UNIVERSITY OF L'AQUILA

MASTER THESIS

Human Movement Recognition using Deep Learning on Visualized CSI Wi-Fi data

Author:
Daria Butyrskaya

Supervisor:
Dr. Phuong T. Nguyen

Erasmus Mundus Joint Master Degree Programme on the Engineering of Data-intensive Intelligent Software Systems

Academic Year 2022/2023

Acknowledgment

First and foremost, I express my sincere gratitude to my advisor, Dr. Phuong T. Nguyen, for his invaluable guidance, unwavering support, and mentorship throughout the course of this research. His insights and patience have been instrumental in shaping this work.

A significant portion of my thesis focused on machine learning for channel state information human recognition research, and I had the privilege of testing the benchmark at Nokia under the supervision of Darbari Mayank (email: mayank.darbari@nokia.com) and Mikko Jarva (email: mikko.jarva@nokia.com). I am immensely thankful to both of them for their expertise, the resources provided, and their guidance. The collaboration and time spent with them have been both enlightening and rewarding.

I reserve the deepest gratitude for my family. To my parents, Tatyana and Vyacheslav, thank you for your unfaltering love, encouragement, and belief in me. Your sacrifices and unwavering support have been the backbone of my journey.

Lastly, to all those who have contributed directly or indirectly to this work and are not mentioned here, please know that your contributions are sincerely appreciated.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Preamble

Wireless signal transmission is an intricate process, significantly influenced by the environment within which it operates. Notably, the mobility of various elements within this environment, such as the parts of a human body, distinctly modifies the manner in which these signals are reflected. These alterations subsequently cause changes in Channel State Information (CSI) data captured by Wi-Fi routers [29].

Intriguingly, specific human behaviors can be detected through a meticulous examination of the data streams from CSI. These behaviors, representing diverse activities, can be identified by processing the data streams and juxtaposing them against predefined models. The recognition of these activities hinges on discerning patterns within the CSI data, reflecting the relationship between human movement and the variation in channel state information [14].

A variety of techniques have been developed to explore and understand these patterns, with machine learning emerging as the most popular and effective tool [7, 9, 10]. Machine learning techniques are harnessed to develop sophisticated models capable of correlating variations in channel state information with specific human movements. These correlations enable the prediction and identification of human activities based on changes in CSI data.

This research focuses on further exploring this intriguing intersection of human activity, wireless signal processing, and machine learning. It aims to provide a deeper understanding of these correlations and develop more effective models for human activity recognition.

More specifically, with this work we attempt to to explore new way of using the CSI data in Deep Learning tasks. That is by using the visualized amplitude of signals and correlate them to certain activities.

1.2 Motivation

Over the past ten years, there has been an expansion in the diversity of sensing technologies that are accessible for use in smart home health monitoring systems. In dense setups, standard technologies like passive infrared sensors (PIRs) [34] for motion detection and magnetic switches for tracking doors have typically been used, with commercial offers.

These sensors generate straightforward activation events that may be utilized to monitor activity and presence levels in the house. Numerous fundamental sensors are likewise regarded as ambient because they are not seen as intrusive by locals. These sensors produce relatively little data, despite the fact that their ambient nature is a strength [11].

Cameras and wearables are popular sensor technologies that can create richer data representations, but they can also seen as being excessively intrusive for various jobs. This lowers their total uptake as a result, and residents may alter their behavior if they are always aware that they are being watched.

Addressing these challenges calls for innovative solutions that can balance non-invasiveness and data-rich capabilities [34]. This thesis presents such a solution, exploring the use of Channel State Information (CSI) derived from ubiquitous Wi-Fi radio signals to facilitate human movement recognition in a smart home setting [11].

Instead of utilizing traditional sensor technologies or relying on potentially intrusive devices, this work proposes a novel approach that directly correlates recorded frequencies with specific movements. The technique harnesses vision techniques to classify the action performed, providing a data-rich yet non-invasive method for efficient health monitoring.

This unprecedented exploration not only addresses the limitations of existing technologies but also heralds a significant advancement in CSI Human Movement Recognition research. By opening new avenues in the field, the research seeks to maximize the potential applications and effectiveness of ambient sensing technologies.

1.3 Thesis Structure

This thesis is structured to address the above research objectives. Chapter 2 provides some background related the topics under consideration. The relevant literature is reviewed in Chapter 3. We describe in detail the proposed approach in Chapter 4. Afterward, Chapter 5 reports and analyzes an empirical evaluation with concrete experimental results. Finally, Chapter 6 sketches future work, and concludes the thesis.

Chapter 2

Background

In this chapter, we review the background related to systems based on Wi-Fi devices. After that, we go in further details on the recognition of activities from Wi-Fi signals, as well as capturing Channel State Information (CSI).

2.1 Device-based Systems

Before the advent of more modern sensing technologies, such as Wi-Fi signals, device-based systems were the predominant method used for activity recognition and health monitoring. With these systems, the user would be required to wear a device equipped with motion sensors [5], typically an accelerometer and a gyroscope.

The device would collect motion data from the user, forming the basis for understanding and interpreting their activity. The analysis and extraction of features from this sensor data could occur directly on the wearable device or be transmitted to a server for more robust processing. Leveraging supervised learning techniques, these features were then categorized, providing insights into the user's activities and behaviors.

This type of monitoring, often referred to as active monitoring, required constant user participation and compliance [22]. While effective in providing data for activity recognition, the need for a user to consistently wear a device could be burdensome. This posed a significant challenge for many passive activity recognition applications, where it might not be practical or possible for the person to carry any sensor or wireless device at all times.

In these situations, the device-based approach revealed its limitations, driving the need for more ambient, non-intrusive technologies for effective activity recognition and health monitoring. The advent of Wi-Fi signal-based recognition systems emerged as a solution to these challenges.

Camera-based systems emerged as an alternative to device-based systems for detecting passive activity. However, these technologies come with their own set of challenges. Primarily, they necessitate a line-of-sight (LOS), severely restricting their range and efficacy [20].

Beyond the practical constraints, camera-based systems also stir privacy concerns. The intrusive nature of constant video surveillance may infringe upon people's comfort and sense of privacy, making these systems unsuitable for many settings.

In light of these limitations, the need for a more flexible, less intrusive approach to passive activity monitoring becomes evident. This has led to a growing interest in wireless signal-based systems.

Such systems, particularly those utilizing Wi-Fi signals, present a promising solution to these issues [1]. Their capacity to function without direct line-of-sight and

ability to respect the privacy of individuals make them an attractive alternative for passive activity recognition and health monitoring.

2.2 Wi-Fi Activity Recognition

Given the limitations of device-based and camera-based systems, Wi-Fi-based human activity recognition systems have emerged as a promising alternative for passive monitoring. These systems typically comprise a Wi-Fi access point (AP) and one or more Wi-Fi-enabled devices strategically distributed across the environment [23].

During human activity, body movements influence the propagation of wireless signals in the environment, creating unique signatures in the signal patterns. This movement influences the system's multi-path profile, causing modifications that can be tracked and analyzed.

Such changes in the multi-path profile of the system are specifically reflected in the Channel State Information (CSI) [28]. Thus, Wi-Fi-based human activity recognition systems can be designed using either existing CSI-implemented systems or be adapted from traditional systems with the necessary updates.

The key advantage of Wi-Fi-based systems is that they utilize the ubiquitous presence of Wi-Fi signals in modern environments, making them highly practical and less intrusive. By tracking and analyzing the unique signal patterns caused by human movement, these systems open new avenues for passive activity recognition and monitoring.

2.2.1 Systems that require hardware modifications

Wi-Fi-based human activity recognition systems have come a long way, evolving to include advanced signal extraction methodologies. In one such approach, the traditional Wi-Fi system undergoes a hardware modification, giving rise to the WiFi Universal Software Radio Peripheral (USRP) system [25, 36].

This modified system operates by employing a frequency modulated carrier wave, aimed at quantifying the Doppler shift in the orthogonal frequency-division multiplexing (OFDM) signals caused by human movement [18]. Because the Doppler shift and the distance from the target are interconnected, the USRP system can estimate the target's location by measuring the Doppler shift.

The Doppler shift exhibits a positive value when body parts move towards the receiver and a negative value when they move away. Considering a 5 GHz system, a gesture moving at 0.5 m/s would typically generate a Doppler shift of about 17. To capture such subtle shifts, the system needs to monitor the frequency changes in the narrowband pulses derived from the received signal, which vary by only a few Hertz.

After the conversion from narrowband 802.11 to wideband 802.11, the system undergoes a series of steps to extract and utilize the Doppler information [21]:

1. **Extraction of Doppler Information**: Here, the system employs a quick Fourier transform to develop a Doppler profile from the data within a half-second window. This window is then adjusted by 5 ms, and the process is repeated, producing a method known as the Short-Time Fourier Transform (STFT). Considering the typical speed of human movement (0.25 m/s to 4 m/s), only the FFT outputs within the 8 Hz to 134 Hz frequency range (the Doppler shifts at 5 GHz) are considered.

2. **Segmentation of STFT data**: This step aims to identify different patterns within the Doppler data. A single gesture might comprise multiple segments, each exhibiting positive or negative Doppler shifts, or even a combination of both. By observing the energy fluctuations within a brief period, the system can determine the start and end of a segment.

3. **Classification**: Classification is relatively straightforward - segments are categorized into three types: those with positive Doppler shifts, those with negative shifts, and those with both. Each gesture is represented by a series of numbers, which are then compared against training sequences for classification.

The same principle that powers multiple-input multiple-output (MIMO) receivers is used to distinguish reflected signals caused by multiple people moving in the area. The challenge lies in identifying the weight matrix that maximizes the Doppler of each antenna segment, achieved through iterative algorithms.

2.2.2 Systems that do not require hardware modifications

In contrast to the above mentioned USRP software radio-based systems, which require hardware modification, several research initiatives have sought to leverage commercial Wi-Fi APs in their original form. These solutions capitalize on other metrics, such as Channel State Information (CSI), to detect and illustrate the dynamic changes in the environment caused by human movement [16].

One example of this approach involves wireless devices using IEEE 802.11n/ac standards, which utilize Multiple Input Multiple Output (MIMO) systems. Incorporating MIMO technology offers multiple advantages, such as enhancing diversity gain, array gain, and multiplexing gain, all while minimizing co-channel interference.

IEEE 802.11 standard utilizes Orthogonal Frequency-Division Multiplexing (OFDM) [2] as its modulation scheme, where the total bandwidth is distributed among multiple orthogonal subcarriers. Due to the limited bandwidth allotted to each subcarrier, each experiences what is referred to as flat fading. OFDM effectively diminishes the small-scale fading feature of the channel by ensuring that each subcarrier only has to deal with flat fading.

In a Wi-Fi channel leveraging MIMO, OFDM further dissects the channel into multiple subcarriers. To measure the CSI, the Wi-Fi transmitter dispatches Long Training Symbols (LTFs) within the packet preamble. These LTFs contain predefined symbols for each subcarrier. Upon receiving the LTFs, the Wi-Fi receiver estimates the CSI matrix using the received signals and the original LTFs.

Hence, such systems utilizing commercial Wi-Fi APs and CSI estimation offer a unique approach to human activity recognition, without the need for any hardware modification.

The MIMO system at any time instant can be modeled as follows:

$$y_i = H_i x_i + n_i \quad \text{for} \quad i \in \{1, ..., S\} \text{ where}$$

- **S** is the number of OFDM sub-carriers.
- $x_i \in \mathbb{R}^{M^T}$ is the transmit signal vector for the i-th sub-carrier.
- $y_i \in \mathbb{R}^{M^R}$ is the received signal vector for the i-th sub-carrier.

