Department of Mathematics and Computer Science
Interconnected Resource-aware Intelligent Systems Research Group

Towards domain agnostic Wi-Fi CSI gesture classification

Master's Thesis

Yvan Putra Satyawan

Supervisors:
Your First Committee Member
Your second Committee Member
Your Third Committee Member

Draft version

Eindhoven, February 2023

Abstract

THIS IS MY ABSTRACT

Abstract needs to be completed

Preface

We choose to write this thesis. We choose to write this thesis, in this semester and do the other things, not because they are easy, but because they are hard.

actually write a proper preface

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this ever more connected climate that we find ourselves in, IoT devices everywhere are adding little conveniences to our every day lives. With IoT devices becoming ever more common and reaching a forecasted 27 billion devices by 2025 [5], the dream of ubiquitous computing and sensing is transitioning from a mere dream to the reality of our every day lives. Additionally, approximately 19% of new devices bought in 2020 also utilize some form of Wi-Fi radio for communications with a forecasted increase to 24% by 2025.

It is clear that this trend towards integrating computing technology into everyday objects will only accelerate in the future. The increased convenience and efficiency may be the biggest boon of such technologies. For example, smart thermostats can predict heating requirements and adjust accordingly, leading to lower heating costs in a house while maintaining the convenience of having a well heated space.

With all these connected devices becoming and edge computing ability comes ubiquitous sensing, enabling new modalities of interaction and improving data collection and analytics. It is now possible to envision households with complete presence detection coverage, for example, through the use of smart motion sensors, enabling increased efficiency by intelligently identifying which rooms require heating and lighting and which do not. Always-on voice control systems, such as Amazon Alexa and Google Assistant speakers are also increasingly common, making a connected AI-assistant only one call away. The always-connected nature of these sensors also enable the gathering and analysis of vast amounts of user data, potentially providing valuable insights into various aspects of our lives.

One challenge that has continued to plague ubiquitous devices is the lack of a ubiquitous user interface which does not require input devices. It is not too common to use the end device itself as the input. Even in the case of voice control systems, a dedicated smart speaker is still required and must be placed in every room from which interaction is desired. For example, in the case of smart lights or smart thermostats, the end device would be the

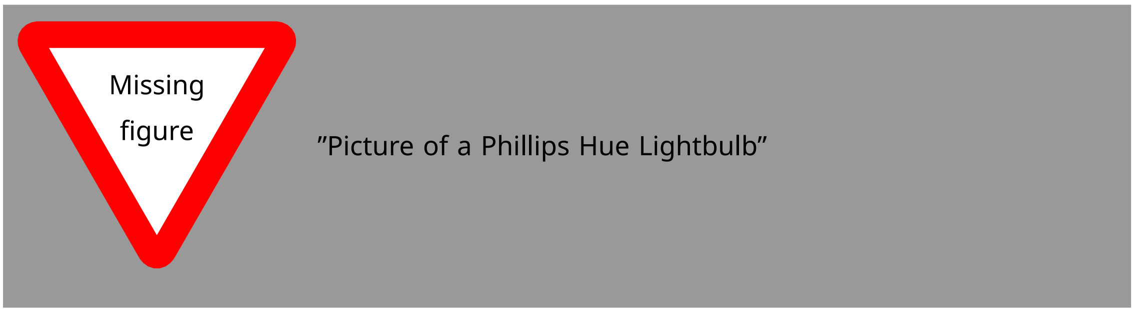

Figure 1.1: A Phillips Hue Lightbulb, one example of an IoT smart lightbulb with an integrated Wi-Fi Radio

light bulbs and heating system, respectively. In both cases, a separate control unit, the light switch or thermostat, respectively, is still required.

With these challenges in mind, Wi-Fi-based sensing provides one potential solution. Many IoT devices already contain some sort of Wi-Fi radio, such as the Phillips Hue Light bulb in Figure 1.1, and it would be a rather safe assumption to make that spaces with IoT devices would also have some sort of Wi-Fi infrastructure in place as well. With this in mind, the idea of a gesture-based interface based around Wi-Fi becomes rather appealing. The use of Channel-State Information (CSI) also enables of finer-grain signals to be extracted from consumer-grade Wi-Fi radios, enough to enable reliable gesture recognition [1]. This method does, though, suffer from the domain-shift problem, achieving the best accuracy only in cases with a prediction model fine-tuned to a specific person and environment is used.

