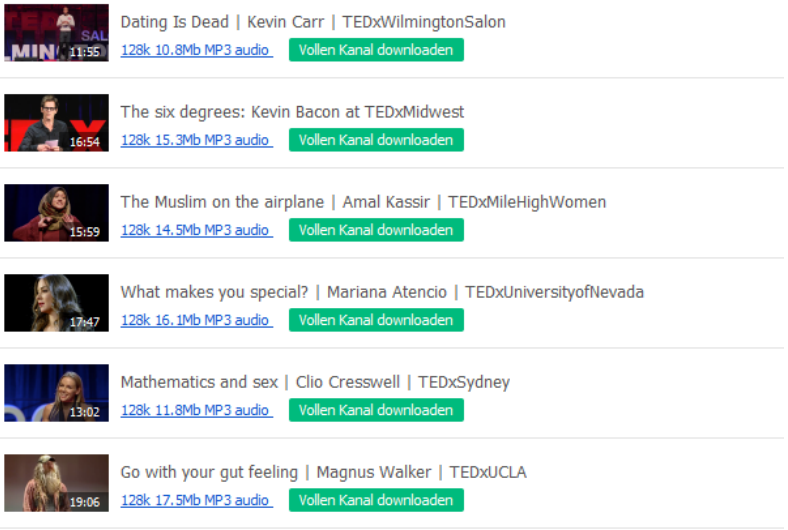


Deep dream image of class „Hen“ from Alex Net

Evaluation and Validation

Validation

We use the following data to validate the CNN performance:



Spec Images:

- 3 different female speakers: 2714
- 3 different male speakers: 2817

The CNN reached an accuracy of **93.03%**.

Evaluation and Validation

Validation

The CNN reached an accuracy of **84.25%**.

Real / Predicted	Pred. Female	Pred Male
Real Female	2446	267
Real Male	604	2212

NOTE: Data was NOT cleaned! This means that overlapping noise like “introduction music”, “applause” and “laughter” is present in the data and will be classified either as female or male due to the missing “NULL” class.

However, this problem is solved when combining it with a Music-Speech-Surrounding Noise detection system.

Evaluation and Validation

Online

Runtime classify function: ~0.007sec (Lenovo ThinkStation P700)

Evaluation and Validation

Online

Online Runtime Evaluation on Lenovo ThinkStation P700

- Calculate Feature: ~0.008 sec
- Classify Function: ~0.007 sec

Introduction

AI Services – Realized since March 2017 (1)

Gender Recognition based on the speaker's voice

Detection of siren sounds in urban environments

Detection of baby scream in home environments

Detection of doorbell sounds in home environments

Environment type recognition based on a single image

Music genre recognition based on sound snipets

Speaker recognition based on large training data and unknown speaker clustering based on voice samples

*Intelligent noise reduction**

*Classification of music, speech and noise based on sound snipets**

* Student work

Overwiew

1. Problem Description
2. Dataset
3. Deep Learning Algorithm
4. Evaluation and Validation
5. Summary

Problem Description

Siren Detection in Urban Environments

People love to enjoy their favorite music when they are at home. Besides high-end sound systems and Bluetooth speakers, headphones are very often used in home environments for several reasons.

Families with babies for example, do not want to disturb the nap of their kids. And in big residential buildings, one would like to remain the good relation with his neighbors. So headphones are a good way to enjoy music and not to disturb anybody around you.

However, a big disadvantage of listening to loud music with headphones is, that you will not recognize what happens in your environment.

This work shows a method that is able to detect baby scream and doorbell sounds based on audio processing and deep learning. Consequently, a smart headphone could use this technique to recognize such important events and to inform the user by reducing the volume automatically.

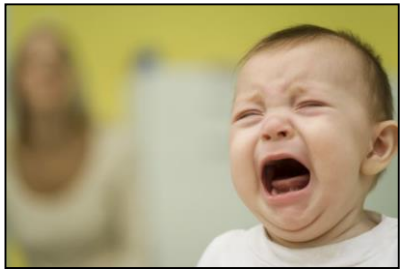
Data Set

Youtube

The analysis presented are based on a data set which was downloaded from Youtube.

The training / test data consists of sounds of the following classes:

- Baby Scream (about 28hours – partly repeating)
- Doorbell sounds (about 32hours – partly repeating)
- Home Environment Sounds – NULL / Noise class (about 35hours – partly repeating)

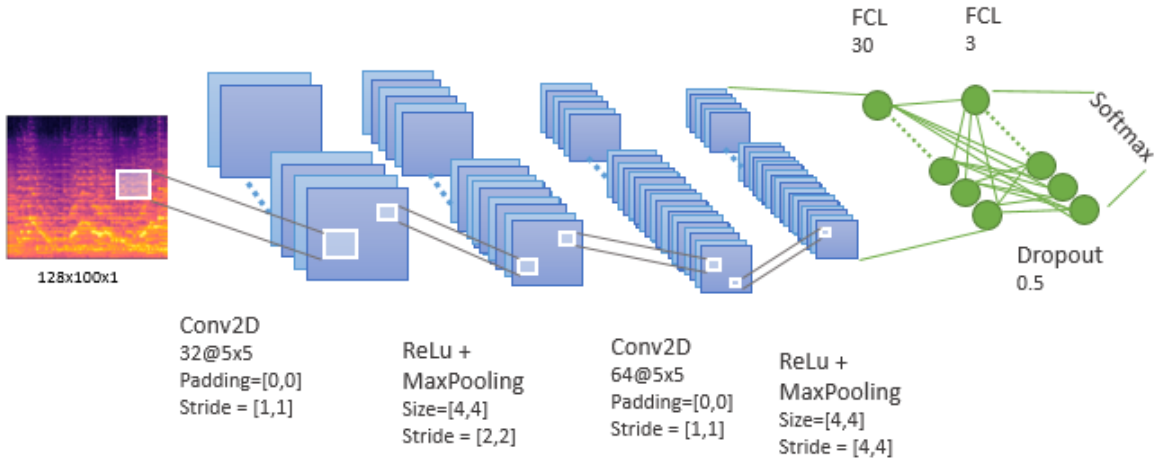


About 25hours of data was used for training and the remaining samples for testing.

Deep Learning

A CNN Approach

The CNN has the following structure:



As input we use 128bin Mel-spectrogram images based on 16kHz sampling rate, 1024 FFT size, 160 samples hop size, dynamic range compression and one second snippets of non-overlapping spectrogram pieces.

Deep Learning

A CNN Approach (2)

The training options are as described in the following:

```
Method: Stochastic gradient descent with momentum
Max Epochs: 3
Initial Learn Rate: 0.0001
Learn Rate Drop Factor: 0.02
Learn Rate Drop Period: 10
MiniBatch Size: 128
```

Deep Learning

Evaluation and Validation

The CNN trains as following:

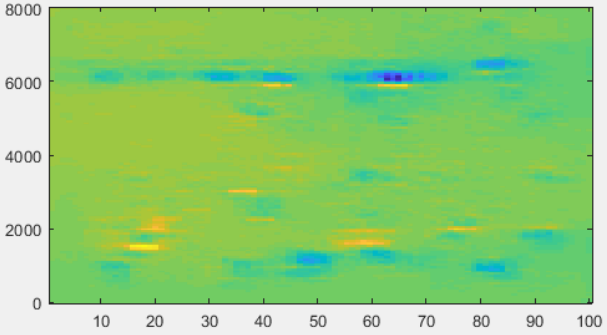
Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
1	1	3.84	1.0987	35.16%	1.00e-04
1	50	9.37	1.0749	50.00%	1.00e-04
1	100	13.44	1.0027	72.66%	1.00e-04
1	150	17.51	0.6457	84.38%	1.00e-04
1	200	21.59	0.4137	84.38%	1.00e-04
1	250	25.67	0.2974	89.84%	1.00e-04
1	300	29.78	0.2051	93.75%	1.00e-04
1	350	33.92	0.2210	94.53%	1.00e-04
		...			
3	5100	490.52	0.0174	100.00%	1.00e-04
3	5150	496.63	0.0165	100.00%	1.00e-04
3	5200	502.74	0.0051	100.00%	1.00e-04
3	5250	508.83	0.0091	100.00%	1.00e-04
3	5300	514.81	0.0048	100.00%	1.00e-04
3	5350	520.70	0.0113	100.00%	1.00e-04
3	5400	526.61	0.0066	100.00%	1.00e-04
3	5450	532.48	0.0026	100.00%	1.00e-04
3	5500	538.37	0.0124	99.22%	1.00e-04
3	5550	544.26	0.0038	100.00%	1.00e-04
3	5600	550.14	0.0076	100.00%	1.00e-04
3	5650	556.01	0.0048	100.00%	1.00e-04
3	5700	561.87	0.0181	99.22%	1.00e-04
3	5750	567.78	0.0118	100.00%	1.00e-04
3	5800	573.65	0.0029	100.00%	1.00e-04
3	5850	579.32	0.0011	100.00%	1.00e-04
3	5900	583.45	0.0029	100.00%	1.00e-04
3	5950	587.54	0.0020	100.00%	1.00e-04
3	6000	591.60	0.0039	100.00%	1.00e-04
3	6050	595.80	0.0197	99.22%	1.00e-04
3	6100	599.99	0.0071	100.00%	1.00e-04
3	6150	604.23	0.0200	98.44%	1.00e-04
3	6200	608.45	0.0062	100.00%	1.00e-04
3	6250	612.61	0.0041	100.00%	1.00e-04
3	6300	616.73	0.0043	100.00%	1.00e-04
3	6327	619.07	0.0041	100.00%	1.00e-04

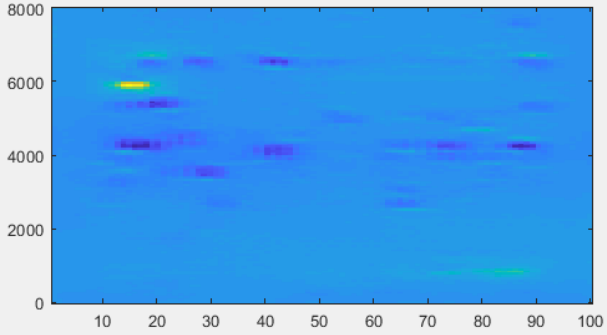
- Optimizer: Stochastic gradient descent with momentum
- Max Epochs: 3
- Initial Learn Rate: 0.0001
- Learn Rate Drop Factor: 0.02
- Learn Rate Drop Period: 10
- MiniBatch Size: 128

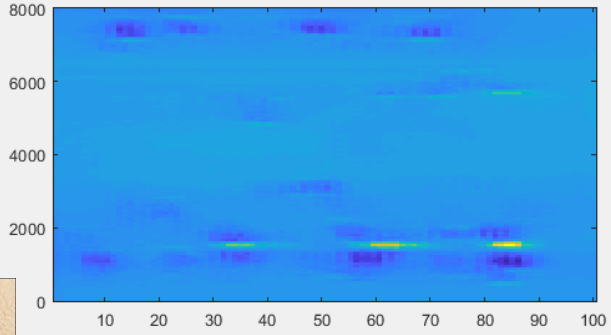
Evaluation and Validation

Deep Dream

The deep dream images (high activations) for each class







Evaluation and Validation

Evaluation

The CNN is evaluated with spec which were excluded from the training phase.

Baby Scream: 11.027sec
Doorbell: 25.744sec
EnvironmentSounds: 37.050sec

The CNN reached an accuracy of **99.75%**.

Real / Predicted	Pred. EnvSound	Pred. BabyScream	Pred. Doorbell
Real EnvSound	10981	12	34
Real BabyScream	0	25744	0
Real Doorbell	121	14	36915

Validation

Data was recorded in a real-live scenarios using the audio recorder App of an iPhone 6s. Only EnvironmentSounds and DoorBell was evaluated due to a missing baby.

EnvSounds: daily living sounds (e.g. having a shower, washing machine, driving a car, walking outside, walk inside a mall, doing home work,…)

Doorbell: Different door bells

Real / Predicted	Pred. EnvSound	Pred. BabyScream	Pred. Doorbell
Real EnvSound	2241	0	7
Real BabyScream	0	0	0
Real Doorbell	96	0	30

Note: Doorbell sounds were included in environment sound. To do a more reliable evaluation the doorbell data must be isolated.

Deep Learning

Outlook

- Do a more detailed validation based on a huge data set including various doorbell sounds and baby scream
- Validate a event spotting approach instead of a sample based recognition

Introduction

AI Services – Realized since March 2017 (1)

Gender Recognition based on the speaker's voice

Detection of siren sounds in urban environments

Detection of baby scream in home environments

Detection of doorbell sounds in home environments

Environment type recognition based on a single image

Music genre recognition based on sound snipets

Speaker recognition based on large training data and unknown speaker clustering based on voice samples

*Intelligent noise reduction**

*Classification of music, speech and noise based on sound snipets**

* Student work

Overwiew

1. Problem Description
2. Dataset
3. Deep Learning Algorithm
4. Evaluation and Validation
5. Summary

Problem Description

Siren Detection in Urban Environments

People love to hear music with their portable music players when they are on the roads. Nowadays you can see lots of people walking through the city and enjoying their favorite music using headphones. Although the sound experience is great, the fact that they are isolated from their surrounding with respect to the cognition of environmental sounds is very dangerous.

Consequently, people are not aware of warn signals such as car horns or sirens which could bring them in dangerous situations.

The following approach shows a concept to overcome this problem. A deep learning technique is used to analyze the environment sound and categorize them in “Silence”, “Urban Environment Noise” and “Siren”.

A smart headphone can automatically lower the sound if it detects a siren and hence the user gets aware of a possible dangerous situation.

Data Set

Youtube

The analysis presented are based on a data set which was downloaded from Youtube.

The data consists of sounds of the following classes:

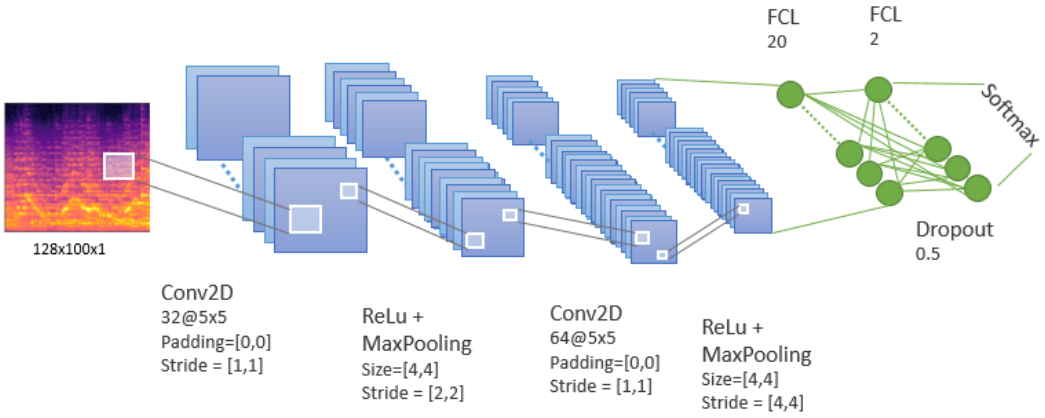
- Urban environment noise
- Siren (German ambulance)

Deep Learning

A CNN Approach

As shown in related work we use a CNN to differentiate between siren sound and urban noise.

The CNN has the following structure:



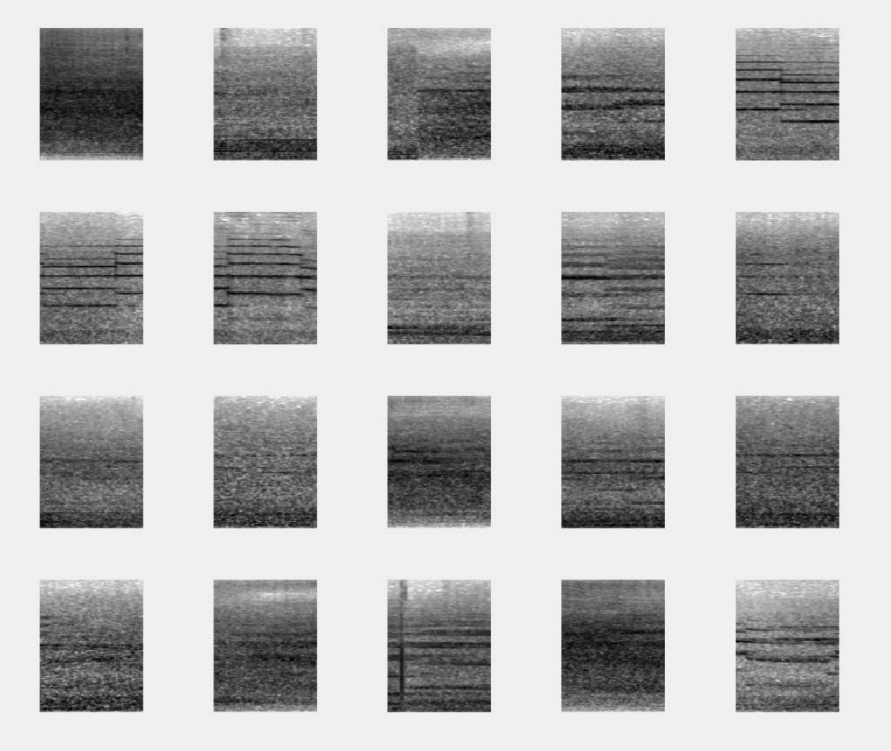
As input we use 128bin Mel-spectrogram images based on 16kHz sampling rate, 1024 FFT size, 160 samples hop size, dynamic range compression and one second snippets of non-overlapping spectrogram pieces.

Deep Learning

A CNN Approach (2)

All in all 4.086 spec images for siren sound and 4.899 spec images for urban background noise were computed.

Due to the small data set, the CNN is trained with 4.000 spectrogram images for each class which were randomly chosen.



Deep Learning

A CNN Approach (2)

The training options are as described in the following:

```
Method: Stochastic gradient descent with momentum
Max Epochs: 15
Initial Learn Rate: 0.0001
Learn Rate Drop Factor: 0.02
Learn Rate Drop Period: 10
MiniBatch Size: 200
```

Evaluation and Validation

The CNN trains as following:

```
Training on single GPU.
Initializing image normalization.
|=========================================================================================|
|     Epoch    |   Iteration  | Time Elapsed |  Mini-batch  |  Mini-batch  | Base Learning|
|              |              |   (seconds)  |     Loss     |   Accuracy   |     Rate     |
|=========================================================================================|
|            1 |            1 |         0.62 |       0.6949 |       48.00% |     1.00e-04 |
|            2 |           50 |        20.74 |       0.6760 |       58.00% |     1.00e-04 |
|            3 |          100 |        41.11 |       0.6412 |       59.50% |     1.00e-04 |
|            4 |          150 |        61.50 |       0.5142 |       82.00% |     1.00e-04 |
|            5 |          200 |        81.99 |       0.3439 |       88.00% |     1.00e-04 |
|            7 |          250 |       102.45 |       0.2669 |       91.50% |     1.00e-04 |
|            8 |          300 |       122.92 |       0.2881 |       91.00% |     1.00e-04 |
|            9 |          350 |       143.39 |       0.2266 |       93.00% |     1.00e-04 |
|           10 |          400 |       163.88 |       0.1949 |       94.50% |     1.00e-04 |
|           12 |          450 |       184.46 |       0.2084 |       94.50% |     2.00e-06 |
|           13 |          500 |       204.98 |       0.2456 |       93.50% |     2.00e-06 |
|           14 |          550 |       225.50 |       0.2078 |       94.00% |     2.00e-06 |
|           15 |          600 |       245.96 |       0.2057 |       94.00% |     2.00e-06 |
|=========================================================================================|
Elapsed time is 247.818337 seconds.

accuracy =

    0.9675
```

Evaluation (1)

The CNN is evaluated with remaining spec images from all speakers. The data was excluded from the training phase.

NOTE: The test data set is very small and hence not really significant – we would like to refer to the validation chapter to get reliable results.

Siren: 86
UrbanSounds: 899

The CNN reached an accuracy of **96.75%**.

Real / Predicted	Pred. Siren	Pred US
Real Siren	75	11
Real US	21	878

The net structure (Matlab data file) has a size of 887kB.

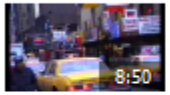
Evaluation and Validation

Validation

We use the following data to validate the CNN performance:

[NUR LED BLAULICHT] Feuerwehr + Rettungsdienst + Polizei Frankfurt/Main im Dauereinsatz

128k 5.4Mb MP3 audio Vollen Kanal downloaden



New York - the noisy city - 1988 - Manhattan

128k 8Mb MP3 audio Vollen Kanal downloaden

Ambulance VZA Amsterdam (collection)

128k 2.3Mb MP3 audio Vollen Kanal downloaden

Spec Images:

- Siren sounds: 412
- Urban noise: 526

Evaluation and Validation

Validation

The CNN reached an accuracy of **79,91%**.

Real / Predicted	Pred. Siren	Pred US
Real Siren	349	62
Real US	126	399

NOTE: Due to the small data set the CNN was not able to generalize very well. Some urban sound noises were wrongly predicted as siren sound. Whereas only a small set of siren sound was misclassified as urban noise.

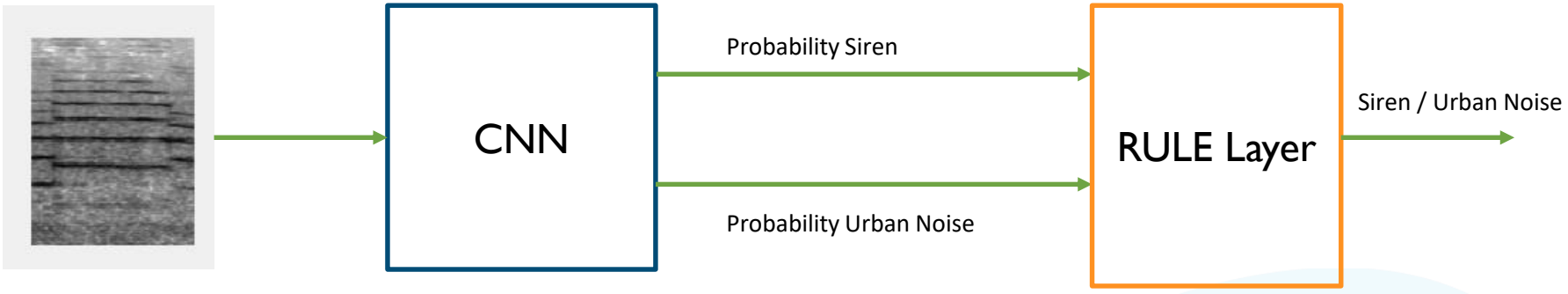
To overcome this problem and to void the need to collect more training data the following approach is realized.

Evaluation and Validation

Improve recognition rate by keeping a small training set

The approach shown is based on the following facts:

1) A siren lasts usually several seconds. This implies that a siren is only detected if a specific number of consecutive siren events was recognized
2) Investigations have shown that almost all siren sounds are recognized with a very high probability. So many false classified siren sounds can be removed by applying a recognition probability threshold



Min detected consecutive Siren events = 3
Min probability threshold for a siren event = 92%

Evaluation and Validation

Online

Runtime classify function: ~0.007sec (Lenovo ThinkStation P700)

Evaluation and Validation

Remarks

The rule layer is able to compensate the small training set to a certain degree only. Due to the fact that the application should be used as a life guard service, we strongly recommend to enhance the training data and consequently the recognition rate of the system by:

- Collecting more siren and urban environment sounds
- Generate the Siren sounds automatically

Introduction

AI Services – Realized since March 2017 (1)

Gender Recognition based on the speaker's voice

Detection of siren sounds in urban environments

Detection of baby scream in home environments

Detection of doorbell sounds in home environments

Environment type recognition based on a single image

Music genre recognition based on sound snipets

Speaker recognition based on large training data and unknown speaker clustering based on voice samples

*Intelligent noise reduction**

*Classification of music, speech and noise based on sound snipets**

* Student work

Overwiew

1. Problem Description
2. Dataset
3. Deep Learning Algorithm
4. Evaluation and Validation
5. Summary

Problem Description

Scene Analysis based on a single Image

Smart speakers are more and more present in our daily environment. Such speakers are able to understand what we are saying and to react properly. Hence it is possible to ask the speaker to provide information like the current weather or to do some work for us like to order products from web shops.

Beside the range of smart services, the audio quality is still of high importance as well. Due to the portability of the speaker, the acoustic scene can significantly change from one moment to the other. Consequently, the speaker must be able to realize in which type of surrounding it is currently located to adapt its parameter in order to deliver the best audio possible.

This approach shows a scene analysis service which is based on a single image. The image of the room (taken by an integrated camera or sent to the speaker via the smartphone) is analyzed and classified in several typical surroundings which can be used to adapt the speaker's sound quality (e.g. equalizing).

Data Set

MIT places database

The dataset was downloaded from the MIT Places project page (http://places2.csail.mit.edu/index.html)

We used 70.000 images to train the classifier to distinguish between 14 typical rooms / places. Each class consists of 5.000 images.

The image size is 256x256

Data Set

MIT places database

The following classes are considered:

- Auditorium
- Bathroom
- Bedroom
- Childs room
- Conference room
- Corridor
- Dinning room
- Garage
- Kitchen
- Living room
- Office
- Park
- Swimming Pool
- Terrace











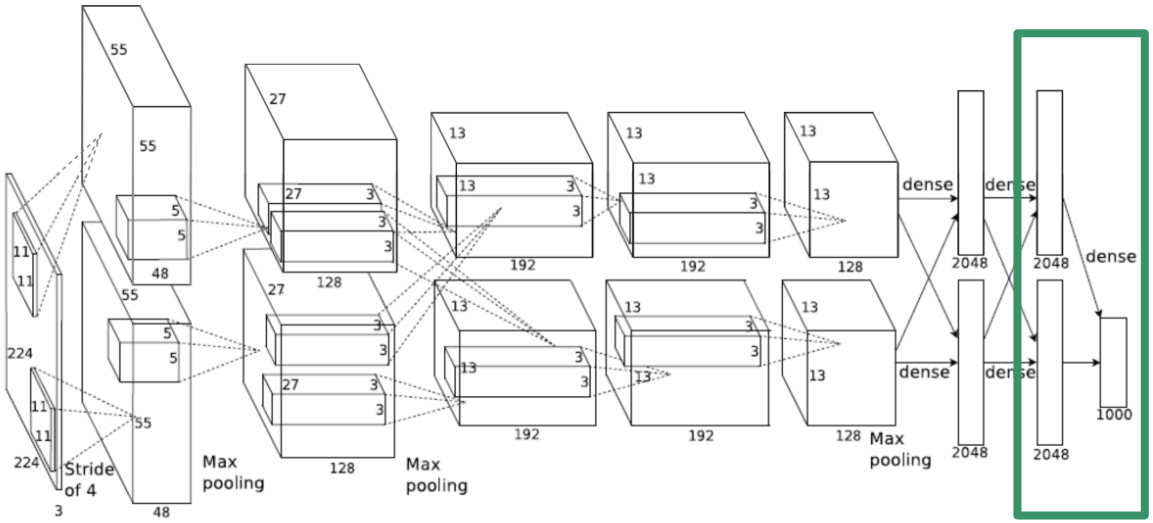



Deep Learning

A CNN Approach

This approach uses a transfer-learning technique to make use of already existing well trained image classification networks.

The well known AlexNet was adapted to the problem described in this work. More detailed, the last two layers of AlexNet were modified and adapter to solve our problem of recognizing 14 scenes.



Alex Net

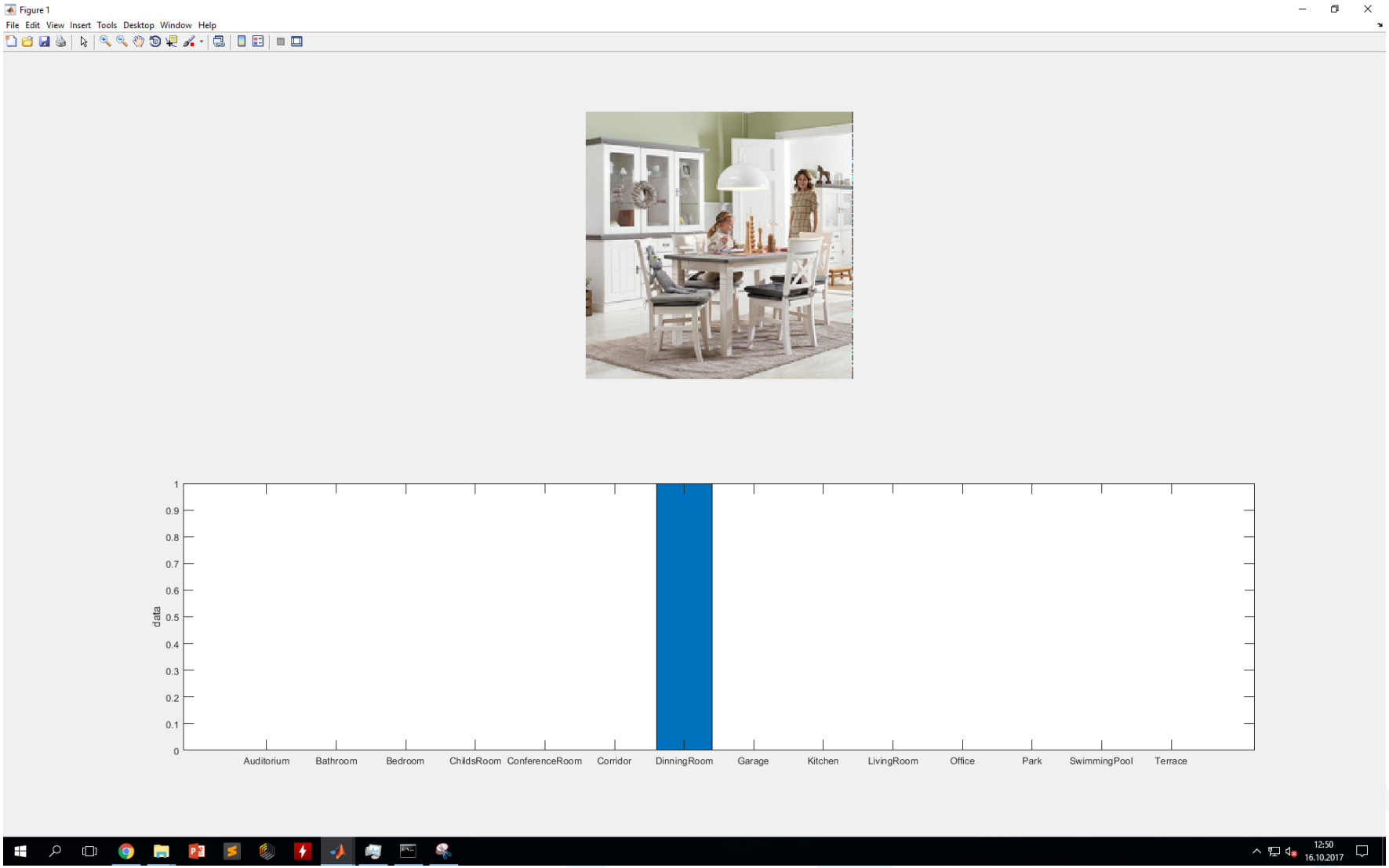
Evaluation

The system was validated in a first shot with a very small dataset (25 images). Half of the images were downloaded from GoogleImages and the other half was shot with an iPhone in and around the HARMAN building in Straubing.