- H_i is the channel matrix for the i-th sub-carrier.
- n_i is the noise vector for the i-th sub-carrier.

By dividing the output signal by a known sequence of input, also known as the pilot, one can estimate the complex value channel state matrix (also known as CSI, channel state information) for the i-th subcarrier H_i. Simply said, CSI describes the manner in which wireless signals spread from the certain carrier frequencies, from the transmitter to the receiver.

FIGURE 2.1: CSI signal propagation.

Multi-path channels, transmit/receive processing, and hardware/software mistakes all have an impact on the observed CSI in real-world WiFi systems. Baseband-to-baseband CSI as measured is:

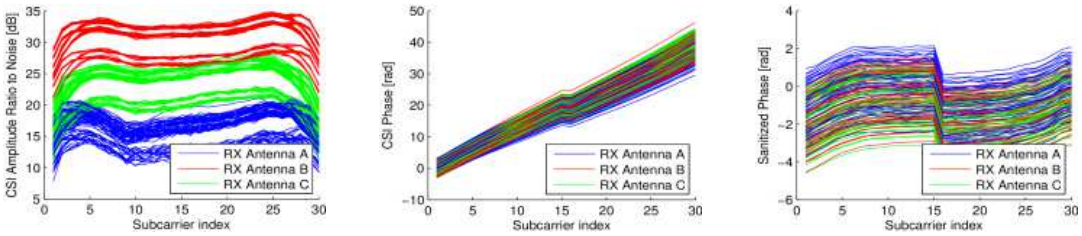
$$
\begin{aligned}
H_{i,j,k} = & \underbrace{\sum_{n=1}^{N} a_n e^{-j2\pi d_{i,j,n} f_k / c}}_{\text{Multi-Path Channel}} \\
& \times \underbrace{e^{-j2\pi\tau_i f_k}}_{\text{Cyclic Shift Diversity}} \\
& \times \underbrace{e^{-j2\pi p f_k}}_{\text{Sampling Time Offset}} \\
& \times \underbrace{e^{-j2\pi\eta(f_k'/f_k - 1)f_k}}_{\text{Sampling Frequency Offset 1}} \\
& \times \underbrace{q_{i,j} e^{-j2\pi\zeta_{i,j}}}_{\text{Sampling Frequency Offset 2}}
\end{aligned}
\tag{2.1}
$$

While the amplitude of CSI generally serves as a reliable metric for feature extraction and classification, it is subject to alterations due to changes in transmission power and transmission rate adaptation. Nevertheless, the impact of burst noise can be mitigated through the application of filtering techniques.

Contrastingly, the phase of a Wi-Fi system is susceptible to a variety of error sources, including but not limited to Sampling Frequency Offset (SFO) and Carrier Frequency Offset (CFO).

- The **CFO** primarily stems from the lack of synchronization between the transmitter and receiver clocks, which essentially translates to differences in their central frequencies. For a duration of 50ms, the CFO in the 5 GHz Wi-Fi band can reach up to 80 kHz, causing an 8π phase change. Consequently, the phase changes induced by bodily movements - which are typically less than 0.5π - may remain undetected from the CSI phase. The second source of error, namely the SFO, is introduced by the receiver's Analog-to-Digital Converter (ADC).

- The **SFO** introduces a distinct error for each subcarrier, as it varies based on the subcarrier index. Given the unknown CFO and SFO, the utility of raw phase information can be limited. Nevertheless, a linear transformation technique has been suggested to exclude the CFO and SFO from the calibrated phase - a process also known as "phase sanitization" [24]. This can be exemplified in a scenario where the Wi-Fi transmitter and receiver are located in close proximity, under Line-Of-Sight (LOS) conditions. In such a case, the CSI amplitude, phase, and sanitized CSI phase can be plotted against the subcarrier index as depicted in the figure below.



(A) CSI amplitude (B) CSI phase (C) Sanitized CSI phase

FIGURE 2.2: CSI measured in LOS condition for three antennas as a function of subcarrier index: (a) amplitude of CSI, (b) phase of CSI, (c) Sanitized phase of CSI. Adapted from Sen et al. [24].

The CSI amplitude exhibits various clusters but is generally steady. Since the SFO expands with subcarrier index, as seen in Fig. b, the raw phase rises with subcarrier index. Phase sanitization will lessen the phase change brought on by SFO. As illustrated in Figure 4.7, a time series of CSI matrices characterizes MIMO channel fluctuations in the time, frequency, and spatial domains. The CSI matrix, which represents the amplitude attenuation and phase shift of multi-path channels, is a 3D matrix H C NMK for a MIMO-OFDM channel with M transmit antennas, N receive antennas, and K subcarriers. In comparison with the Received Signal Strength Indicator, CSI offers a lot more data (RSSI).

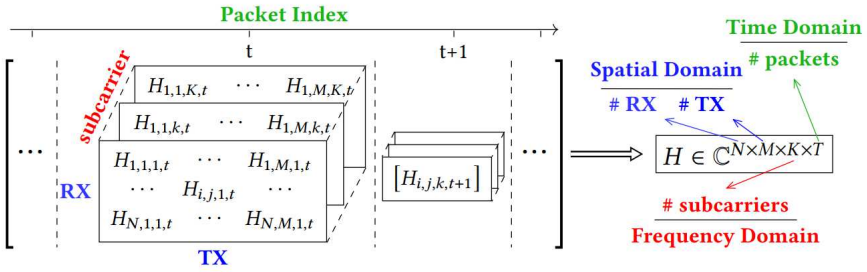


FIGURE 2.3: The 4D CSI tensor is a time series of CSI matrices of MIMO-OFDM channels. It captures multi-path channel variations, including amplitude attenuation and phase shifts, in spatial, frequency, and time domains. Adapted from Yousefi et al. [36].

2.3 Channel State Information

Although Channel State Information (CSI) has been a component of WiFi since the inception of the IEEE 802.11n standard, not all commercial WiFi cards report it [27].

The most recognized tool for CSI measurements is the 802.11n CSI Tool. This tool reports compressed CSIs from 802.11n-compatible WiFi networks using Intel 5300 WiFi devices, and includes instructions for CSI processing, along with source code for MATLAB. A similar utility, OpenRF, was adapted from the 802.11n CSI Tool.

The Atheros CSI Utility provides Channel State Information data from Qualcomm Atheros WiFi cards. Notably, the Atheros CSI Tool has 52 subcarriers for a 20MHz WiFi channel, compared to the 30 of the 802.11n CSI Tool. Both tools support the 2.4GHz and 5GHz frequencies. Software Defined Radio (SDR) platforms, such as the Universal Software Radio Peripheral (USRP) and Wireless Open Access Research Platform (WARP), offer CSI measurements at 2.4GHz, 5GHz, and 60GHz frequencies. Additionally, Nexmon, a C-based firmware patching framework, can monitor traffic and perform other functions on Broadcom/Cypress WiFi chips, including those used in Raspberry Pi devices for experiments.

TABLE 2.1: CSI capturing tools.

	Atheros CSI	**Nexmon**
Platform	Linux distributions	Linux distributions
Customization	Possible	Highly Possible
Chips	Atheros, Intel	Broadcom/Cypress FullMAC WLAN chip
Main Goal	Captures and analyzes channel state information from wireless networks that use Atheros-based wireless network card	Firmware patching framework for users to modify the firmware of WiFi devices that use the Broadcom/Cypress FullMAC WLAN chip, adding new features and functionality such as the ability to capture and analyze CSI data.
Format output	.dat	.pcap

It is crucial to note that the setup and location of the experiment significantly affect the results. CSI, a complex signal, encapsulates the properties of the wireless channel, such as its attenuation, phase, and frequency response. The receiver determines these channel characteristics by analyzing a known signal received from the transmitter. When the transmitter and receiver are in close proximity, the signal experiences less interference or attenuation, which typically results in more accurate and reliable CSI readings. This accuracy is due to increased signal strength and decreased probability of reflections or multipath propagation.

Conversely, if the transmitter and receiver are far apart or separated by walls or other obstacles, the readings may become less accurate. This inaccuracy is due to increased likelihood of reflections and multipath propagation, as well as increased attenuation and interference of the signal. For optimal results, the antenna should generally be placed 1 to 2 meters above the ground.

2.3.1 Challenges and Issues

During the process of recording CSI WiFi data, various challenges can be encountered [38]. One of these challenges is due to the nature of random communications and the requirement for persistent sensing. The rate at which CSI is estimated in

wireless communication often differs from the rate required by the CSI-based sensing system. This can lead to issues such as a low sampling rate when extracting CSI from an Access Point (AP) beacon, or an uneven sampling interval when extracting CSI from data frames.

An additional challenge in Inaudible Sound through Air Communication (ISAC) systems is the unsynchronized transceiver problem. Since both transmitting and receiving devices use their hardware clock for timing and carrier generation, unsynchronized transceiving devices can create a random phase offset in CSI.

Furthermore, the actual CSI in a communication system is not only the wireless channel of air transmission but also involves multiple processing modules at the transmitter and receiver. This complexity can make CSI-based sensing techniques, which aim to sense human information by monitoring changes in wireless channels, more challenging to implement.

Moreover, in non-contact human sensing, the system often comprises a transmitter and multiple receivers at different locations. Many applications, such as motion tracking and gesture recognition, require the collection of CSI from multiple receivers. However, since CSI can only be generated when two transceivers communicate, it is difficult for non-participating devices to estimate CSI simultaneously. This makes the design of an ISAC system that enables multiple devices to estimate CSI simultaneously a challenging task.

Finally, while the ability of wireless signals to propagate through obstacles like walls allows CSI-based sensing to detect activities in a through-wall manner, this also introduces significant privacy and security concerns. For example, malicious hackers and attackers could leverage the received CSI signals to infer a target's daily activities and potentially launch attacks.

2.3.2 Different formats

Every chipset has a unique data format, but fundamentally, all readings share similarities. It's important to remember that different tools yield different outputs, leading to varying data formats which necessitate diverse processing techniques.

For instance, data from Intel cards are represented in a .dat format. The usual process for handling this data involves using MATLAB to extract the data into a more straightforward .csv format, which simplifies further analysis and manipulation.

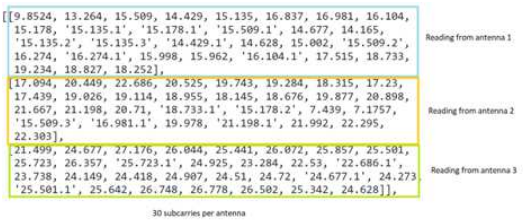


FIGURE 2.4: Atheros/Intel data format.

On the other hand, data from Raspberry Pi cards is often produced in a .pcap format. This format is essentially a container for packets captured on a computer network, such as a WiFi or Ethernet network. Because .pcap files are primarily used for traffic data, they can be processed by custom scripts like CSIKit, among others.

FIGURE 2.5: Raspberri Pi data format.

In short, understanding the specific data format produced by each chipset is vital for determining the best methods for processing and analyzing the data. Consequently, the structure of these files—whether they're .dat or .pcap—should be thoroughly understood to allow for effective data handling and analysis.

Chapter 3

Literature Review

3.1 Wi-Fi Signal Recognition

There has been a significant body of research utilizing Channel State Information (CSI) data across a variety of fields, including recognition tasks, pattern detection, vision, and environmental analysis [39]. Among the pioneer works employing this type of data is a system known as E-Eyes [31].

