MPA-MLF - Final project

Classification of room occupancy

Date: 20.3.2023

1 Task description

The project's task is to classify the number of persons in the room from 60 GHz signal transmission. The dataset consists of snapshots of signals in the delay-Doppler domain that represent the reflections from the targets (humans/machines) at some distance from the receiver moving with some velocity. To explain, the faster the object moves, the higher the Doppler frequency shifts it generates. The farther from the receiver the person is, the higher the delay it generates. Examples of delay-Doppler snapshots are shown in Fig. 1 for one, two, and three persons in the room. Note that these examples represent almost ideal records, while noise, reflections, missed targets, etc may distort the real records. The dataset is created in figures in .png format, where each file represents one two-dimensional snapshot in the delay-Doppler domain.

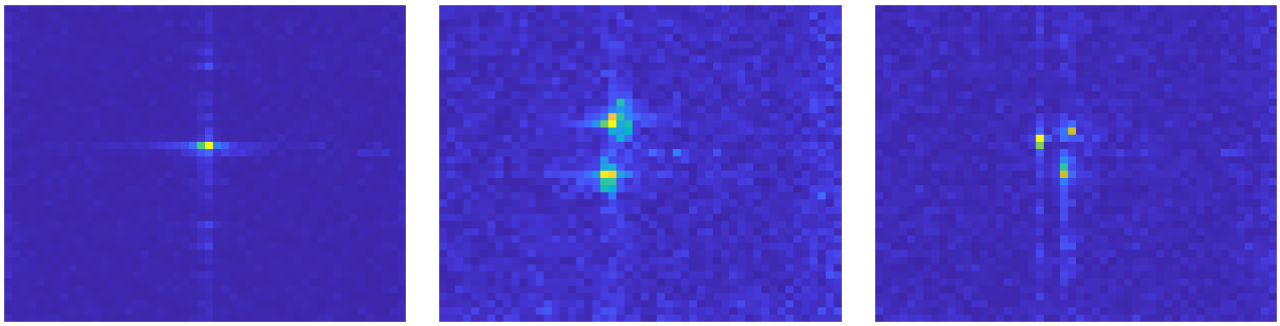

Figure 1: Examples of IDEAL snapshots for one (left), two (middle), and three (right) persons in the room. Note that many of the real records will not be so perfect. But it's life...

The goal is to classify the number of persons present in the room from a given snapshot. Thus, one of the following four classes may be valid:

- Machine only, zero person in the room
- One person in the room
- Two persons in the room
- Three persons in the room.

You are provided with the test and train files as you are used to from your last project. All files can be found here: Link to the Datasets . You are provided with four different files:

1. **x_train.zip** - training dataset,
2. **y_train.csv** - ground truth values for the training data
3. **x_test.zip** - data for testing
4. **submission_example.csv** - example of data format that is accepted by kaggle

NOTE: be careful about the numbering. Input images are numbered from 1 and labels from 0

2 Steps

Your task is to make a model for classification on the dataset described above. We do not prescribe the type of ML model. You can use any architecture we discussed during the semester. Try to achieve the highest testing accuracy possible. Tune your model's performance using any techniques we discussed, like Data augmentation, Regularization, Batch normalisation, etc. You can use any Hyperparameter tunning algorithm to find the best hyperparameters and the model structure (but it is not required).

3 Submission and grading

You are required to work in pairs. Please make your pairs on your own and write down the teams in the following document: Link to the document Could you create your teams until **24.3.2023**. You are expected to write a report(a maximum of 10 pages) that describes the essential steps and strategies you used to build a model with the highest possible testing accuracy.
Your solutions will be submitted in the following different ways:

- **Report, e-learning**. You will upload your report into the e-learning. Please upload your report in .pdf format
- **Model predictions, kaggle** You are required to test your results in the Kaggle competition, link: Link to the competition . You are limited to 15 submissions a day, so please start early. The correct format of your submissions can be seen in the *submission_example.csv* file. The score you see is calculated using only 50 % of randomly selected images (The Public Leaderboard). The score of the other 50 % (The Private Leaderboard) is hidden for you and will be displayed after the submission deadline. Select two of your predictions before dl to evaluate the results on the private leaderboard. If you don't do so, they will be selected randomly.
- **Code, GitHub, e-learning**. Please create a new folder in the repo you have used for MPA-MLP. Do not push the input dataset to your GitHub repo. **Link the folder with the final code to your report**.

The deadline for the report submission is **10.4.2024**. The deadline for submission of the results to Kaggle is **8.4.2024**. After the Kaggle dl is passed, the private leader board score will be displayed. Please include scores from both the public and private leaderboards in your reports.

4 General comments

- The report should have all the necessary formalities that a report of a similar type has (Introduction, problem description, the main body of your work, conclusions…., all figures and tables should have labels and numbering and should be referenced in the text, etc..). You should be familiar with this from your bachelor's thesis.
- Do not include the screenshots of your code in the report. If you wish to describe some of your algorithms, use pseudo-code/flow charts.., Do not describe well-known algorithms.
- Don't present your work in the report as plain text. Use graphs, figures, and tables to show and present the relevant information.
- You are required to do all of your coding in Python. You don't have to do all coding from scratch, but you are all allowed to use Python's libraries and frameworks (For instance, *Keras*, *PyTorch*, *Scikit-learn.*.).
- Usage of Google Collab is strongly recommended but not required.
- Ensure you properly describe your work in your report; we are more interested in the process of your work than in a correct result.