The following slides show the results (image + probability for each class).

Deep Learning

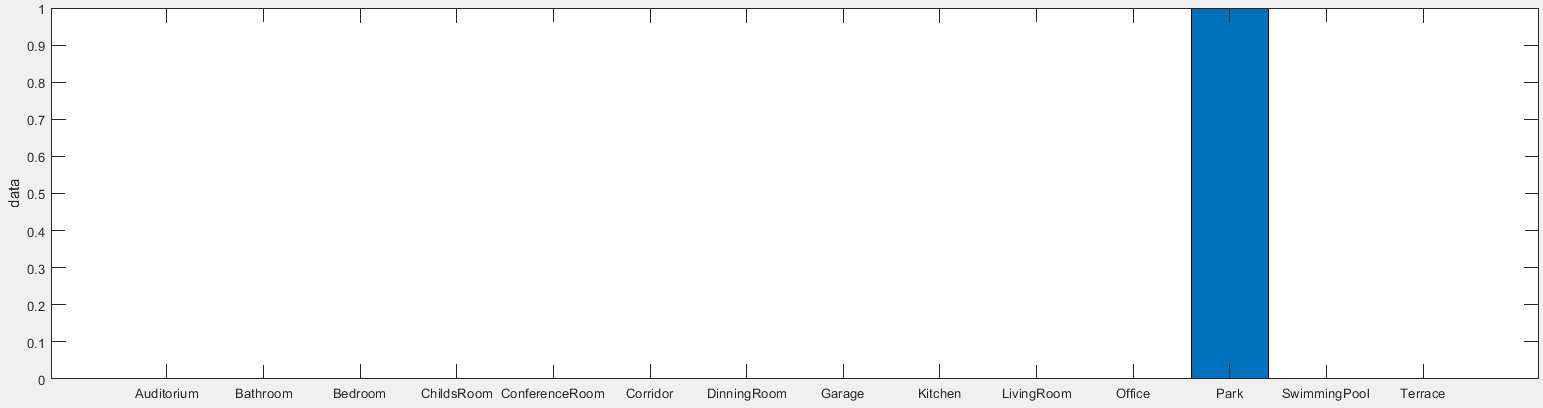
Evaluation



Deep Learning

Figure 1

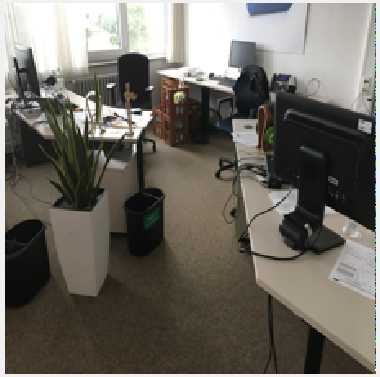
File Edit View Insert Tools Desktop Window Help

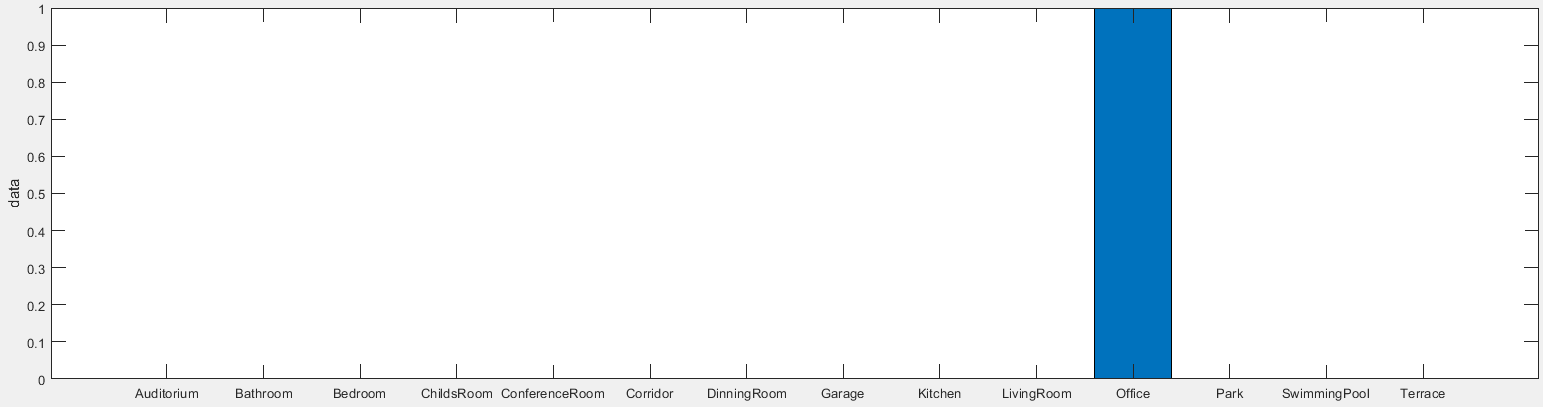


Deep Learning

Figure 1

File Edit View Insert Tools Desktop Window Help

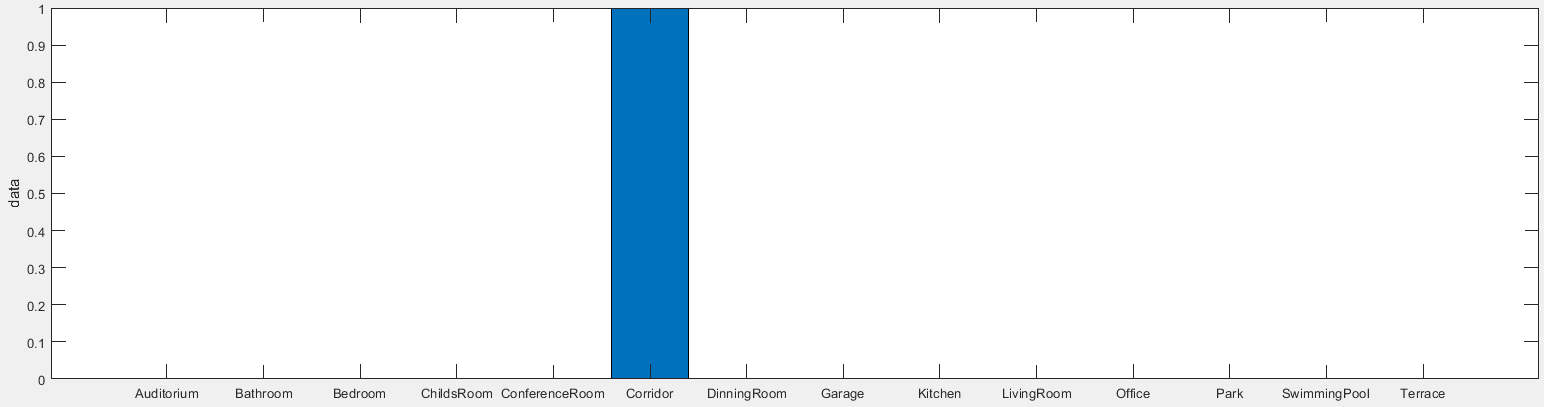




Deep Learning

Figure 1

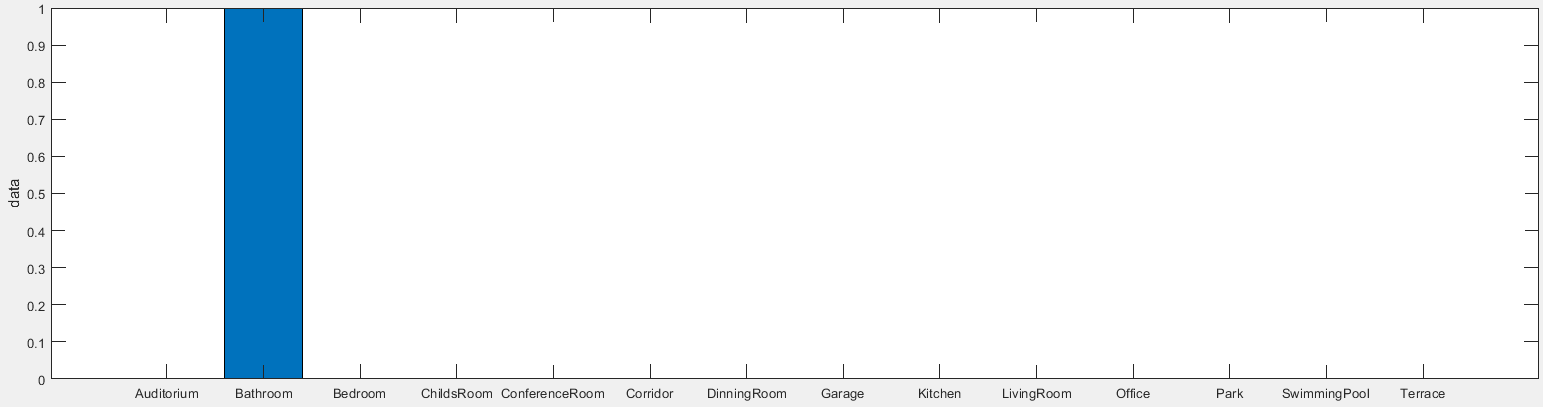
File Edit View Insert Tools Desktop Window Help



Deep Learning

Figure 1

File Edit View Insert Tools Desktop Window Help

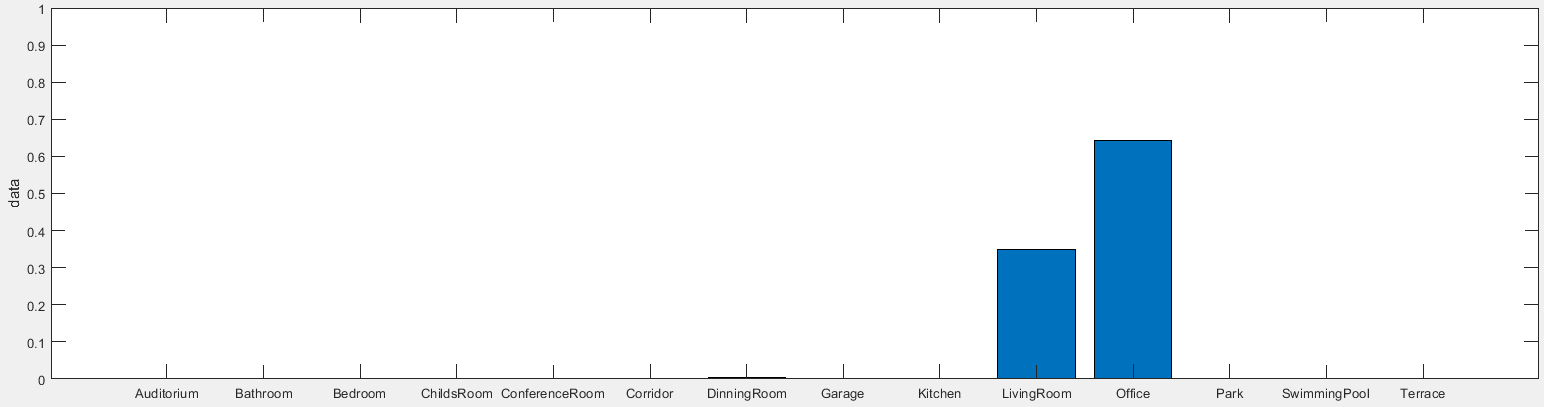


12:51
16.10.2017

Deep Learning

Figure 1

File Edit View Insert Tools Desktop Window Help

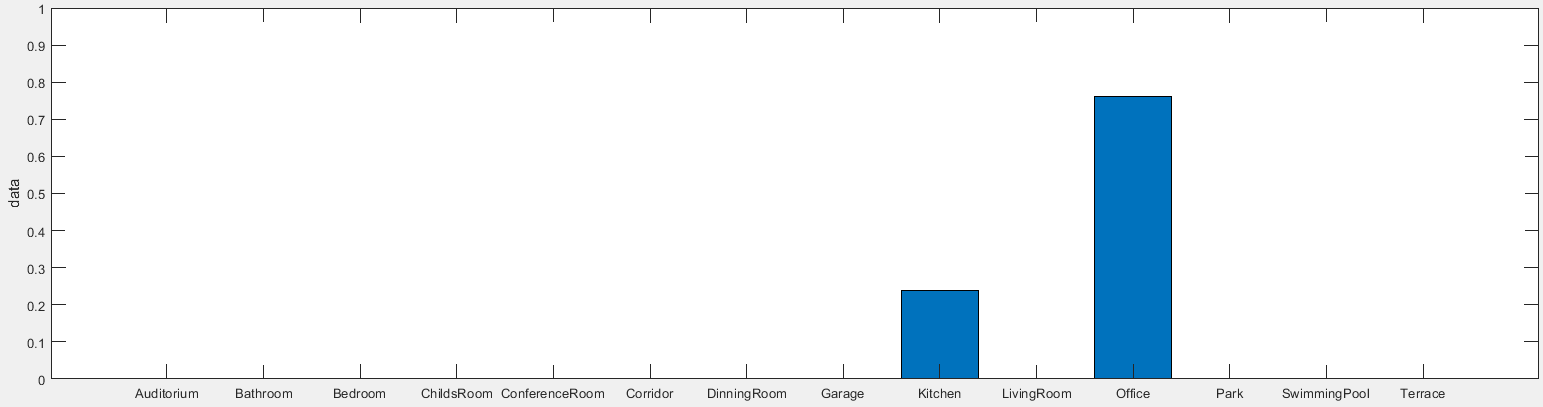


12:51
16.10.2017

Deep Learning

Figure 1

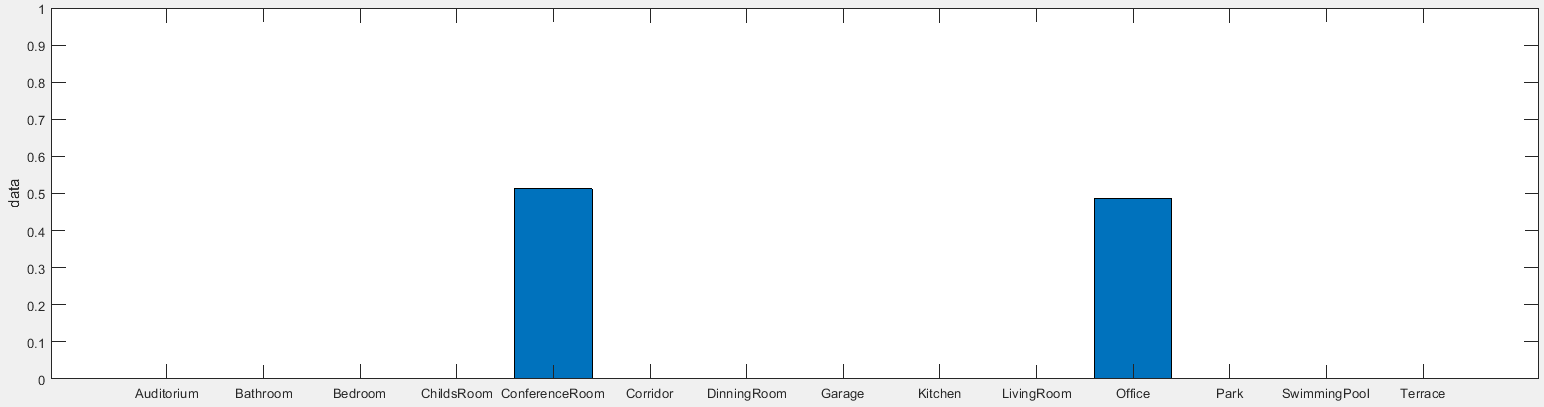
File Edit View Insert Tools Desktop Window Help



Deep Learning

Figure 1

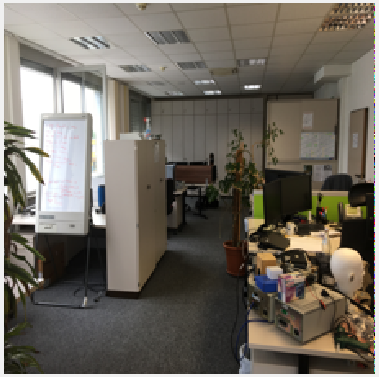
File Edit View Insert Tools Desktop Window Help

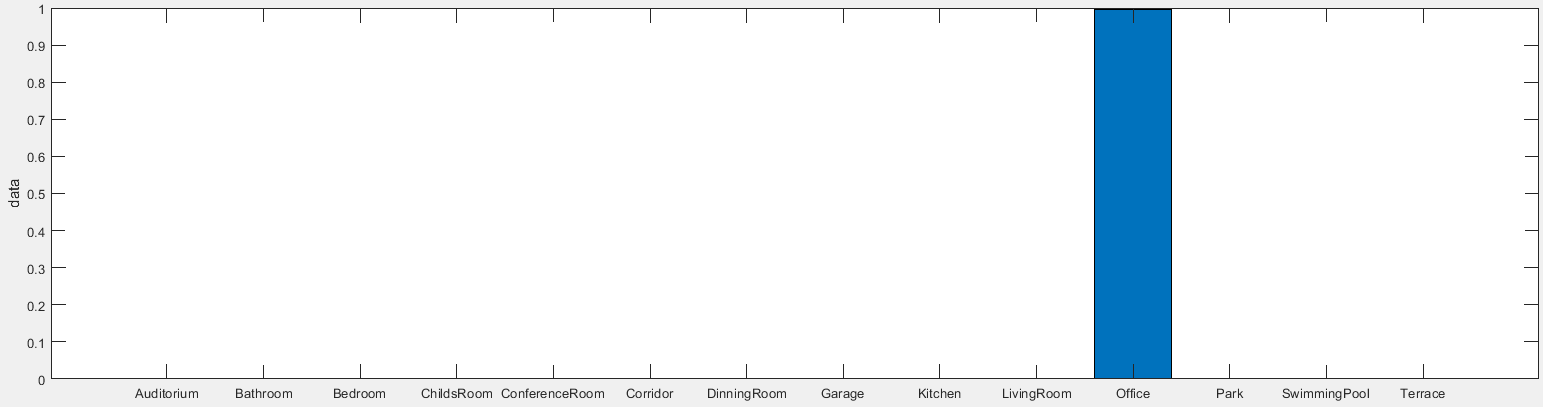


Deep Learning

Figure 1

File Edit View Insert Tools Desktop Window Help



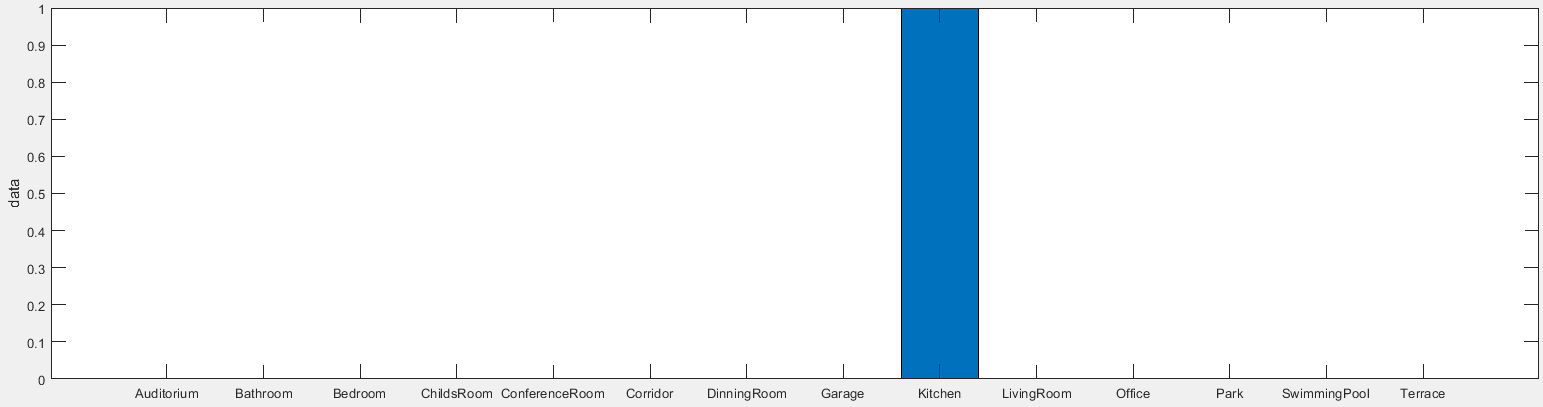


12:51
16.10.2017

Deep Learning

Figure 1

File Edit View Insert Tools Desktop Window Help

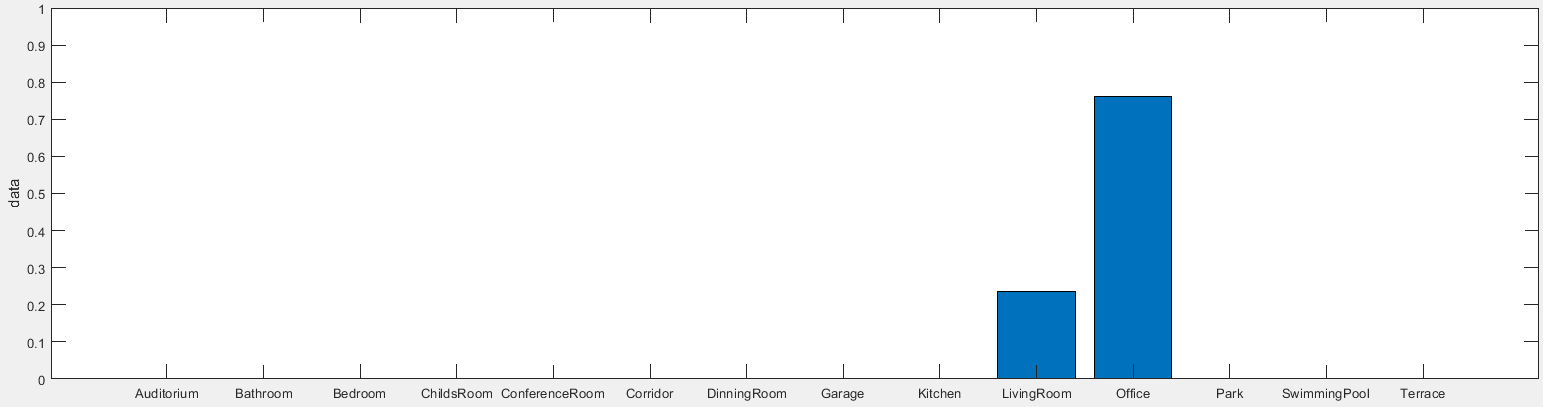


12:51
16.10.2017

Deep Learning

Figure 1

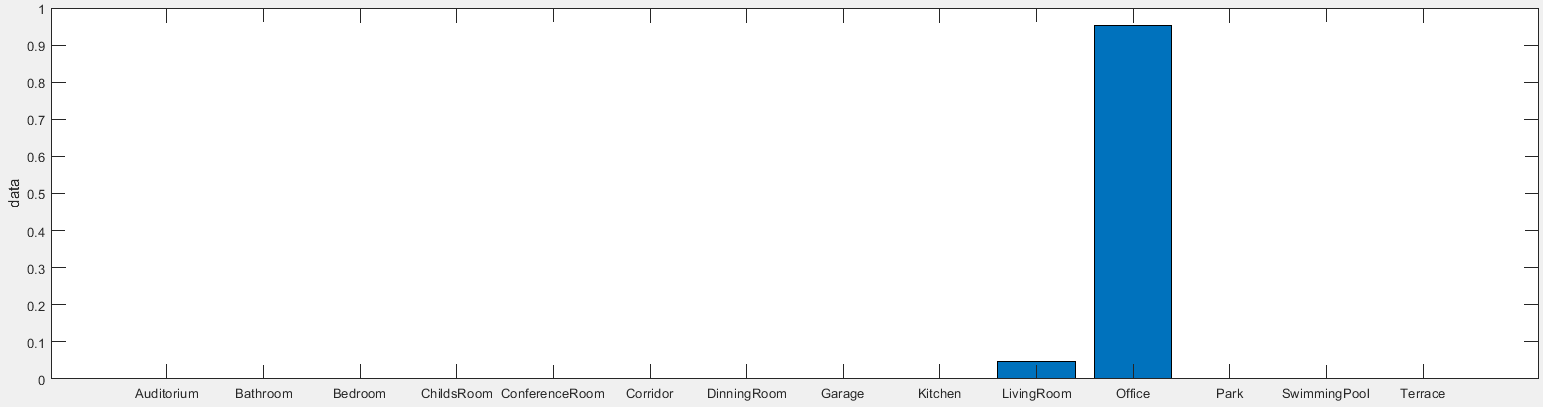
File Edit View Insert Tools Desktop Window Help



Deep Learning

Figure 1

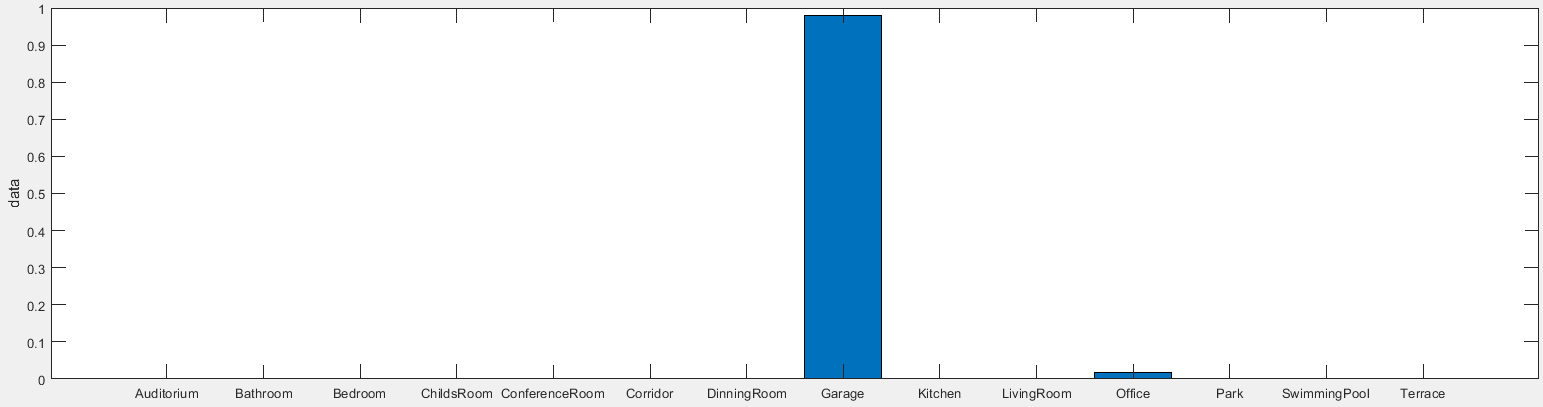
File Edit View Insert Tools Desktop Window Help



Deep Learning

Figure 1

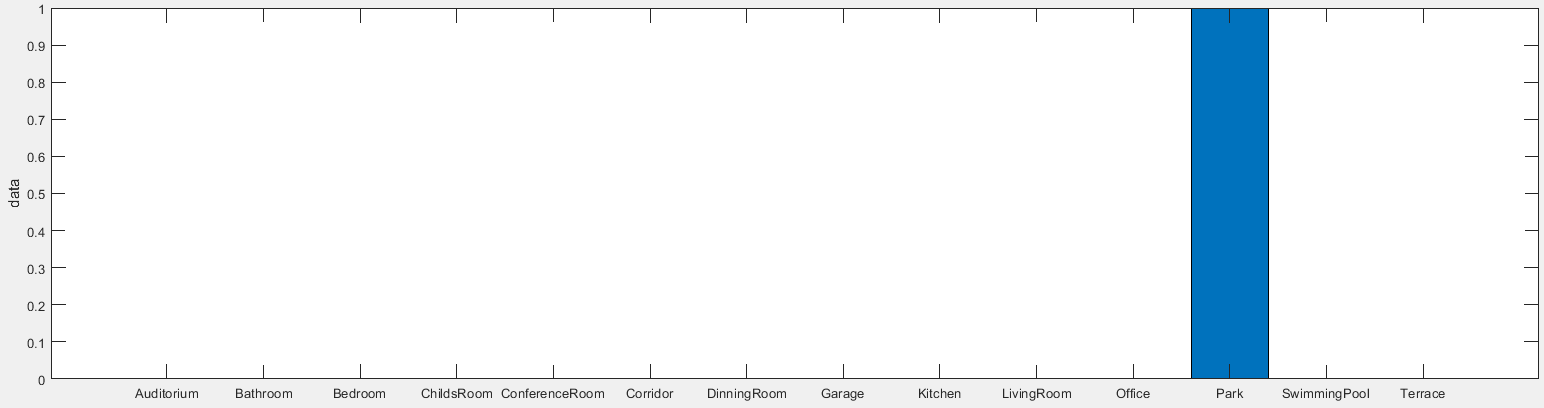
File Edit View Insert Tools Desktop Window Help



