EE4675 Object Classification with Radar

Classification Project

Bob van Nifterik- 4558421
Jurgen Wervers - 4599136

June 17, 2022

1 introduction

Classification of human actions is useful in many places with privacy concerns such as an elderly care home. In these situations it may not be allowed to monitor residents with cameras. However, with the help of radars it could still be detected when an emergency occurs such as the fall of an elderly person. In this report we classify human actions based on radar measurements. We compare the k-nearest neighbours (KNN), support vector machine (SVM), bootstrap aggregation (bagging), and convolutional neural network (CNN) algorithms. For the KNN, SVM and bagging algorithms, we use and compare different procedures of feature creation and selection. We create features both based on the centroid and bandwidth of the spectrogram as well as features based on Chebyshev polynomial expansions of the cadence velocity diagram (CVD). We apply feature selection in combination with k-fold cross validation and data augmentation. We find that the bagging algorithm in combination with data augmentation is able to achieve the highest accuracy of the considered models.

2 Methodology

2.1 Data

We use the Radar signatures of human activities dataset of the University of Glasgow [1]. The dataset consists of a set of measurements done over several time windows. The data is collected using a FMCW radar at the operating frequency of 5.8 Ghz, a bandwidth of 400 MHz and 1ms chirp duration. The measured data consists of 1754 individual measurement that are provided as a vector of 1280000 complex values. Each individual measurement is of one specific action of a person. The actions are: walking, sitting down, standing up, picking up an object, drinking water, and falling. Each vector is reshaped into chirps, resulting in a 128x10000 complex valued matrix. We apply a Fourier transform to get the range-time data. We then apply a moving target indication(MTI) filter to filter out clutter from the measurement. We then use the filtered range-time data to get spectrograms represented by a 800x481 complex valued matrix.

2.2 Feature creation

The data-set consists of a 800x481 matrix for each sample. This makes that each sample contains 384800 complex values.This amount of values is too large for most conventional machine learning algorithms. Most individual values also provide little to none information about the specific class of the sample. Therefore, the data needs to be transformed into data that is suitable for our algorithms. This will be done by creating so called features of the different samples of data. These features should represent discriminating properties. In feature creation, the goal is to create and select features that have these properties. We want to find features that say as much as possible about the class the sample belongs to. In this document different methods are proposed for creating and selecting features and the results are compared.

2.2.1 Centroid

The first features extracted from the spectrograms are the centroid and bandwidth as used in [2]. The centroid of the spectrogram is an indicator for the centre of gravity of the micro-Doppler signature. The bandwidth gives an estimate of the bandwidth around the centroid [3]. The centroid and bandwidth are given by

$$f_c(j) = \frac{\sum_i f(i)F(i,j)}{\sum_i F(i,j)} \tag{1}$$

and

$$B_c(j) = \sqrt{\frac{\sum_i \left(f(i) - f_c(j)\right)^2 F(i,j)}{\sum_i F(i,j)}}, \tag{2}$$

respectively. $F(i,j)$ denotes the value in the ith Doppler and jth time bin of the spectrogram. We divide the spectrogram into equally sized segments and compute the centroid and bandwidth of each segment, we then use the first four moments, i.e., mean, variance, skewness, and kurtosis of these segments as features.

2.2.2 CVD

In [4], the Cadence Velocity Diagram (CVD) is purposed to extract micro Doppler features. The CVD is a measure on how often certain Doppler velocities repeat. The cadence of some moving object can therefore be estimated by the CVD. The CVD is defined as the Fourier transform over each frequency bin of the time Doppler velocity data. It can be calculated using (3) which corresponds to taking the discrete Fourier transform of the absolute value of the short-time Fourier transform(STFT).

$$\Delta(\nu, \varepsilon) = \sum_{k=0}^{K-1} |\,\mathrm{STFT}(\nu, k)| e^{-j2\pi k\varepsilon/K} \tag{3}$$

Because some of the activity classes of interest do have a characteristic cadence, the CVD could be an intuitively good representation of the data in relation to their corresponding classes. An example of a characteristic cadence would be the in the time Doppler frequency plot visible harmonics of someone walking. These harmonics, generated by moving arms, legs, etc. will be represented as straight lines in the CVD, with the location of these lines corresponding to the cadance frequency. To translate this into features, the average of the cvd values are projected on the time axes and frequency axes, resulting in a single line. A well performing single value feature would be to take the M=2 strongest peaks from both of these lines [4]. Other proposed techniques for extracting features out of the CVD profile are: Maximum of main peak, energy of the main peak, Intensity of the main peak in the CVD and the Most significant Doppler frequency in the CVD [5].

