# Classification of room occupancy

Author: Kristyna Pijackova
ID: 211534
Class: MPA-MLF 2023

## Introduction

This project deals with a classification of a number of persons that are present in a room based on 60 GHz transmission. The available dataset consists of snapshots of signals in the delay-Doppler domain representing reflections from humans at some distance from the receiver moving with some velocity. The dataset includes CSV files and colored and black-and-white figures that all contain the same information. Examples of the ideal snapshot are shown in Fig. 1.

The goal of the project is to classify the number of persons in the room from those files, sorting the snapshots into three categories - one/two, or three persons in the room.

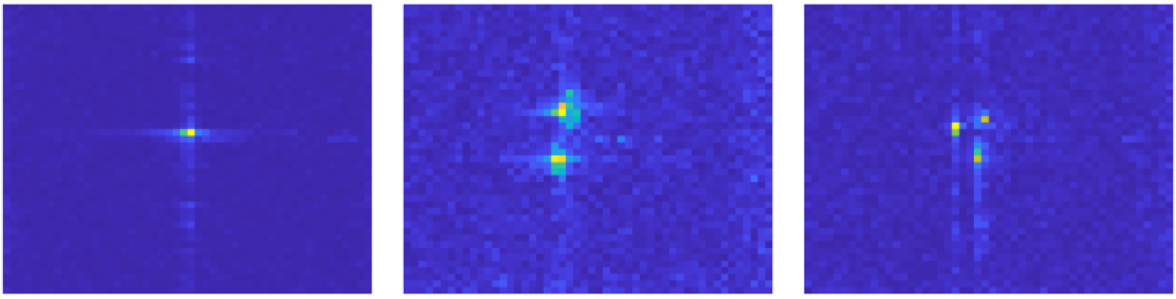

Fig. 1 - Example of ideal snapshots for one, two, and three persons in the room (from left to right.)

## Methodology

### Data cleaning

For the completion of the task, I chose the CSV file with the numerical snapshot values. Each file consists of a numpy array of the shape (45, 51, 1) and one additional column with file indexes. The training and testing data includes some files, that are corrupted and contain two decimal points instead of one.

To deal with this issue, I load each CSV file as panda DataFrame, remove the index column, as the information it is redundant, and check for invalid values that are then repaired by removing the second decimal point. All of the files are stored together in a .npy format for both training and testing sets. Since this process takes a longer time, the data was saved and the code uses this new version of the data.

## Data preprocessing

Before training, the data still needs some preprocessing. The training labels, which are from 1 to 3 are changed by subtracting 1 from them to be in the range 0 to 2 and encoded with one-hot encoding. The numerical values of the training and testing set are normalized by dividing all the values by the maximal value found in the training set. Finally, the training dataset is split into an 80:20% ratio to create training and validation sets.

## Model building and training

Since the provided data are images, I decided to try out a simple convolutional neural network with 3 convolutional layers, 2 max-pooling layers in between and 3 dense layers with dropout between them, out of which the last one is for the final classification.

Although this model with randomly chosen parameters seemed to work fine, I've decided to include Bayes optimization and optimize the kernel sizes and a number of filters for the convolutional layers, as well as a number of units for the dense layers with the help of keras-tuner library.

The final, best-performing model with the optimized parameters can be seen in Fig. 2 and Table 1.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d_3 (Conv2D) | (None, 41, 47, 120) | 3120 |
| max_pooling2d_2 (MaxPooling2D) | (None, 20, 23, 120) | 0 |
| conv2d_4 (Conv2D) | (None, 18, 21, 40) | 43240 |
| max_pooling2d_3 (MaxPooling2D) | (None, 9, 10, 40) | 0 |
| conv2d_5 (Conv2D) | (None, 5, 6, 56) | 56056 |
| flatten_1 (Flatten) | (None, 1680) | 0 |
| dense_3 (Dense) | (None, 96) | 161376 |
| dropout_2 (Dropout) | (None, 96) | 0 |
| dense_4 (Dense) | (None, 80) | 7760 |
| dropout_3 (Dropout) | (None, 80) | 0 |
| dense_5 (Dense) | (None, 3) | 243 |

```
Total params: 271,795
Trainable params: 271,795
Non-trainable params: 0
```