Deep Learning

Figure 1

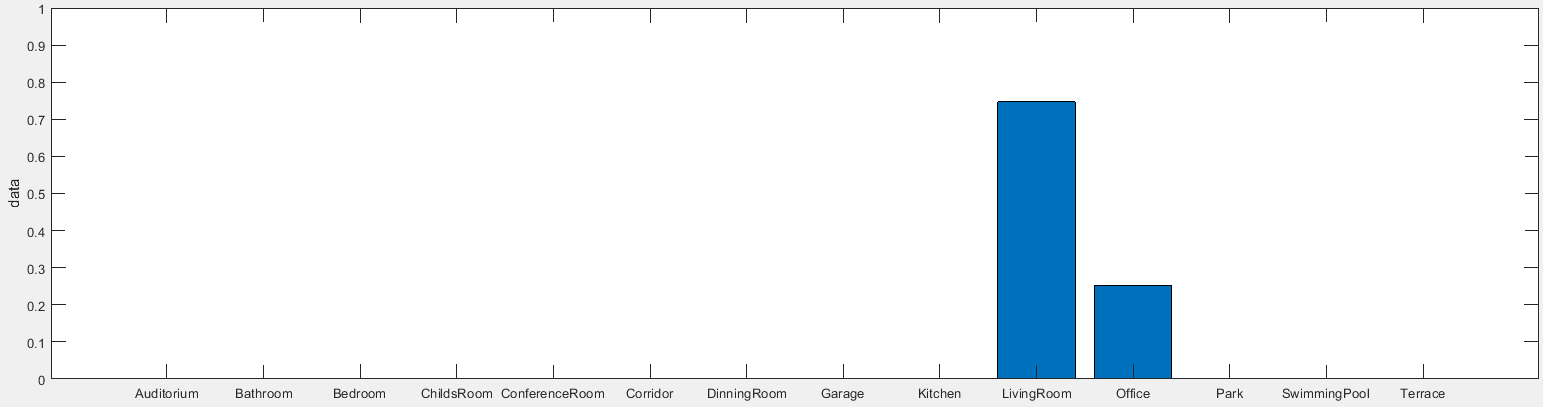
File Edit View Insert Tools Desktop Window Help



Deep Learning

Figure 1

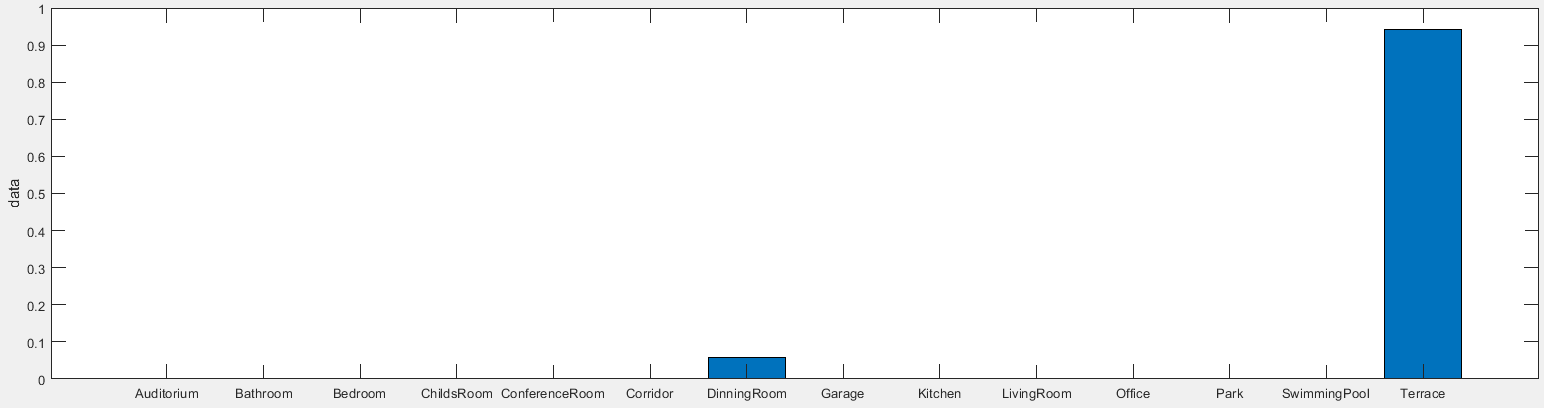
File Edit View Insert Tools Desktop Window Help



Deep Learning

Figure 1

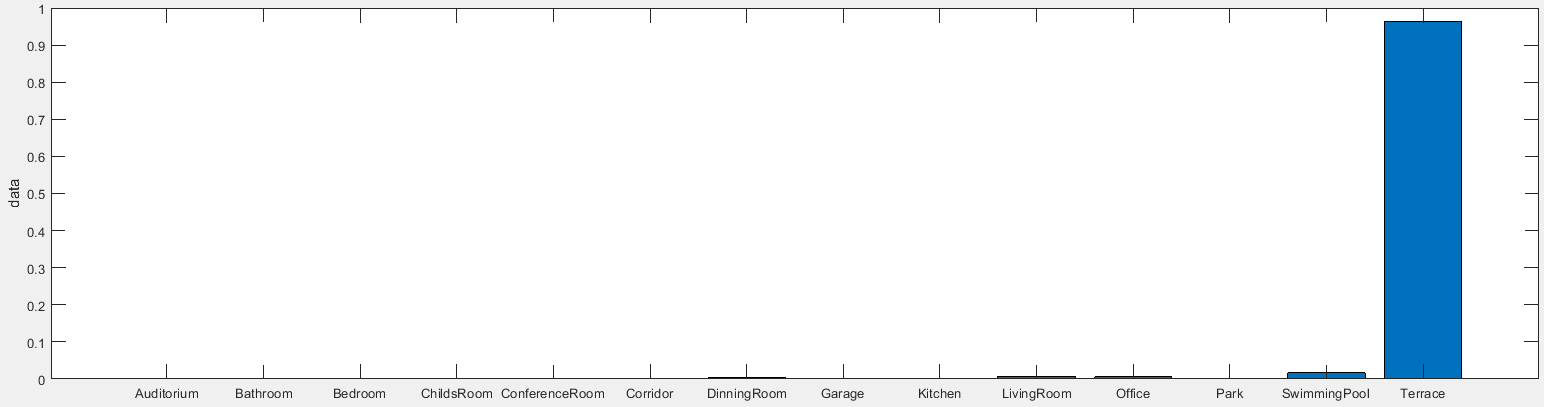
File Edit View Insert Tools Desktop Window Help



Deep Learning

Figure 1

File Edit View Insert Tools Desktop Window Help

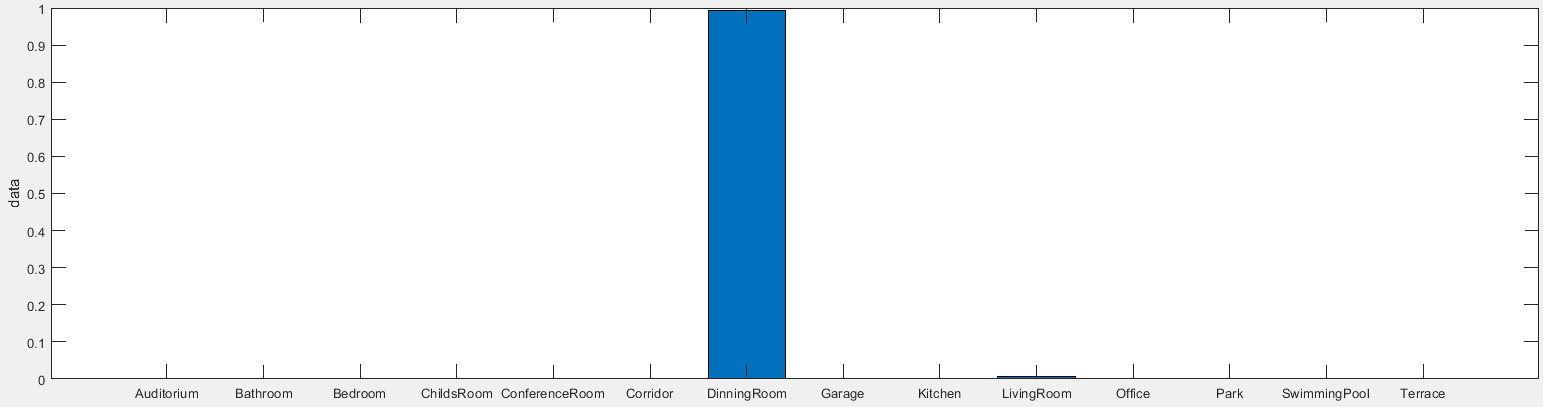


12:51
16.10.2017

Deep Learning

Figure 1

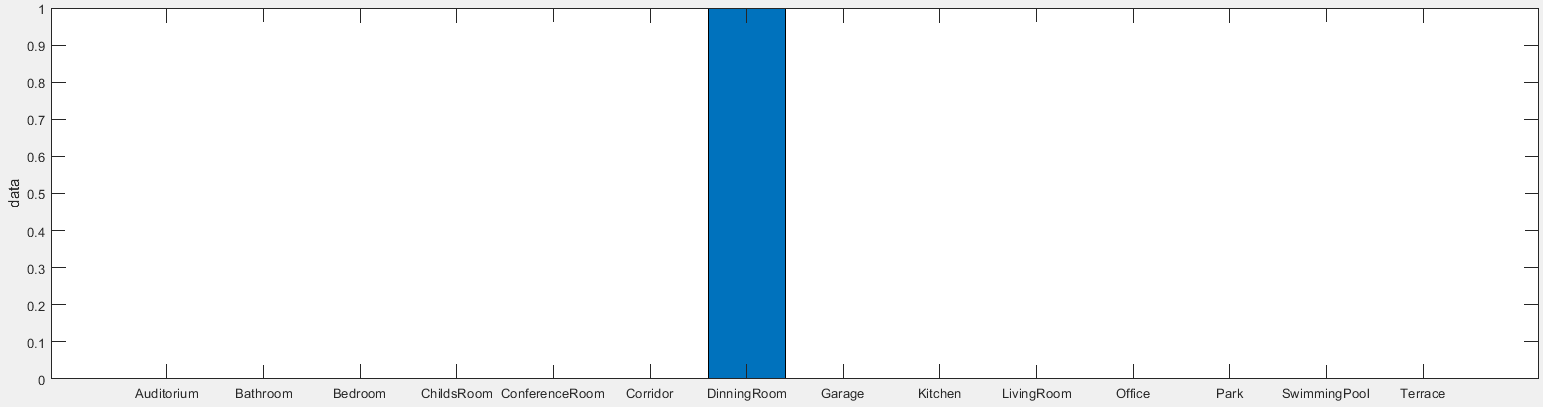
File Edit View Insert Tools Desktop Window Help



Deep Learning

Figure 1

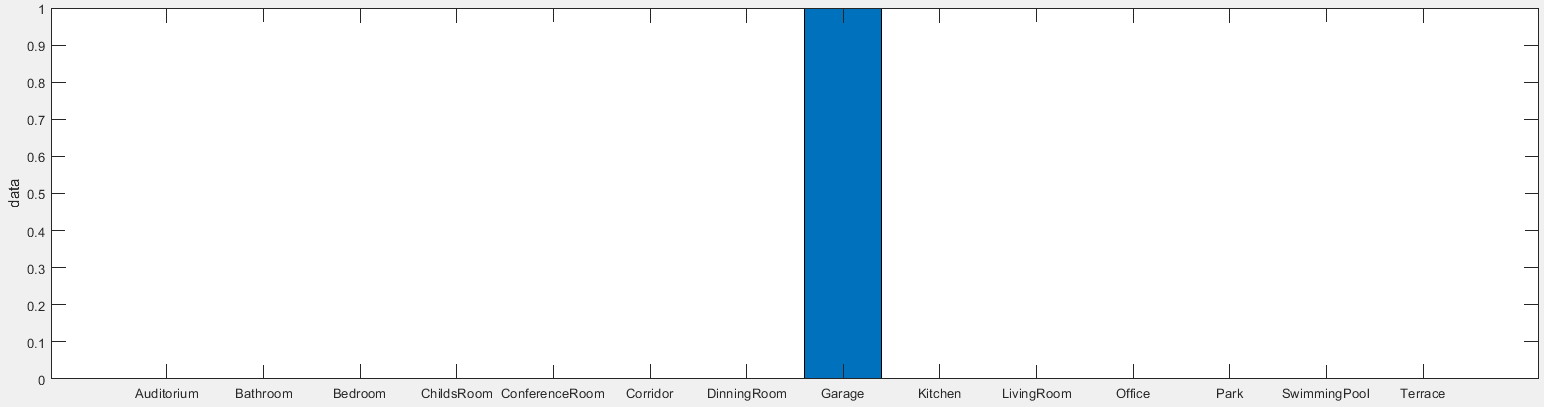
File Edit View Insert Tools Desktop Window Help



Deep Learning

Figure 1

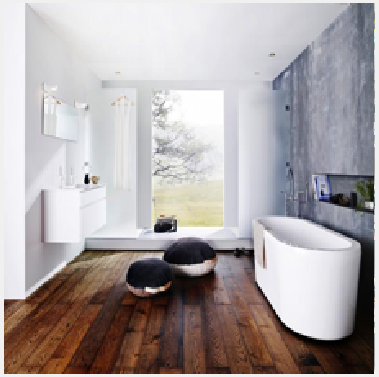
File Edit View Insert Tools Desktop Window Help

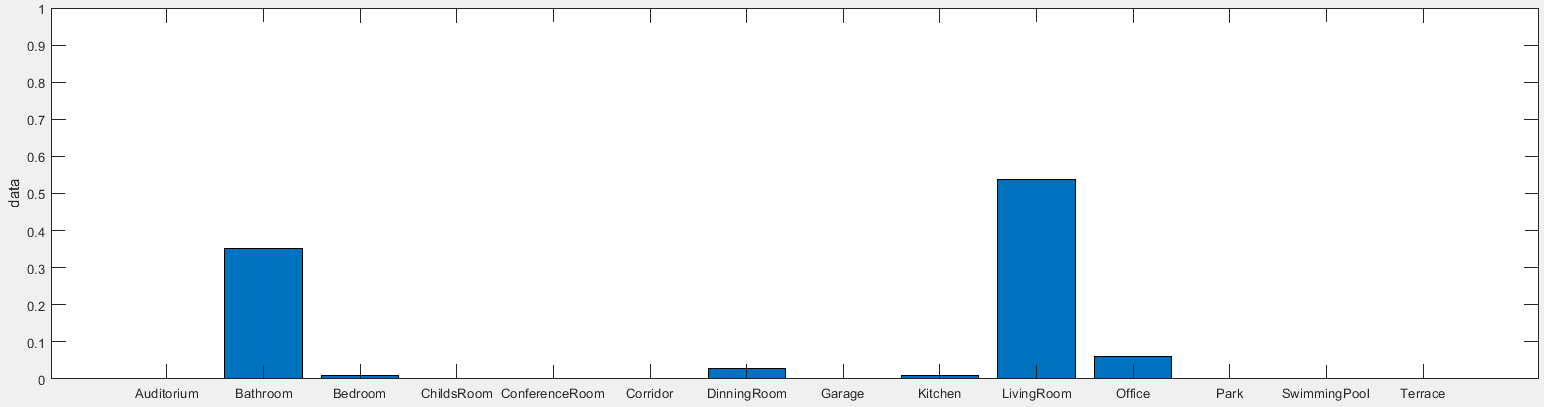


Deep Learning

Figure 1

File Edit View Insert Tools Desktop Window Help

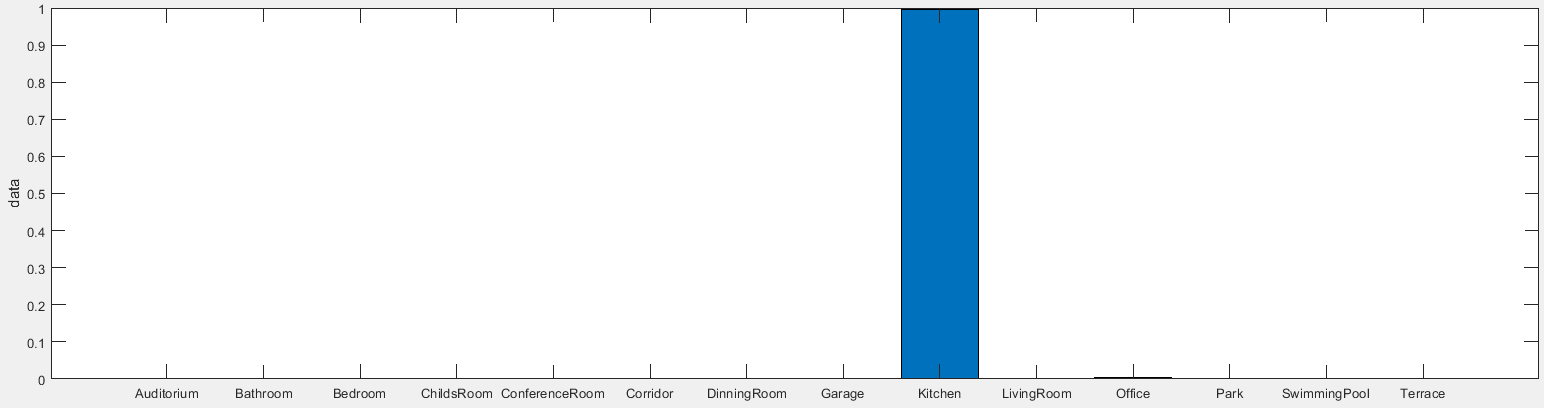




Deep Learning

Figure 1

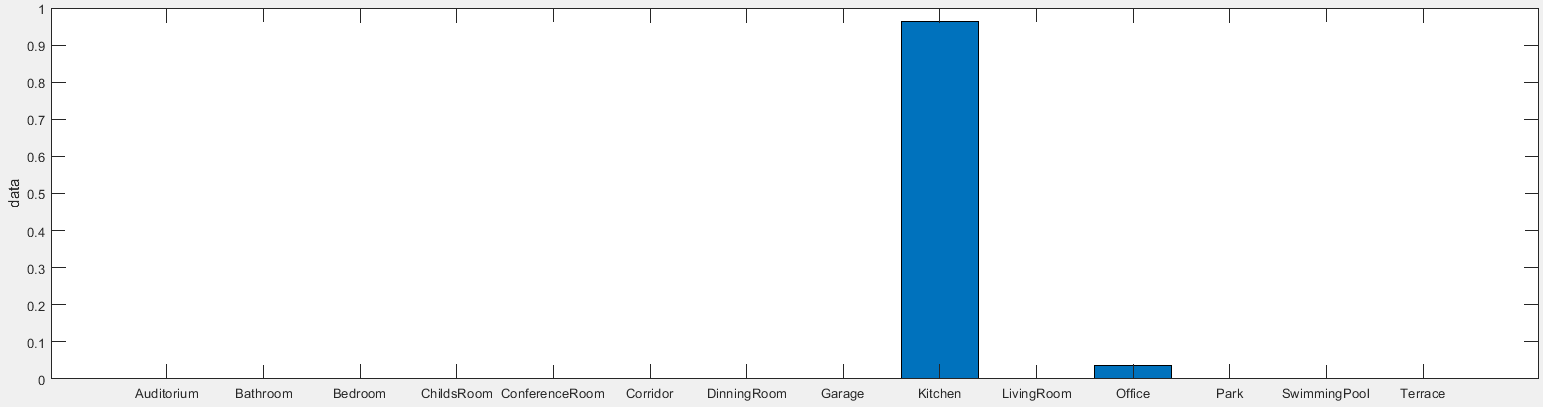
File Edit View Insert Tools Desktop Window Help



Deep Learning

Figure 1

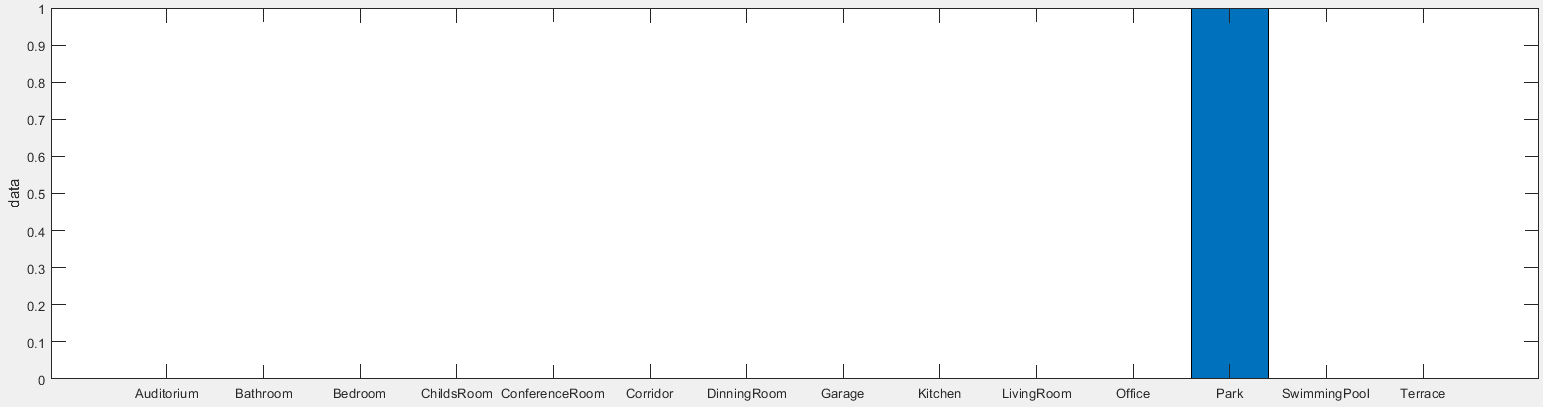
File Edit View Insert Tools Desktop Window Help



Deep Learning

Figure 1

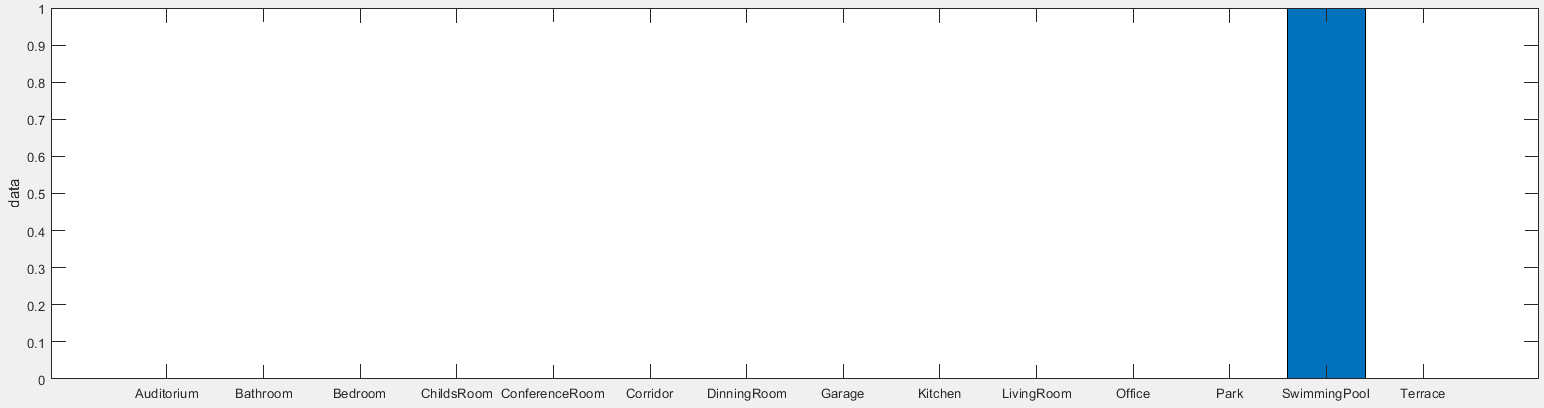
File Edit View Insert Tools Desktop Window Help



Deep Learning

Figure 1

File Edit View Insert Tools Desktop Window Help



12:51
16.10.2017

Deep Learning

Evaluation

Accuracy – Top1 Score: 88%
Accuracy – Top2 Score: 100%

AI Services – Realized since March 2017 (1)

Gender Recognition based on the speaker's voice

Detection of siren sounds in urban environments

Detection of baby scream in home environments

Detection of doorbell sounds in home environments

Environment type recognition based on a single image

Music genre recognition based on sound snipets

Speaker recognition based on large training data and unknown speaker clustering based on voice samples

*Intelligent noise reduction**

*Classification of music, speech and noise based on sound snipets**

* Student work

Overwiew

1. Problem Description
2. Dataset
3. Deep Learning Algorithm
4. Evaluation and Validation
5. Summary

Problem Description

Gender Recognition based on Audio Analysis

Smart speakers are more and more present in our daily environment. Such speakers are able to understand what we are saying and to react properly. Hence it is possible to ask the speaker to provide information like the current weather or to do some work for us like to order products from web shops.

So far there is no speaker on the market which provides the possibility to restrict services to specific users by automatically identifying them by their voice. This work shows a way to do speaker identification by voice based on a large training data set for each speaker.

Obviously for real time applications it is unrealistic to have such data sets. Therefore, it is analyzed how well unknown speakers can be clustered without any additional training data. As a result, the system is able to distinguish between different unknown speakers and can perform a speaker identification by asking for the speakers name as soon as a new unknown speaker was recognized.

Data Set

LibriSpeech

The analysis presented are based on the LibriSpeech data set which is available for free under: http://www.openslr.org/12/

"LibriSpeech is a corpus of approximately 1000 hours of 16kHz read English speech, prepared by Vassil Panayotov with the assistance of Daniel Povey. The data is derived from read audiobooks from the LibriVox project, and has been carefully segmented and aligned."

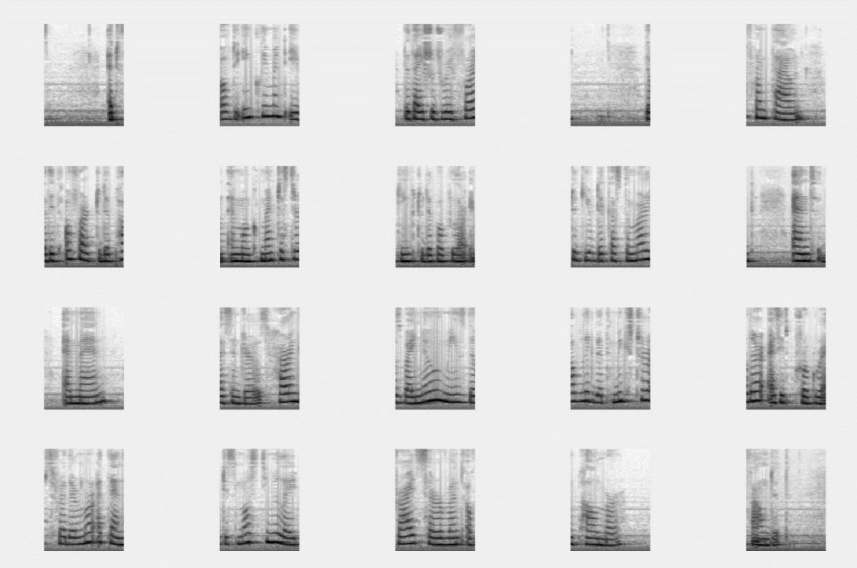
Training / Test Data: 400 different speakers, all in all 184hours, 70% split

Clustering: 5 different speakers (not included in the training / test data set)

Speaker Identification - Features

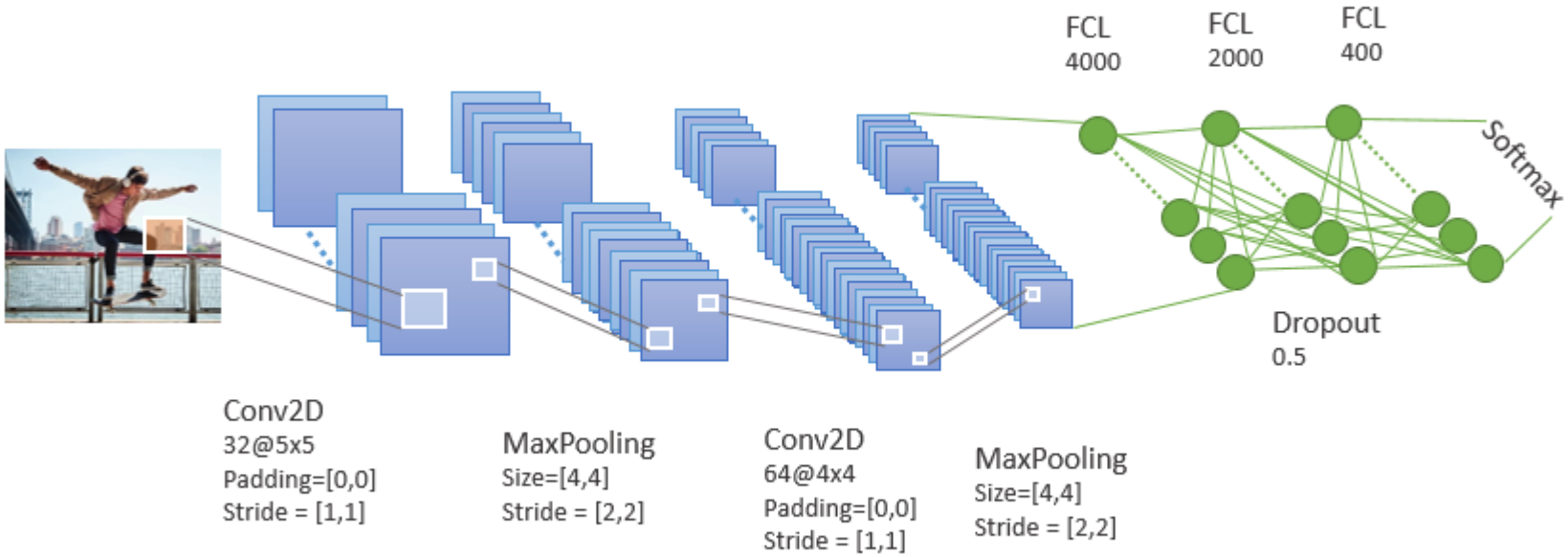
Feature:

As input we use 128bin Mel-spectrogram images based on 16kHz sampling rate, 1024 FFT size, 160 samples hop size, dynamic range compression and one second snippets of non-overlapping spectrogram pieces.