E-Eyes stands out for its innovative use of environmental-based profiles. These profiles function as benchmarks, with recorded activities compared against them to facilitate identification. To be more specific, E-Eyes leverages Intel WiFi Link 5300 NICs to identify loosely-defined activities—those that do not generate a consistent, repetitive signal frequency that can be interpreted as thresholds for strictly defined activities. Such activities could include cooking dinner in front of the stove, eating at the dining table, exercising on a treadmill, or working at a desk.

Once these activities are identified, a machine learning model classifies the readings. If the observed activity doesn't align with any known profile, semi-supervised learning techniques are used to update the profiles, enhancing the system's adaptability.

However, like most systems, E-Eyes comes with certain limitations. Firstly, the system is designed primarily to classify the activities of a single individual at a time. Secondly, E-Eyes, like many Wi-Fi recognition systems that utilize CSI data, has been tested predominantly in idealized situations. Real-world factors, such as background noise from pets or infants, were not factored into the experimental conditions.

Despite these limitations, E-Eyes demonstrated impressive efficacy, achieving an accuracy rate of 92 % in experiments conducted in two differently sized apartments using a single CSI recording device. While this approach performs commendably and is cost-effective in terms of processing, the histogram method employed is sensitive to environmental changes. This sensitivity could limit the system's utility for certain applications where environmental conditions are likely to fluctuate.

In the continually evolving domain of human speech recognition, researchers have identified a new application for Channel State Information (CSI). In a recent study, the focus has shifted from the traditional use of acoustic sensors—commonly placed near the individual speaking—to the exploration of radio signals as an alternate means for speech recognition. This alternative approach proves particularly advantageous in noisy environments, where acoustic sensors may struggle to function efficiently.

The innovative aspect of this method is evident in its ability to capture the nuanced movements of the mouth through the reflection of these radio signals. Such a technique allows for the identification and management of even the smallest changes

in mouth patterns, referred to as 'micro-motions'. The novelty of this approach is encapsulated in the 'Wi-Hear' system that harnesses the power of CSI data. This is a departure from previous detection methodologies that relied on Doppler shifts or radar technology.

In the context of the 'Wi-Hear' system, 'mouth motion profiles' are created by recognizing changes in the signal reflections triggered by mouth movements. One of the main hurdles encountered in this approach is the complexity inherent like mouth movements. These movements are 'non-rigid', implying that the pronunciation of each letter is unique, involving varying speeds and directions—unlike the more predictable, rigid full-body movements like walking or jumping.

Additionally, the system integrates a multi-tracking mechanism, which, during trials, demonstrated an accuracy rate of up to 73 % across three test subjects. Even more impressively, the system achieved a 91 % accuracy rate in recognizing sentences containing no more than six words. Such a significant level of accuracy holds immense promise for the future, particularly for individuals with disabilities. The potential for them to control various devices using voice commands could open up an entirely new avenue of independence and accessibility.

Another paper [37] not only offers a comprehensive overview of the implementation of Channel State Information (CSI) data recognition systems, but it also outlines a suite of machine learning models employed for this purpose.

Drawing a parallel to the field of speech recognition, where Hidden Markov Models (HMM) have traditionally been utilized for classification, the authors view the problem of activity recognition through a similar lens. Nonetheless, the use of deep recurrent neural networks (RNN) is proposed as an alternative to HMM. Training RNN, however, presents challenges due to issues related to vanishing or exploding gradients. To circumvent these issues, the authors suggest using Long Short-Term Memory (LSTM) models for activity recognition in lieu of conventional machine learning techniques like HMM. The application of LSTM, in this context, diverges from the feature extraction methods typical of other approaches like CARM.

There are two key advantages to employing LSTM models. First, LSTM models are capable of automatically extracting features, thereby eliminating the need for preliminary data preprocessing. Second, LSTM models can retain temporal state information related to the activity, thereby enabling the model to differentiate between similar activities. For instance, the LSTM model can discern between the activities "Lay down" and "Fall", given that "Lay down" involves a sequence of "Sit down" followed by "Fall". The memory capabilities of LSTM models can significantly aid in accurately recognizing such activities.

For the purpose of their research, the authors curated their own dataset, which has since been widely adopted by the community. The experimental setup involved an indoor office area equipped with a commercial Intel 5300 NIC, operating at a sampling rate of 1KHz. During the experiment, a person was asked to perform an activity within a 20-second time frame in Line-of-Sight (LOS) conditions, remaining stationary at the start and end of this period. The dataset comprises data from six individuals, each performing six activities—namely, "Lay down, Fall, Walk, Run, Sit down, Stand up"—with each activity repeated across 20 trials. The results demonstrated that the accuracy of activity recognition using this approach exceeded 75 % for all activities.

The CARM research paper [30] utilizes refined feature extraction techniques and machine learning methods to improve recognition capabilities.

In the initial stage, the system tackles the issue of noise within CSI. CSI often contains substantial noise that makes it challenging to clearly distinguish features

of different activities. Therefore, it's essential to first filter out this noise and then pinpoint specific features to classify using machine learning algorithms. While certain approaches like Butterworth low-pass filters are commonly utilized for noise reduction, they may fail to produce a smooth CSI stream due to the prevalence of high bandwidth burst and impulse noises in CSI. A more effective solution for noise reduction has been found in Principal Component Analysis (PCA). This method reduces the dimensionality of a system, leveraging the principle that the bulk of the signal information is focused on a subset of features. In the application of CARM, the first principal component is purposely excluded to mitigate the noise, while the subsequent five principal components are employed to extract features. This approach ensures that valuable information stemming from the dynamic reflection of the mobile target is retained, as it's also captured in the other principal components. Following PCA de-noising of the CSI data, specialized features are extracted for subsequent classification.

The process of feature extraction forms the next pivotal step. A prevalent method for extracting features from a signal involves its transformation to another domain - the frequency domain being a common choice. The transformation process employs the Fast Fourier Transform (FFT), which is a highly efficient implementation of the Discrete Fourier Transform (DFT). In this procedure, a window of a particular number of CSI samples is chosen, followed by the application of FFT on each segment through window sliding. This strategy, known as the Short-Time Fourier Transform (STFT), enables the detection of frequency changes in a signal over time. Moreover, the Discrete Wavelet Transform (DWT) is deployed to extract features from CSI over time. DWT yields high time resolution for activities characterized by high frequencies and high frequency resolution for those with slower speeds. Each DWT level indicates a frequency range, with the lower levels encompassing higher frequency information and the higher levels accommodating lower frequencies.

DWT offers distinct advantages over STFT. Notably, it presents an excellent compromise in the time and frequency domain, and it effectively minimizes the size of the data, rendering it more amenable for machine learning algorithms.

Within the CARM framework, a 12-level DWT is applied to decompose the five principal components (after the initial component is excluded). The resulting five DWT values are subsequently averaged. Every 200 milliseconds, CARM derives a 27-dimensional feature vector that includes three types of features: the energy in each wavelet level (representing the intensity of varying speed movements), the difference at each level between consecutive 200ms intervals, and the torso and leg speeds estimated using Doppler radar technique. These features serve as inputs for the final classification algorithm.

This paper unveils SenseFi [35], a notable benchmark, and model zoo library tailored for WiFi Channel State Information (CSI) sensing using deep learning.

Initially, the authors acquaint the readers with a variety of leading deep learning models. The suite includes multilayer perceptron (MLP), convolutional neural network (CNN), recurrent neural network (RNN), as well as variants of RNN, CSI transformers, and CNN-RNN. They explore the prowess of these models in CSI feature learning and establish their suitability for WiFi sensing tasks.

In a core section of the paper, they scrutinize and benchmark these models across three distinct WiFi human activity recognition data sets. These data sets encompass both raw and processed CSI data collected via the Intel 5300 CSI tool and the Atheros CSI tool. The accuracy and efficiency of these models are compared and discussed, offering insights into their potential application in real-world situations.

For the purposes of data collection, the Intel 5300 NIC and the Atheros CSI Tool are primarily employed, alongside freely accessible data sources. Given the focus on learning-based methods, the authors chose the most common data modality - amplitude only - and the innovative BVP modality, known for being domain-invariant. Moreover, they delve into the effectiveness of several learning strategies for WiFi CSI data such as supervised learning, transfer learning, and unsupervised learning, all of which are common strategies in WiFi sensing applications.

The researchers selected two public CSI datasets, gathered using the Intel 5300 NIC, for their study. To further validate the effectiveness of deep learning models on CSI data from various platforms, two additional datasets - NTU-Fi HAR and NTU-Fi Human-ID - were collated using the Atheros CSI Tool and a bespoke embedded IoT system.

The findings reveal that the ResNet-18 model achieves the best accuracy of 98.11 % on the UT-HAR dataset, with the CNN-5 model coming in a close second. Although the simpler CNN-5 model demonstrates strong results across all datasets, the more sophisticated ResNet-18 fails to generalize on the Widar dataset, a point further elaborated in Section V-F of the paper. Interestingly, the BiLSTM model records the highest performance on the two NTU-Fi benchmarks.

From these findings, a number of insightful observations emerge. Firstly, the MLP, CNN, GRU, LSTM, and Transformer models all perform satisfactorily across all benchmarks. Additionally, MLP, GRU, and CNN models show stable and superior performance when compared to other models. In contrast, the deeper network series of ResNet models, while performing well on UT-HAR and Widar, do not surpass the simple CNN on the NTU-Fi dataset. This suggests that increasing the number of network layers does not necessarily improve performance, which is a departure from findings in visual recognition tasks. The improvement margin compared to the simpler CNN-5 model is also found to be quite limited.

Another important observation is that the RNN model is outperformed by both LSTM and GRU models. The Transformer model also shows a weakness in scenarios where there is a limited amount of training data, as was the case with the NTU-Fi Human-ID dataset. Lastly, model performance varies across different datasets, highlighting the fact that some datasets, like Widar, are more challenging than others.

Human activity recognition (HAR) models, when trained in a specific environment or domain, often encounter difficulties in generalizing to different or unseen domains. Traditionally, one way to address this challenge is to acquire vast amounts of data from each unique domain. However, this approach is not always practical because collecting such data can be labor-intensive, time-consuming, and might lead to a poor user experience. The research paper in discussion addresses this challenge by aiming to facilitate device-free, location-independent HAR using a minimal number of samples [8]. The underlying concept is to ensure that a model trained with data samples from a source domain can be efficient and accurate in a target domain, even when presented with very few data samples from that domain.

To tackle this, the authors turn to the principles of few-shot learning and meta-learning. Few-shot learning typically deals with the challenge of understanding new concepts with very little data, while meta-learning, or "learning to learn," focuses on using knowledge from one task to help learn another. Drawing inspiration from these paradigms, the research employs metric learning. The core idea is that even if there are no universally stable features to describe actions across different locations, it's still possible to categorize them by maximizing the differences between classes and minimizing the differences within the same class.

A significant portion of the research is dedicated to understanding the effects of location on wireless signal transmission, a crucial component of their HAR model. The authors extensively investigate how the same human activity at different locations impacts the transmission of wireless signals. This includes an analysis of signals received on various antennas and subcarriers at different sampling rates. To purely understand the role of location, without the interference of other variables, data collected from an anechoic chamber—a space that prevents reflections of either sound or electromagnetic waves—is utilized.