With this thesis, we aim to explore the use of CNN architectures and domain-shift mitigation methods to improve the state-of-the-art in Wi-Fi CSI-based gesture classification. Specifically, we will look at using preprocessing methods to transform the input signal from CSI into an image using table-to-image and signal-to-image transformations, the use of traditional signal processing algorithms to process the incoming CSI signal, and the use of Reinforcement Learning (RL) to perform domain auto-labeling and provide the CNN classifier with additional information.

1.1 Context and Background

IoT devices are without a doubt increasingly prevalent in everyday life. The Atlas building at the Technical University of Eindhoven (TU/e), for example, uses centrally networked lights for all of its lighting fixtures powered through Power over Ethernet (PoE) and this is at least partially credited as a reason the building has the best efficiency of any academic building in Europe when it was constructed [4].

All over the building, presence detection, in the form of motion detection sensors, is also used to automatically set appropriate lighting levels for each room. This is just one example of how ubiquitous computing and sensing has now entered the mainstream and is no longer a dream of a few enthusiasts and IoT evangelists. Developments in AI and big data processing has also made the usefulness of ubiquitous computing much more evident, legitimizing its use in everyday objects.

Finally, the deployment of 5G networks in densely populated areas is working towards enabling faster speeds and lower latency, essential in many ubiquitous computing and sensing applications.

With these advances, the question has shifted towards what sort of interface we should utilize to provide an always-available non-intrusive experience for the users. One possible solution is gesture-based interfaces. With any ubiquitous computing and sensing product, especially in the consumer space, minimal setup on the user's part is desired; otherwise the product will not become something which is widely accepted and used.

Wi-Fi gesture recognition can solve these issues, providing a gesture-based interface requiring potentially zero additional setup requirements. As a bonus, this would also be a low-cost solution which many IoT devices already having the necessary hardware for regardless.

There also exists a task group for Wireless Local Area Network (WLAN) sensing, called 802.11bf, within the IEEE 802.11 working group, the group which sets the standards for WLAN, with members from large companies including Huawei, Qualcomm, and Meta [3]. This shows there is genuine interest in the industry to utilize WLAN for these purposes. With an approval date set for September 2024, it is clear that WLAN sensing is not just some theoretical possibility confined to a lab, but rather a very real technology that may soon become widespread.

To make such an approach possible, we utilize machine learning (ML) to process the incoming CSI data and classify user gestures. Wi-Fi technology, when boiled down, is just a really complex radio and what is radar but a different form of very complex radio. It naturally, or not so naturally, follows then, can Wi-Fi be used for remote sensing analogously to radar technology? The answer to this question, according to [1] and [2], the answer is a resounding yes!

However, ML suffers from degraded performance when faced with domain-shifts. When dealing with new, unseen users and environments, gesture classification accuracy degrades significantly. As such, factors to mitigate this domain-shift problem are required in any citation needed implementation outside of a pristine laboratory setting.

1.2 Motivation

There are significant potential commercial and practical benefits to enabling domain-agnostic Wi-Fi gesture classification.

From a commercial perspective, the 802.11bf standard, which standardizes the hardware technology required for Wi-Fi sensing, is on the cusp of being released. If we are to take advantage of these new technologies, it is imperative that its practical applications be researched.

I don't like the wording here

Wi-Fi sensing is also unique in that the hardware for it is already commercially available and widespread, though not necessarily standardized. As such, it is in a unique place where it is an almost completely software-based solution for preexisting, mass-adopted hardware enabling completely new modes of interaction. This makes it unique and, ultimately, much more commercially desirable as a technology to be adopted.

With respect to domain-agnosticism, the motivating factor is that the largest hurdle for Wi-Fi gesture classification is its loss of performance in new, unseen domains. A model that cannot adapt to such issues will inevitably be nonviable for commercial adaptation lest the end-user be required to perform the same gesture hundreds of times in various positions after installing every single smart IoT device.

We also wish to further the state-of-the-art in domain-agnostic models. The results of this thesis is not only applicable to the field of Wi-Fi gesture classification. Domain-agnostic systems are required for many industry applications, such as [what]. Figure out what The conclusions of this thesis will hopefully be able to advance the field and provide new insight into what future avenues of research may end up being fruitful.