2.2.3 Chebyshev polynomials

For image representation, orthogonal moments can be used. The general idea of representing an image in moments is to project an original image function on a certain space formed by a set of basis functions [6]. This can then be represented by a feature vector containing the moments belonging to the basis functions. With the moments and basis functions together, the image could be retrieved. The quality is defined by the order and type of orthogonal moments and basis function used [6]. The idea behind this technique on generating features is that you still have relatively much of information about the image compressed to just a few values.

There are several orthogonal moments methods used in the field of image representation for the creations of feature vectors [6]. For this project, the Chebyshev moment expansion will be used to create a feature vector on the micro Doppler data. At first, the CVD is calculated by taking the Discrete Fourier transform on the STFT of the measured data. Then, the moment expansion is done up to a certain order. The expansion is given as

$$T_{l,h} = \frac{1}{\tilde{\rho}(l,L)\tilde{\rho}(h,H)} \sum_{x=0}^{N_{\mathrm{DFT}}-1} \sum_{y=0}^{N_{\mathrm{CVD}}-1} \tilde{t}_l(x)\tilde{t}_h(y)|\Delta(y,x)|, \tag{4}$$

where $l + h$ denotes the order. $\tilde{t}_l(x)$ is given by

$$\tilde{t}_l(x) = \frac{t_l(x)}{\beta(l,L)}. \tag{5}$$

in which $t_l(x)$ denotes the Chebyshev polynomial of order l evaluated at x and is given by

$$t_l(x) = l! \sum_{k=0}^{l} (-1)^{l-k} \left(\begin{array}{c} L-1-k \\ l-k \end{array} \right) \left(\begin{array}{c} l+k \\ l \end{array} \right). \tag{6}$$

$\beta(l,L)$ is a scaling factor introduced for numerical stability and computed as

$$\beta(l,L) = L^l. \tag{7}$$

In (4), $\rho(l,L)$ denotes the amplitude factor and is given as

$$\rho(l,L) = (2l)! \left(\begin{array}{c} L+l \\ 2l+1 \end{array} \right). \tag{8}$$

The final feature vector is then a collection of moments, denoted by

$$\boldsymbol{F} = \left[T_{0,0}, T_{0,1}, \ldots, T_{l,h}\right]. \tag{9}$$

The usage of Chebyshev moments in comparison to pseudo-Zernike moments as proposed in [7] has the advantage to be directly defined on a discrete set, which makes them easier to implement. Chebyshev moments generally perform better when using a K-NN algorithm and Gaussian SVM algorithm as well [8].

2.3 K-nearest neighbours

The first considered algorithm is the KNN algorithm, as described by, e.g,, [9]. KNN is a non-parametric, supervised machine learning model. The KNN algorithm uses the euclidean distance between data to make classification predictions. The key assumption of the algorithm being that data points corresponding to a certain class have a lower euclidean distance within the class than compared to datapoints of other classes. If we make a classification with a KNN model, we look at the k closest data points to the data point we want to classify and classify the data point as the most frequently occurring class within those k nearest neighbours. In this report, the number of neighbours is set to be seven.

2.4 Support vector machine

In the support vector machine algorithm, boundaries between each classification are created. The created boundaries are called hyper planes. The algorithm places these hyper planes as far as possible from the nearest data points of the classes it tries to separate. The closest points are then called the support vectors of the algorithm. The SVM algorithm provides predictions by looking at which side of the hyperplane the new datapoint is positioned. SVM algorithms are binary classification algorithms. In this report, they are extended to multiclass classification algorithms by combining individual SVM algorithms in a tree-based fashion. The hyper planes of the SVM algorithm are always linear by construction. However, we can introduce non-linear relations by making use of a kernel function. In this report, we always use the Gaussian kernel when using SVM algorithms.