Speaker Identification – CNN

Net Structure:



Deep Learning

Speaker Identification – Training

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
1	1	4.27	6.0305	0.00%	0.0010
1	50	8.87	6.0055	1.56%	0.0010
1	100	13.38	5.8499	0.00%	0.0010
1	150	17.90	5.5162	2.34%	0.0010
1	200	22.45	5.2605	3.91%	0.0010
1	250	26.99	4.7085	10.16%	0.0010
1	300	31.54	3.8035	20.31%	0.0010
1	350	36.11	2.9509	36.72%	0.0010
1	400	40.68	2.4201	41.41%	0.0010
1	450	45.24	2.3105	42.19%	0.0010
1	500	49.81	1.8072	50.00%	0.0010
1	550	54.41	1.7081	60.94%	0.0010
1	600	59.06	1.9858	58.59%	0.0010
1	650	63.76	1.8385	58.59%	0.0010
1	700	68.50	1.7142	60.16%	0.0010
1	750	73.28	1.3474	70.31%	0.0010
1	800	78.08	1.2700	65.63%	0.0010
1	850	82.90	1.3391	63.28%	0.0010
1	900	87.66	1.2418	64.84%	0.0010
1	950	92.45	1.2353	70.31%	0.0010
1	1000	97.29	1.1013	71.09%	0.0010
1	1050	102.17	1.2481	72.66%	0.0010
1	1100	107.01	1.0146	78.13%	0.0010
1	1150	111.78	0.9628	75.78%	0.0010
1	1200	116.65	1.1874	75.78%	0.0010

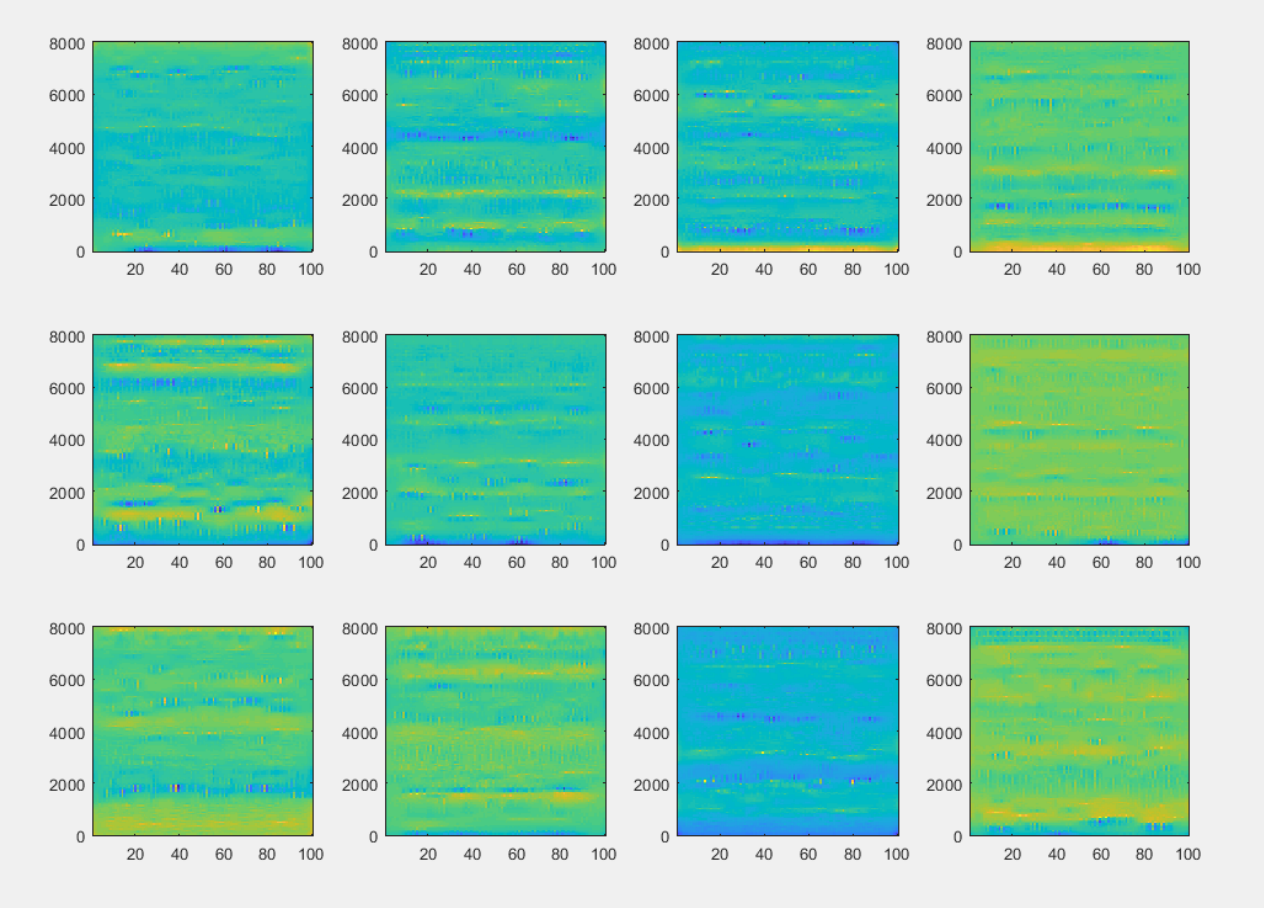
...

5	14950	1469.07	0.0658	97.66%	2.00e-05
5	15000	1474.24	0.0980	96.88%	2.00e-05
5	15050	1479.42	0.0748	97.66%	2.00e-05
5	15100	1484.58	0.1303	95.31%	2.00e-05
5	15150	1489.78	0.0644	96.09%	2.00e-05
5	15200	1494.92	0.0741	96.88%	2.00e-05
5	15250	1500.03	0.0420	99.22%	2.00e-05
5	15300	1505.18	0.1322	96.88%	2.00e-05
5	15350	1510.31	0.0236	99.22%	2.00e-05
5	15400	1515.38	0.0616	98.44%	2.00e-05
5	15450	1520.51	0.1202	96.88%	2.00e-05
5	15500	1525.63	0.0334	98.44%	2.00e-05
5	15550	1530.62	0.0964	96.88%	2.00e-05
5	15600	1535.73	0.0283	98.44%	2.00e-05
5	15625	1538.22	0.0111	100.00%	2.00e-05

Optimizer: SGDM
Epochs: 5
Learn Rate: 0.001
Min Batch Size: 128
LearnRateDrop: Factor=0.02; Period=3

Speaker Identification – Training (2)

Deep Dream Image – High activations for 12 speakers



Speaker Identification – Evaluation

The CNN is evaluated with the test data set introduced (400 speakers, 30% of all data, not included in training – one sample = 1 sec of speech)

The CNN reached an accuracy of **76,50%** for **400** different speaker

Due to the high number of classes the confusion matrix is not shown.

Speaker Clustering

Data Set:

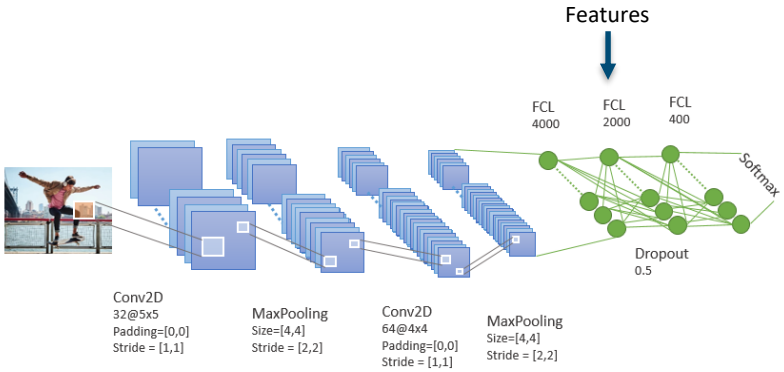
- Take set of different speakers (not included in the training / test data set)

Features:

- Take activations from CNN layer FCL_2 (= 2000 features)

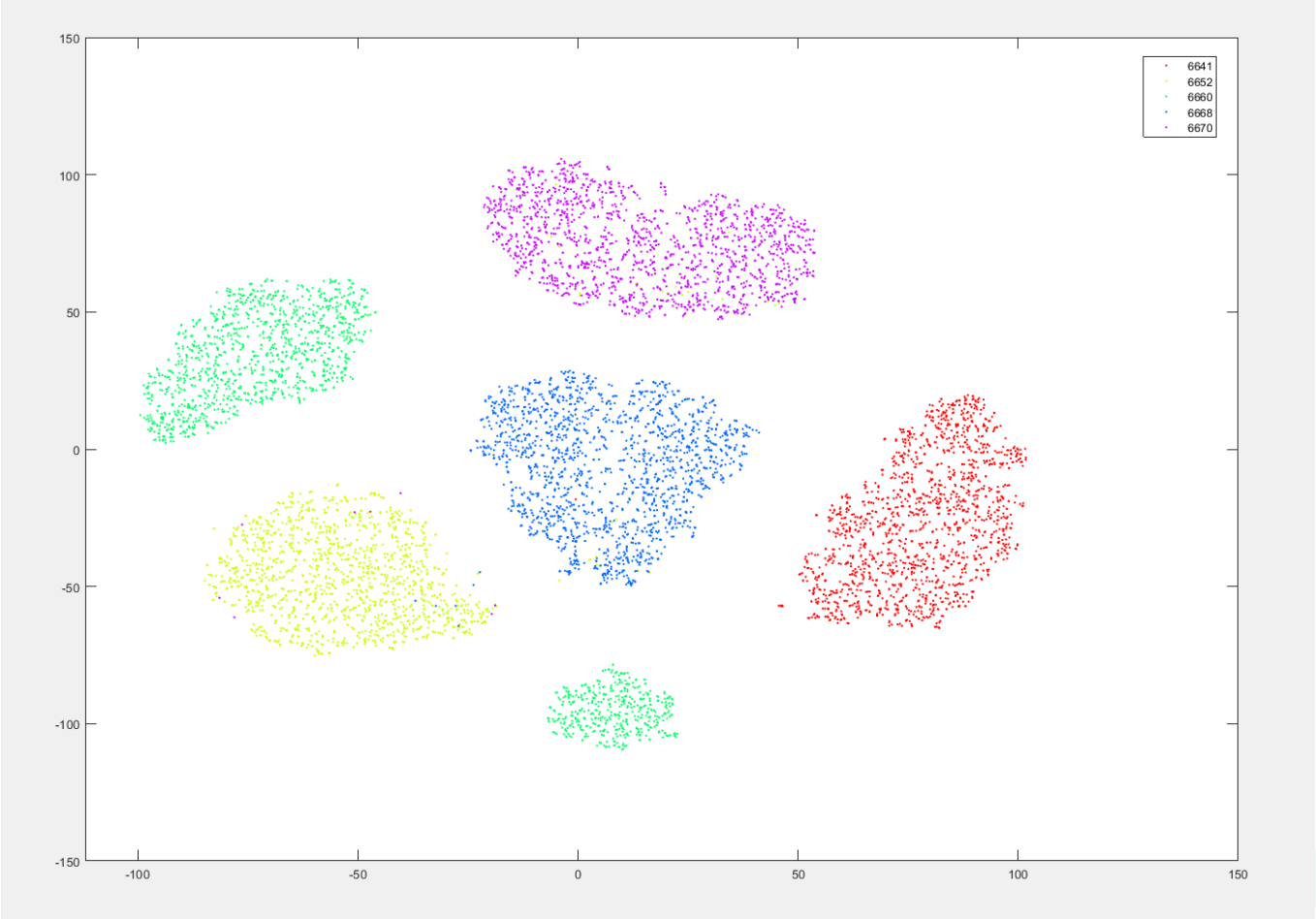
Clustering:

- T-SNE algorithm is used to perform the dimension reduction and the clustering.



Deep Learning

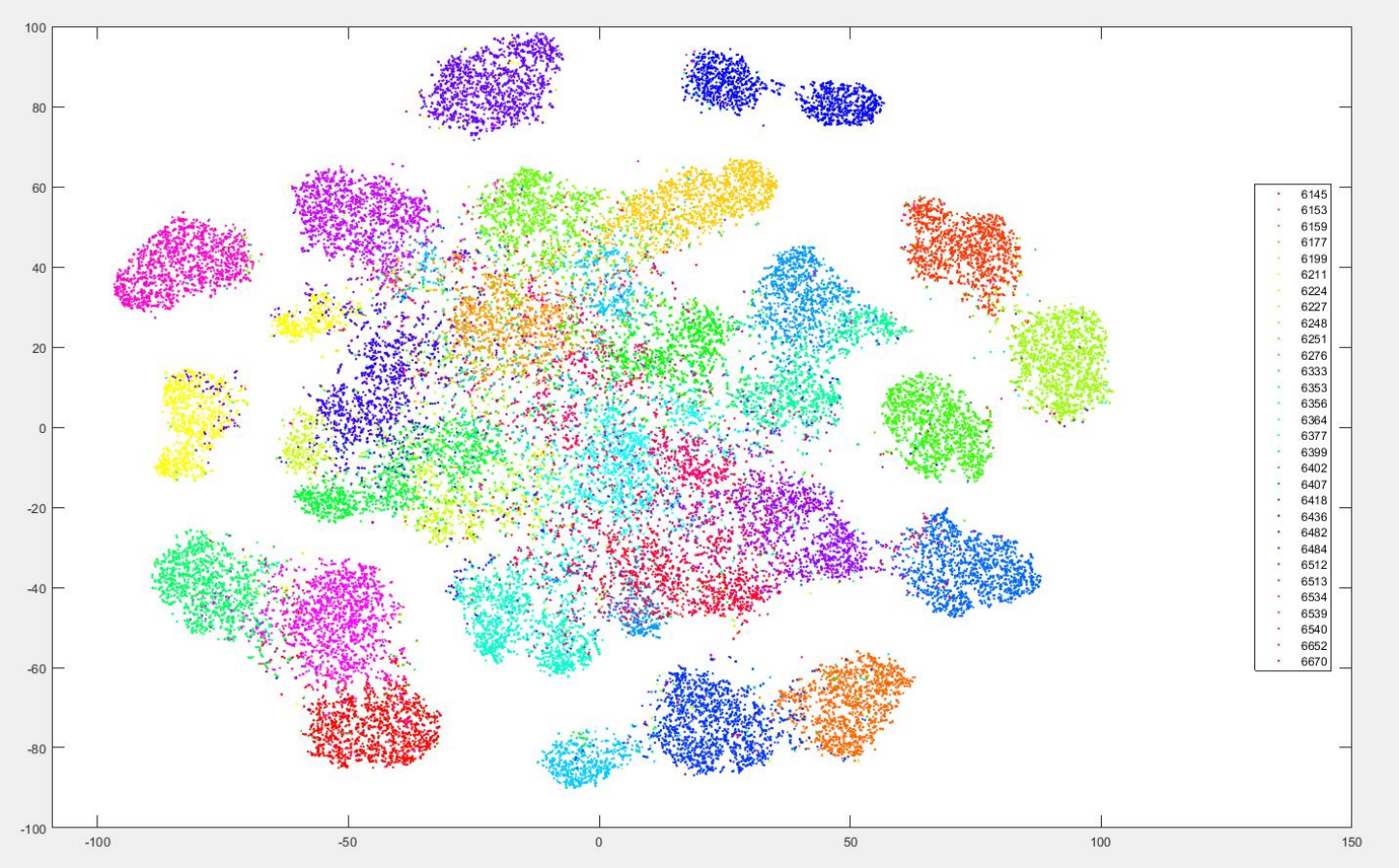
Speaker Clustering – Results for 5 unknown speakers



Error: 1.5751

Deep Learning

Speaker Clustering – Results for 30 unknown speakers



Error: 2.3618

Deep Learning

Outlook

- Try different methods for dimension reduction and clsutering methods in terms of real-time application
- Realize real-time validation system
- Evaluate the impact of noise

AI Services – Realized since March 2017 (1)

Gender Recognition based on the speaker's voice

Detection of siren sounds in urban environments

Detection of baby scream in home environments

Detection of doorbell sounds in home environments

Environment type recognition based on a single image

Music genre recognition based on sound snipets

Speaker recognition based on large training data and unknown speaker clustering based on voice samples

*Intelligent noise reduction**

*Classification of music, speech and noise based on sound snipets**

* Student work

Overwiew

1. Problem Description
2. Dataset
3. Deep Learning Algorithm
4. Evaluation and Validation
5. Summary

Problem Description

Gender Recognition based on Audio Analysis

Smart speakers are more and more present in our daily environment. Such speakers are able to understand what we are saying and to react properly. Hence it is possible to ask the speaker to provide information like the current weather or to do some work for us like to order products from web shops.

Smart speakers will turn form active to pro-active systems soon. This means, that the speaker will recognize what happens within its environment and the device will act autonomously on a specific situation. The most important sensor of a smart speaker to capture the environment is a microphone array.

This work describes the automatic recognition of music genre played by other music players within the close vicinity of a mic array. Based on this information the device is able to recognize the favorite music genres of people and is consequently able to adapt playlists and to propose new songs to purchase automatically.

Besides, the music genre played can be used to help to understand the current mood of the listener which is a valuable information in smart home and in-car applications. Mood recognition opens the door to turn a smart device into a person's "best-friend".

Genre Types

Selection

The following 6 genre types were investigated.

Rock *Electro* *HipHop*

Latino *Classic* *Jazz*

Data Set Generation

For each genre type about 6hours and 30min of music was collected. The data was downloaded from Youtube where Mixes between 30min and 3 hours were chosen.

The data was split into training and test data. The first 70% of each mix was used for training and the remaining 30% for testing.

Training data:	39.489 samples (based on 5sec snippets)
Test data:	16.920 samples (based on 5sec snippets)

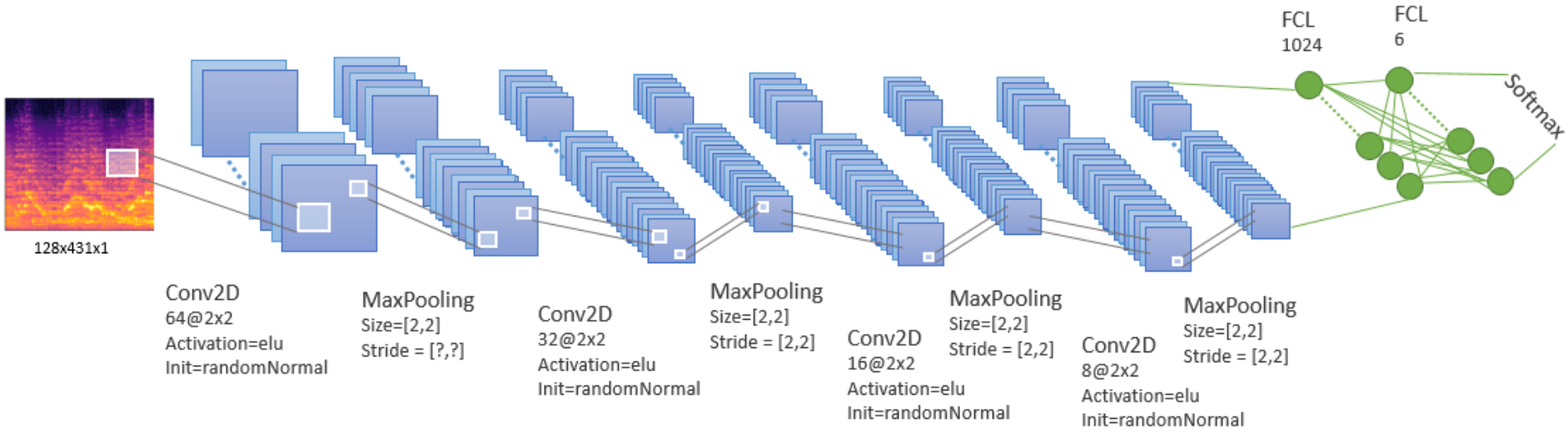
Features

The following features were calculated based on the dataset introduced. Therefore, 5sec segments were used with an overlapp of 50%. The audio data was sampled with 44.1kHz, 16bit.

- Mel Frequency Spectograms (128 bins)
- Chroma Energy Normalized (12 bins)
- Spectral Contrast (7 bins)
- Tempogram Histogram (384 bins)

Deep Learning

Mel Spectograms – CNN Net Structure




```
## Compile model
epochs = 5
lrate = 0.001
Optimizer = Adam(lr=lrate, beta_1=0.9, beta_2=0.999, epsilon=1e-08, decay=0.0)
loss='categorical_crossentropy',
metrics=['accuracy']
```

Evaluation – Mel Spectograms

A convolutional networt approach was used and optimized in terms of net structure and net parameters.

	Precision	Recall	F1
Classic	0.89	0.88	0.88
Electro	0.77	0.77	0.77
Rock	0.66	0.83	0.73
HipHop	0.83	0.74	0.78
Jazz	0.62	0.66	0.64
Latino	0.95	0.75	0.84

Evaluation – Mel Spectograms (2)

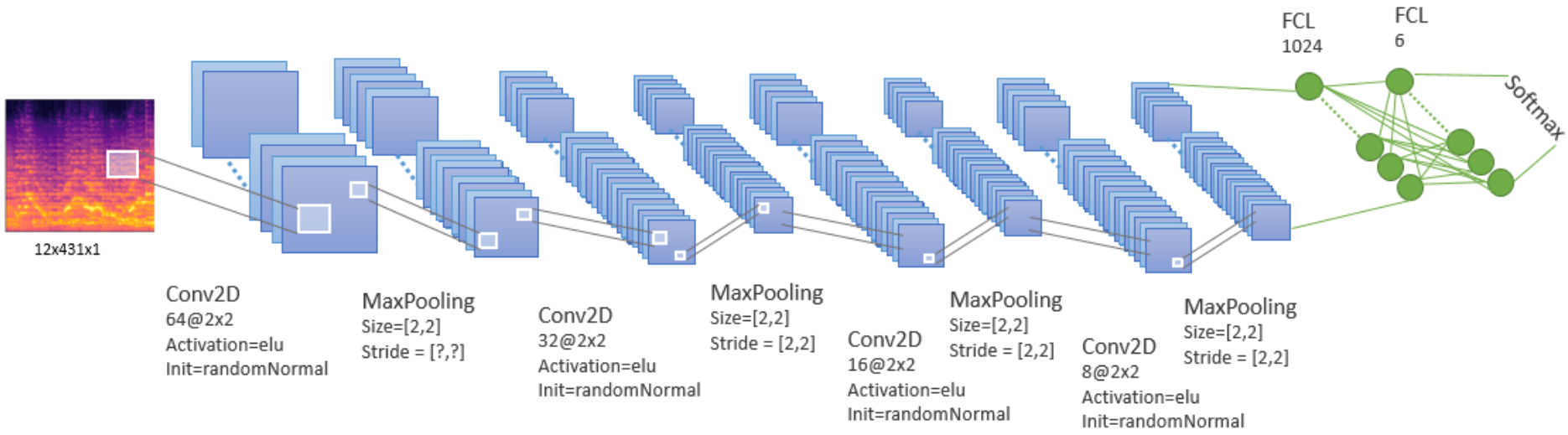
A convolutional network approach was used and optimized in terms of net structure and net parameters.

	Classic	*Electro*	*Rock*	*HipHop*	*Jazz*	*Latino*
Classic	2491	39	15	0	295	3
Electro	29	2267	386	79	159	18
Rock	103	126	2273	8	222	1
HipHop	22	195	324	2097	203	10
Jazz	149	179	368	174	1868	76
Latino	6	134	103	161	277	2060

Accuracy: 77.16%

Deep Learning

Chromagram – CNN Net Structure



Compile model
epochs = 10
lrate = 0.001
Optimizer = Adam(lr=lrate, beta_1=0.9, beta_2=0.999, epsilon=1e-08, decay=0.0)
loss='categorical_crossentropy',
metrics=['accuracy']

Evaluation – Chromagram

A convolutional network approach was used and optimized in terms of net structure and net parameters.

	Precision	Recall	F1
Classic	0.77	0.64	0,70
Electro	0,68	0,63	0,65
Rock	0,59	0,70	0,64
HipHop	0,63	0,72	0,67
Jazz	0,38	0,34	0,36
Latino	0,62	0,62	0,62

Deep Learning

Evaluation – Chromagram (2)

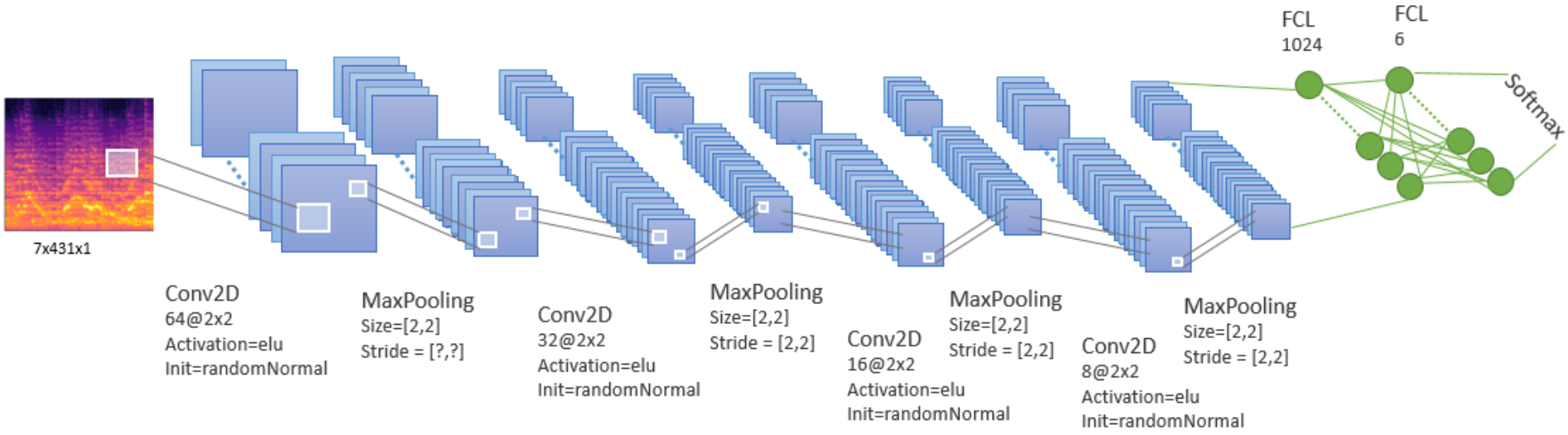
A convolutional network approach was used and optimized in terms of net structure and net parameters.

	Classic	*Electro*	*Rock*	*HipHop*	*Jazz*	*Latino*
Classic	1828	101	281	8	498	127
Electro	210	1855	289	184	222	178
Rock	62	254	1906	90	327	94
HipHop	14	312	274	2052	117	82
Jazz	225	154	336	583	959	557
Latino	33	63	149	353	432	1711

Accuracy: 60.94%

Deep Learning

Spectral Contrast – CNN Net Structure



Compile model
epochs = 5
lrate = 0.001
Optimizer = Adam(lr=lrate, beta_1=0.9, beta_2=0.999, epsilon=1e-08, decay=0.0)
loss='categorical_crossentropy',
metrics=['accuracy']

Evaluation – Spectral Contrast

A convolutional network approach was used and optimized in terms of net structure and net parameters.

	Precision	Recall	F1
Classic	0.92	0,89	0,91
Electro	0,91	0,41	0,56
Rock	0,46	0,85	0,60
HipHop	0,84	0,66	0,74
Jazz	0,57	0,65	0,61
Latino	0,92	0,79	0,85

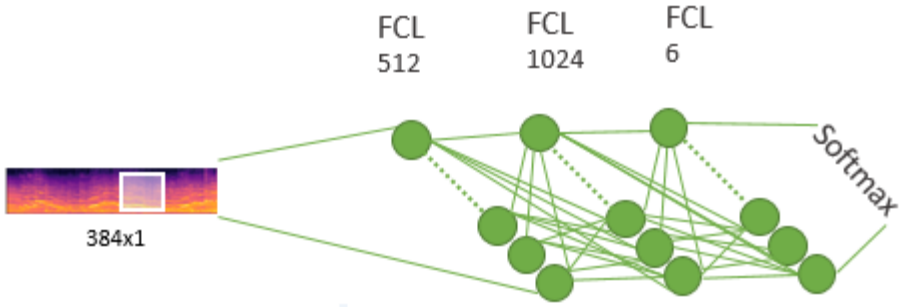
Evaluation – Spectral Contrast (2)

A convolutional network approach was used and optimized in terms of net structure and net parameters.

	Classic	*Electro*	*Rock*	*HipHop*	*Jazz*	*Latino*
Classic	2532	12	44	0	255	0
Electro	25	1196	1270	181	212	54
Rock	18	15	2334	12	346	8
HipHop	16	31	540	1876	363	25
Jazz	142	49	560	127	1842	94
Latino	6	18	279	37	226	2175

Accuracy: 70.66%

Tempogram Histogram – CNN Net Structure




```
## Compile model
epochs = 15
lrate = 0.001
Optimizer = Adam(lr=lrate, beta_1=0.9, beta_2=0.999, epsilon=1e-08, decay=0.0)
loss='categorical_crossentropy',
metrics=['accuracy']
```

Evaluation – Tempogram Histogram

A convolutional network approach was used and optimized in terms of net structure and net parameters.