Table. 1: Optimized model

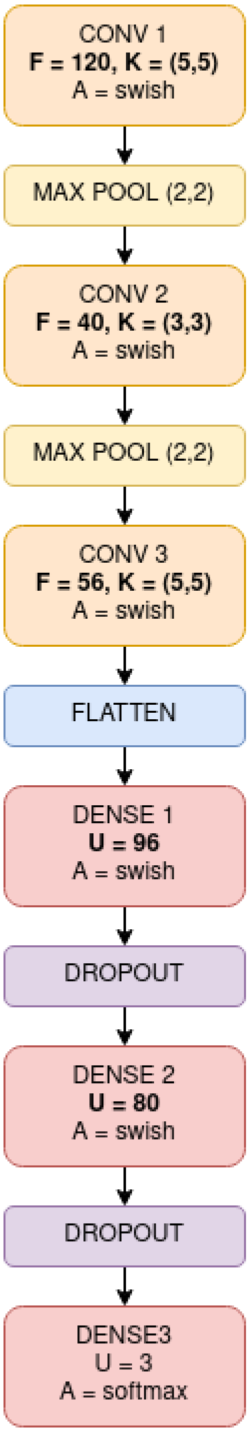


Fig. 2: Optimized model with parameters - bold parameters were optimized with the Bayes Optimization
(F - number of convolutional filters, K - kernel size, A - activation function, U - dense units)

## Model evaluation

Figure 3. shows the accuracy and loss plots for training and validation data. The accuracy on the validation set was 98.007% and 97.830 %  on the test set.

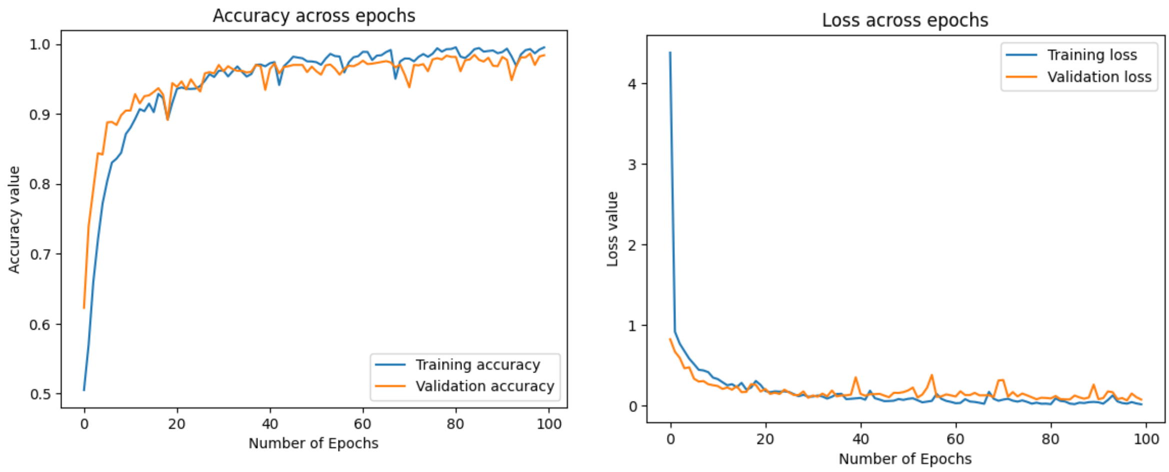


Fig. 3: Accuracy and Loss for training and validation set.