The solution proposed in the research is termed "WiLiMetaSensing." This system integrates meta-learning principles to facilitate few-shot learning in sensing activities. It employs a combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) for effective feature representation. Unlike traditional methods that use LSTM for processing Wi-Fi signals, in this approach, the LSTM's memory capabilities are harnessed to capture and retain significant information from all activity samples. Furthermore, the system introduces an attention mechanism-based metric learning method, which aids in understanding the metric relationships between activities, whether they belong to the same category or different ones.

To validate the effectiveness of WiLiMetaSensing, the authors conduct extensive experiments across various scenarios, including singular locations, mixed locations, and completely location-independent sensing. They also put the system to the test under constraints like reduced sampling rates and fewer subcarriers and antennas. Impressively, the results show that WiLiMetaSensing consistently delivers robust performance. Particularly noteworthy is the system's ability to achieve an average accuracy of 91.11 % when trained on data from four locations, even when provided with just a single sample for testing in other locations. Additionally, the model demonstrates a commendable level of adaptability when working with a reduced number of subcarriers and a lowered sampling rate.

In conclusion, the research introduces WiLiMetaSensing, a pioneering human activity recognition system, which is adept at location-independent sensing in Wi-Fi environments with very few samples. Leveraging the principles of meta-learning, the system can seamlessly apply knowledge from one location to another. With its CNN-LSTM architecture, it ensures effective feature extraction across different locations. The results affirm that WiLiMetaSensing stands as a robust and adaptive solution for HAR, especially in situations where data collection across multiple environments is challenging.

The research under scrutiny delves into an innovative method of human activity recognition by leveraging deep learning models in conjunction with triaxial accelerometers [3]. The study's foundation is built upon the success observed in acoustic modeling using deep learning, emphasizing the potential similarities between patterns found in speech and acceleration data. This correlation is proposed on the belief that both domains manifest analogous temporal fluctuations.

A pivotal highlight of the study is its emphasis on the improved recognition accuracy achieved through the use of deep learning models, as opposed to conventional shallow models. This significant edge in performance is attributed to the deep models' innate capability to autonomously discern and interpret intrinsic features from the acceleration data. Such an automated approach offers a marked advantage, as it effectively eliminates the need for the prevalent practice of manual feature engineering seen in contemporary methods.

The research also sheds light on the potential of semi-supervised learning techniques in the realm of activity recognition. This exploration was necessitated due to

the scarcity of labeled activity datasets. Notably, the study found that generative (or unsupervised) training played a crucial role in optimizing and tuning the weights of deep activity recognition models.

Moreover, the paper underscores a shift in data analysis techniques. Given that accelerometers produce multifaceted data, characterized by its multi-frequency, fluctuating, and aperiodic nature, the study advocates for the use of spectrogram signals instead of raw acceleration data. Such a transition is noted to significantly bolster the deep learning models' capacity to detect nuanced variations in input, thereby improving recognition accuracy.

Another innovative approach presented is the fusion of deep learning with hidden Markov models (HMM), leading to the introduction of a novel hybrid model termed DL-HMM. This model is designed to refine the accuracy of recognizing temporal activity sequences. By amalgamating the hierarchical representations inherent in deep learning with the stochastic temporal sequence modeling capabilities of HMMs, the DL-HMM showcases remarkable efficacy. Experimental evidence from the research supports the model's superiority over traditional HMM-based techniques. Furthermore, the study illustrates that the representations learned from deep activity recognition models are instrumental in approximating the posterior probabilities of HMMs.

The study's experimental framework employed a deep activity recognition model structured with three layers, each encompassing 1000 neurons. This architecture, when tested on the Skoda checkpoint dataset, yielded an impressive accuracy rate of 89.38 %. This marked a notable improvement of 3.38 % compared to a previously established HMM methodology. The potential of the hybrid DL-HMM model was further exemplified, as it achieved an almost impeccable recognition accuracy rate of 99.13 % when temporal correlations within the dataset were leveraged.

In conclusion, this research provides a comprehensive and innovative lens into human activity recognition through triaxial accelerometers. The fusion of deep learning models and established methodologies not only showcases remarkable results but also offers novel insights and avenues for future endeavors in the domain.

The research paper titled "TW-See: Human Activity Recognition Through the Wall with Commodity Wi-Fi Devices" [33] delves into an avant-garde technique of device-free passive human activity recognition by leveraging Wi-Fi signals that penetrate walls. The foundation of this study hinges on the observation that in a majority of indoor settings, especially residences, Wi-Fi signals inevitably pass through walls to reach the designated Access Point (AP). This often results in significant signal attenuation. Experimental evidence from the paper highlights that Wi-Fi signals, particularly those operating at a frequency of 2.4GHz, suffer considerable attenuation when navigating through concrete walls as thick as 18 inches, experiencing a decline of up to 18dB.

Furthermore, the paper sheds light on the deficiencies of conventional denoising methods, such as the low-pass filters and Principal Component Analysis (PCA). Specifically, these methods were found to underperform in scenarios where the transmitter-receiver distance surpasses 3 meters and when concrete walls obstruct all propagation paths, thereby dampening the overall effectiveness of these denoising techniques.

In the pursuit of addressing these challenges, the authors introduced TW-See, a system conceptualized using commonplace Wi-Fi devices, typical to homes or offices, equipped with a singular antenna to function as the transmitter. In contrast, any PC or laptop boasting three antennas NIC is envisioned to act as the receiving end. The actualization of the TW-See system brought forth two primary technical

obstacles. First, the task of extracting a valuable correlation between human activities and Channel State Information (CSI) values from raw CSI measurements turned out to be an intricate challenge, primarily due to the interference caused by walls and other facets of the indoor physical environment, including reflection, diffraction, and scattering. The second hurdle was the segmentation of activities from CSI waveforms. While earlier studies suggested ease in segmenting activities owing to conspicuous amplitude variations in received signals, the introduction of walls into the equation made the changes instigated by human actions on the CSI waveform far less discernible.

To navigate these challenges, the paper proposed two innovative methodologies. The first, known as the Opposite robust PCA (Or-PCA), is a technique tailored to extract a meaningful correlation between human activity and the subsequent shifts in CSI values. A defining feature of Or-PCA, setting it apart from preceding methods, is its focus solely on the CSI alterations induced by human actions. This strategy minimizes the distractions arising from environmental noise and other background interferences. Moreover, this approach emphasizes prioritizing the concentration of correlation on the initial Or-PCA component. The second solution presented is the Normalized Variance Sliding Windows Algorithm, crafted to segment human activities from the Or-PCA waveforms. This algorithm is adept at reducing the impact of minor fluctuations in the Or-PCA waveforms on the activity segmentation process and precisely determining the beginning and cessation times of various activities.

In terms of contributions, the introduction of TW-See is laudable as it represents a pioneering stride in harnessing CSI for human activity recognition through walls using off-the-shelf Wi-Fi devices, eliminating the dependence on specialized equipment. The paper's unique proposition, Or-PCA, not only excels in extracting the desired activity-CSI correlation but also accentuates the correlation's prominence on the first principal component. Following the extraction of activity features based on this correlation, a Back Propagation (BP) neural network is employed to discern and categorize various human activities. Empirical validation of the TW-See system, leveraging commercial Wi-Fi devices with a single antenna at the transmitter's end and a trio of antennas at the receiver's end, exhibited a remarkable average accuracy rate of 94.46% for signals traversing a concrete barrier. In essence, the "TW-See" paper stands as a significant milestone in the domain of human activity recognition, fostering novel applications and prompting further scholarly endeavors in Wi-Fi-centric sensing and activity recognition.

The paper titled "WiHF: Gesture and User Recognition with WiFi" [17] engages in the exploration of gesture recognition while simultaneously identifying unique users, utilizing the omnipresent Wi-Fi signals. This area of study, despite its immense potential, grapples with a myriad of challenges. The foremost challenge arises from the inherent noise associated with Wi-Fi signals. Extracting discernible features from these signals, which not only capture distinct gesture dynamics but also the personalized nuances of how users perform these gestures, remains a daunting task. Consequently, the dual capability of recognizing gestures and user identities has been a difficult feat to accomplish. Another pivotal challenge lies in ensuring that the computational complexity remains within bounds, allowing for real-time gesture and user recognition. Furthermore, the variability with which gestures are performed—different locations, orientations, and environments—complicates the process. Given these variations, Wi-Fi signals recorded during identical gestures can differ considerably, making consistent and accurate recognition across varied domains quite intricate.

To tackle these challenges head-on, the authors present WiHF (WiFi HuFu), an innovative system striving for real-time, cross-domain user-identified gesture recognition using everyday Wi-Fi devices. A notable feature of WiHF is its ability to capture the nuanced motion change patterns resulting from arm gestures. These patterns encompass rhythmic velocity fluctuations and defining pauses. More intriguingly, these patterns maintain their consistency across varied domains. To further optimize computational efficiency, the authors employ the seam carving algorithm, enabling the swift extraction of these motion change patterns. A collaborative dual-task Deep Neural Network (DNN) model forms the cornerstone of WiHF. This model's dual functionality allows it to seamlessly recognize gestures and user identities. The model employs a splitting and splicing scheme, enhancing its cross-domain capability and promoting collaborative learning.

Empirical results serve as a testament to WiHF's prowess. When evaluated on a public dataset, the system showcased remarkable accuracy levels of 97.65% and 96.73% for gesture recognition within a domain and user identification, respectively. Notably, WiHF boasts zero-effort cross-domain gesture recognition capabilities, standing toe-to-toe with leading methodologies in the field. Moreover, in terms of processing time, WiHF paves the way with a 30-fold reduction, affirming its real-time operational capacity.

In essence, "WiHF: Gesture and User Recognition with WiFi" champions several pioneering contributions. The authors have meticulously designed a domain-agnostic motion change pattern for arm gestures, curbing the deployment costs for real-world applications and ushering in a suite of algorithms optimized for real-time execution. The dual-task DNN framework proposed is not only proficient in recognizing gestures and identifying users simultaneously but also offers prospects for expansion into multi-task sensing using wireless signals. The extensive experimental evaluations affirm WiHF's effectiveness and feasibility, marking it as a significant stride in gesture recognition and user identification through Wi-Fi signals. The systematic presentation of the paper, encompassing a review of related works, preliminary observations, system design intricacies, performance evaluations, discussions, and conclusions, offers a comprehensive insight into Wi-Fi-based gesture and user recognition.

The paper entitled "*WiFall: Device-Free Fall Detection by Wireless Networks*" [32] dives deep into the critical challenge of detecting falls, particularly prevalent among the elderly. Falling, as described in the paper, is an uncontrolled, rapid transition of the human body from an upright to a prone position. Emphasizing the gravity of the situation, the authors cite statistics from the Center for Disease Control and Prevention, which note that one in every three adults aged 65 and above experiences a fall at least once annually. These falls often lead to fatal injuries. Given the time-sensitive nature of addressing falls, there is an urgent need for the development of an automated system capable of detecting falls swiftly, thereby facilitating a rapid response.

For a fall detection system to be deemed effective, it must meet a triad of criteria: accuracy, cost-effectiveness, and user-friendliness. Most existing solutions, however, fail to strike a balance between these critical attributes, rendering them unsuitable for widespread home deployment. In light of this, the authors of this paper propose an innovative fall detection mechanism leveraging the ubiquity of Wireless Local Area Networks (WLAN). WLAN, as a platform, promises to deliver an optimal mix of accuracy, affordability, and user engagement while ensuring minimal privacy intrusions. Furthermore, the paper explores the potential of harnessing commercial

wireless products, exploiting the distinct properties of WiFi signals during propagation, for device-free fall detection.