Finally, a Sci-Fi future where all your devices are controlled through your thoughts might seem like a dream, but in reality, we are not too far from this future. The use of Wi-Fi sensing to detect gestures is one step closer to this futuristic dream and in addition to all the potential commercial and practical benefits this technology could bring, the sheer "coolness" of the technology should not be underestimated as a motivation to perform research.

A bit less academic, but hopefully just as viable as a motivation

1.3 Problem Statement

Models already exist to classify gestures through Wi-Fi CSI data. Ideally models are solely influenced by those factors which contribute to the correct classification of the gesture, this is, in reality, not the case. These models are influenced by "domain factors" such as the subject performing the gesture and the environment where this is taking place.

Replace "this is taking place" with something else

These domain factors cause feature domain shifts between the training data and the data encountered during actual use or inference despite both domains containing the phenomenon of interest, for example the gesture we wish to classify. Due to these shifts, performance of the model is degraded. citation needed

There exist proposals to mitigate this degradation including using very large datasets. This is one method especially championed by large companies such as Tesla and OpenAI which have vast resources at their disposal. However, the gathering of such large datasets is only feasible for specific scenarios. In the case of Tesla, for example, having a large fleet of vehicles capable of recording what is essentially supervised training data makes it possible to collect the vast amounts of data required to build a dataset usable for the training of self-driving vehicles. [specific citation needed] Various OpenAI projects, on the other hand, simply scrape large amounts of websites and collect these as part of their dataset. [specific citation needed] Within the scope of our research, due to its nature, such widespread data collection methods are non-viable, and we must resort to more novel approaches towards data-agnosticism.

- It is thus interesting to be able to create a domain-agnostic model
- Most research so far relies on a ground truth domain label to be provided
  - We then use a "discriminator" network which predicts domain labels
  - Adversarial training procedure is then used in a min-max game [citation needed]
- Ma et al. with their CNN → state machine neural network → RNN, trained via RL.
  - This doesn't require domain labels
  - Activity transitions are required though and state machine nn is actually kinda weak when not given enough temporal information + accurate action transition labels
- We hypothesize that we can extend the RL component of Ma et al. to facilitate domain auto-labeling and eliminate the use of a state machine nn by using signal-to-image preprocessing methods.
- To what extent can domain independent feature extraction be achieved through the use of this method, measured by performance metric values in a domain factor leave-out cross validation setting?
- To what extent can a reinforcement learning agent utilize the latent signal representation produced by the CNN to accurately produce a latent representation of the domain space?
- To what extent does changing the domain space latent representation from one-hot to a probability measure change the domain independent feature extraction achieved, through ...?
- To what extent can the signal-to-image transformation replace the state machine nn presented in ma et al.?

Chapter 2

Literature Review

This is the first real chapter, this depends on your thesis structure.

Chapter 3

Methodology

- The learning agent will, given the latent feature embedding produced by the CNN as its environment observation, produce a latent embedding of the domain space as a 1D vector as its action
- an FCN is used as the classifier with two groups of two heads

Chapter 4

Conclusions

Write your conclusions here.

Bibliography

[1] Fadel Adib and Dina Katabi. "See through walls with WiFI'. In: *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*. 2013, pp. 75–86.

[2] Kevin Chetty, Graeme E Smith, and Karl Woodbridge. "Through-the-wall sensing of personnel using passive bistatic wifi radar at standoff distances". In: *IEEE Transactions on Geoscience and Remote Sensing* 50.4 (2011), pp. 1218–1226.

[3] Rui Du et al. "An overview on IEEE 802.11 bf: WLAN sensing". In: *arXiv preprint arXiv:2207.04859* (2022).

[4] Technical University of Eindhoven. *Atlas most sustainable education building in the world Atlas most sustainable education building in the world Atlas most sustainable education building in the world*. Apr. 2019. URL: `https://www.tue.nl/en/our-university/tue-campus/buildings/atlas/atlas-most-sustainable-education-building-in-the-world/`.

[5] Mohammad Hasan. *State of IoT 2022 Number of connected IoT devices growing 18% to 14.4 billion globally*. URL: `https://iot-analytics.com/number-connected-iot-devices/` (visited on 05/18/2022).