	Precision	Recall	F1
Classic	0,51	0,52	0,52
Electro	0,92	0,71	0,8
Rock	0,40	0,51	0,45
HipHop	0,74	0,86	0,80
Jazz	0,50	0,35	0,41
Latino	0,80	0,88	0,84

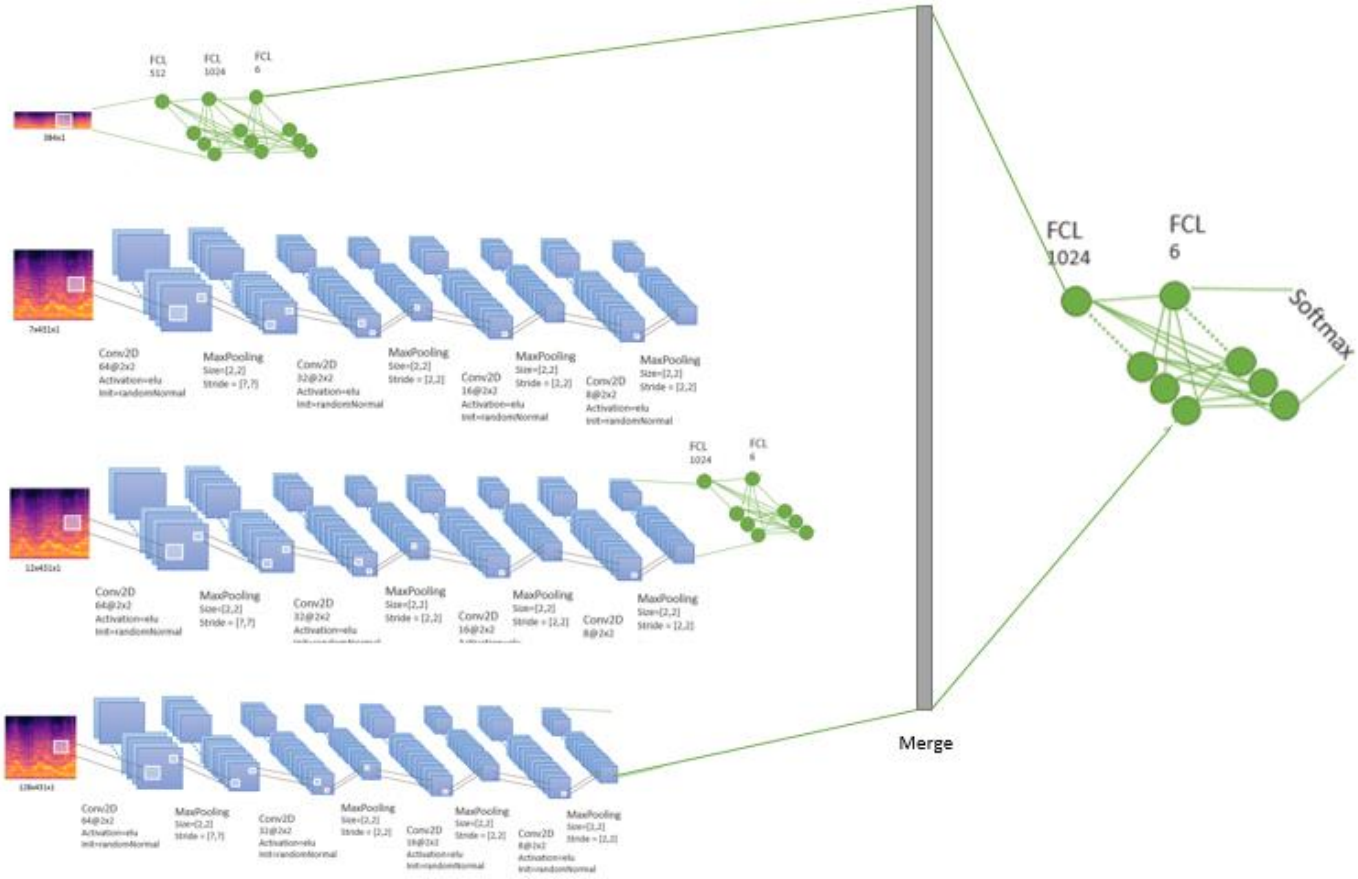
Evaluation – Tempogram Histogram (2)

A convolutional network approach was used and optimized in terms of net structure and net parameters.

	Classic	*Electro*	*Rock*	*HipHop*	*Jazz*	*Latino*
Classic	1483	4	879	89	325	63
Electro	268	2092	173	129	150	126
Rock	796	97	1399	78	329	34
HipHop	30	0	139	2449	135	98
Jazz	280	70	763	428	995	278
Latino	32	0	107	137	56	2409

Accuracy: 63.99%

Combined Features – CNN Net Structure



Compile model
epochs = 25
lrate = 0.001
Optimizer = Adam(lr=lrate, beta_1=0.9, beta_2=0.999, epsilon=1e-08, decay=0.0)
loss='categorical_crossentropy',
metrics=['accuracy']

Deep Learning

Evaluation – ALL

A convolutional network approach was used and optimized in terms of net structure and net parameters.

	Precision	Recall	F1
Classic	0.95	0,89	0,92
Electro	0,81	0,90	0,85
Rock	0,74	0,89	0,81
HipHop	0,92	0,87	0,89
Jazz	0,79	0,62	0,69
Latino	0,89	0,92	0,90

Deep Learning

Evaluation – ALL (2)

A convolutional network approach was used and optimized in terms of net structure and net parameters.

	Classic	*Electro*	*Rock*	*HipHop*	*Jazz*	*Latino*
Classic	2530	24	40	0	215	34
Electro	24	2641	133	30	56	54
Rock	19	159	2420	7	99	29
HipHop	9	76	175	2469	48	74
Jazz	60	302	424	154	1746	128
Latino	11	39	84	33	49	2525

Accuracy: 84.70%

Evaluation – Summary

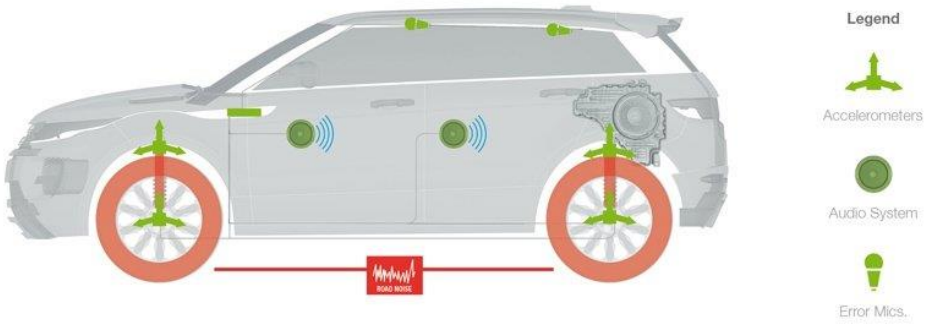
System	Accuracy
CNN – MelSpec	77,16
CNN – Chromagram	60,94
CNN – Spectral Contrast	70,66
CNN – Tempogram Histogram	63,99
Multi-CNN	**84,70**

Next Steps

The next step will be

- to implement a online verification system and to perform an exhaustive performance evaluation.
- To evaluate feature subsets
- To include more genres
- To evaluate LSTMs

AI Services – Realized since March 2017 (II)



Automatic road surface recognition based on car mounted motion sensors

ROAD SURFACE DETECTION

COC AUDIO

HARMAN

Road Surface Detection

The target is to recognize the road surface a car is currently driving on. This information is valuable for several safety applications (e.g. optimize break behavior), automatic road map creation / road condition monitoring and specific applications such as RNC (road noise cancelation).

The following road surface types will be considered:

- Rough asphalt
- Cobblestone
- Normal
- Concrete

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

The idea is to use sensors mounted on the car which are able to detect the road vibrations as these seem to be characteristic for the road surfaces considered.

Hence, high accurate PCB (3 axis) sensors are mounted on the car. The positions are taken from investigations made during the RNC project.

As a first step only a single PCB sensor is used for the evaluations. The position which seems to be most reasonable is: FRONT WHEEL HUB LEFT.

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

The basic idea is that the vibrations of the road are characteristic for their specific type. Consequently acceleration data is used to calculate features which serve as input for machine learning models.

The following features are calcualted based on 4000Hz sampling rate, a sliding window of 4000 samples with an overlapp of 2000 samples.

- Exponential Band (10 features)
- Cepstral Coefficents (10 features)
- Spectral Entropy (1 features)
- Energy (1 feature)
- Mean + Variance (2 features)

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

X-Axis:
#1 - #10: Spectrum in exp bands
#11 - #20: Cepstral Coeficients
#21: Spectral Entropy
#22: Energy
#23: Mean
#24: Variance
#25: Empty

Y-Axis:
#26 - #35: Spectrum in exp bands
#36 - #45: Cepstral Coeficients
#46: Spectral Entropy
#47: Energy
#48: Mean
#49: Variance
#50: Empty

Z-Axis:
#51 - #60: Spectrum in exp bands
#61 - #70: Cepstral Coeficients
#71: Spectral Entropy
#72: Energy
#73: Mean
#74: Variance
#75: Empty

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

In a frist step available data from a Daimler Maybach are used.

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

The data container consits of the following recordings:

Concrete:
Betonkreisel_30kmh (240000 samples)
Betonkreisel_40kmh (240000 samples)
Betonkreisel_50kmh (240000 samples)

Cobblestone:
Kopfsteinkreisel_30kmh (240000 samples)
Kopfsteinkreisel_40kmh (240000 samples)
Kopfsteinkreisel_50km (240000 samples)

Normal:
Ostgerade_60kmk (201600 samples)
Ostgerade_80kmh (142400 samples)
Ostgerade_100kmh (119200 samples)
Ostgerade_120kmh (55200 samples)

Rough:
Westgerade_50kmh (196800 samples)
Westgerade_60kmh (152800 samples)
Westgerade_80kmh (118400 samples)
Westgerade_100kmh (90400 samples)
Westgerade_120kmh (65600 samples)

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Note:

(1) Data are ok for a first try. However, the data basis is not big enough to evaluate the behavior of other than the recorded road surface types (only a single road surface was recorded for each type)
(2) No road was recorded twice with the same velocity.

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

1. Feature Ranking and Selection
2. Simple Machine Learning Models for DSP realization – kNN / Decision Tree
3. Evaluation using 10fold Cross-Validation
4. Evaluation with separate Training and Test Data

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Done with Rapid Miner.

Information Gain – Top 5

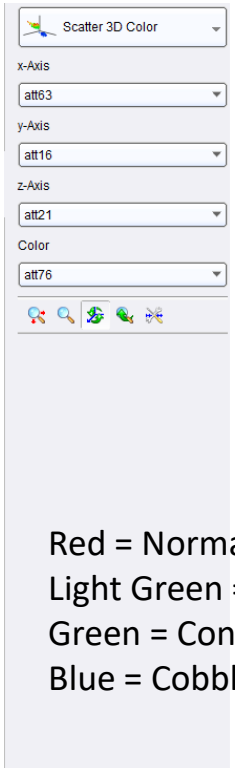
- #63: Z-Achse CC 3
- #16: X-Achse CC 6
- #12: X-Achse CC 2
- #21: X-Achse Spectral Entropy
- #62: Z-Achse CC 2

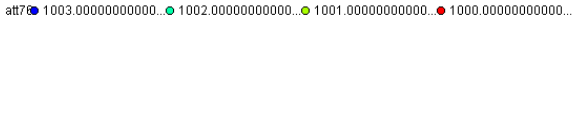
©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

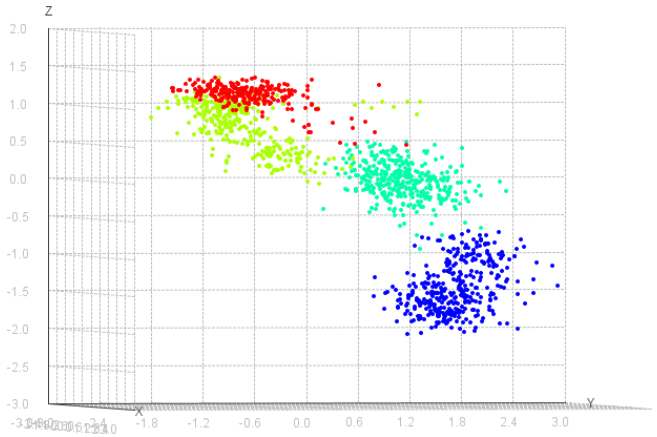
Use the best three features

(#63,#16,#21)





Red = Normal
Light Green = Rau
Green = Concrete
Blue = Cobblestone

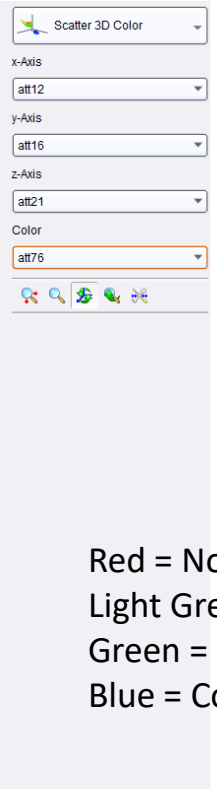


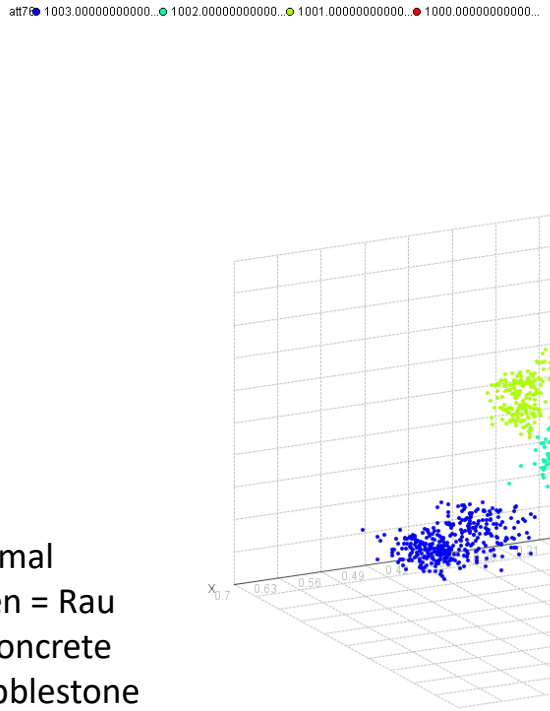
©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

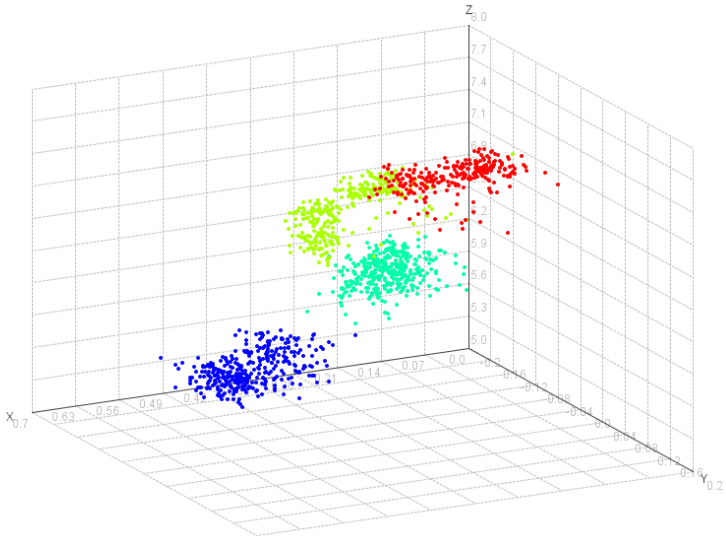
Use one axis only:

X-axis – 3 features
(#12,#16,#21)





Red = Normal
Light Green = Rau
Green = Concrete
Blue = Cobblestone

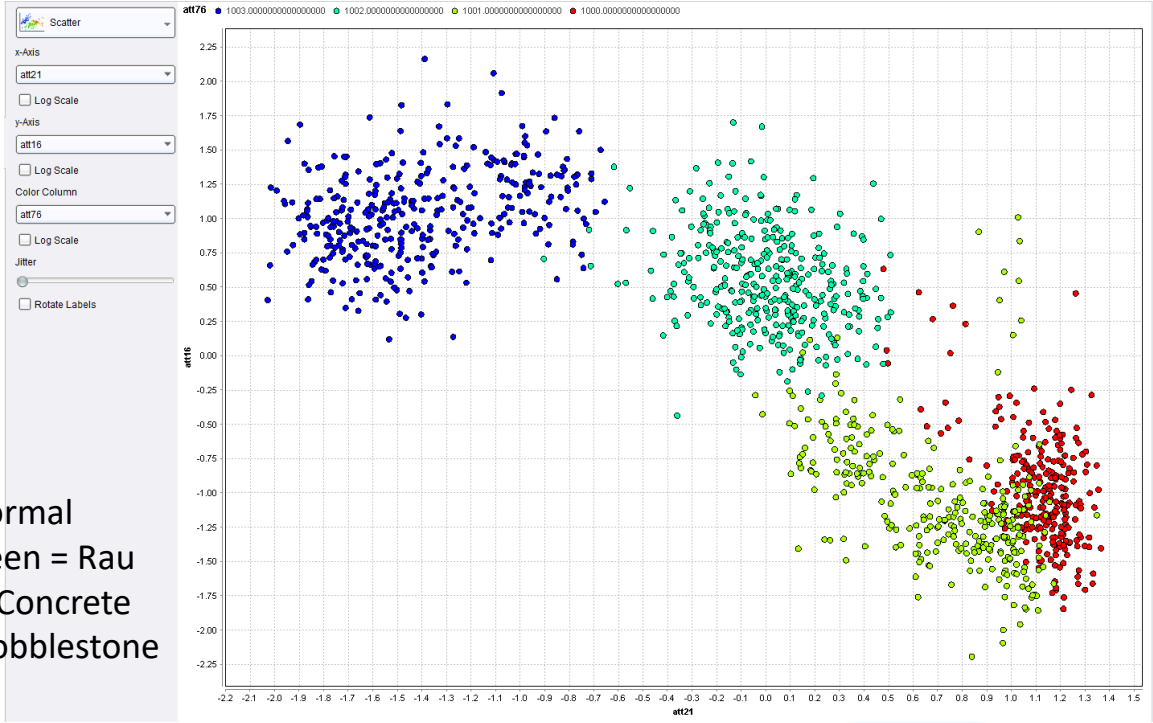


©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Use two features

(#16, #21)



Red = Normal
Light Green = Rau
Green = Concrete
Blue = Cobblestone

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features: #12,#16,#21
Model: kNN (k=10)
10-fold Cross Validation

1000 = Normal
1001= Rau
1002 = Concrete
1003 = Cobblestone

accuracy: 96.21% +/- 1.69% (mikro: 96.21%)

	true 1003.000000000000	true 1002.000000000000	true 1001.000000000000	true 1000.000000000000	class precision
pred. 1003.000000000000	354	1	0	0	99.72%
pred. 1002.000000000000	0	350	5	1	98.31%
pred. 1001.000000000000	0	3	284	22	91.91%
pred. 1000.000000000000	0	0	16	230	93.50%
class recall	100.00%	98.87%	93.11%	90.91%	

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features: #12,#16,#21
Model: DecisionTree (max depth=4)
10-fold Cross Validation

1000 = Normal
1001= Rau
1002 = Concrete
1003 = Cobblestone

accuracy: 93.95% +/- 6.63% (mikro: 94.00%)

	true 1003.0000000000000	true 1002.0000000000000	true 1001.0000000000000	true 1000.0000000000000	class precision
pred. 1003.0000000000000	353	1	0	0	99.72%
pred. 1002.0000000000000	1	348	13	2	95.60%
pred. 1001.0000000000000	0	5	259	21	90.88%
pred. 1000.0000000000000	0	0	33	230	87.45%
class recall	99.72%	98.31%	84.92%	90.91%	

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features: #16,#21
Model: kNN (k=10)
10-fold Cross Validation

1000 = Normal
1001= Rau
1002 = Concrete
1003 = Cobblestone

accuracy: 92.89% +/- 2.10% (mikro: 92.89%)

	true 1003.000000000000(	true 1002.000000000000(	true 1001.000000000000(	true 1000.000000000000(	class precision
pred. 1003.000000000000(	351	4	0	0	98.87%
pred. 1002.000000000000(	3	344	3	3	97.45%
pred. 1001.000000000000(	0	4	271	40	86.03%
pred. 1000.000000000000(	0	2	31	210	86.42%
class recall	99.15%	97.18%	88.85%	83.00%	

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features: #16,#21
Model: DecisionTree (max depth=4)
10-fold Cross Validation

1000 = Normal
1001= Rau
1002 = Concrete
1003 = Cobblestone

accuracy: 79.06% +/- 4.50% (mikro: 79.07%)

	true 1003.0000000000000	true 1002.0000000000000	true 1001.0000000000000	true 1000.0000000000000	class precision
pred. 1003.000000000000	354	3	0	0	99.16%
pred. 1002.000000000000	0	346	5	3	97.74%
pred. 1001.000000000000	0	5	300	249	54.15%
pred. 1000.000000000000	0	0	0	1	100.00%
class recall	100.00%	97.74%	98.36%	0.40%	

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

10fold Cross Validation showed that the recognition problem considered can be solved with an accuracy of about 96% using a simlpe kNN classifier. In a next step, the data will be splitted into a training set and a dedicated test set (orange):

Concrete:
Betonkreisel_30kmh (240000 samples)
Betonkreisel_40kmh (240000 samples)
Betonkreisel_50kmh (240000 samples)

Cobblestone:
Kopfsteinkreisel_30kmh (240000 samples)
Kopfsteinkreisel_40kmh (240000 samples)
Kopfsteinkreisel_50km (240000 samples)

Normal:
Ostgerade_60kmh (201600 samples)
Ostgerade_80kmh (142400 samples)
Ostgerade_100kmh (119200 samples)
Ostgerade_120kmh (55200 samples)

Rough:
Westgerade_50kmh (196800 samples)
Westgerade_60kmh (152800 samples)
Westgerade_80kmh (118400 samples)
Westgerade_100kmh (90400 samples)
Westgerade_120kmh (65600 samples)

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features: #12,#16,#21
Model: DecisionTree (max depth=4)
Train und Test Set

1000 = Normal
1001= Rau
1002 = Concrete
1003 = Cobblestone

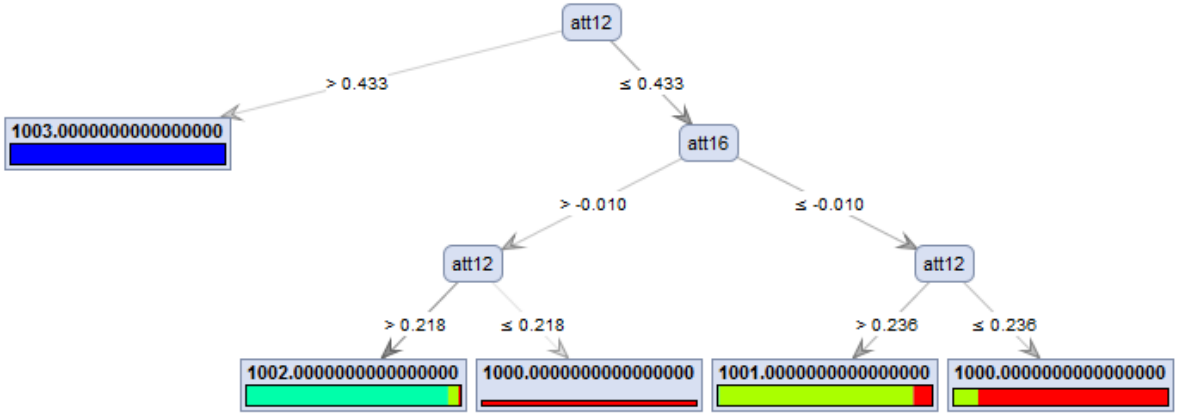
accuracy: 98.90%

	true 1003.0000000000000	true 1002.0000000000000	true 1001.0000000000000	true 1000.0000000000000	class precision
pred. 1003.0000000000000	118	1	0	0	99.16%
pred. 1002.0000000000000	0	117	1	0	99.15%
pred. 1001.0000000000000	0	0	55	0	100.00%
pred. 1000.0000000000000	0	0	2	70	97.22%
class recall	100.00%	99.15%	94.83%	100.00%	

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

1000 = Normal
1001= Rau
1002 = Concrete
1003 = Cobblestone



©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features: #12,#16,#21
Model: kNN (10)
Train und Test Set

1000 = Normal
1001= Rau
1002 = Concrete
1003 = Cobblestone

accuracy: 96.98%

	true 1003.0000000000000	true 1002.0000000000000	true 1001.0000000000000	true 1000.0000000000000	class precision
pred. 1003.0000000000000	118	3	0	0	97.52%
pred. 1002.0000000000000	0	109	0	0	100.00%
pred. 1001.0000000000000	0	6	56	0	90.32%
pred. 1000.0000000000000	0	0	2	70	97.22%
class recall	100.00%	92.37%	96.55%	100.00%	

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

In the following it is investigated if the trained Maybach model can be used for other cars as well without any adaptations.

Normal:
LarkinsRd_32kph_SportMode1 (120.000 samples)

Rough:
KensingtonRd_65kph_SportMode (120.000 samples)
KensingtonRd_80kph_SportMode (120.000 samples)

Concrete:
CabotDr_50kph_SportMode (120.000 samples)
CabotDr_50kph_sportMode1 (120.000 samples)

Cobblestone:
None

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features: #12,#16,#21
Model: kNN (10)
Trained on Maybach, tested on SRT

1000 = Normal
1001= Rau
1002 = Concrete
1003 = Cobblestone

accuracy: 40.69%

	true 1002.0000000000000	true 1001.0000000000000	true 1000.0000000000000	true 1003.0000000000000	class precision
pred. 1002.0000000000000	13	0	0	0	100.00%
pred. 1001.0000000000000	4	105	0	0	96.33%
pred. 1000.0000000000000	0	0	0	0	0.00%
pred. 1003.0000000000000	99	11	58	0	0.00%
class recall	11.21%	90.52%	0.00%	0.00%	

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features: #12,#16,#21
Model: kNN (10) + NORMALIZED Data
Trained on Maybach, tested on SRT

1000 = Normal
1001= Rau
1002 = Concrete
1003 = Cobblestone

accuracy: 51.03%

	true 1002.0000000000000	true 1001.0000000000000	true 1000.0000000000000	true 1003.0000000000000	class precision
pred. 1002.0000000000000	72	0	28	0	72.00%
pred. 1001.0000000000000	9	76	0	0	89.41%
pred. 1000.0000000000000	7	40	0	0	0.00%
pred. 1003.0000000000000	28	0	30	0	0.00%
class recall	62.07%	65.52%	0.00%	0.00%	

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

It can be seen that the Maybach model can not be applied to the SRT as it is. But how well can the SRT be classified alone?

Normal:
LarkinsRd_32kph_SportMode1 (120.000 samples)

Rough:
KensingtonRd_65kph_SportMode (120.000 samples)
KensingtonRd_80kph_SportMode (120.000 samples)

Concrete:
CabotDr_50kph_SportMode (120.000 samples)
CabotDr_50kph_sportMode1 (120.000 samples)

Cobblestone:
None

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features: #12,#16,#21
Model: kNN (k=10)
10 fold Crossvalidation

1000 = Normal
1001= Rau
1002 = Concrete
1003 = Cobblestone

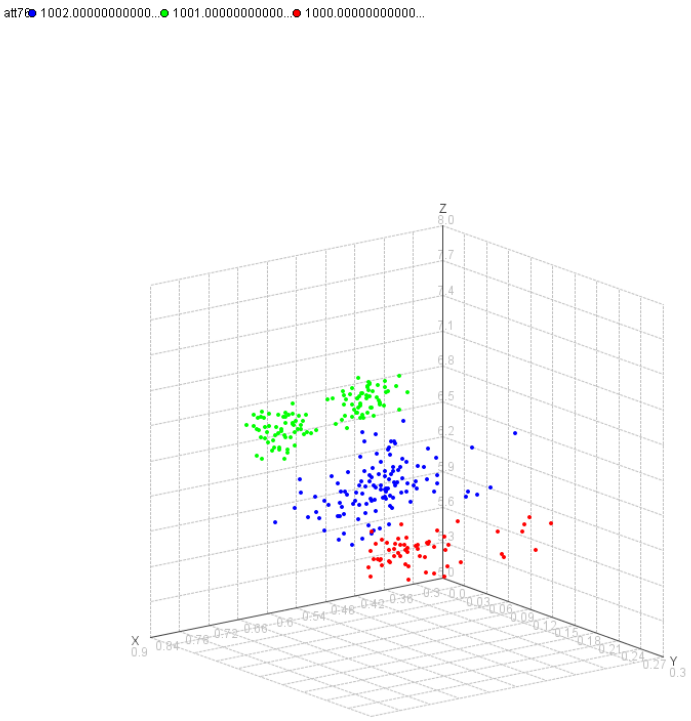
accuracy: 100.00% +/- 0.00% (mikro: 100.00%)

	true 1002.0000000000000000	true 1001.0000000000000000	true 1000.0000000000000000	class precision
pred. 1002.0000000000000000	116	0	0	100.00%
pred. 1001.0000000000000000	0	116	0	100.00%
pred. 1000.0000000000000000	0	0	58	100.00%
class recall	100.00%	100.00%	100.00%	

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features: #12,#16,#21



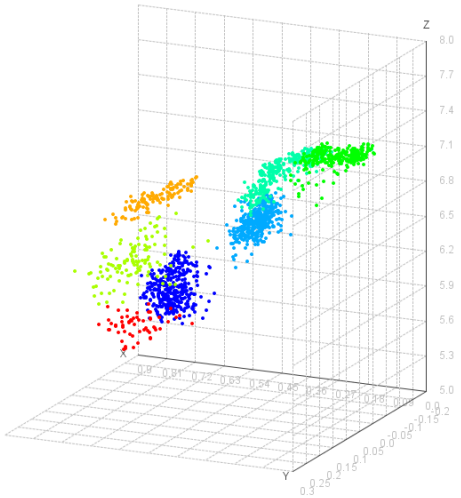
1000 = Normal
1001= Rau
1002 = Concrete
1003 = Cobblestone

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features: #12,#16,#21

1000 / 2000 = Normal
1001 / 2001= Rau
1002 / 2002 = Concrete
1003 / 2003 = Cobblestone



©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Bad performance because of different car or different roads?

Normal:
None

Rough:
Harman_30kmh
Harman_50kmh
Bruecke_SR_60kmh

Concrete:
None

Cobblestone:
None

Features: #12,#16,#21
Model: kNN (10) + NORMALIZED Data
Trained on Maybach, tested on SRT

accuracy: 0.00%

	true 1001.000000000000	true 1003.000000000000	true 1002.000000000000	true 1000.000000000000	class precision
pred. 1001.000000000000	0	0	0	0	0.00%
pred. 1003.000000000000	152	0	0	0	0.00%
pred. 1002.000000000000	2	0	0	0	0.00%
pred. 1000.000000000000	0	0	0	0	0.00%
class recall	0.00%	0.00%	0.00%	0.00%	

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Bad performance because of different car or different roads?