Highlighting the existing research paradigms, the authors point out that while the past decade has seen a flurry of research activity around WiFi-based localization, motion detection, and object tracking, the relationship between human activities and wireless signal dynamics remains underexplored. Existing radio propagation models fall short when applied to complex human activities, calling into question their utility in capturing the nuances of human movement.

To bridge this gap, the paper introduces a refined radio propagation model tailored for indoor environments, factoring in the disruptions caused by human activities. It delves into the theoretical underpinnings of the radio propagation model during a fall and presents a nuanced model for scenarios where the user is in a Non-Line of Sight (NLOS) location. At the heart of the proposed solution is WiFall—a passive, device-free fall detection system that leverages Channel State Information (CSI) as a key metric. The authors also define an 'effective range area', representing the spatial extent covered by WiFall's communication links, which becomes the focal point for evaluating the system's performance.

A key contribution of this paper lies in its advocacy for the use of CSI for device-free fall detection. Leveraging the temporal stability and frequency diversity inherent in CSI, WiFall is posited as a cost-effective solution, seamlessly integrating with existing wireless infrastructures. For genuine unintrusiveness, the authors combine motion detection via anomaly detection algorithms with sophisticated classifiers such as one-class Support Vector Machines and the Random Forest algorithm. This combination not only detects falls but also classifies varied human activities.

The efficacy of WiFall is substantiated through extensive evaluations carried out in three distinct indoor environments, using commonly available 802.11n Network Interface Cards on laptops. Impressively, in all tested scenarios, WiFall demonstrates an average detection accuracy rate of 94% with Random Forest-based classification. The authors also emphasize the system's potential scalability in recognizing a broader spectrum of human activities.

The methodical structure of the paper offers readers a logical flow—starting from related works, delving into foundational knowledge, elucidating WiFall's design intricacies, detailing evaluation methodologies, presenting the results, and culminating in conclusions and discussions. In essence, "WiFall: Device-Free Fall Detection by Wireless Networks" pioneers a novel approach to fall detection, potentially revolutionizing elderly care and offering peace of mind to caregivers worldwide.

Chapter 4

Implementation

In this chapter, we describe in detail the proposed framework for the recognition of humans' activities from Wi-Fi signal.

4.1 Model Pipeline

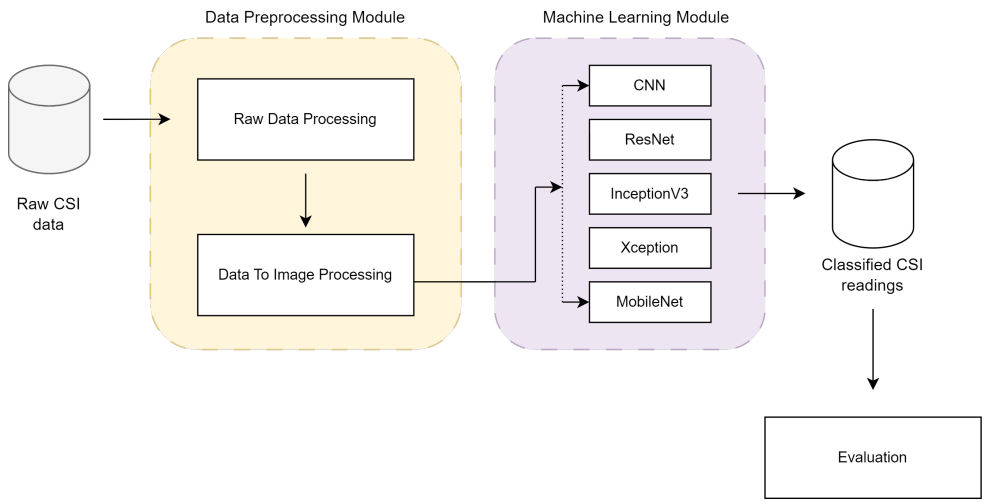


FIGURE 4.1: Model pipeline.

At the beginning of the pipeline, raw Channel State Information (CSI) data is ingested. CSI data typically provides detailed information about the wireless communication channel, capturing various parameters that describe the environment's characteristics.

In the **Data Preprocessing Module**, there's an initial step that involves cleaning, normalizing, and transforming the raw CSI data. Common preprocessing tasks can include removing outliers, smoothing, or filtering to reduce noise. This ensures the data is in an appropriate format and quality before undergoing further transformations. Following this, in the **Data to Image Processing Block**, the CSI amplitude readings are transformed into image format. This transformation might involve creating a heatmap-like representation of the CSI data or some other type of visualization that effectively captures the nuances of the data in an image format.

Moving on to the **Machine Learning Module**, the processed image data from the previous step is now used as input for machine learning models. Five different models are trained on this data. This could be a mix of various architectures suited for image data, such as Convolutional Neural Networks (CNNs), or other models adept at handling image inputs. Each model will try to learn patterns in the CSI data

and subsequently classify the data based on its learning. After training, the models are tested on unseen data to determine how well they've learned and to predict the classification of the CSI readings.

Lastly, the performance of each of the five models is evaluated. This is typically done using metrics such as accuracy, precision, recall, F1-score, and ROC curves, among others. The evaluation determines the effectiveness of each model in classifying the CSI readings and provides insights into potential improvements or the selection of the best-performing model for deployment.

In essence, this pipeline takes raw CSI data, transforms it into a format conducive for machine learning models (in this case, images), and then utilizes multiple machine learning models to classify the data. The performance of these models is then assessed to determine their effectiveness.

4.2 Setting Up

4.2.1 Dataset

The data set chosen for this project is a publicly available data set, shared in [36]. The authors conducted the experiments for the dataset collection within an indoor office area, with the transmitter (Tx) and receiver (Rx) positioned 3 meters apart under line-of-sight (LOS) conditions. The Rx was fitted with a commercial Intel 5300 Network Interface Card (NIC) that has a sampling rate of 1 kHz.

During the experimental process, an individual began to move and engage in an activity for a span of 20 seconds in LOS conditions, while maintaining a stationary position at the start and end of this period. Concurrently, the authors recorded video footage of these activities to facilitate the labeling of the data.

The dataset assembled by the authors encompasses six individuals, each performing six different activities. These activities, classified as "Lie down, Fall, Walk, Run, Sit down, Stand up," were each replicated in 20 trials.

From Figures 4.2 and 4.3, it is evident that the dataset is fairly balanced due to the near-equal distribution of the classes.

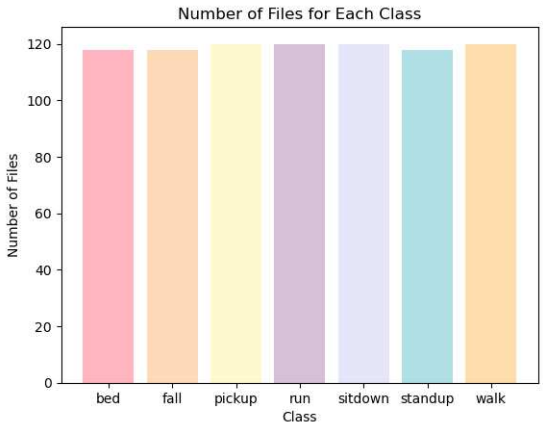


FIGURE 4.2: Class distribution of the dataset.

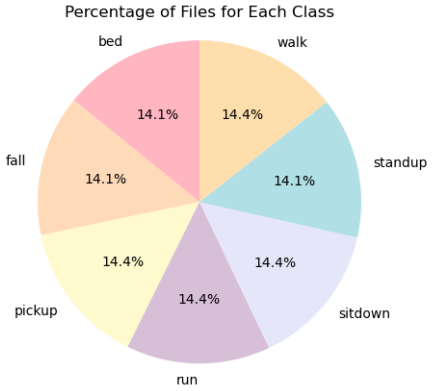


FIGURE 4.3: Class distribution of the dataset.

4.2.2 Environment and Limitations

The limitations placed on the scope of the experiment were primarily driven by the university's inability to provide the requisite computational resources. Consequently, the dataset chosen for this study, despite being publicly available, is relatively small in size to accommodate processing on a personal machine.

In addition, models that are more computationally demanding, such as Long Short-Term Memory (LSTM) networks and others, were not utilized due to these hardware restrictions.

As for the technical setup, the experiments were conducted using the Anaconda environment, specifically through Jupyter Notebook. The use of Jupyter Notebook allowed for an interactive and user-friendly interface despite the constraints of running the experiments on a personal machine.

4.2.3 Expected Challenges

Utilizing computer vision to categorize the visualized recordings of amplitude changes in Channel State Information (CSI) WiFi data can entail a multitude of challenges.

Firstly, the quality of visualization can be a significant issue. If the visualized amplitude of the CSI data is filled with noise or distortions, it could result in incorrect feature extraction and consequently a dip in classification accuracy.

On top of that, the features within the visualized data might be overly complex or subtle, rendering the learning and correct classification by the computer vision model a daunting task. The amplitude changes might not always demonstrate clear or easily identifiable features, complicating the categorization process further.

Environmental changes pose another challenge, as CSI WiFi data is acutely sensitive to the surroundings where it is captured. Any alteration in the physical environment, such as moving objects or shifts in humidity, could influence the amplitude, thereby introducing additional intricacies into the data.

The volume of data derived from amplitude changes in CSI WiFi data can be substantial, particularly in real-time or long-duration recordings. This can heighten the computational requirements for visual classification exponentially.

Furthermore, the model might struggle with robustness to noise or fluctuations in the WiFi signal. Interference like high-frequency noise could deteriorate the visualized amplitude data's quality, making it even more challenging to maintain the accuracy of classifications.

The requirement for training data could pose another obstacle. Collecting ample and representative training data might prove to be a tough task, and any lack of a diverse, extensive, and accurately labeled dataset could lead to overfitting or poor generalization of the computer vision model.

Finally, real-time processing might be a requisite depending on the application, but such tasks can be computationally intensive in the field of computer vision, potentially leading to latency issues.

In summary, these potential issues underscore the necessity for careful planning and execution when using computer vision to classify visualized recordings of amplitude changes in CSI WiFi data.

4.2.4 Mitigating Expected Challenges

Improving the quality of visualized data is the first step and can be achieved through various signal processing techniques such as filtering and smoothing. These methods aid in reducing noise and distortions which could hamper the learning process. Next, we can handle the complexity of features through advanced feature engineering and selection techniques. This simplification process can be further bolstered by using deep learning models like convolutional neural networks (CNNs) which are adept at automatically extracting and learning useful features from the data.

Another aspect to consider is the influence of environmental variations on the data. To mitigate this, we can strive to make the model more robust to environmental changes by diversifying our data collection across different environments and conditions. In addition, employing data augmentation techniques can enhance the model's ability to generalize, ensuring that it performs well even under unfamiliar conditions.

When dealing with the large volumes of data associated with CSI WiFi recordings, efficient data handling techniques become crucial. Methods such as data streaming and mini-batch processing allow us to handle these large volumes effectively. For computational load concerns, we can resort to more powerful hardware or leverage the power of distributed computing.

Noise robustness is another significant factor to consider. By utilizing robust machine learning models, we can ensure reliable performance even in the presence of noisy data. Techniques like denoising autoencoders or noise-cancellation algorithms provide effective ways to mitigate the impact of noise on our model's performance.

One of the key challenges lies in meeting the requirements for large and diverse training datasets. This can be overcome by employing techniques like data augmentation, transfer learning, and semi-supervised learning. These methods allow us to maximize the utility of our available data and generate new data to train our model effectively.