## Activations, Filters, and Feature Maps

With convolutional layers, we can visualize the learned filters of the network and see how they get activated on the image that is passed through them, as the filters are basically weights with the size of our chosen kernel size. Our first layer has one feature filter, as we are using 1 channel input, as we go further in the network, the number of filters grows, but I will demonstrate only the first out of many…

The activations from and feature maps are shown on the test image that can be seen in Fig. 4. Fig. 5 shows which filter and features got activated the most across the 3 convolutional layers (the brighter, the higher the activation.) In Figure 6 we can see some of the filters from the first convolutional layer. This layer had a kernel (5,5) and we see the different weights in each of them. In Figure 7 we can see one of the filters and feature maps from the second convolutional layer for filters 12 to 15, which, based on the activation map from Fig. 5 (middle figure), contributes the most to the final decision for this image scenario. And finally, Fig. 8 shows feature maps, which capture the result of applying filters to the input, of the second layer.

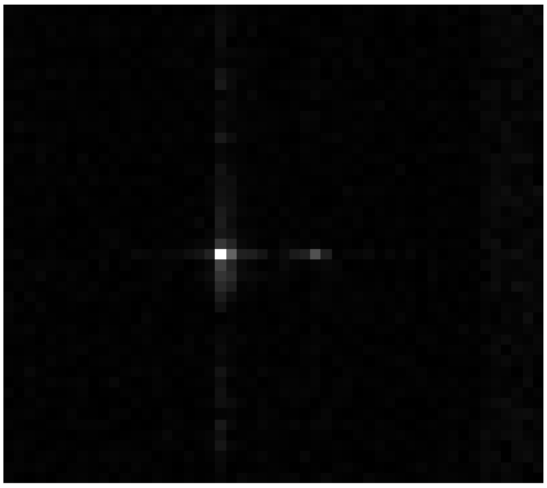

Fig. 4: Image from the test set for the visualization of feature maps and activations