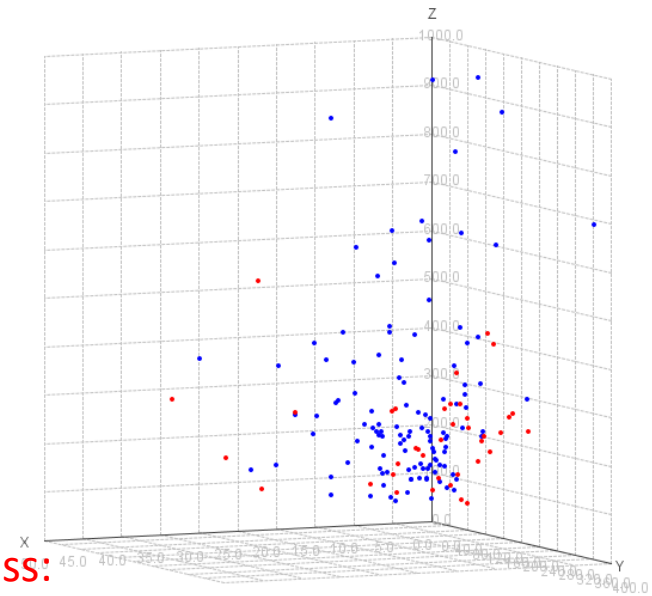
Normal:
None

Rough:
Harman_30kmh
Harman_50kmh
Bruecke_SR_60kmh

Concrete:
None

Cobblestone:
None

Features: #12,#16,#21



First Guess:
Bad performance because of different car!

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

What happens if data is transferred to the ENVELOP domain? Will it be possible to use a trained model for car X without any modification for car Y?

1) Will relevant features change?

att63	1
att21	0.882
att12	0.871
att37	0.868
att23	0.804
att13	0.787
att56	0.786
att6	0.769
att62	0.764
att46	0.751

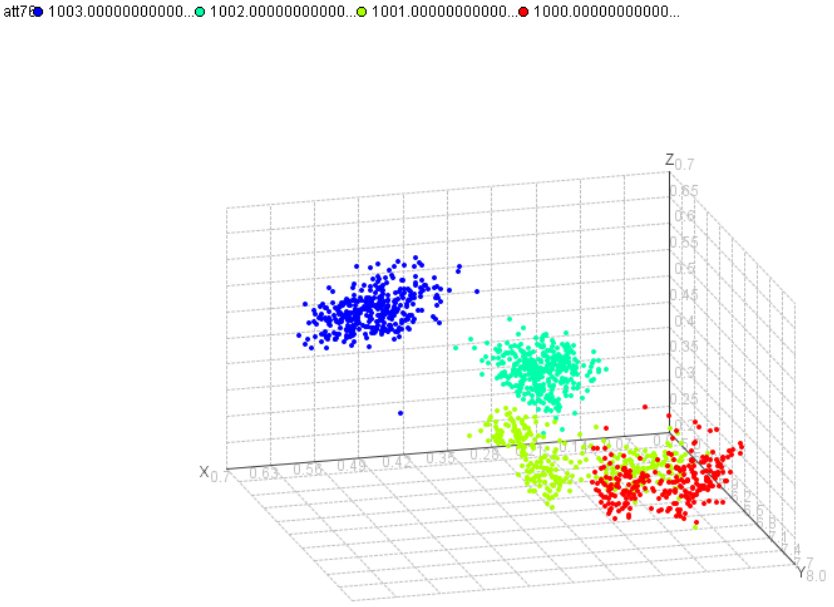
Top 5 (Original)

- #63: Z-Achse CC 3
- **#16: X-Achse CC 6 – missing!**
- #12: X-Achse CC 2
- #21: X-Achse Spectral Entropy
- #62: Z-Achse CC 2 – low ranked

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

What happens if data is transferred to the ENVELOP domain? Will it be possible to use a trained model for car X without any modification for car Y?



Features: #63, #21, #12

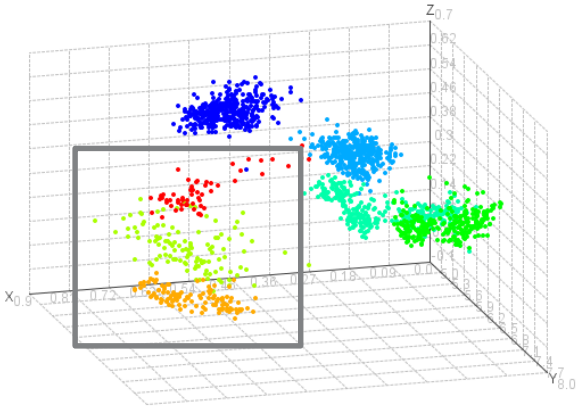
©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

What happens if data is transferred to the ENVELOP domain? Will it be possible to use a trained model for car X without any modification for car Y?

2) Compare data with SRT

SRT Data still separated!

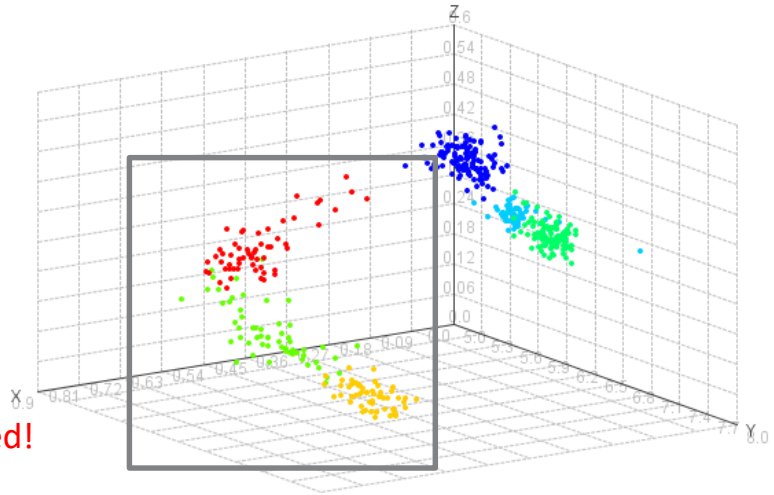


©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

What happens if data is normalized for each car?

#12,#63,#21



SRT Data still separated!

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

The investigations showed that the recognition of road surfaces is very promising if the model is trained for a specific car. The trained model for car x can not be applied to car y (if the sensor is mounted on the position considered).

Data normalization and transformation into Envelop domain does not improve the portability of trained models from one car to another.

It seems that different roads having the same road surface will have the same impact on the features introduced.

However, the data basis is very small and restricted and the results can be seen as a first try only!

To get a reliable result much more data must be recorded including different roads having the same road surface type.

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Daimler GL data set includes data from different roads having the same road type. The following investigations should show the impact of different roads having the same road type on the classification.

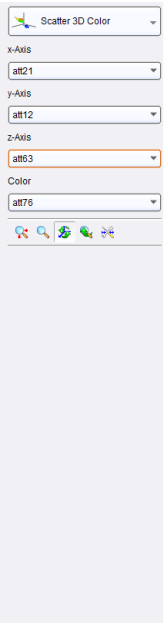
©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

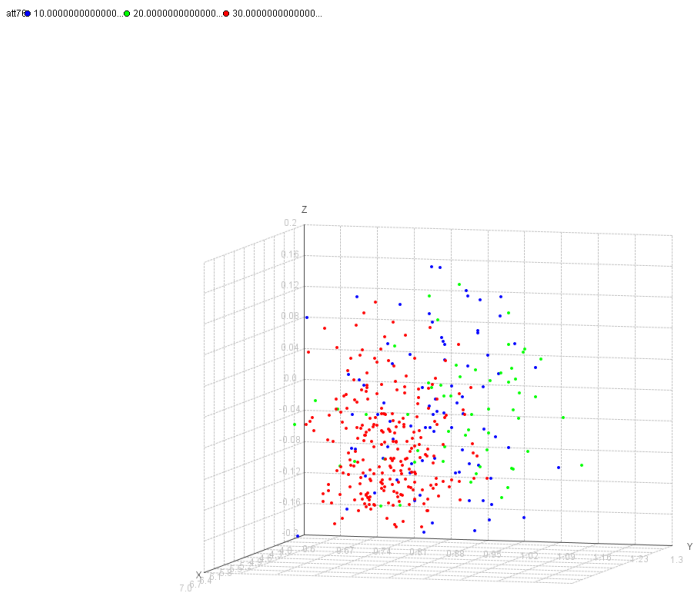
Road Surface Detection

Features: #12,#63,#21

Cobblestone: 3 different roads:

- Bogen (blue)
- Hunderdorf (green)
- Kopfsteinkreisverkehr (red)





DATA IS NOT SEPARABLE

→ Different roads show similar behavior

→ No impact on the classifier is expected!

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Data Set GL

Cobblestone: 3 different roads:

- Bogen (3x)
- Hunderdorf (2x)
- Kopfsteinkreisverkehr (2)

Rough: 3 different roads:

- Westgerade (4x)
- Puchhof – Oifing (5x)

Beton: 2 different roads:

- Ostgerade (3x)
- Betonkreisverkehr (3x)

Normal: 2 different roads:

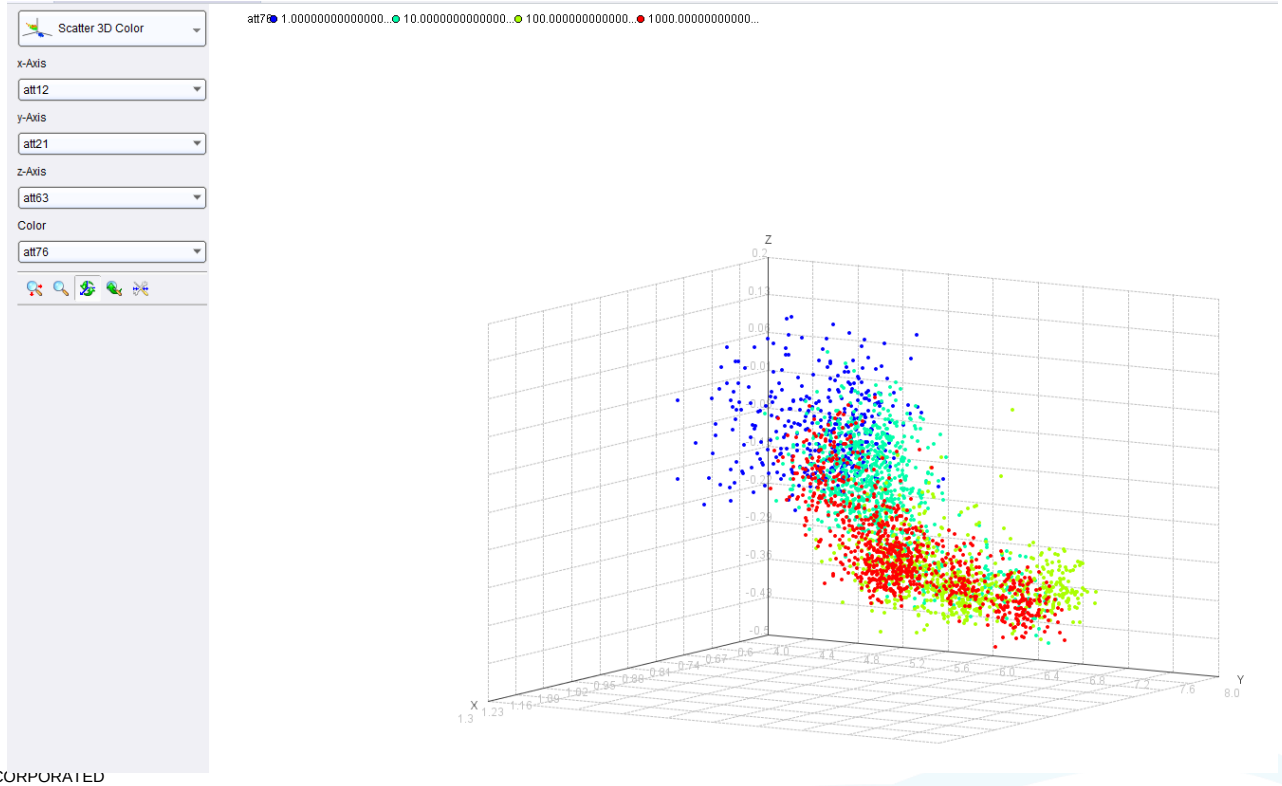
- A3 (9x)
- Roa-Oifing (2x)

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features: #12,#63,#21

Cobblestone (blue)
Beton (green)
Normal (light green)
Rough (red)

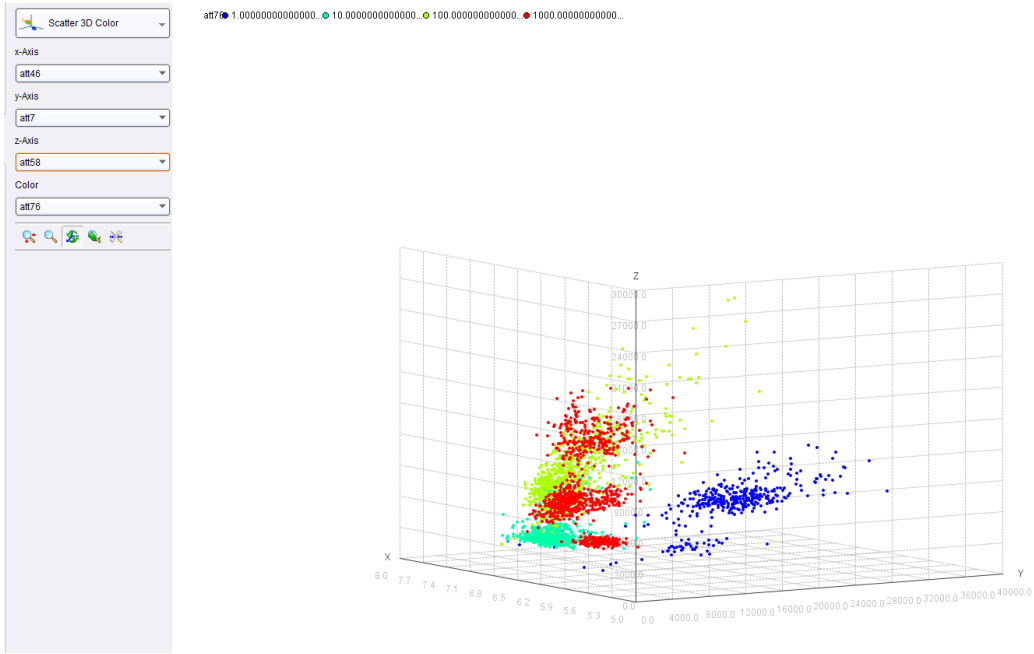


©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features:

#46(y-Axis: Spectral Entropy)
#7 (x-Axis: Exponential Bands – Spectrum)
#58 (z-Axis: Exponential Bands – Spectrum)



Cobblestone (blue)
Beton (green)
Normal (light green)
Rough (red)

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features:

#46 #7 #58

10fold-Crossvalidation

Table View Plot View

accuracy: 76.87% +/- 1.58% (mikro: 76.86%)

	true 1.0000000000000	true 10.000000000000	true 100.00000000000	true 1000.0000000000	class precision
pred. 1.000000000000	349	8	6	9	93.82%
pred. 10.00000000000	10	679	12	35	92.26%
pred. 100.0000000000	2	14	469	218	66.71%
pred. 1000.000000000	16	93	210	606	65.51%
class recall	92.57%	85.52%	67.29%	69.82%	

Knn (10)

!Normal VS Rough!

accuracy: 65.90% +/- 1.14% (mikro: 65.90%)

	true 1.0000000000000	true 10.000000000000	true 100.00000000000	true 1000.0000000000	class precision
pred. 1.000000000000	359	4	0	3	98.09%
pred. 10.00000000000	7	599	9	21	94.18%
pred. 100.0000000000	0	0	1	0	100.00%
pred. 1000.000000000	11	191	687	844	48.70%
class recall	95.23%	75.44%	0.14%	97.24%	

Tree (Max d=5)

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Features: #46 #7 #58 #33 #9 #21

10fold-Crossvalidation

Knn (10)

accuracy: 86.70% +/- 2.47% (mikro: 86.70%)

	true 1.0000000000000	true 10.000000000000	true 100.00000000000	true 1000.0000000000	class precision
pred. 1.000000000000	367	3	0	3	98.39%
pred. 10.00000000000	5	757	23	6	95.70%
pred. 100.0000000000	1	22	527	138	76.60%
pred. 1000.000000000	4	12	147	721	81.56%
class recall	97.35%	95.34%	75.61%	83.06%	

Tree (Max d=5)

accuracy: 65.32% +/- 1.82% (mikro: 65.31%)

	true 1.0000000000000	true 10.000000000000	true 100.00000000000	true 1000.0000000000	class precision
pred. 1.000000000000	364	4	0	4	97.85%
pred. 10.00000000000	8	624	9	66	88.26%
pred. 100.0000000000	0	1	1	0	50.00%
pred. 1000.000000000	5	165	687	798	48.22%
class recall	96.55%	78.59%	0.14%	91.94%	

Tree (Max d=8)

accuracy: 69.11% +/- 3.39% (mikro: 69.12%)

	true 1.0000000000000	true 10.000000000000	true 100.00000000000	true 1000.0000000000	class precision
pred. 1.000000000000	365	5	0	3	97.86%
pred. 10.00000000000	7	638	7	28	93.82%
pred. 100.0000000000	0	14	78	27	65.55%
pred. 1000.000000000	5	137	612	810	51.79%
class recall	96.82%	80.35%	11.19%	93.32%	

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

Neural Net

10fold-Crossvalidation

Features: #46 #7 #58 #33 #9 #21

accuracy: 89.22% +/- 1.51% (mikro: 89.22%)

	true 1.0000000000000	true 10.000000000000	true 100.00000000000	true 1000.0000000000	class precision
pred. 1.000000000000	365	3	0	4	98.12%
pred. 10.00000000000	6	756	12	9	96.55%
pred. 100.0000000000	1	19	510	45	88.70%
pred. 1000.000000000	5	16	175	810	80.52%
class recall	96.82%	95.21%	73.17%	93.32%	

Features: #46 #7 #58

accuracy: 84.90% +/- 2.67% (mikro: 84.90%)

	true 1.0000000000000	true 10.000000000000	true 100.00000000000	true 1000.0000000000	class precision
pred. 1.000000000000	363	1	0	1	99.45%
pred. 10.00000000000	8	740	26	22	92.96%
pred. 100.0000000000	1	35	457	82	79.48%
pred. 1000.000000000	5	18	214	763	76.30%
class recall	96.29%	93.20%	65.57%	87.90%	

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

TRAIN / TEST SPLIT

Features: #46 #7 #58 #33 #9 #21

Data Set GL

Cobblestone: 3 different roads:

- Bogen (3x)
- Hunderdorf (2x)
- Kopfsteinkreisverkehr (2)

Rough: 3 different roads:

- Westgerade (4x)
- Puchhof – Oifing (5x)

Beton: 2 different roads:

- Ostgerade (3x)
- Betonkreisverkehr (3x)

Normal: 2 different roads:

- A3 (9x)
- Roa-Oifing (2x)

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

TRAIN / TEST SPLIT

Features: #46 #7 #58 #33 #9 #21

accuracy: 68.02%

	true 1.0000000000000	true 10.000000000000	true 100.00000000000	true 1000.000000000	class precision
pred. 1.000000000000	235	0	0	0	100.00%
pred. 10.00000000000	0	344	78	78	68.80%
pred. 100.0000000000	0	0	0	145	0.00%
pred. 1000.000000000	1	10	98	293	72.89%
class recall	99.58%	97.18%	0.00%	56.78%	

Knn (10)

accuracy: 55.30%

	true 1.000000000000000	true 10.00000000000000	true 100.0000000000000	true 1000.000000000000	class precision
pred. 1.000000000000000	226	1	0	0	99.56%
pred. 10.00000000000000	0	294	101	195	49.83%
pred. 100.000000000000	0	13	10	142	6.06%
pred. 1000.00000000000	10	46	65	179	59.67%
class recall	95.76%	83.05%	5.68%	34.69%	

Neural Net

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

TRAIN / TEST SPLIT – 48kHz Data, Window Size = 4000 (1/12 sec)

Features: #46 #7 #58 #33 #9 #21

accuracy: 68.02%

	true 1.0000000000000	true 10.000000000000	true 100.00000000000	true 1000.000000000	class precision
pred. 1.000000000000	235	0	0	0	100.00%
pred. 10.00000000000	0	344	78	78	68.80%
pred. 100.0000000000	0	0	0	145	0.00%
pred. 1000.000000000	1	10	98	293	72.89%
class recall	99.58%	97.18%	0.00%	56.78%	

Knn (10)

accuracy: 38.62%

	true 1.000000000000000	true 10.00000000000000	true 100.0000000000000	true 1000.000000000000	class precision
pred. 1.000000000000000	2426	230	130	576	72.16%
pred. 10.00000000000000	10	1638	1221	1760	35.39%
pred. 100.000000000000	55	1	1	1986	0.05%
pred. 1000.00000000000	385	2445	804	1977	35.23%
class recall	84.35%	37.97%	0.05%	31.39%	

Neural Net

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

TRAIN / TEST SPLIT – 3 classes (rough fused with normal)

Features: #46 #7 #58 #33 #9 #21

accuracy: 87.13%

	true 1.0000000000000000	true 10.0000000000000000	true 100.0000000000000000	class precision
pred. 1.0000000000000000	235	0	0	100.00%
pred. 10.0000000000000000	0	344	154	69.08%
pred. 100.0000000000000000	1	10	538	98.00%
class recall	99.58%	97.18%	77.75%	

Knn (10)

accuracy: 65.91%

	true 1.0000000000000000	true 10.0000000000000000	true 100.0000000000000000	class precision
pred. 1.0000000000000000	216	1	0	99.54%
pred. 10.0000000000000000	0	296	359	45.19%
pred. 100.0000000000000000	20	57	333	81.22%
class recall	91.53%	83.62%	48.12%	

Neural Net

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

Road Surface Detection

TRAIN / TEST SPLIT

Features: Best 3 out of ALL sensor features (#559, #534, #234)

accuracy: 66.61%

	true 1.0000000000000000	true 10.0000000000000000	true 100.000000000000000	true 1000.00000000000000	class precision
pred. 1.0000000000000000	230	5	11	0	93.50%
pred. 10.000000000000000	0	311	33	44	80.15%
pred. 100.00000000000000	0	0	3	162	1.82%
pred. 1000.0000000000000	6	38	129	310	64.18%
class recall	97.46%	87.85%	1.70%	60.08%	

Knn (10)

©2017 HARMAN INTERNATIONAL INDUSTRIES, INCORPORATED

AI Services – Realized since March 2017 (1)

Gender Recognition based on the speaker's voice

Detection of siren sounds in urban environments

Detection of baby scream in home environments

Detection of doorbell sounds in home environments

Environment type recognition based on a single image

Music genre recognition based on sound snipets

Speaker recognition based on large training data and unknown speaker clustering based on voice samples

*Intelligent noise reduction**

*Classification of music, speech and noise based on sound snipets**

* Student work

AI Services – Realized since March 2017 (1)

Gender Recognition based on the speaker's voice

Detection of siren sounds in urban environments

Detection of baby scream in home environments

Detection of doorbell sounds in home environments

Environment type recognition based on a single image

Music genre recognition based on sound snipets

Speaker recognition based on large training data and unknown speaker clustering based on voice samples

*Intelligent noise reduction**

*Classification of music, speech and noise based on sound snipets**

* Student work


HARMAN
A SAMSUNG COMPANY
SINGLE CHANNEL NOISE REDUCTION USING DEEP LEARNING TECHNIQUES
GABRIEL HÜLSER
ON-GOING WORK
166

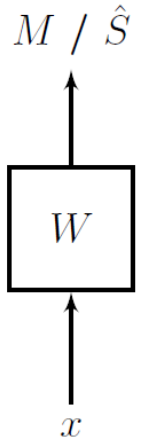
INTELLIGENT NOISE REDUCTION

Idea: Learning a mapping-function from noisy features to clean features (spectral masks)

→ Regression problem

Input / Features: Log Power Spectrum, Cochleagramm, Mel Spektrum…

Output / Targets: Binary Masks, Soft Masks, straight estimation of clean aplitude spectrum



mask or enhanced features

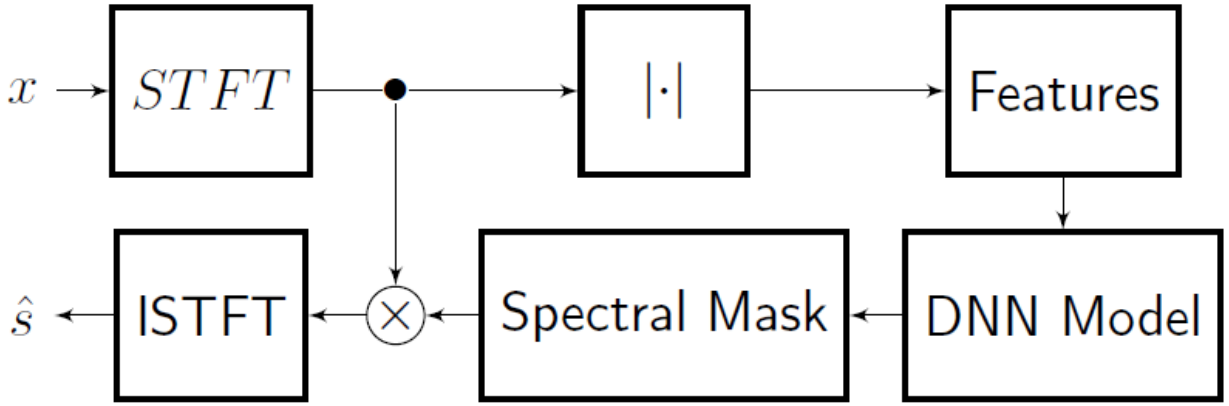
network

noisy data

INTELLIGENT NOISE REDUCTION

Reconstructed Signal:
Element-wise multiplication of noisy STFT and
Estimated spectral mask

$$\hat{s}[n] = ISFTF\left(M \odot STFT\left(x[n]\right)\right)$$

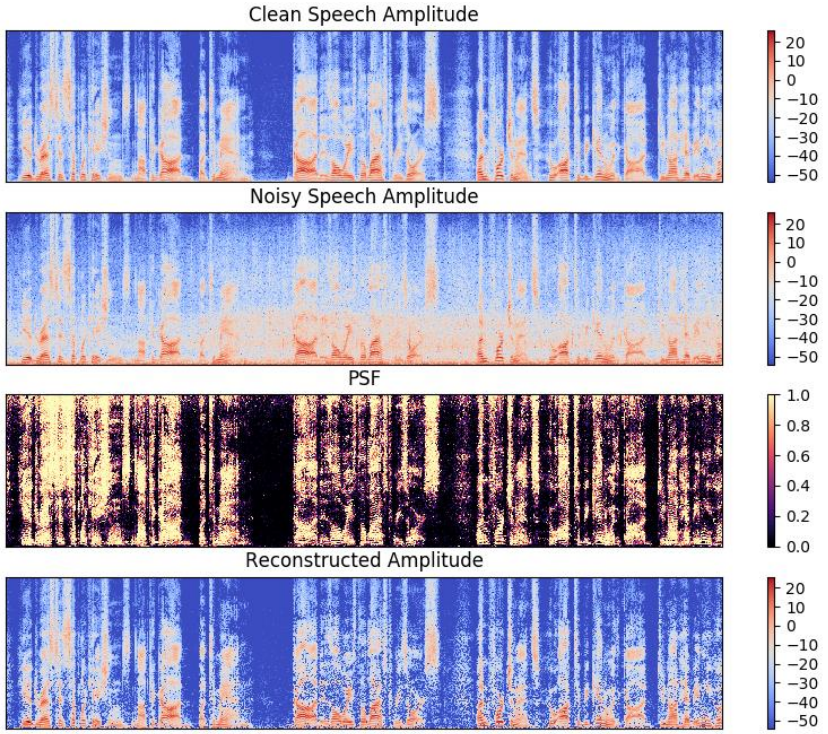


INTELLIGENT NOISE REDUCTION

Phase Sensitive Filter:

$$M^{\text{psf}} = \Re\left(\frac{S}{X}\right) = \Re\left(\frac{|S|}{|X|}e^{j(\Theta_S - \Theta_X)}\right) = \frac{|S|}{|X|}\cos\left(\Theta_S - \Theta_X\right)$$

$$M^{\text{psf}*} = \begin{cases} 1, & \text{for } M^{\text{psf}} > 1 \\ 0, & \text{for } M^{\text{psf}} < 0 \\ M^{\text{psf}}, & \text{else} \end{cases}$$



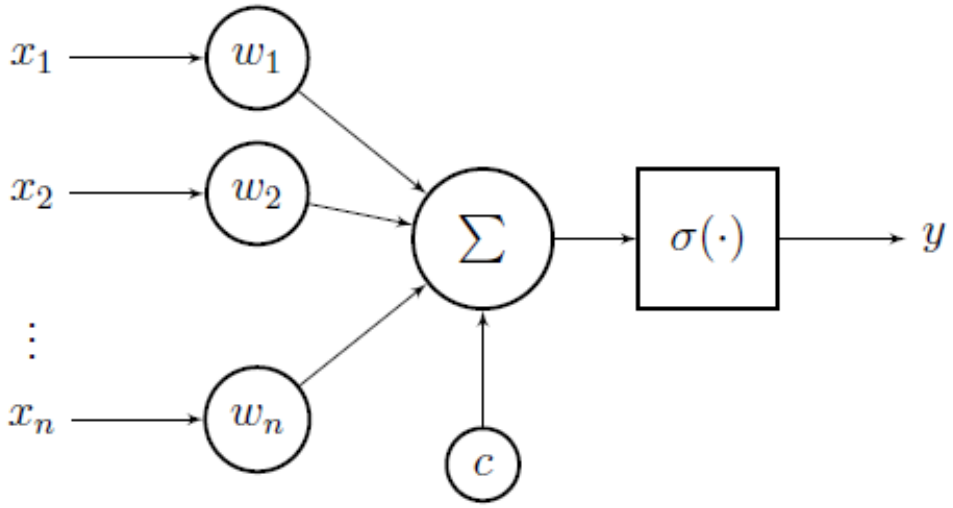
INTELLIGENT NOISE REDUCTION

Mathematical modell based on artificial neurons

Linear combination of inputs + bisas
Afterwards (non linear) transformation

$$a = \sum_{i=1}^{N} w_i x_i + c$$

$$y = \sigma(a)$$



INTELLIGENT NOISE REDUCTION

Organized in several layers => Deep Learning

$$\mathbf{h}_1 = \sigma\left(\mathbf{W}_1^T\mathbf{x} + \mathbf{c}_1\right)$$

$$\mathbf{y} = \sigma\left(\mathbf{W}_2^T\mathbf{h}_1 + \mathbf{c}_2\right) = \sigma\left(\mathbf{W}_2^T\left(\sigma\left(\mathbf{W}_1^T + \mathbf{c}_1\right) + \mathbf{c}_2\right)\right)$$

$$\mathbf{h}_k = \sigma\left(\mathbf{W}_k^T\mathbf{h}_{k-1} + \mathbf{c}_k\right)$$