Lastly, for applications where real-time processing is essential, we must focus on optimizing our algorithms for speed. Efficient algorithm design, coupled with the use of hardware acceleration techniques (such as GPUs), and edge computing can significantly reduce computational time, making real-time processing a feasible goal.

By implementing these strategies, we can effectively navigate the challenges that arise when using computer vision to classify visualized recordings of changes in the amplitude of CSI WiFi data.

4.3 Data Processing

4.3.1 Raw Data Processing

This initial script is responsible for the import of data from a multitude of CSV files of raw CSI recordings, conducting the requisite transformations, and subsequently exporting the transformed data into new CSV files. The script is bifurcated into two primary components:

- The first component, the 'dataimport()' function, oversees the importation of two distinct categories of CSV files. The initial category, referred to as 'input', comprises numerical data which is subjected to reshaping and subsequent stacking. The secondary category, 'annotation', contains string-based data which is converted into numerical arrays pursuant to a predefined threshold criterion.
- The principal segment of the script is tasked with generating file paths for the input and annotation data, corresponding to an array of activities (including, but not limited to, bed, fall, pickup, run, sitdown, standup, walk). The script invokes the 'dataimport()' function for each correlated pair of file paths, before compiling the returned processed data into newly created CSV files."

Here is a systematic analysis of the program's operations:

1. The code initiates two vacant arrays, designated as xx and yy, intended for storing the input and annotation data respectively.
2. The input data is imported from CSV files that are specified by 'path1'. The program iterates over each file, extracts the data from the CSV file, and transmutes it into a numpy array, referred to as tmp1.
3. The code then implements a sliding window methodology on the input data. It iterates across the data in window units of a predefined size, window size, moving in strides of 'slide size'. For every window, the necessary values are culled and appended to the x2 array.
4. The x2 array is merged with the xx array, accumulating all the input data.
5. The xx array is reconfigured to maintain a specific number of rows and columns.
6. The code proceeds to import the annotation data from CSV files, specified by path2. The process emulates Step 2, wherein the annotation data is read from the CSV file and transposed into a numpy array, tmp2.
7. A sliding window approach is applied to the annotation data. Each window quantifies the frequency of different labels and assigns a particular binary label based on the threshold. The computed labels are stored in the y array.
8. The y array is then amalgamated with the yy array, capturing all the annotation data.
9. The function finally returns the xx and yy arrays.
10. Post the definition of the dataimport function, the code transitions into the main section, creating a directory labeled "input files" if it is non-existent.

11. The code then cycles through a list of labels: bed, fall, pickup, run, sitdown, standup, and walk. For each label, it constructs file paths for the input and annotation CSV files based on the label. It also formulates output file names for xx and yy contingent upon the window size and threshold.

12. The code calls the dataimport function with the input and annotation file paths, assigning the returned values to x and y.

13. It then transcribes the contents of x to the output CSV file designated by outputfilename1, and the contents of y to the output CSV file defined by outputfilename2.

14. Upon completion of the label processing, it outputs a message indicating the successful execution of the task.

4.3.2 Data to Image

The secondary script's purpose is to render data from multiple Channel State Information recordings stored in modified previously CSV files as heatmaps, and subsequently save these graphical representations to a specified directory. It comprises:

The 'visualize()' function, which utilizes the pandas library to convert a CSV file into a DataFrame. It then extracts a segment of the data, specifically the amplitude data, and generates heatmaps from this data using the matplotlib library. These heatmaps are stored as PNG images in a predefined output directory.

- The visualize files function scans a designated directory for CSV files whose names commence with "input". It then invokes the visualize function for each of these files.

- The principal segment of the script designates the folder path and the output directory for the visualize files function, following which it calls this function.

4.4 Choice of Models

4.4.1 Convolutional Neural Networks (CNNs)

In the vast arena of machine learning and artificial intelligence, Convolutional Neural Networks (CNNs) stand out as one of the most powerful tools for image processing and computer vision tasks. These neural networks have played a pivotal role in achieving state-of-the-art results in areas such as image and video recognition, self-driving cars, and many other domains that involve visual data.

Originating from biological processes observed in the human brain, CNNs are specifically designed to recognize patterns [15]. The human visual cortex contains specialized cells responsive to light patterns in specific regions of the visual field, and CNNs draw inspiration from this concept. Unlike traditional neural networks that treat input data as flat vectors, CNNs maintain the spatial hierarchy of the data, ensuring that pixels closer to each other are treated as related entities.

A typical CNN architecture is composed of multiple layers, each having a distinct role:

- Convolutional Layers: This is the cornerstone of CNNs. The layer scans the input image with a small, sliding window called a filter or kernel, processing small chunks of the image at a time. As the filter slides (or convolves) around

the input image, it produces a feature map, emphasizing features like edges or textures.

- Activation Function: Post convolution, an activation function like the Rectified Linear Unit (ReLU) introduces non-linearity to the model. This allows the network to capture complex patterns and representations.
- Pooling Layers: These layers reduce the spatial dimensions of the data, making computations faster and reducing the risk of overfitting. The most common technique, max-pooling, retains the maximum value from a section of the feature map.
- Fully Connected (FC) Layers: After several convolutional and pooling layers, the architecture typically has one or more FC layers. These layers flatten the high-level features learned in the previous layers and combine them, leading to the final output. In image classification, for instance, the final FC layer will output the probabilities for each class.
- Dropout and Normalization: These techniques are not layers in the traditional sense but they are crucial for enhancing the performance and stability of CNNs. Dropout is a regularization method that randomly drops out neurons during training, preventing any single neuron from becoming too decisive. Normalization methods, like Batch Normalization, make training faster and more stable by scaling the activations.

Training a CNN involves inputting an image, letting it pass through the network, and comparing the output to the ground truth using a loss function. The difference (or error) is then propagated back through the network using a technique called backpropagation, adjusting the weights of the filters and neurons to reduce the error. This process is iteratively performed using numerous images until the model converges to a point where it can recognize patterns and features in unseen images.

One of the key reasons CNNs excel in visual tasks is their ability to automatically and adaptively learn spatial hierarchies of features. Early layers might capture simple features like edges and textures, but as data moves through the network, higher layers can recognize more complex features, such as shapes or even entire objects.

In recent years, with advancements in computing power and the proliferation of large labeled datasets, deeper and more complex CNN architectures like VGG, ResNet, and Inception have emerged, pushing the boundaries of what's possible in computer vision.

CNNs have transformed the way machines perceive visual data. Their biologically-inspired architecture and ability to learn intricate patterns autonomously make them a key player in the ongoing AI revolution. Whether it's detecting objects in real-time video feeds, diagnosing medical images, or powering the vision of robots, CNNs will continue to be at the forefront of technological advancements in visual computing.

4.4.2 ResNet

Deep Learning, a subset of Machine Learning, has transformed a multitude of domains from natural language processing to computer vision. As researchers attempted to increase the depth of neural networks to improve accuracy, they faced challenges. The direct increase in depth often led to issues of vanishing gradients

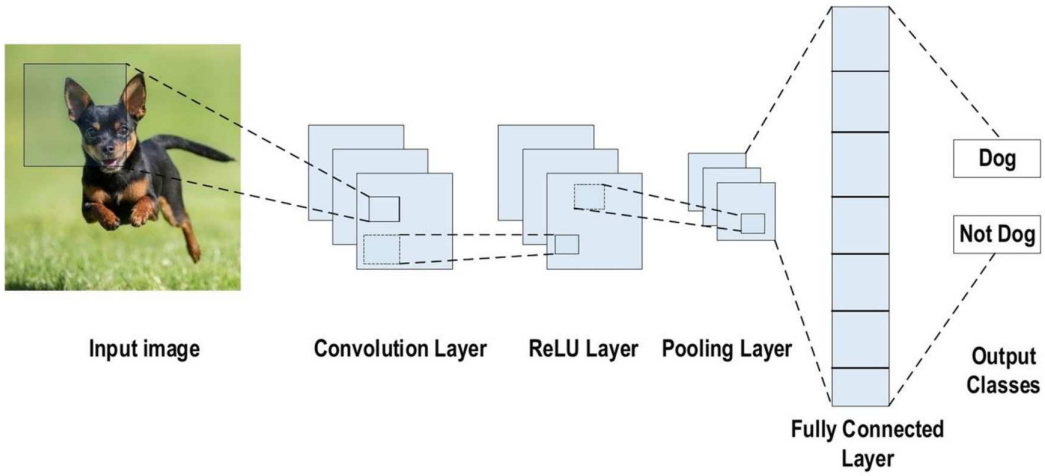


FIGURE 4.4: CNN example, extracted from [4].

and degraded performance, contradicting the common intuition that deeper networks should perform at least as well as their shallower counterparts.

Enter ResNet, or Residual Networks, a groundbreaking architecture introduced by Kaiming He and his colleagues in their 2015 paper, "Deep Residual Learning for Image Recognition.' [12] This model introduced the concept of "skip connections" or "shortcuts" to counteract the aforementioned issues.

At the heart of the ResNet architecture lies the residual block. Instead of trying to learn an underlying mapping directly, ResNet aims to learn the residual (or difference) between the input and the desired output. This residual mapping is typically easier to optimize.

A residual block is represented as:

$$F(x) = H(x) - x \tag{4.1}$$

Where $F(x)$ is the residual mapping, $H(x)$ is the desired underlying mapping, and x is the original input. The reformed equation becomes:

$$H(x) = F(x) + x \tag{4.2}$$

The added x is the skip connection, bypassing one or more layers.

Skip connections, or shortcut connections, are the main innovation in ResNet. They allow the gradient to be directly back-propagated to earlier layers, mitigating the vanishing gradient problem. This direct path ensures that even if the weights of the added layers are initialized as zeros, the identity function can still be easily learned.

To make the network computationally efficient, especially for deeper architectures like ResNet-152, a bottleneck design was employed. Instead of stacking two large-size kernel layers (like 3x3), they used three layers for the residual block: a 1x1 layer, a 3x3 layer, and another 1x1 layer. The 1x1 layers are responsible for reducing and then increasing (restoring) dimensions, leaving the 3x3 layer a bottleneck with smaller input/output dimensions.

Various depths of ResNets have been experimented with, like ResNet-18, ResNet-34, ResNet-50, ResNet-101, and ResNet-152. The depth indicates the number of layers in the network. Empirically, deeper ResNets have shown better performance, though diminishing returns are seen as depth increases significantly.

Towards the end of the network, instead of using fully connected layers, a global average pooling layer is employed across the height and width dimensions, resulting in a fixed-size vector regardless of image input size. This vector is then passed to the final softmax layer for classification.

ResNet has achieved state-of-the-art results on various benchmarks and competitions. Notably, in the ImageNet Large Scale Visual Recognition Challenge 2015, ResNet set a new record, reducing the error rate by a significant margin.

Beyond its remarkable performance, ResNet's true legacy is its influence on subsequent deep learning architectures. The idea of skip connections has been integrated, adapted, and built upon in numerous subsequent models and papers.

ResNet has played a transformative role in the deep learning landscape. By introducing the concept of residual learning and skip connections, it enabled the training of much deeper networks than previously thought feasible, paving the way for further innovations in the field. Whether used in its original form or as a foundation for new architectures, the principles behind ResNet will continue to shape the future of deep learning research and applications.