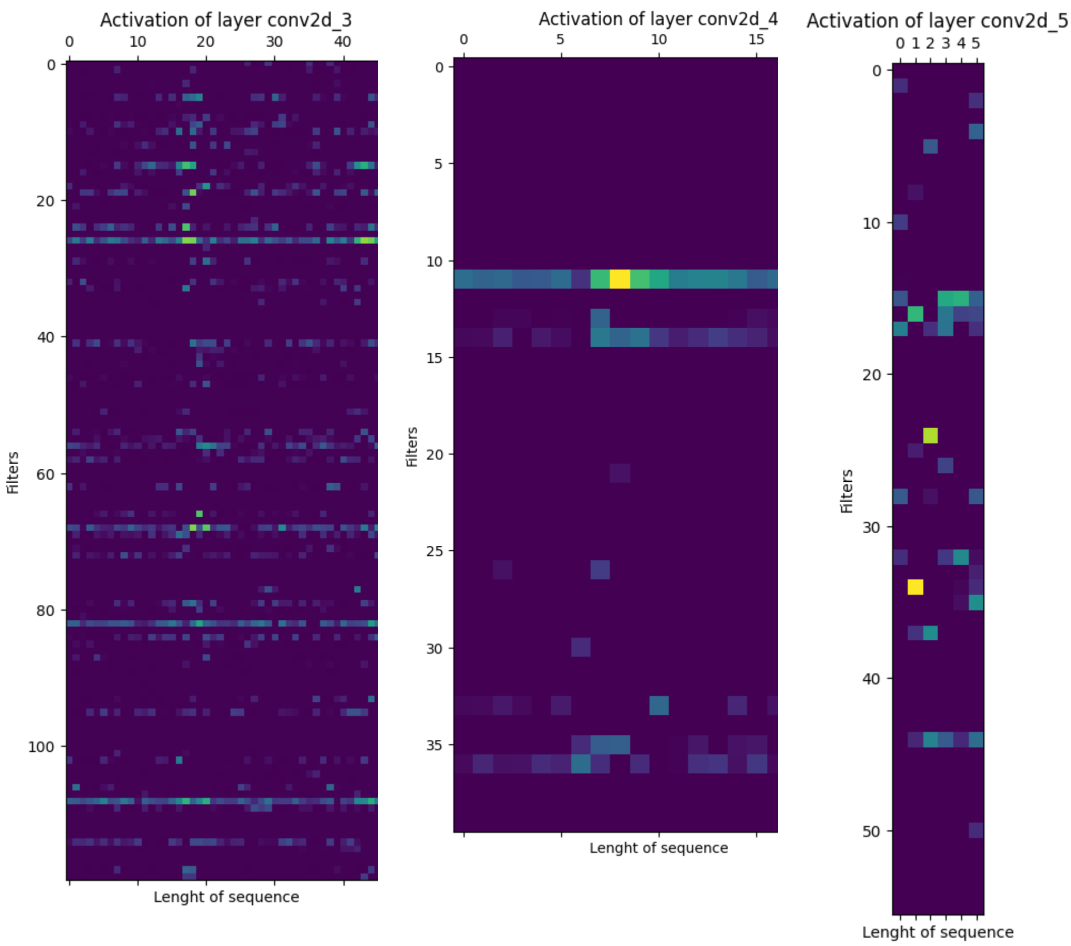


Fig. 5: Activations of filters in 1st, 2nd and 3rd convolutional layers (brighter spots mean higher activation)