2.5 Bootstrap aggregation

In bootstrap aggregation as described by [10], we use ensemble learning. Ensemble learning algorithms are a class of algorithms in which we combine multiple weak learners such as decision trees in order to make more powerful regression or classification predictions. In the bagging algorithm specifically, we bootstrap many replicas of the original dataset and fit decision trees on these replicas. Each of the replicated dataset is bootstrapped to be the same size as the original dataset. I.e., we sample N samples from the original dataset with replacement, with N being the size of the original dataset. The descicion trees fitted on the replicas of the dataset could also be replaced by random forests to improve performance [11]. After fitting all weak learners, predictions can be made by combining predictions of all fitted weak learners. The predictions are weighted by using the error that each individual weak learner makes in training. Bagging reduces the likelihood of overfitting, as it will inherently give a low weight to features that explain little of the data [12].

2.6 K-fold cross validation

To reduce bias and variance in our estimates, we apply k-fold cross validation. In k-fold cross validation, we first split the data in a train and a test set. The train set is then used to train and tune the model, while the test set is kept unseen until the model is fully developed and then used to evaluate the final performance. In k-fold cross validation, we split the training data into k sets. If we then want to evaluate the performance of the model for, e.g., a certain hyperparameter, we train the model k times, each time leaving one of the sets out as a validation set. after each of the k times of training the model, we use the left out set to validate and save the performance. In the end, we can take the average of the performance in the k folds as an estimate of the expected performance of the model with that hyperparameter.

2.7 Data augmentation

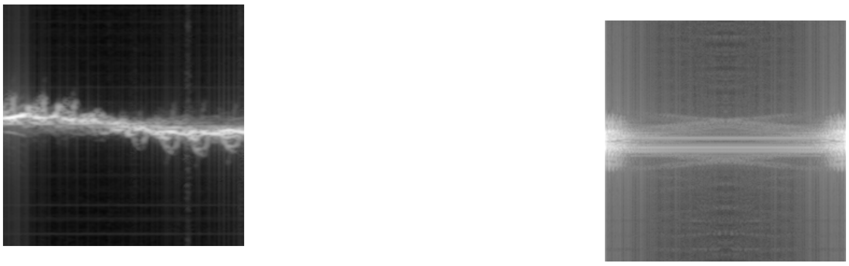
Machine learning algorithms tend to perform better when they have more training data available. In data augmentation, we use the original data to generate extra data. As we use spectrograms to generate all features used in the machine learning algorithms, we will apply data augmentation directly to the spectrograms. When working with spectrograms in data augmentation, it is important to keep the physical meaning of the spectrograms in mind. We cannot just flip the spectrogram in time, as the physical meaning of, e.g., a fall will be lost. Therefore, we only flip all data once over the Doppler axis. This way, the spectrogram still holds the physical representation of the action, just reversed in direction.

2.8 Feature selection

To reduce the amount of features that the classification framework takes into account, feature selection is applied. This way, we can achieve a more general model, consequently preventing overfitting of the model. Due to the fact that the model needs to take less features into account after feature selection, the computational costs are reduced for training, validating and using the classifier. Four of the most used methods in feature reduction are brute-force, sequential forward selection(SFS), sequential backward elimination(SBE) and filter methods. In the case of brute-force, SFS or SBE, we need to choose a performance metric. Such a metric could be accuracy of the algorithm. In the brute-force method, all possible combinations of features are evaluated and the combination with the best performance according to the provided performance measure is selected. In SFS, a feature is added step by step, in each step the feature that increases the performance measure the most is selected. The SFS method stops when there is no feature available that improves the performance measure any further. In SBE, we start with all features and remove a single feature in each step. The feature that is removed is the feature that, when left out, improves the performance measure the most. The SBE method stops when there is no feature that, when removed, improves the performance measure any further. The brute-force method is computationally very expensive as we need to train models with all possible combinations of features. The brute-force method does however provide a global optimum. Both the SFS and SBE methods are computationally less expensive compared to brute-force, but can only provide a local optimum. Filter methods try to remove features that are highly correlated to other features. These features will likely provide little additional explanation when the other correlated features are present.

2.9 CNN

The data, as used before, can be transformed into images. In [13] a CNN is used for pattern recognition based on micro-doppler data that is translated into image data. In this case, already existing convolutional neural networks are used like Alexnet and GoogleNet. For this project an implementation of a pretrained Googlenet is made. GoogleNet is a convolutional neural network that is 22 layers deep, where the input is a 224x224x3 RGB image. To get the data into the right input format; first, the Time Doppler-velocity spectrogram is transformed to a greyscale image as greyscale images have been shown to have improved classification accuracy over coloured images when used for radar image classification [14]. The greyscale image is then rescaled to the desired 224x224 pixels. This greyscale image is then copied over three dimensions as the input for the neural net required a 3 dimensional RGB image. This is done for both the spectogram and the CVD. Two random samples of the greyscale image used as input for GoogleNet are shown in Figure 1.