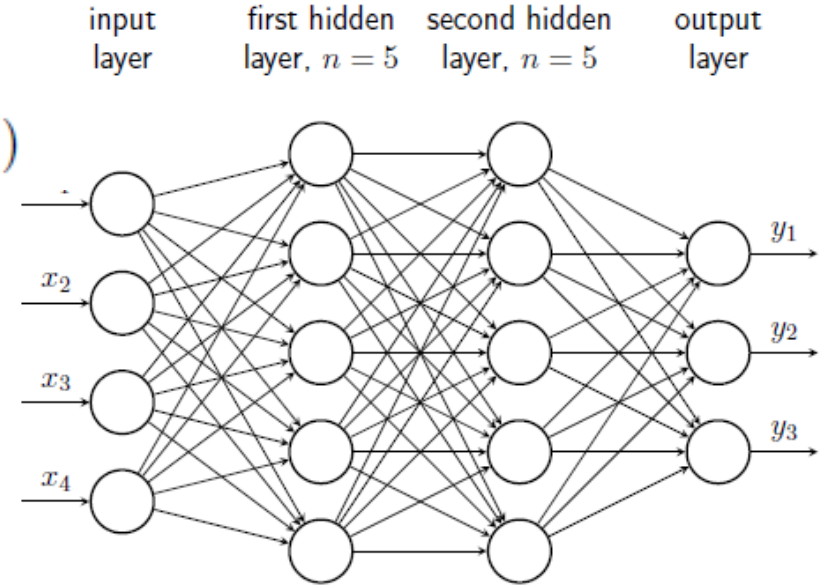


Figure 4 – Deep feed forward neural network with two hidden layers

INTELLIGENT NOISE REDUCTION

Training with pair of input and output vectors {**x, t**}
Minimize error function

$$y = f(x; \Theta)$$

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmin}}\, \mathcal{L}(f(\mathbf{x}; \Theta), \mathbf{t})$$

$$\mathcal{L}_{MSE}(f(\mathbf{x}; \Theta), \mathbf{t}) = \frac{1}{N} \sum_{i=1}^{N} (f(\mathbf{x}_i; \Theta) - \mathbf{t}_i)^2 = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{y}_i - \mathbf{t}_i)^2$$

INTELLIGENT NOISE REDUCTION

HARMAN

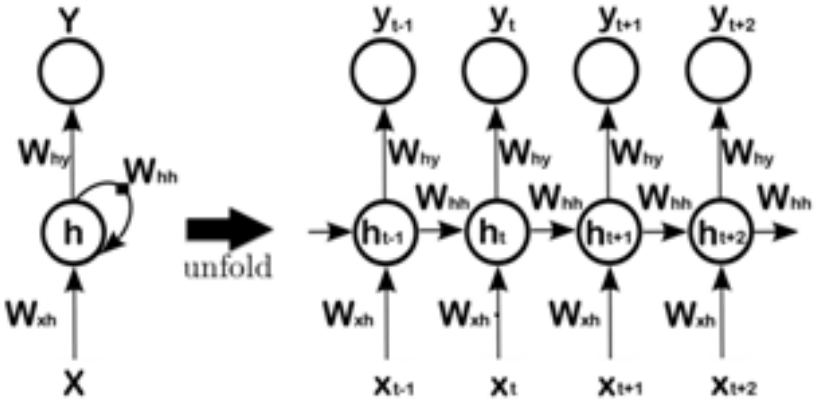
Parameter optimization using Gradient Descent and variants (scg, RMSProp, Adam, Adagrad)
Numerical calculation of the gradient with backpropagation

Gradient Descent: $$\Theta^{(t+1)} = \Theta^{(t)} - \eta \nabla \mathcal{L}(f(\mathbf{x}_i; \Theta^{(t)}), \mathbf{t}_i)$$

Stochastic Gradient Descent: $$\Theta^{(t+1)} = \Theta^{(t)} - \eta \nabla \sum_{i=1}^{N_\beta} \mathcal{L}_i(f(\mathbf{x}_i; \Theta^{(t)}), \mathbf{t}_i)$$

INTELLIGENT NOISE REDUCTION

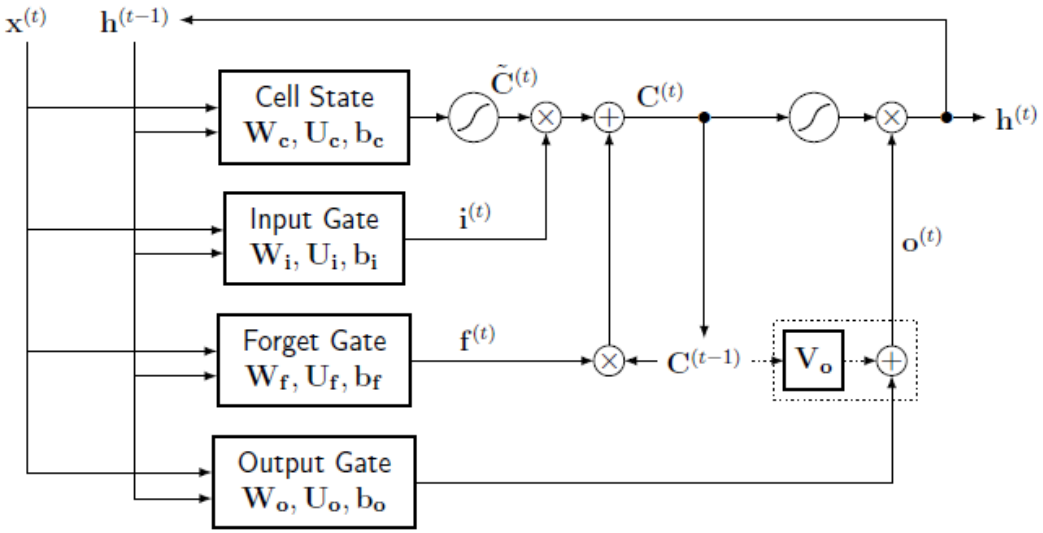
Using history information (States)
Promising model for time series (Audio!)
Problem: hard to optimize => Solution: Long short-term memory (LSTM) Networks



INTELLIGENT NOISE REDUCTION

$$\tilde{\mathbf{C}}^{(t)} = \tanh\left(\mathbf{W}_c\mathbf{x}^{(t)} + \mathbf{U}_\mathbf{c}\mathbf{h}^{(t-1)} + \mathbf{b}_\mathbf{c}\right)$$
$$\mathbf{i}^{(t)} = \sigma\left(\mathbf{W}_i\mathbf{x}^{(t)} + \mathbf{U}_\mathbf{i}\mathbf{h}^{(t-1)} + \mathbf{b}_\mathbf{i}\right)$$
$$\mathbf{f}^{(t)} = \sigma\left(\mathbf{W}_f\mathbf{x}^{(t)} + \mathbf{U}_\mathbf{f}\mathbf{h}^{(t-1)} + \mathbf{b}_\mathbf{f}\right)$$
$$\mathbf{o}^{(t)} = \sigma\left(\mathbf{W}_o\mathbf{x}^{(t)} + \mathbf{U}_\mathbf{o}\mathbf{h}^{(t-1)} + \mathbf{b}_\mathbf{o} + \mathbf{V}_\mathbf{o}\mathbf{C}^{(t-1)}\right)$$

$$\mathbf{C}^{(t)} = \mathbf{i}^{(t)} \odot \tilde{\mathbf{C}}^{(t)} + \mathbf{f}^{(t)} \odot \mathbf{C}^{(t-1)}$$
$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \odot \tanh\left(\mathbf{C}^{(t)}\right)$$



INTELLIGENT NOISE REDUCTION

Dataset:

- Speech: 360 hours ‚librispeech' Corpus
- Noise data: 29 hours QUT-Noise + TUT-Noise + free-sound.org
  => 8 categories: car, city, cocktail_party, home, nature, office, traffic, transient

=> Training set with 300 speakers, each 50 sec,
mixed with eight categries,
4 different SNR {10, 5, 0, -5} dB

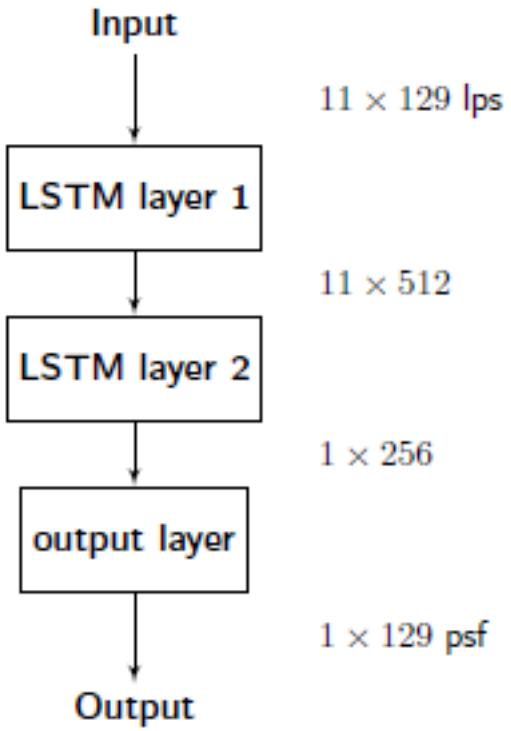
= 48000 sets = 130h
= 39,000,000 data points (32ms Frames, 50% Overlap)
about 500GB Features (16Khz)

INTELLIGENT NOISE REDUCTION

LSTM Netzwerk 8kHz:

- features:
  11 frames x 129 log power spectral bins
  mean variance normalization
- 512 + 256 hidden units
- targets:
  1 frame x 129 psf mask
  corresponding to last input frame (11th)
- Batch size 128
- Dropout 0.2

Cortana Accuracy Score: 40% -> 65% (after upsampling to 16kHz)

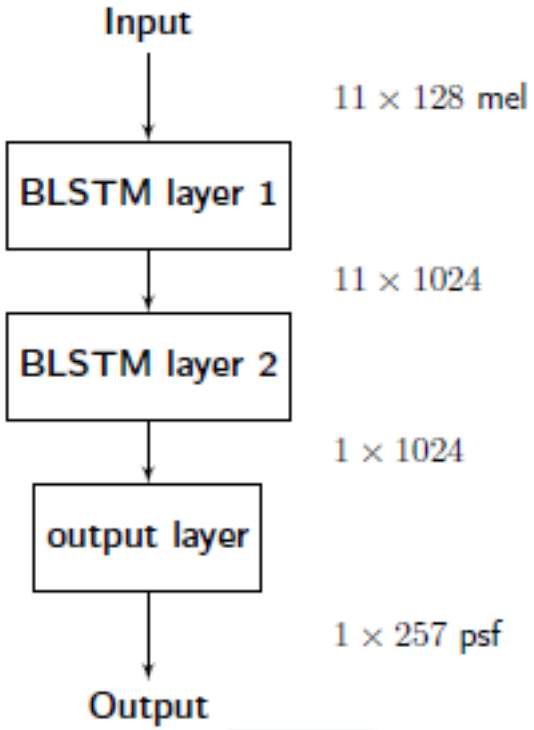


INTELLIGENT NOISE REDUCTION

LSTM Netzwerk 16kHz:

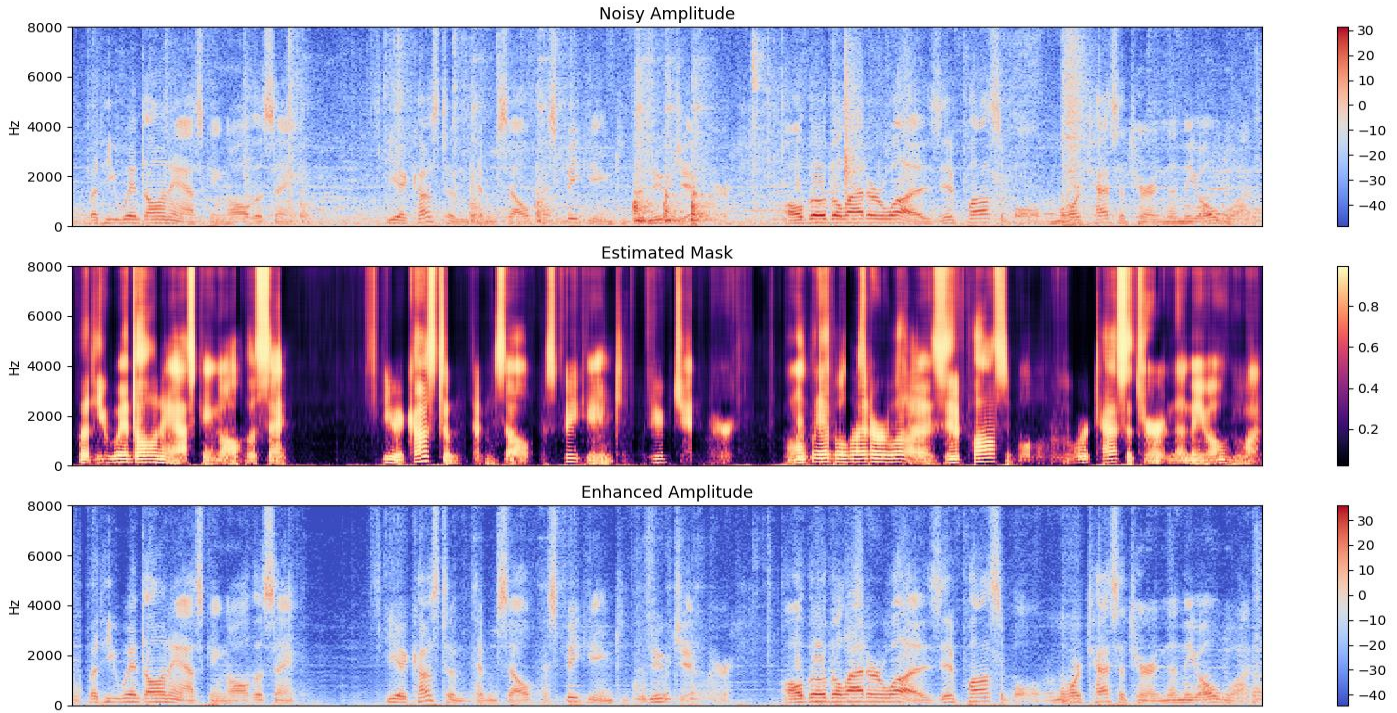
- features:
  11 frames x 128 mel bins
- 512 + 512 hidden units, bidirectional!
- targets:
  1 frame x 257 psf mask
  corresponding to 6th input frame
- Batch size 128
- Dropout 0.2

Cortana Accuracy Score: 40% -> 71%
ITU-T P.862 PESQ ~ +0.48



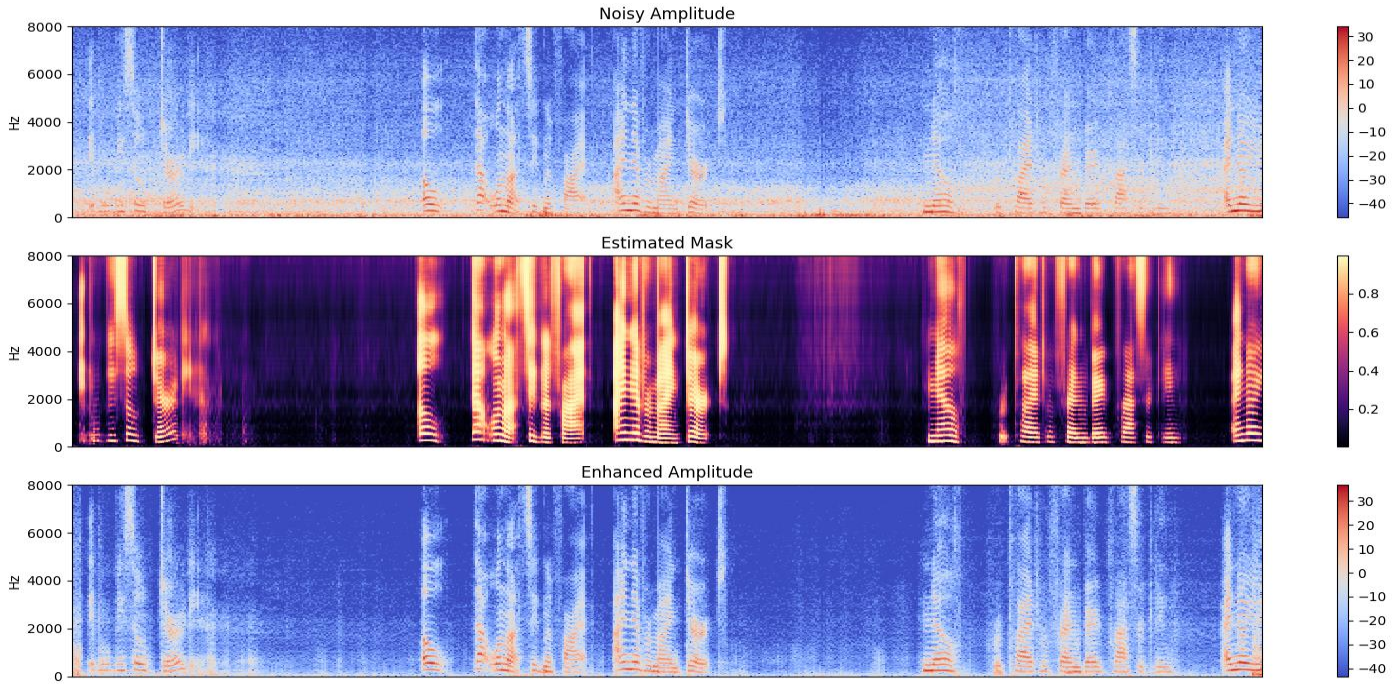
INTELLIGENT NOISE REDUCTION

BABBLE NOISE:



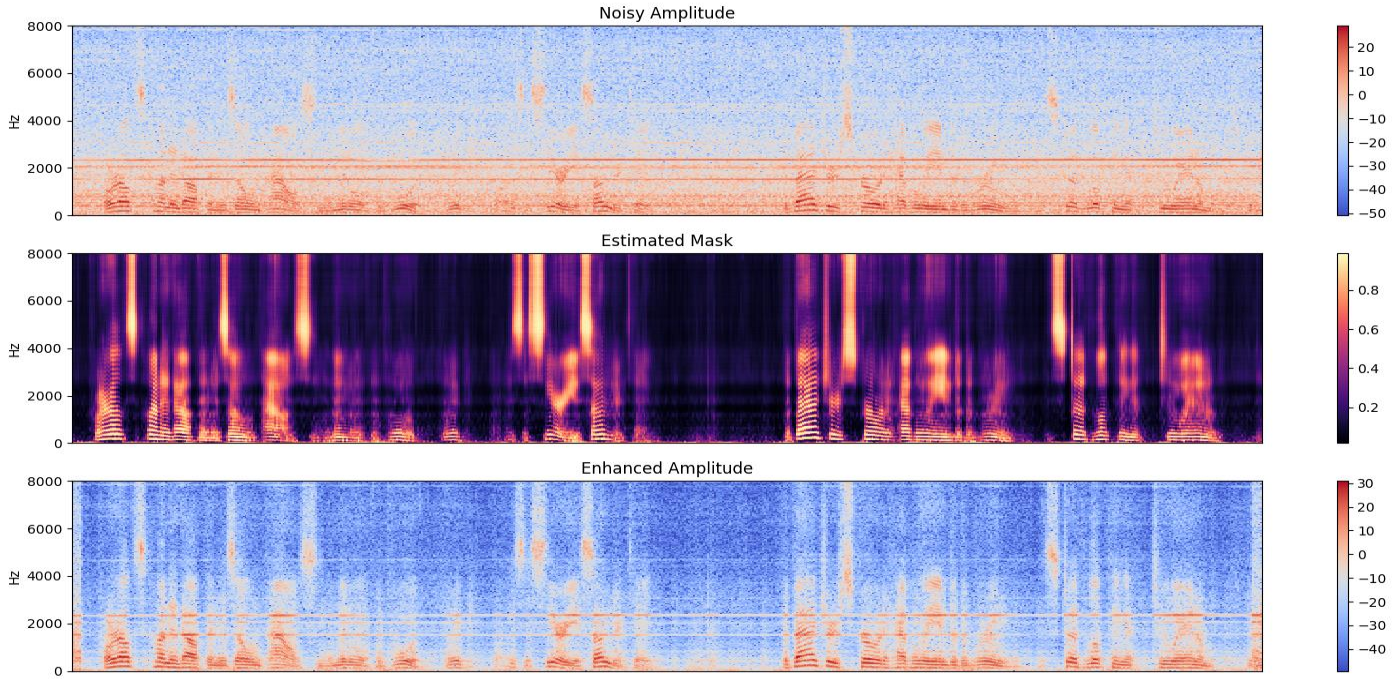
INTELLIGENT NOISE REDUCTION

CAR NOISE:



INTELLIGENT NOISE REDUCTION

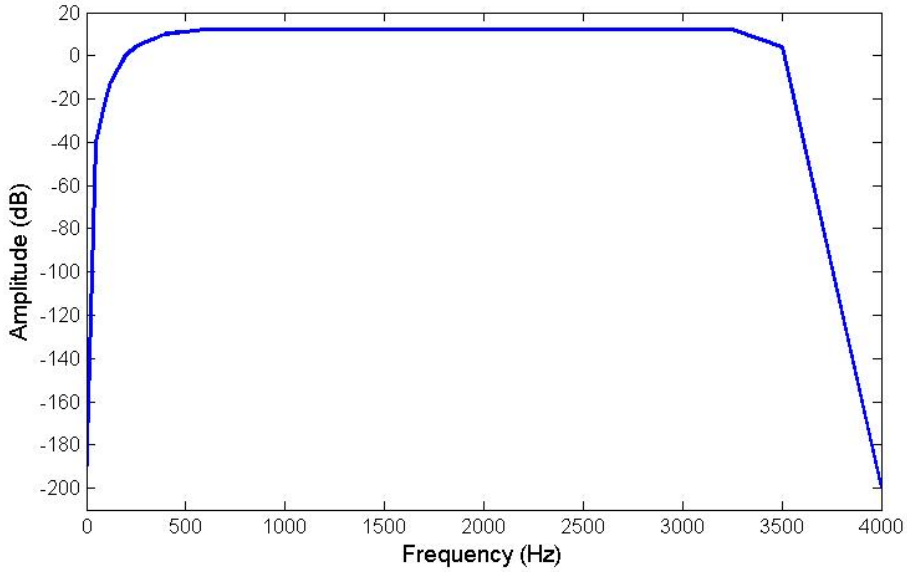
Engine Noise:



INTELLIGENT NOISE REDUCTION

Problems:

- Band limited signals (NOIZEUS)
- reverberation, time luting
- Very complex models
  (BLSTM about 9 million parameter)



INTELLIGENT NOISE REDUCTION

Outlook:

- Optimize net structure
- Simulate different rooms and microphones
- Adapted error function
- Use phase information

AI Services – Realized since March 2017 (I)

Gender Recognition based on the speaker's voice

Detection of siren sounds in urban environments

Detection of baby scream in home environments

Detection of doorbell sounds in home environments

Environment type recognition based on a single image

Music genre recognition based on sound snipets

Speaker recognition based on large training data and unknown speaker clustering based on voice samples

*Intelligent noise reduction**

*Classification of music, speech and noise based on sound snipets**

* Student work

AI Services – Realized since March 2017 (1)

Gender Recognition based on the speaker's voice

Detection of siren sounds in urban environments

Detection of baby scream in home environments

Detection of doorbell sounds in home environments

Environment type recognition based on a single image

Music genre recognition based on sound snipets

Speaker recognition based on large training data and unknown speaker clustering based on voice samples

*Intelligent noise reduction**

*Classification of music, speech and noise based on sound snipets**

* Student work


HARMAN
A SAMSUNG COMPANY
SPEECH, MUSIC AND ENVIRONMENTAL NOISE CLASSIFICATION
FELIX ROTHMUND
ON-GOING WORK

Overview

Speech Music and Environmental Noise

1. Problem Description and Introduction
2. Dataset
3. Machine Learning - Support Vector Machine
4. Deep Learning - Convolutional Neural Net
5. Evaluation and Outlook

Problem Description

Speech Music and Environmental Noise

With the rising presence of voice activated devices, we start to expect human-level perception of our virtual assistants. Speech, Music and Environmental Noise classification could make the assistants more context-aware (“Is there someone speaking? Did someone call for help?”) and allow for automatic logging of everyday activities (“What was the song I was listening to this morning? Can you replay my conversation with John last night?”).

Speech and Music classification has been successfully used for various tasks in audio signal processing, such as content-aware data compression techniques or mode-switching for de-noising algorithms in hearing aids.

It has also worked well for automatic annotation and segmentation of multimedia data,
e.g. helping to auto-tag large datasets and to finds segments containing music for accurate royalty fee estimations.

However, most proposed algorithms were designed for clean and well-leveled signals. This work describes the automatic classification of an audio data stream into the categories Music, Speech and Environmental Noise with special emphasis on degraded audio data, i.e. far-field microphone array recordings.

Machine Learning

Approaches

(Classic) Machine Learning:

- Hand-made descriptive features (Zero-Crossing Rate, Spectral Centroid, …)
- Feature-based Classifiers, such as Naïve Bayes (NB), Support Vector Machines (SVM), k-Nearest Neighbors (kNN), Multi-Layer Perceptron (MLP), …
- Good performance for 'clean' Speech-Music Discrimination (>95% accuracy for audio clips of 1s - 2.5s length)
- Performance drops significantly for multi-class classification (i.e. Speech-Music-Noise)
- Not so well for 'degraded' signals (highly dependent on feature-selection)
- Not so much training data required

Deep Learning:

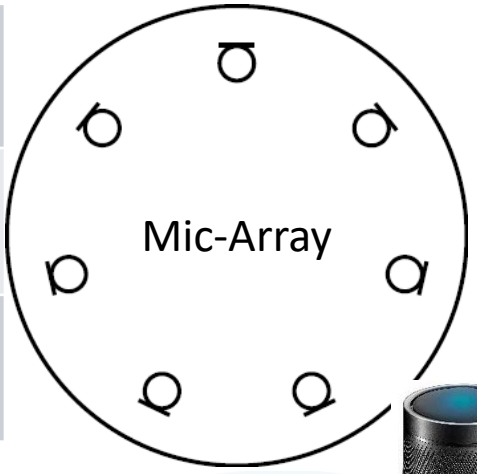
- Features are learned by the algorithm itself
- Lots of training data required
- unparalleled performance even for domain offset (train with clean data, test with degraded and vice-versa)
- Hard to say *why* it works
- Computationally demanding

Speech, Music and Environmental Noise

Data Set Generation

- Collected ‘raw’ audio data from various sources containing lots of different genres / speakers / noise types

Class	Train	Validation	Test (Offline)	Test (Online)
Music	~50h	~10h	20min	
Speech	~50h	~10h	20min	
Noise	~50h	~10h	20min	



Recorded ‘real-life‘ data-set with Cortana-Speaker

Speech, Music and Environmental Noise

Descriptive Feature Set

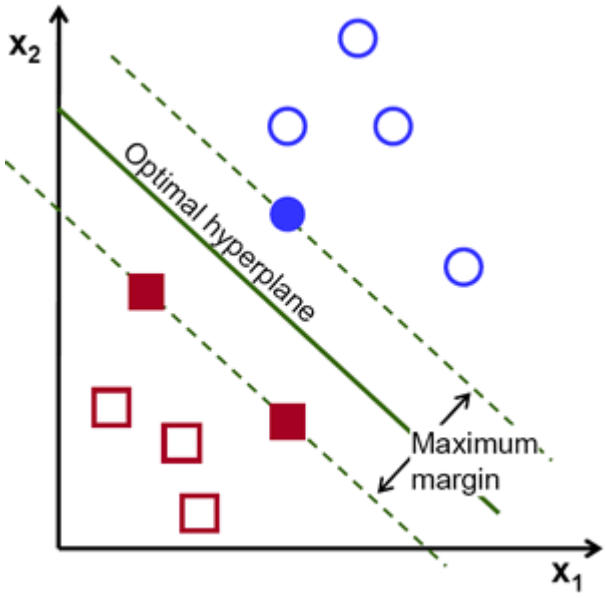
Feature	Desciption	#
f0	Fundamental Frequency	1
harmonicity	Harmonicity	1
tristimulus	Tri-stimulus	3
mfccs	Mel Frequency Cepstral Coefficients	12
zcr	Zero-Crossing-Rate	1
ste	Short-Time Energy / Root Mean Square (RMS)	1
spec_centroid	Spectral Centroid	1
spec_spread	Spectral Spread	1
spec_rolloff	Spectral Roll-Off	1
spec_flux	Spectral Flux	1
chroma_ent	Chromatic entropy	1
Total		24

→ **FRAME-LEVEL FEATURES:**
Computed 100 times per second

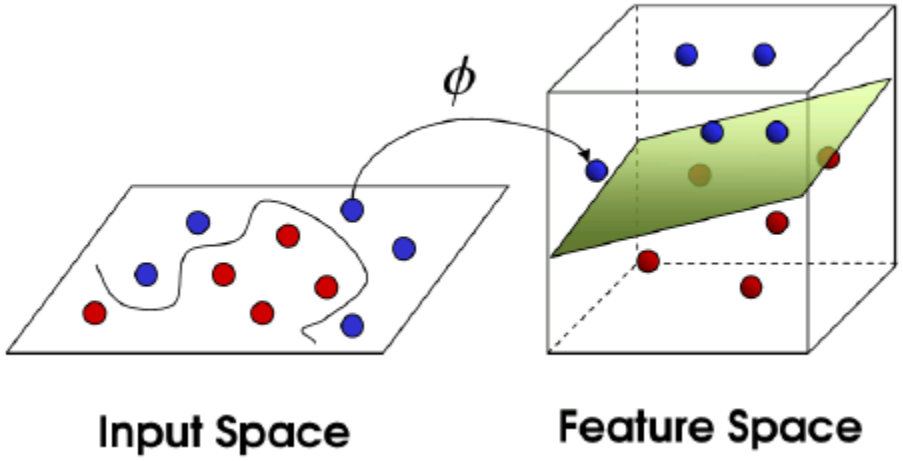
→ **CLIP-LEVEL:**
Combine 100 frames to clips of 1s
e.g. mean, variance, …

Support Vector Machine

Basic Concept:
Find a Hyperplane that maximizes the margin between two classes



Kernel Trick:
Transform Feature Space so that classes are separable



Support Vector Machine

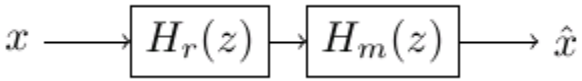
Experiments with non-linear SVM (rbf kernel):
Accuracy rate of up to 96 % for clip length of 1s for 'clean' Validation set

BUT very poor performance in real-time tests with microphone array.