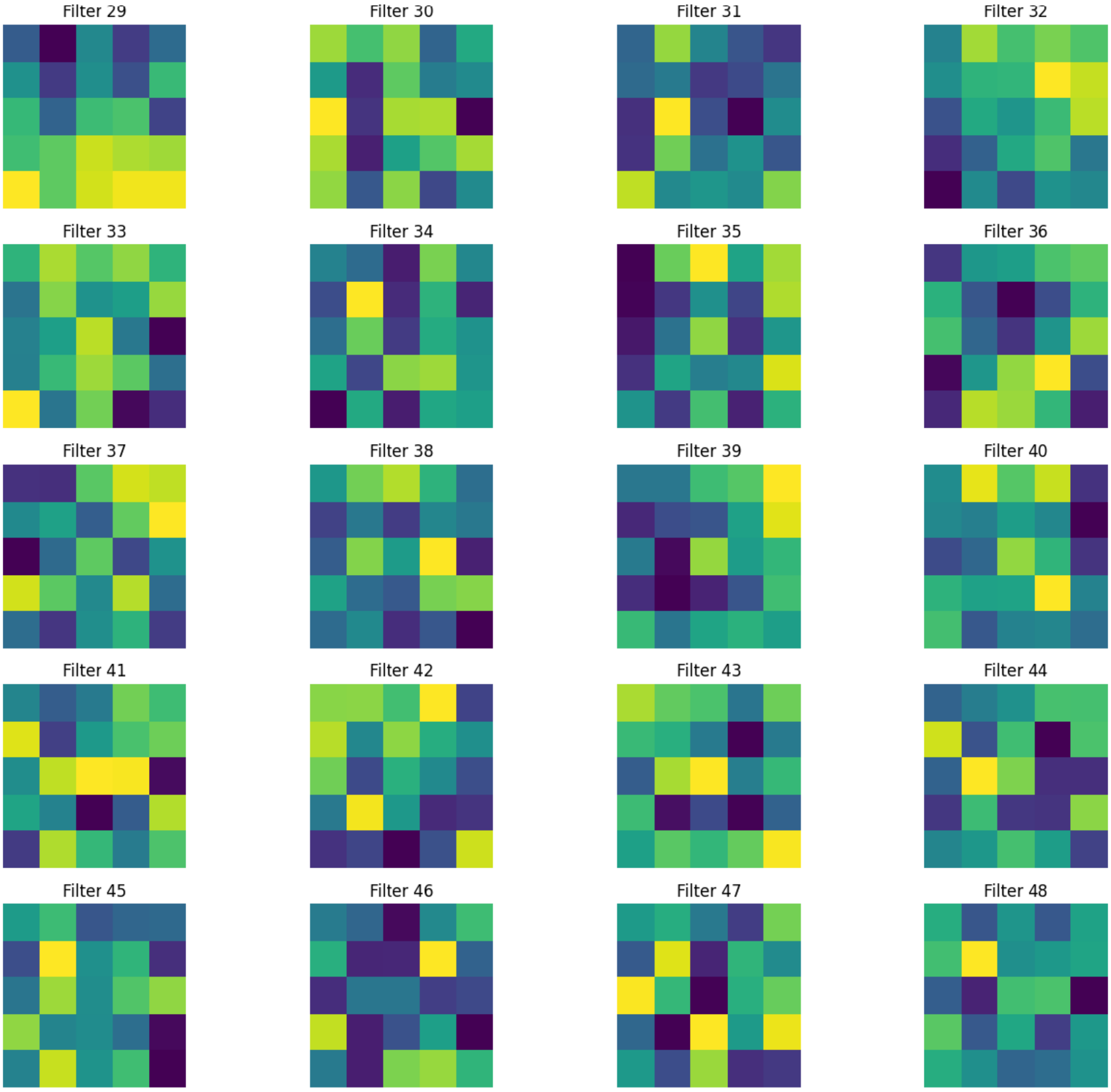


Fig. 6: Example of filters from the first convolutional layer with kernel size (5,5)