(a) Greyscale image of the Doppler velocity spectogam of a person walking

(b) Greyscale image of the CVD of a person walking

Figure 1: Image data used as input for GoogleNet

3 Results

To evaluate the proposed feature extraction methods and machine learning algorithms, we set up an empirical experiment. The Data and the pre-processing are described in subsection 2.1.The data consists of 1751 radar measurements of one of the following six actions: walking, sitting down, standing up, picking up an object, drinking water, and falling. We randomly split the data into a train and a test set with 85% of the dataset belonging to the train set, which is used for training and validating of the model. The other 15% of the data is kept as unseen test set. For the KNN, SVM and bagging algorithm, we train and test each model eight times. We first train each model four times with Chebychev polynomials up to order 16, of which one time without extra processing. One time with SFS feature selection with accuracy as performance measure. k-fold cross validation is applied in each step of the SFS feature selection method. One time without feature selection and with data augmentation, and finally with both SFS feature selection and data augmentation. These four cases are repeated for the centroid features, in which we use 10 segments. For the CNN, we have four test cases. The first being that we train the CNN based on images of the spectrogram. In the second case we train on images of the spectrograms, but apply data augmentation as well. In the third case, we train the CNN on images of the CVD. In the fourth case, we again train on CVD images, but also use data augmentation. The results of all experiments in terms of accuracy on the unseen test set are provided in Table 1.

Table 1: Test set accuracy scores of different models under with different feature processing. The processing abbreviations, none: no processing, FS: feature selection, performed by SFS, DA: data augmentation, FS+DA: feature selection and data augmentation. CNN has its own part in the table as it creates its own features based on an image of the spectrogram and CVD.

Features	Chebychev				Centroid			
Processing	None	FS	DA	FS+DA	None	FS	DA	FS+DA
KNN	0.7300	0.7338	0.6312	0.6521	0.6730	0.8441	0.6121	0.8023
SVM	0.8213	0.7643	0.6027	0.6616	0.7719	0.8365	0.6407	0.7681
Bagging	0.7681	0.7452	0.6616	0.6730	0.8707	0.8707	0.9715	0.9030
Input	Spectrogram				CVD			
CNN	0.8384	-	0.7502	-	0.6350	-	0.5878	-

The results show that for KNN, the centroid based features with feature selection performs best with an accuracy of 84.41%. For both the Chebyshev polynomial based features and the centroid based features, data augmentation seems to decrease performance. an explanation for this decrease in performance is that since KNN compares euclidean distance and we are flipping the spectrograms, we create more overlap of the classes in the feature space.

The SVM algorithm also performs best in the case of centroid based features combined with feature selection, with an accuracy of 83.65%. The Chebyshev polynomial based features without any additional processing also seem to perform wel with an accuracy of 82.12%. The same relation with data augmentation is found as in the case of KNN.

For the bagging algorithm, the centroid based features all tend to perform well. The centroid based features combined with data augmentation performs best of all tested models with an accuracy of 97.15%. We see that in most cases, feature selection increases the accuracy of the model, except for the bagging algorithm. This is likely caused that the bagging algorithm is very robust to overfitting and tends to find relations between features and the data that may not be found by simpler algorithms such as KNN and SVM [12].

As for the CNN, the highest accuracy is achieved when using the spectrogram image and no data augmentation. The results show that, in both cases, the argumentation of data does not increase the performance of the CNN. The CVD performs significantly worse compared to the spectrogram. A possible explanation for this result would be the resolution of the images. The lower frequencies are now blurred and hard to distinguish as can be seen in figure 1b. These lower frequencies do contain the cadences of some class specific movement. The poor distinguishability between these frequencies might cause the worse performance.

3.1 Final pipeline

We base the final classification pipeline on the results as given in Table 1. The best performing model is the bagging model with data augmentation and without feature selection. Therefore the final pipeline is to create spectrograms from the radar data, segment the spectrogram into ten segments and compute the centroid and bandwidth of each segment. Then within each segments, the first four moments, e.g., mean, variance, skewness and kurtosis of both the centroid and the bandwidth are computed. This results in 80 features for each measurement. These features will then be provided to the bagging model that has been trained on the complete and augmented dataset. The results of this pipeline are given in the excel file provided with the report.