Possible solutions:

- Find more robust feature set
- Generate better training data that resemble real-life signals better

Augment training data
by degrading with room and microphone impulse responses



→ improved, but still poor real-life performance

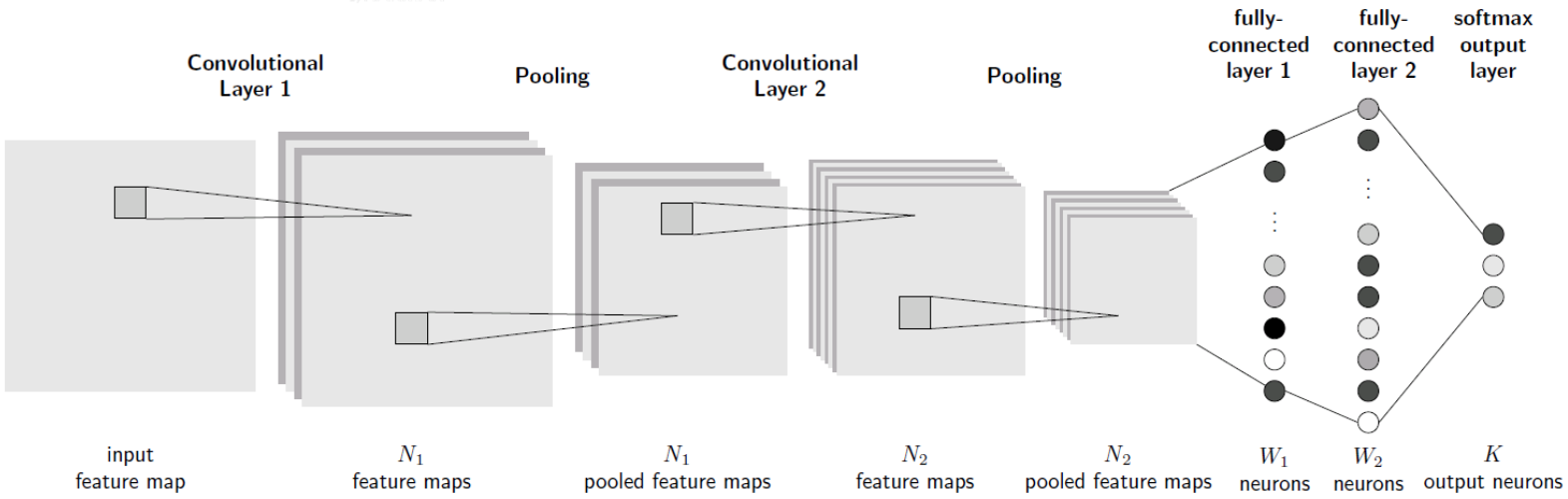
Convolutional Neural Net

Basic Concept:

Convolutional Layer: Find patterns in two-dimensional input image by comparing it segment-wise to different learned filter kernels

Pooling Layer: Reduce size of feature map by combining neighboring pixels

Multi-Layer-Perceptron: *'vanilla'* Artificial Neural Net / Non-linear classifier



2 **Convolutional Layers** with subsequent **Max-Pooling** for feature extraction

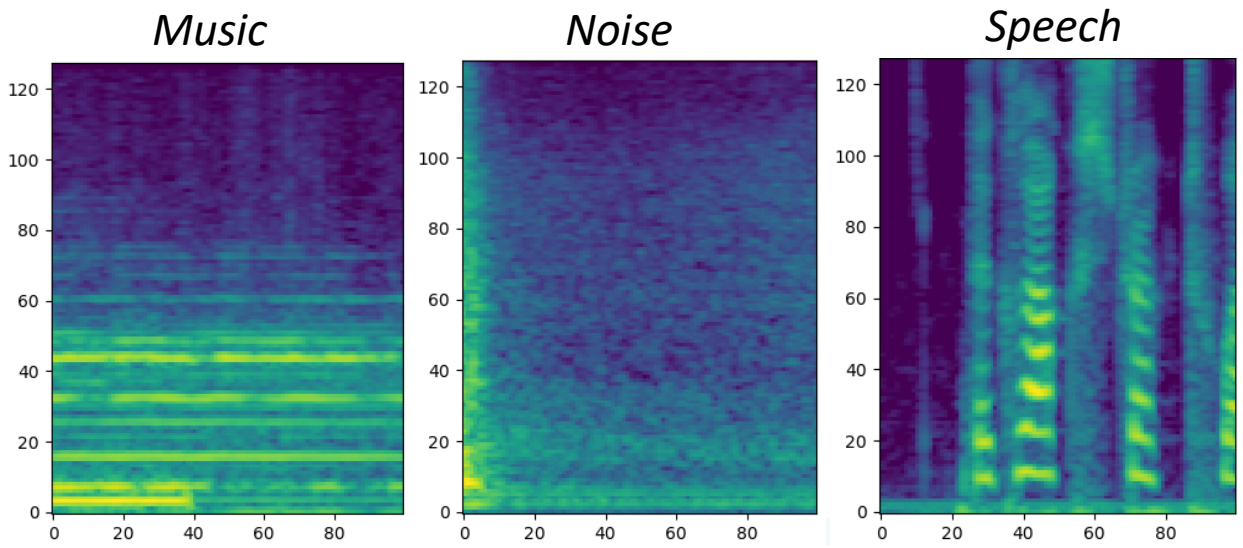
Multi-Layer Perceptron (MLP) with 2 fully connected layers and 1 output layer

Convolutional Neural Net

What is a good input representation?

- *(linear) Log Power Spectrogram (LPS) in dB*
  excellent performance for high resolution, drops significantly for lower resolution

- *Mel-Spectrogram in dB*
  Just as good, if not better with ¼ of frequency resolution (e.g. 128 instead of 512)
  performance drops less rapidly for lower frequency resolution



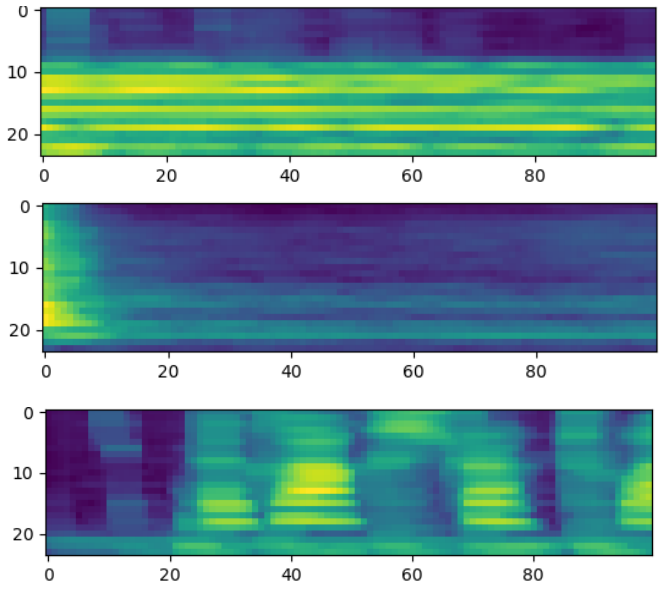
Mel-spectrogram with 128 mel bins and 100 time frames for 1s of audio, sampled at 16kHz

Convolutional Neural Net

What is a good input representation?

- *Mel Frequency Cepstral Coefficients (MFCCs)*
  very compact signal representation, however no sharp edges in resulting image resulting in poor performance

- *Cochleagram*
  models nervous activation on cochlea (inner ear),
  Good performance for relatively low frequency resolution (e.g. 24 gammatone channels)



In general:

temporal vs. frequency resolution:

- Clips shorter than 1s did worsen performance significantly
- 50 time-frames per second or more were found to be sufficient

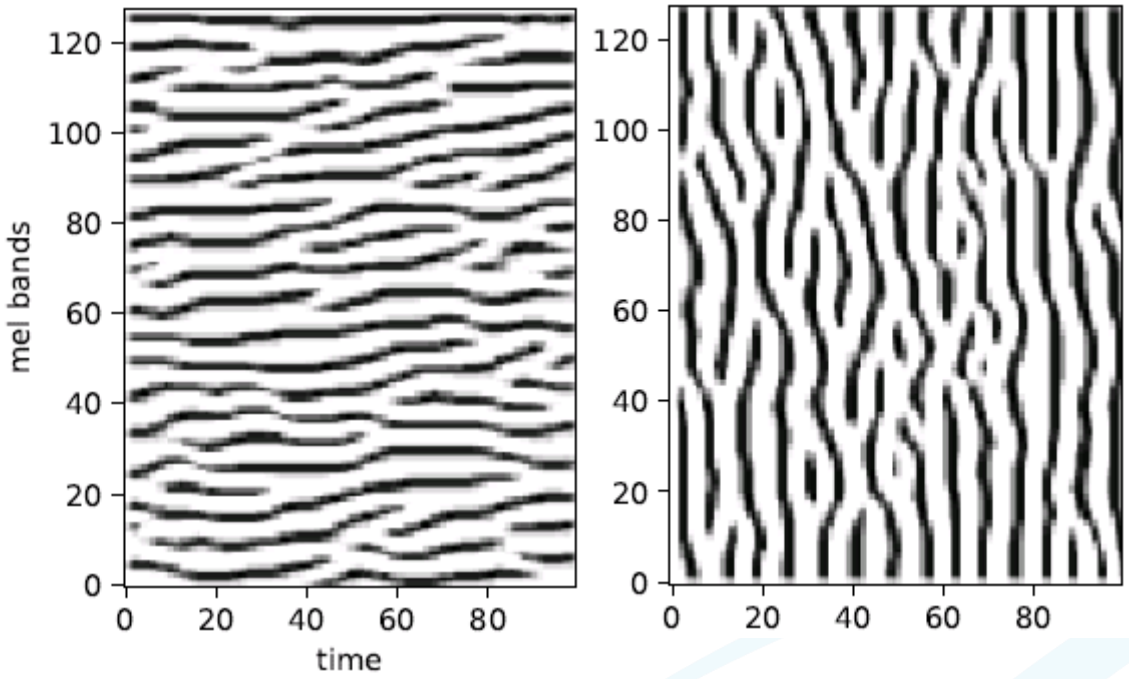
For final selection:

- Which data is available anyways or needed for other steps in the signal path? (e.g. mel-spectrogram for speaker identification and clustering, …)

Convolutional Neural Net

What does the neural net learn?
There is no way to tell exactly WHY a neural works, the underlying mechanisms cannot be easily understood.

However, convolutional filter kernels can be visualized by (iteratively) generating an image that maximizes the respective neuron output.



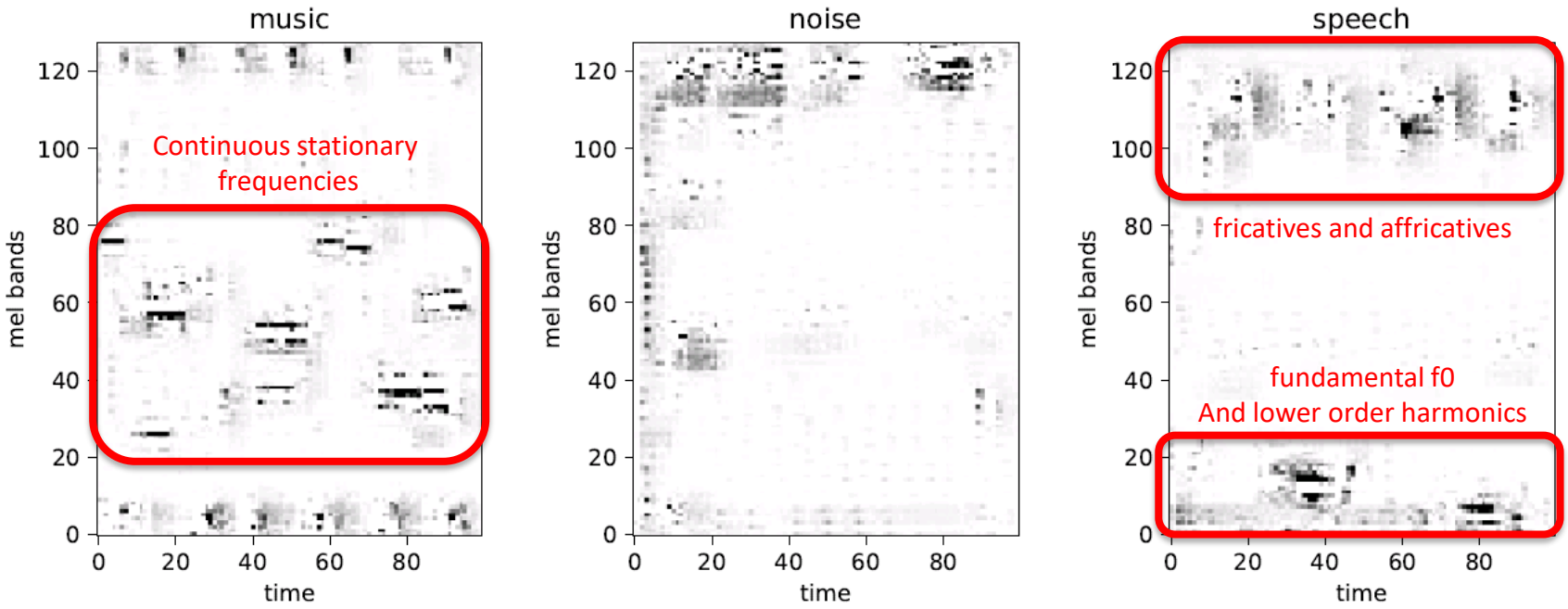
First layer filter kernel visualizations:
The left convolutional filter detects horizontal lines, such as fundamental frequency and its harmonics

The right filter detects vertical lines, such as transient audio events.

Convolutional Neural Net

What does the neural net learn?

So called deep dream images can be generated that give an indication of what the neural net thinks is speech, music or noise:

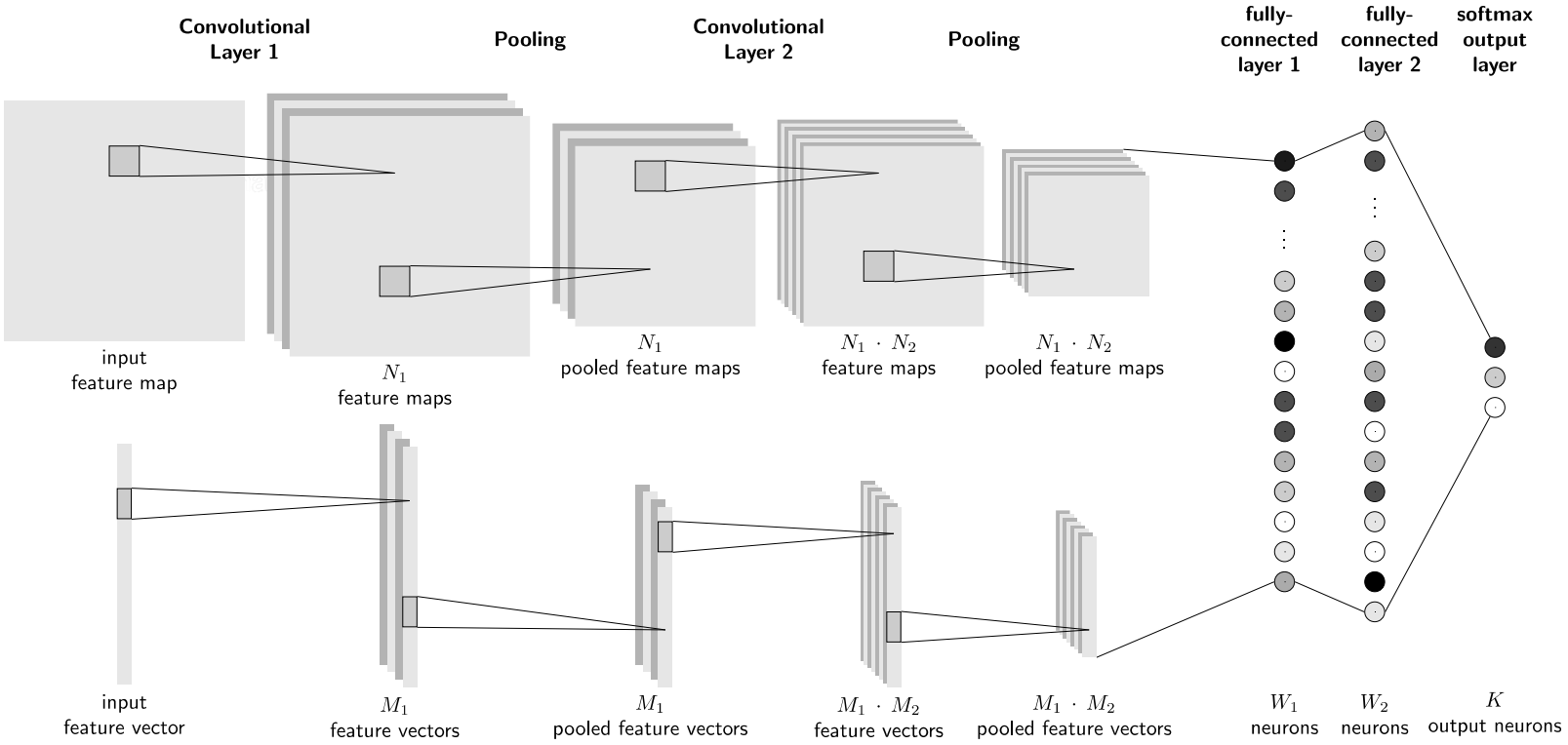


However, the quality of deep dram images does not necessarily say something about the system's performance.

Convolutional Neural Net

Experimenting with Model Structure:

Add parallel Feature path



→ e.g. add Zero-Crossing Rate (ZCR) as parallel feature, did increase accuracy for ‚clean' data noticeably (2%), but not for target domain (Mic-Array)

Speech, Music, Environmental Noise

Offline-Evaluation

Comparing CNN and SVM

using F-measure: combines recall and precision into one value:

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

TRAIN WITH CLEAN DATA

	CLEAN	DEGRADED	TARGET
CNN (128mel)	97	64	65
SVM (rbf)	96	67	40

TRAIN WITH DEGRADED DATA

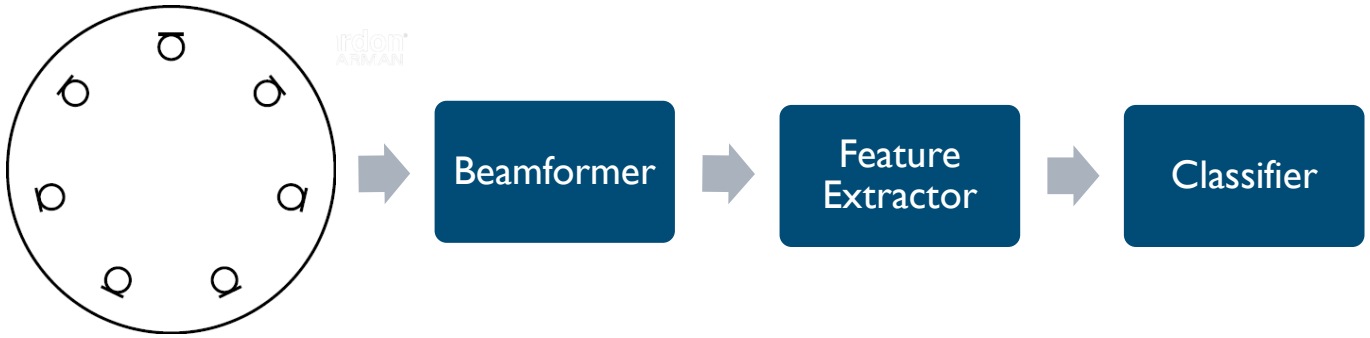
	CLEAN	DEGRADED	TARGET
CNN (128mel)	97	95	92
SVM (rbf)	92	92	50

preliminary results (early November)

Deep Learning

Online-Evaluation

Real-time implementation using pyAudio in Python
Buffer-Size 1s, gets updated 12x per second




```
In [1]: runfile('F:/pythonAudio/test_beamformer.py', wdir='F:/pythonAudio')
0  -  miniDSP ASIO Driver  - Input Channels:  10  - Device Index:  12
Using TensorFlow backend.
Start Live Demo:
M: 0.02 ---------- | N: 0.97 ########## | S: 0.00 ----------
```

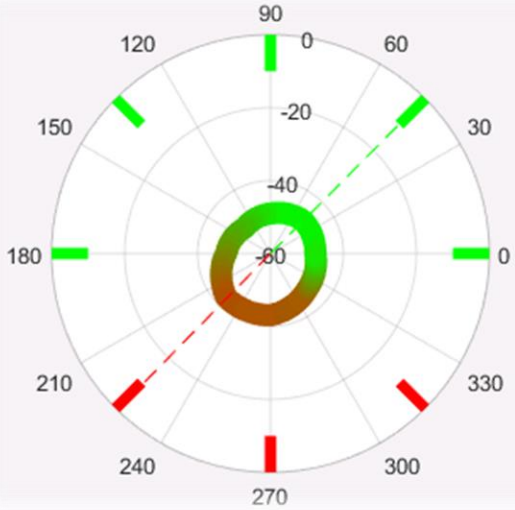
Note that the CNN classifier outputs a probability between 0 and 1 for each class, whereas an SVM will only output the predicted class

Deep Learning

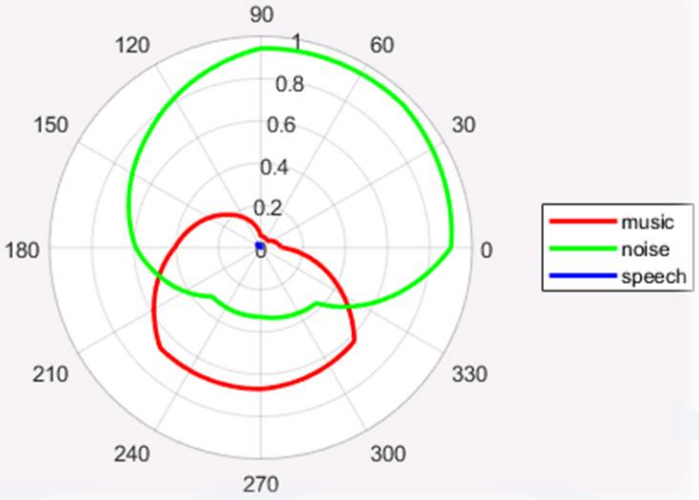
Online-Evaluation

→ **Beamformer and Classifier allow classification and tracking of different sources**

Estimated beam classes and estimated source direction



(interpolated) directional classification score / probability



Live Demo

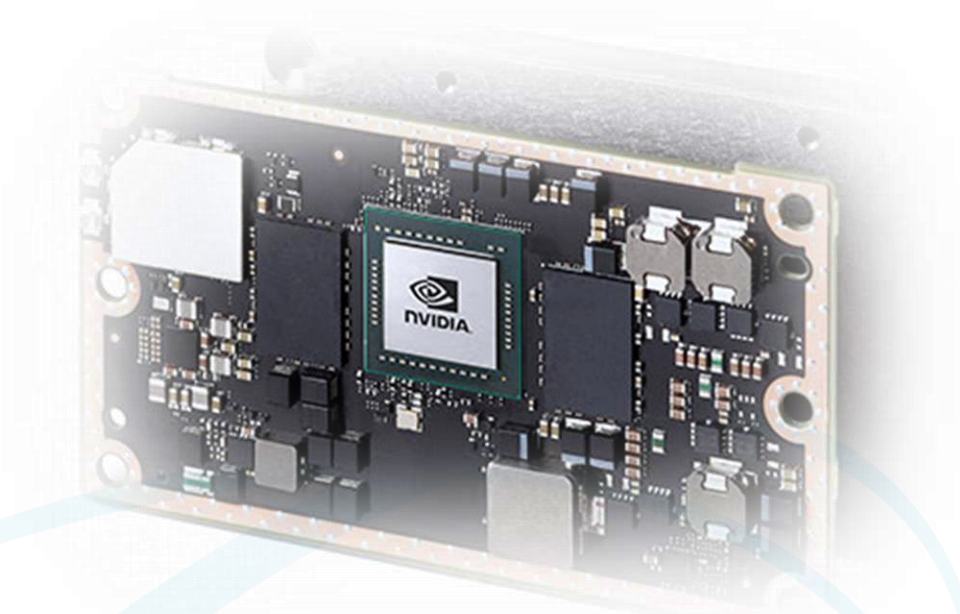
Feasibility

Convolutional Neural Nets are **computationally demanding**, especially when compared to other classifiers like SVM or kNN.

However, parallelization during training (and prediction) allows for much larger training sets and performance is more robust.

With the use of CNNs in more and more applications, such as autonomous driving and face recognition, hardware manufacturers are offering more and more **custom made AI chips** for these applications.

Optimization on DSP:
The CNN used in this work is rather simple, more complex CNNs have been successfully optimized for embedded systems, using optimization techniques like fixed point conversion and smart data re-organization.



Outlook

Summary:

A robust real-time classification of ‘real-world’ audio data into the categories *Music, Speech* and *Environmental Noise i*s feasible using CNNs. This can be a valuable pre-processing step for many tasks, such as speaker identification and clustering, source separation and de-noising.

Next Steps:

- Final Feature Selection and Optimization for SVM
- Evaluate Model Complexity and Performance Gain quantitatively
- Evaluation of different input images (Melspec vs. Cochleagram)
- Summarize Findings in Thesis

AI Services – Realized since March 2017 (1)

Gender Recognition based on the speaker's voice

Detection of siren sounds in urban environments

Detection of baby scream in home environments

Detection of doorbell sounds in home environments

Environment type recognition based on a single image

Music genre recognition based on sound snipets

Speaker recognition based on large training data and unknown speaker clustering based on voice samples

*Intelligent noise reduction**

*Classification of music, speech and noise based on sound snipets**

* Student work

Next Steps

Next Steps

AI Services – Smart Speakers

Gender Recognition based on the speaker's voice

Detection of baby scream in home environments

Environment type recognition based on a single image

Speaker recognition based on large training data and unknown speaker clustering based on voice samples

*Classification of music, speech and noise based on sound snipets**

Music genre recognition based on sound snipets

Detection of siren sounds in urban environments

Detection of doorbell sounds in home environments

*Intelligent noise reduction**

Restore the rich, original sound of music that gets lost during the digital audio compression process with Deep Learning

Emotion Recognition based on voice

Smart echo canceller

Key-Word / Name detection in different sound environments

...

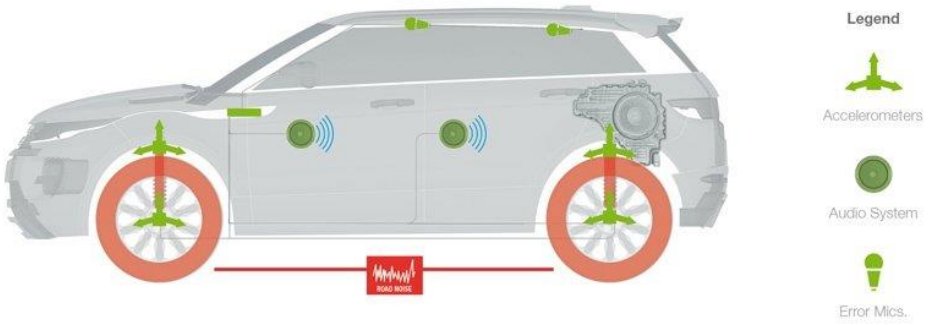
** Student work*

Next Steps

AI Services – Automotive

Sound based detection of car malfunctions / car parts wear

Automatic road surface recognition based on car mounted motion sensors



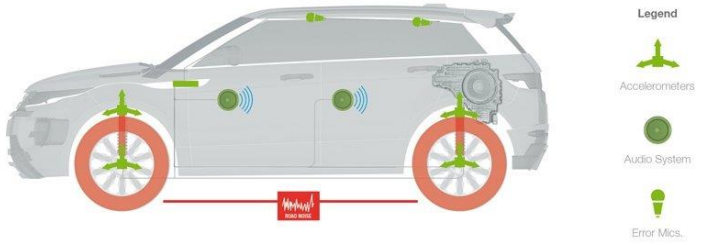
Next Steps

AI Services – Human Behavior Recognition

Daily Live Behavior (based on SAMSUNG smartphone sensors)

Connect everything and learn the user's behavior / situation in order to realize a pro-active support system – the „best friend" of the user / Private life assistent

Smart Home Information (based on smart SAMSUNG and HARMAN devices)



Car Behavior (based on car data provided by HARMAN headunits)

Next Steps

AI Services – Web Mining

Use the WEB as big data source and realize smart systems to identify trends / important activities of our competitors in early stages in order to be able to react faster to technology changes and to understand new business cases


HARMAN
A SAMSUNG COMPANY
THANK YOU