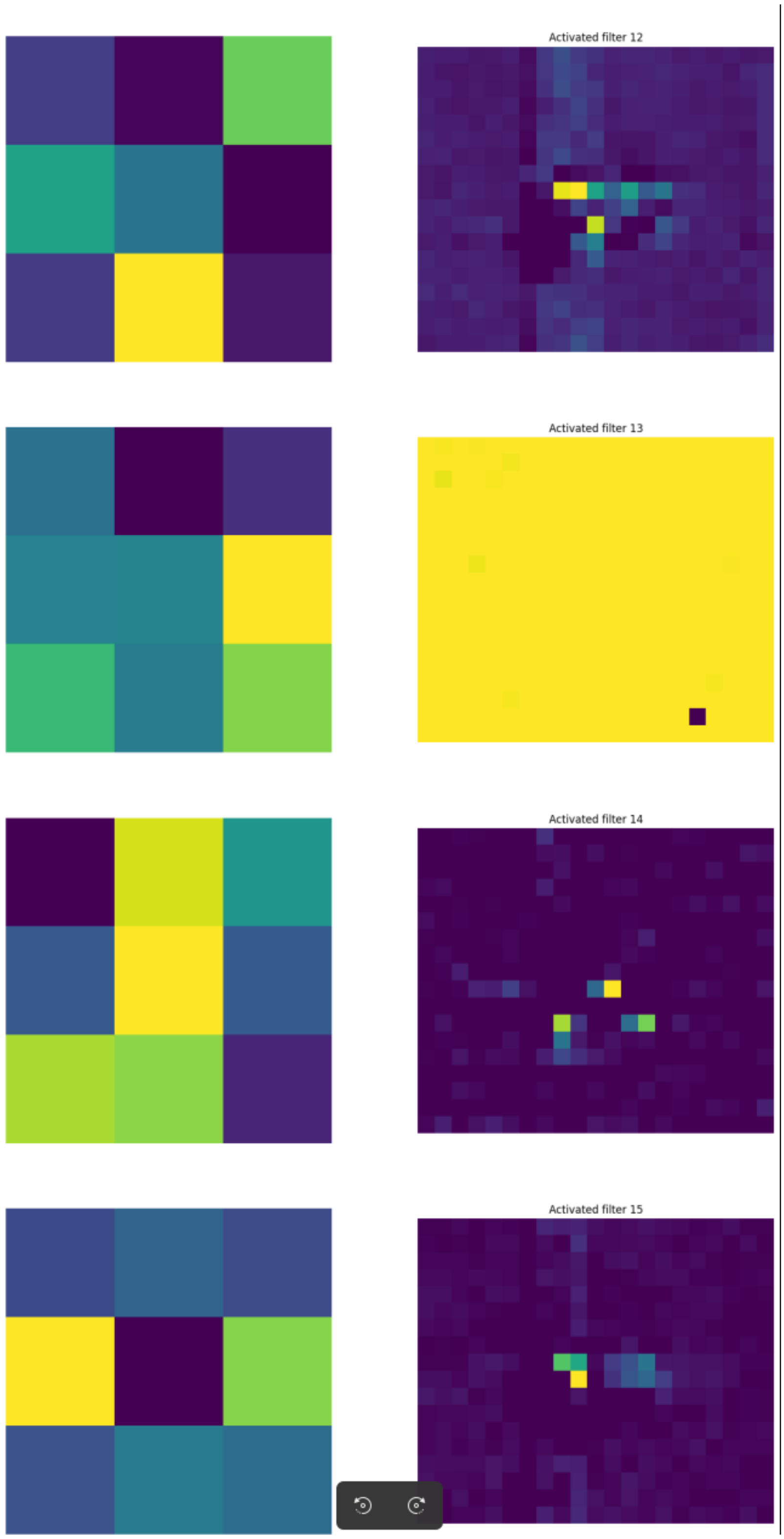


Fig. 7: Example of filters and feature maps that contributed the most in the second convolutional layer (see the bright patterns in Fig 5. in the middle figure between filters 10 and 15)

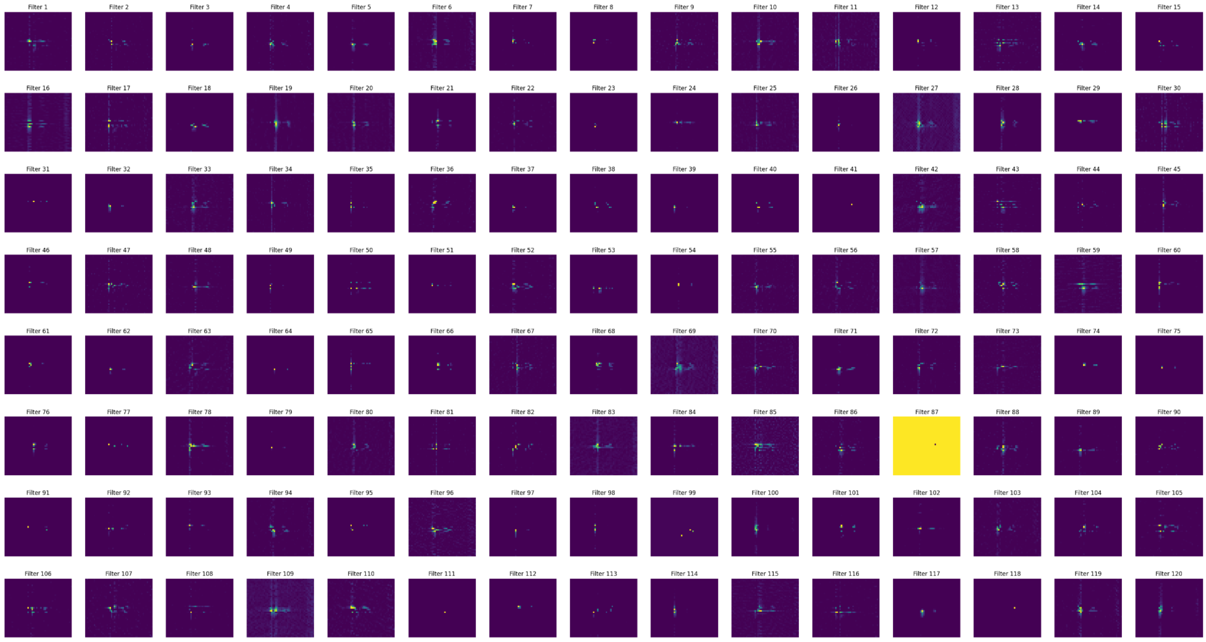

Fig. 8: Feature maps (filters activated by the passed image from Fig. 4) after the first convolutional layer

## Conclusion

This project was focused on finding a suitable machine or deep learning model for a classification problem focused on identifying a number of persons in the room from measured 60 GHz transmission data converted to a delay-Doppler domain. The optimized model was a convolutional neural network with 3 convolutional layers, 2 max-pooling layers, and 3 dense layers with dropout.  The final score reported on the test data was 98.309 %.