[6] Chip Huyen. *Designing Machine Learning Systems*. USA: O'Reilly Media, 2022. ISBN: 978-1801819312.

Appendix A

Plan

Complete by	Task
30 February	Initial infrastructure setup. This involves building the train-validate-test loops, building the data ingest/transformation pipelines, and building the model-building pipelines. This is feasible since much of the code will be taken from the IRIS Seminar project.
15 March	Integration of all modules as well as integration of hyperparameter optimization and training-tracking code complete. Initial training/debugging of the network can begin.
30 March	Parallelization of reinforcement-learning and deep-learning components of the network are complete. This involves making it possible to run the reinforcement-learning and the deep-learning models on separate computers during training, if this improves performance.
15 April	Initial results should be done by now, bugs will be found and “beta-testing” of the framework is in full swing. All results at this point are taken with a massive grain of salt since something will cause the results to be wrong, speaking from experience.
30 April	Initial good results should be available. Deeper investigations through hyperparameter optimization and changes to the model architecture should now start or is already ongoing.
30 May	Final results from the experiments should be completed.
30 June	First draft of the paper is done.
7 July	Second draft of the paper is done.
14 July	Final draft of the paper is done.
21-28 July	Somewhere in these two weeks, the thesis defense should take place.

Table A.1: Project timeline by task completion date

Table A.1 provides a timeline of the project and planning on when each part of the project should be completed by.

Appendix B

Risk Assessment

Risks identified for each phase of the plan and mitigation options are investigated in this section. Risk mitigation strategies are built from the author's previous experience working in an AI research lab and as a data scientist and software developer in industry. Additionally, strategies are also developed from theory learned in his computer science bachelor's.

Initial infrastructure setup During the initial infrastructure setup, the entire pipeline will be developed. This includes a modular data-ingest/transformation pipeline and model-building pipeline. Risks include data availability/usability and bugs in the data pipeline code. Additionally, the model will be completely modular and not prebuilt, increasing the chances of a bug appearing during this building phase but increasing flexibility of the model being investigated.

Data availability/usability refers to the fact that while the data is a public and published dataset, it is nonetheless quite large and it would not be feasible to download the entire dataset and place it on TU/e's HPC server. This means that some way to compress the data must be done. It is also possible that the transformed version of the data can be compressed more efficiently and this is what will end up being the dataset we work with for the majority of the project.

The data pipeline is also modular, allowing for data augmentation to be added "on-the-fly" instead of being hard-coded. This increases flexibility, but introduces the risk of bugs in unexpected circumstances. Mitigation factors include having written similar pipelines multiple times in the past and reuse of old, known-good code from the IRIS Seminar project and 2AMM10 Deep Learning course. Additionally, a similar approach has been use in previous research that we have completed, and we have significant experience in similarly modular pipelines.

Similarly, the model is built only at run-time, allowing for more flexibility and the possibility to fine-tune the model architecture using hyperparameter optimization techniques. This increases the chance that a good model architecture is chosen and strengthens the reasoning behind the chosen model architecture through empirical performance. Mitigation factors are the same as for the data pipeline.

Module Integration The most complex part of this infrastructure is to ensure that all modules work well together and there are no bugs in the hand-off step between modules.

To mitigate these factors, we take some advice from Chip Huyen's book Designing Machine Learning Systems [6]. To ensure that errors are not made during this integration, data flow will be closely monitored and visualized at every step through a UI, such that it is easy to see if anything went wrong at any step. The application of this will essentially be most of the interim steps being given some sort of output so we can visualize their result and track how data is transformed throughout the entire process.

Parallelization This is potentially unnecessary and may take up time that could be better used elsewhere. The idea is essentially that it might make sense to have the reinforcement-learning model and the gesture-classification model run on separate computers and having them communicate through some network.

Mitigation for this being unnecessary is providing only a limited amount of time to do this and the understanding that if this seems too difficult/may take too long, then we will immediately shelf the idea.

General bugs As with any software-based project there will inevitably be bugs in the code. Software engineering principle which lead to fewer bugs, such as proper use of debugging tools (but not test suites as we don't believe they will be necessary for a project with a limited scope such as this), will be used throughout work on this thesis. Additionally, use of "magic numbers" will be limited and as many parameters as appropriate will be assigned from variables.