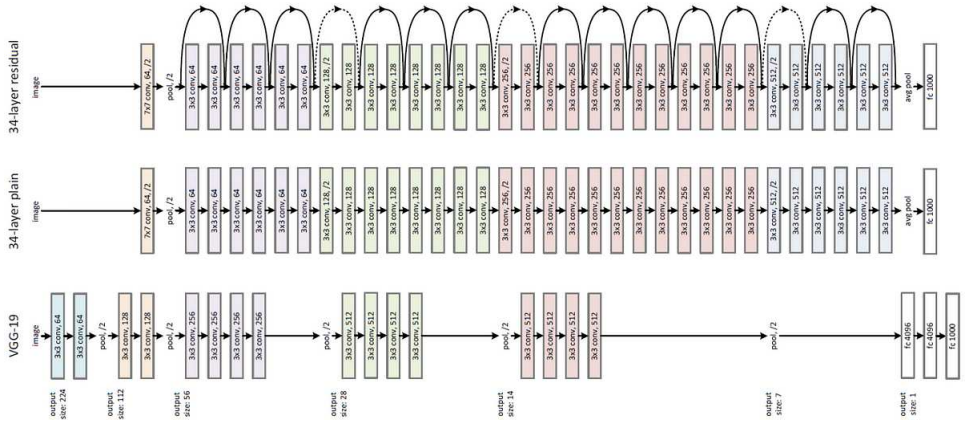


FIGURE 4.5: ResNet example, extracted from He et al. [12].

4.4.3 InceptionV3

In the rapidly evolving landscape of deep learning architectures, Google's Inception models, specifically InceptionV3, have garnered considerable attention due to their performance and efficiency. InceptionV3, introduced in the paper "Rethinking the Inception Architecture for Computer Vision" by Szegedy et al. [26], stands out because of its depth, width, and ability to minimize computational cost while maximizing performance.

The primary motivation behind the Inception architecture was to find a model that achieves good performance on a computational budget. This is achieved through careful convolutional operation design to ensure maximum utility of the model's operations, and consequently, better utilization of computing resources.

Central to the Inception architecture is the "Inception module." It's predicated on the idea that instead of committing to one particular convolution operation size, the model computes multiple types and scales of convolutions in parallel. This module typically involves parallel paths. Each path may involve a 1x1, 3x3, or 5x5 convolution or max pooling, followed by 1x1 convolutions to reduce dimensionality. The outputs of these paths are concatenated and form the input to the next layer.

One of the novel ideas in the Inception architecture is the use of 1x1 convolutions for dimensionality reduction before more expensive operations like 3x3 or 5x5 convolutions. This results in significant computational savings without substantial loss in the capacity of the model.

InceptionV3 introduced the concept of factorizing larger convolutions into smaller ones. For instance, a 5x5 convolution is factorized into two 3x3 convolutions. This not only reduces computational complexity but also adds another nonlinear layer, making the decision function richer.

An interesting feature of the Inception models is the inclusion of auxiliary classifiers. Located at intermediate points in the network, these classifiers propagate their gradient during training. The rationale is to provide an additional regularizing effect and to ensure that middle parts of the network can make useful predictions, mitigating the vanishing gradient problem in very deep networks.

Instead of using pooling operations for reducing grid size (and thus spatial dimensions), InceptionV3 introduced a more complex mechanism: it uses convolutional layers with strides. This choice ensures that the model doesn't lose out on important features that pooling might discard.

InceptionV3 was trained on the ImageNet Large Scale Visual Recognition Challenge dataset. A variety of techniques, including RMSProp optimizer, label smoothing, and batch normalization, were used to stabilize the training of these very deep networks. The model achieved a top-5 error rate of 3.5% on the 2015 ILSVRC challenge.

The InceptionV3 model, due to its efficiency and top-tier performance, was widely adopted in many visual recognition tasks beyond just classification. Its architectural principles served as a foundation for subsequent models and have been incorporated, in some form, into later deep learning models.

InceptionV3 was a testament to the fact that neural network architectures could be both deep and efficient. Through a series of architectural innovations, such as factorized convolutions and the novel use of 1x1 convolutions for dimensionality reduction, InceptionV3 established a new standard in computational efficiency for state-of-the-art performance. Today, its design principles continue to influence the development of deep learning models in the field of computer vision.

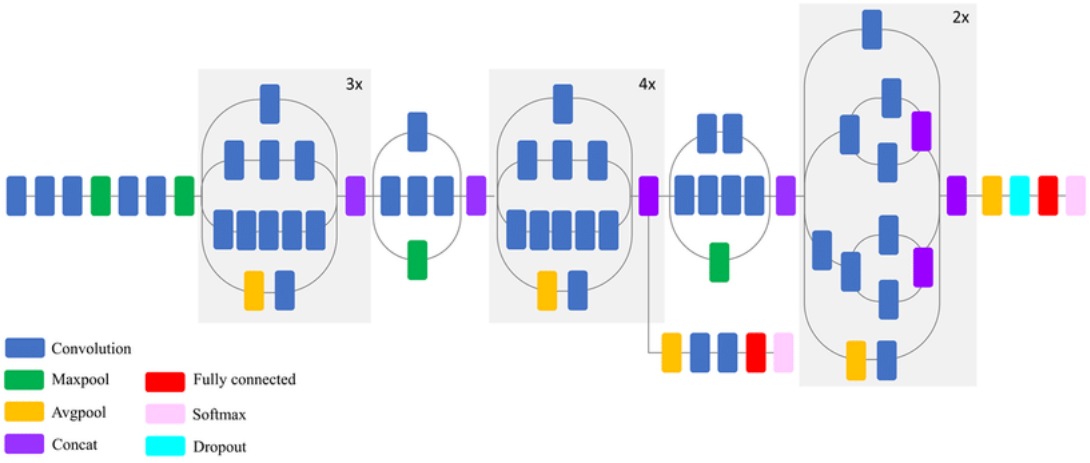


FIGURE 4.6: InceptionV3, extracted from Mahdianpari et al. [19].

4.4.4 Xception

The domain of deep learning has witnessed a slew of architectures, each with its unique features and optimizations. One such architecture is Xception, a name derived from "Extreme Inception." Proposed by François Chollet [6], the creator of the Keras deep learning library, Xception builds upon the Inception architecture but introduces a novel concept called depthwise separable convolutions.

Inception architectures use a mixture of different kernel sizes (1x1, 3x3, and 5x5) in their modules to process data at various spatial hierarchies simultaneously. While this concept was successful in previous models, Xception questions and modifies the basic structure of the Inception module. Instead of using different-sized kernels, Xception posits that the channel-wise spatial correlations and cross-channel correlations can be entirely decoupled.

The central innovation in Xception is the use of depthwise separable convolutions. This type of convolution operation is a variation of the standard convolution and is composed of two steps:

1. Depthwise Convolution: This involves applying a single convolutional filter per input channel. The depthwise convolution deals with spatial correlations.
2. Pointwise Convolution: Following the depthwise convolution, a 1x1 convolution (called pointwise convolution) is applied, which captures cross-channel correlations.

This decomposition of convolutions results in a substantial reduction in computational cost, while the performance remains competitive.

Xception's architecture consists of:

- An initial set of standard convolutions.
- A series of depthwise separable convolution blocks, which follow the depthwise-pointwise pattern, with some of the blocks having residual connections.
- A final global average pooling, followed by a fully connected layer.

Although the original Xception paper does not extensively focus on residual connections, they play a crucial role in its performance. Residual connections, which involve skipping one or more layers and adding the output from the previous layer to a subsequent one, aid in backpropagation, ensuring that the gradients flow well even in very deep networks.

Xception was evaluated on the ImageNet dataset, among others, and was found to outperform the InceptionV3 model in terms of accuracy, with fewer parameters and computational cost.

The introduction of Xception and its depthwise separable convolutions has had a notable impact on the field of deep learning. Many subsequent architectures have incorporated similar concepts to achieve efficiency.

Xception stands as a testament to the idea that challenging established norms can lead to significant advancements. By decoupling spatial and cross-channel operations and rethinking the fundamental building blocks of deep learning models, Xception achieved superior performance with reduced computational demands.

4.4.5 MobileNet

In the ever-evolving landscape of deep learning, the demand for efficient and lightweight models that can operate on devices with limited computational resources has surged.

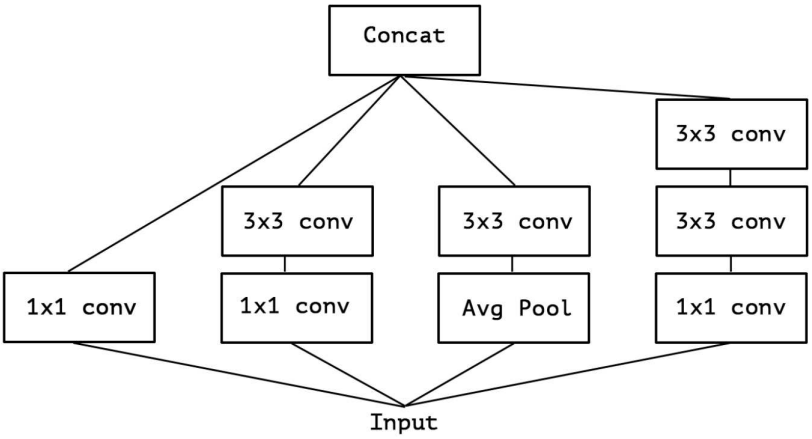


FIGURE 4.7: Xception example, extracted from Chollet [6].

Recognizing this need, Google researchers introduced MobileNet [13], an architecture optimized for mobile and embedded vision applications.

MobileNet focuses on reducing the computational cost of neural network models without a significant drop in accuracy. This is crucial for mobile devices that have limitations on power and computation. The primary innovation that facilitates this reduction is the use of depthwise separable convolutions.

Traditional convolution layers in neural networks combine filtering (which captures spatial features) and combining (which melds channel-wise information) into a single operation. MobileNet proposes to split this operation into two parts:

1. **Depthwise Convolution:** A spatial convolution is performed independently over each channel of the input. This operation captures the spatial context.

2. **Pointwise Convolution:** This is a 1x1 convolution responsible for constructing new features through computing combinations of the input channels.

Separating the traditional convolution into depthwise and pointwise convolutions dramatically reduces the computational cost.

To provide further flexibility, MobileNet introduces two hyperparameters: width multiplier (denoted as α) and resolution multiplier (ρ). These parameters allow for a trade-off between latency and accuracy.

- **Width Multiplier (α):** It's applied to the input and output channels, providing a reduction in the computational cost and the number of parameters.

- **Resolution Multiplier (ρ):** This alters the input dimensions of the image, consequently adjusting the internal tensor dimensions.

By modulating these multipliers, different variants of MobileNet can be created, each tailored for specific resource constraints.

MobileNet, with its compactness and efficiency, is suitable for a variety of applications, including object detection, facial recognition, and image classification, particularly on mobile devices, embedded systems, and IoT devices. Further variants, such as MobileNetV2 and MobileNetV3, were introduced, each building upon the previous with architectural tweaks and optimizations.

MobileNet represents a significant stride in making deep learning models more accessible and versatile. It underscores the notion that practicality of deploying

models in real-world scenarios is paramount. With the burgeoning growth of edge AI and on-device computations, architectures like MobileNet will continue to hold significance in the foreseeable future.

4.5 Models Tuning

4.5.1 CNN

For initial CNN model we utilize a Convolutional Neural Network (CNN) implemented in Keras. The architecture of the model comprises of three convolutional layers, each followed by a max-pooling layer, and two fully connected layers at the end. The specifics of each layer, including the number of filters, filter size, and activation functions, are detailed in the table below.