4 Discussion and conclusion

In sum, we have compared four different machine learning algorithms with the application of human activity classification. We have considered KNN, SVM, bagging and CNN. We have evaluated the effect of multiple data pre-processing methods such as the creation of spectrograms and CVDs. We have applied data processing methods such as feature selection in combination with k-fold cross validation and data augmentation. We found that the bagging algorithm, combined with data augmentation performs the best of all considered models with an accuracy score of 97.15% on an unseen test set.

Future work could be to extract features out of the CVD instead of only applying Chebyshev polynomial decomposition. E.g., the maximum of the main peak of the CVD, the energy of the main peak or the most significant Doppler frequency in the CVD could be extracted. It would also be interesting to combine these CVD features with the centroid based features.

For the CNN, the input data could be optimised. The resolution is scaled down to get to the 224x224 pixel image, as required by GoogleNet. This down scaling has as downside that some data gets lost that was present before the downscale in resolution. Especially in the CVD images the lower frequencies are hardly visible. This can also bee seen in the results. Zooming in on those specific frequencies before scaling could be a solution to keep adequate resolution on the lower frequencies of interest. However, this still needs to be tested and evaluated.

References

[1] F. Fioranelli, S.A. Shah, H. Li, A. Shrestha, S. Yang, and J. Le Kernec; Radar signatures of human activities University of Glasgow, 1993.

[2] F. Fioranelli, M. Ritchie, and H. Griffiths, "Centroid features for classification of armed/unarmed multiple personnel using multistatic human micro doppler," *IET Radar, Sonar & Navigation*, vol. 10, no. 9, pp. 1702–1710, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1049/iet-rsn.2015.0493

[3] A. Balleri, A. Al-Armaghany, H. Griffiths, K. Tong, T. Matsuura, T. Karasudani, and Y. Ohya, "Measurements and analysis of the radar signature of a new wind turbine design at x-band," *IET Radar, Sonar & Navigation*, vol. 7, no. 2, pp. 170–177, 2013.

[4] S. Bjorklund, T. Johansson, and H. Petersson, "Evaluation of a micro-doppler classification method on mm-wave data," in *2012 IEEE Radar Conference*. IEEE, 2012, pp. 0934–0939. [Online]. Available: http://ieeexplore.ieee.org/document/6212271/

[5] H. Li, A. Mehul, J. Le Kernec, S. Z. Gurbuz, and F. Fioranelli, "Sequential human gait classification with distributed radar sensor fusion," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 7590–7603, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9306810/

[6] S. Qi, Y. Zhang, C. Wang, J. Zhou, and X. Cao, "A survey of orthogonal moments for image representation: Theory, implementation, and evaluation," *ArXiv*, 2021, publisher: arXiv Version Number: 3. [Online]. Available: https://arxiv.org/abs/2103.14799

[7] C. Clemente, L. Pallotta, A. De Maio, J. J. Soraghan, and A. Farina, "A novel algorithm for radar classification based on doppler characteristics exploiting orthogonal pseudo-zernike polynomials," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 417–430, 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7073502/

[8] L. Pallotta, M. Cauli, C. Clemente, F. Fioranelli, G. Giunta, and A. Farina, "Classification of micro-doppler radar hand-gesture signatures by means of chebyshev moments," in *2021 IEEE 8th International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, 2021, pp. 182–187.

[9] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00031305.1992.10475879

[10] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, pp. 123–140, 1996.

[11] ——, “Random forests,” *Machine Learning*, vol. 45, pp. 5–32, 10 2001.

[12] T. G. Dietterich, “Machine-learning research,” *AI Magazine*, vol. 18, no. 4, p. 97, 1997. [Online]. Available: https://ojs.aaai.org/index.php/aimagazine/article/view/1324

[13] S. A. Shah and F. Fioranelli, “Human activity recognition : Preliminary results for dataset portability using FMCW radar,” in *2019 International Radar Conference (RADAR)*. IEEE, 2019, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/9079098/

[14] W. Taylor, K. Dashtipour, S. A. Shah, A. Hussain, Q. H. Abbasi, and M. A. Imran, “Radar sensing for activity classification in elderly people exploiting micro-doppler signatures using machine learning,” *Sensors*, vol. 21, no. 11, p. 3881, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/11/3881