TABLE 4.1: Summary of the CNN Architecture

Layer Type	**Specifications**
Conv2D	32 filters, (3, 3) kernel, ReLU activation
MaxPooling2D	(2, 2) pool size
Conv2D	64 filters, (3, 3) kernel, ReLU activation
MaxPooling2D	(2, 2) pool size
Conv2D	128 filters, (3, 3) kernel, ReLU activation
MaxPooling2D	(2, 2) pool size
Flatten	-
Dense	64 units, ReLU activation
Dense (Output)	7 units (as `num_classes = 7`), Softmax activation

The model is compiled using the Adam optimizer, with a categorical crossentropy loss function.
Training Parameters:

- Number of Epochs: 50
- Batch Size: 32

TABLE 4.2: Results of the machine learning model

Metric	Value
Test accuracy	0.7857
Test precision	0.7857
Test recall	0.7857
Test F1 Score	0.7857

Over a span of 50 epochs, the model shows marked improvement in both training loss and accuracy. Initially, with a significant loss of 5.6856 at epoch 1, it's clear that the model has a limited grasp of the data. However, its understanding sharpens quickly over subsequent epochs. A pivotal observation emerges around epoch 10. Even though the training loss continues its descent, the validation loss begins to fluctuate without any substantial improvement. This fluctuation is a hallmark of overfitting. Overfitting is a predicament where the model becomes excessively intricate, essentially memorizing the training data instead of drawing general insights from it. This tendency towards overfitting is corroborated by the accuracy metrics:

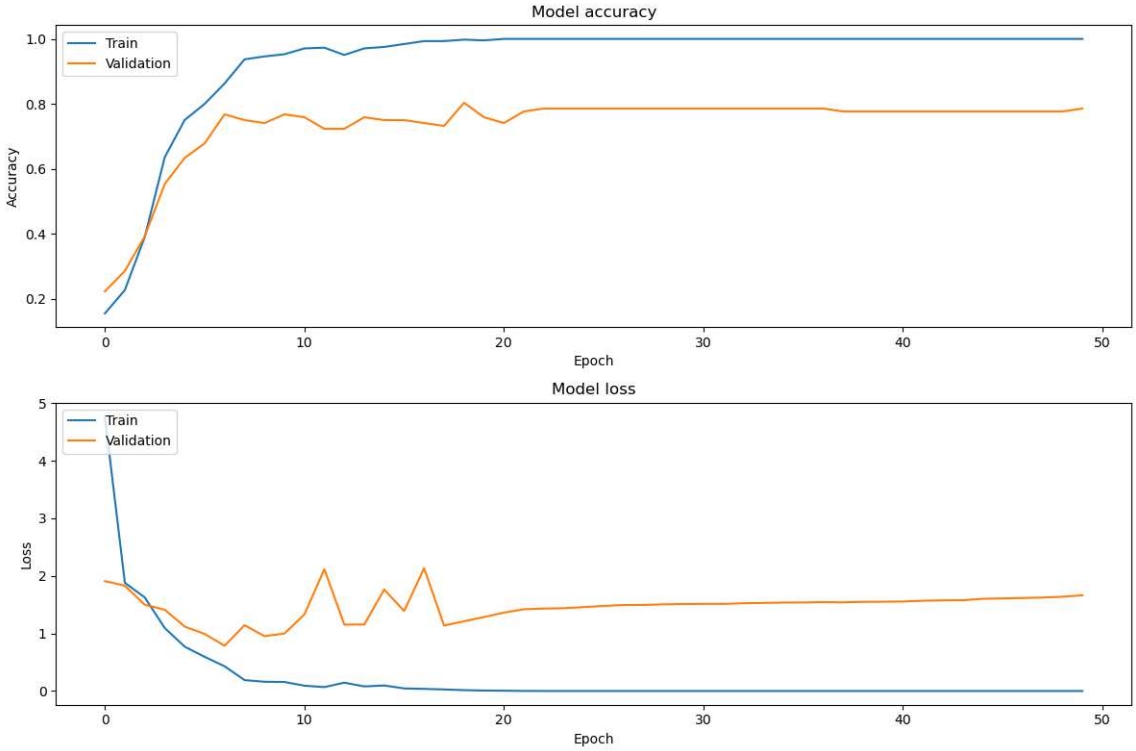


FIGURE 4.8: Plotted loss and accuracy of initial CNN model.

.

while training accuracy nears a flawless 100%, suggesting impeccable performance on the training data, the validation accuracy stagnates, hovering between 77-80%.

The tendency of over-fitting was decreased by following several steps. First was simplifying the model.

Initially, the third convolutional layer utilized 128 filters, which was later reduced to 64 in the updated model, implying a simplification in the feature extraction process. To bolster the model's resilience against overfitting, a dropout layer with a rate of 0.5 was incorporated post the convolutional stack, leading to a random deactivation of 50 % of the neurons during training. Additionally, the fully connected layer, following the convolutional and flattening layers, underwent a modification. Its neuron count was downscaled from 64 to 32, further streamlining the model's complexity. These adjustments were envisaged to strike a balance between the model's capacity and its generalization capabilities.

To furthermore optimize machine learning model's performance, we introduced several modifications to its specifications:

1. **Dropout Rate**: We adjusted the dropout rate to enhance the model's generalization. Previously, the model might have been overfitting the data, which is why a dropout rate of 0.6 was incorporated. This implies that during training, randomly 60% of the neurons are dropped out during each update cycle, preventing any single neuron from becoming overly specialized.

TABLE 4.3: Updated CNN structure.

Layer Type	Specification
Conv2D	32 filters, (3, 3), activation='relu'
MaxPooling2D	(2, 2)
Conv2D	64 filters, (3, 3), activation='relu'
MaxPooling2D	(2, 2)
Conv2D	64 filters, (3, 3), activation='relu'
MaxPooling2D	(2, 2)
Flatten	
Dropout	Rate = 0.5
Dense	32 neurons, activation='relu'
Dense	num_classes, activation='softmax'

2. **Learning Rate Scheduler (ReduceLROnPlateau)**: An adaptive learning rate scheduler has been employed. The settings for this scheduler are as follows:

$$\begin{aligned} \text{monitor} &: \text{val\_loss} \\ \text{factor} &: 0.2 \\ \text{patience} &: 5 \\ \text{min\_lr} &: 0.001 \end{aligned}$$

   The goal is to ensure the model converges more efficiently without getting stuck in local minima.

3. **Early Stopping**: To further prevent overtraining, we've employed an early stopping mechanism with the following criteria:

$$\begin{aligned} \text{monitor} &: \text{val\_loss} \\ \text{patience} &: 5 \end{aligned}$$

   This halts the training if no improvement is seen after 5 consecutive epochs.

These enhancements aim to bolster the model's robustness, prevent overfitting, and optimize the training process.

TABLE 4.4: Results of the machine learning model.

Metric	Value
Test accuracy	0.8125
Test precision	0.8108
Test recall	0.8035
Test F1 Score	0.8071

4.5.2 ResNet

The image classification architecture harnesses the capabilities of the renowned ResNet50 model, a deep convolutional neural network that was initially trained on the extensive ImageNet dataset. By leveraging the ResNet50 model, we are poised to benefit from the rich features it has learned, which can serve as a solid foundation for our classification task.

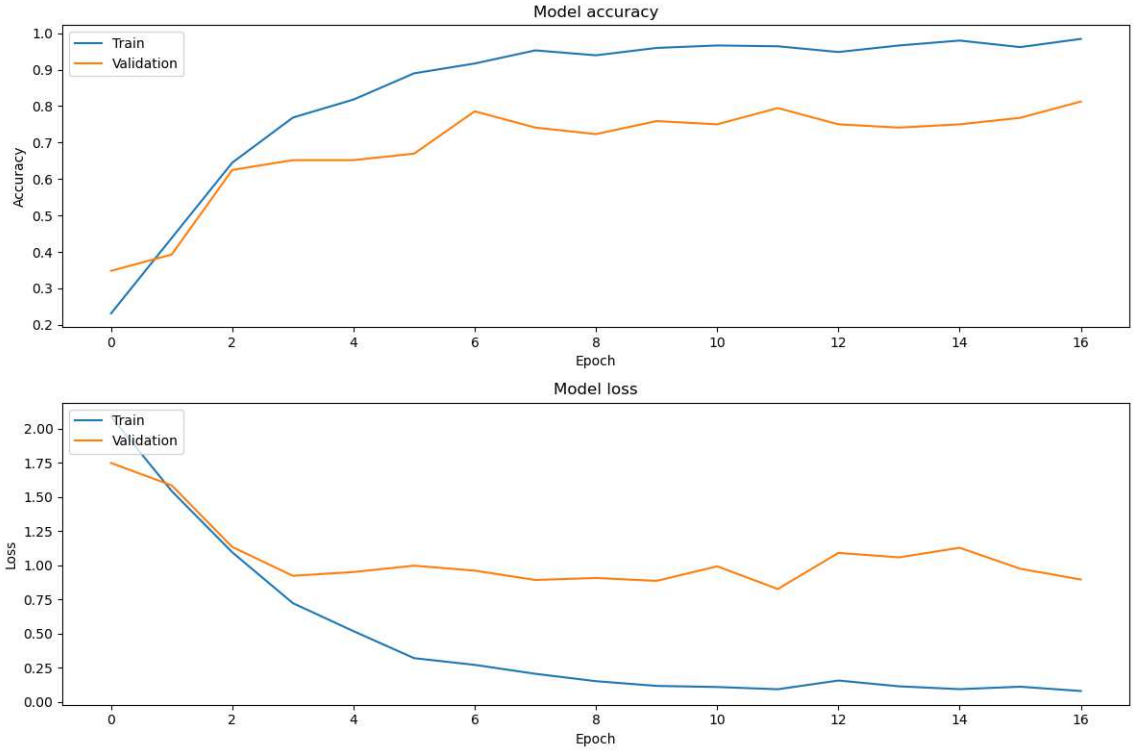


FIGURE 4.9: Plotted loss and accuracy of final CNN model.

To adapt this base model to our specific classification requirements, a sequence of custom layers is added on top of the ResNet50 structure. The output from the base model is first flattened to convert it into a one-dimensional vector. Following this, two dense layers, each with 1,024 neurons and adopting the ReLU activation function, are introduced. These layers play a pivotal role in discerning high-level features from the image dataset. To combat the risk of overfitting, especially given the complexity of the network, dropout layers are strategically positioned after each dense layer. These layers randomly nullify 50% of their input units during training, acting as a regularization mechanism.

The culmination of the model is marked by a dense layer with 7 neurons paired with a softmax activation function. This configuration suggests that the task is set out for categorizing images into one of seven distinct categories. The model, in response, will output a probability distribution spanning these seven categories.

A vital step before the model's training is ensuring that the weights from the pretrained ResNet50 are not inadvertently modified during our training process. This is achieved by freezing all the layers in the ResNet50 base model, ensuring that only the newly added layers will undergo weight adjustments.

For the training process, the model is compiled with the Adam optimization algorithm. The choice of the categorical crossentropy loss is deliberate, given its efficacy for multi-class classification tasks. To keep track of the model's training prowess, a trio of metrics, including accuracy, precision, and recall, are monitored. Furthermore, to optimize the training duration and prevent unnecessary epochs that don't contribute to performance enhancement, an early stopping mechanism is incorporated. This callback vigilantly monitors the validation loss and will halt training if there's an absence of improvement over a span of five epochs.

In assessing the deep learning model's training logs, several critical observations

TABLE 4.5: Results of the ResNet machine learning model

Metric	Value
Test accuracy	0.0714
Test precision	0.0
Test recall	0.0
Test F1 Score	0.0

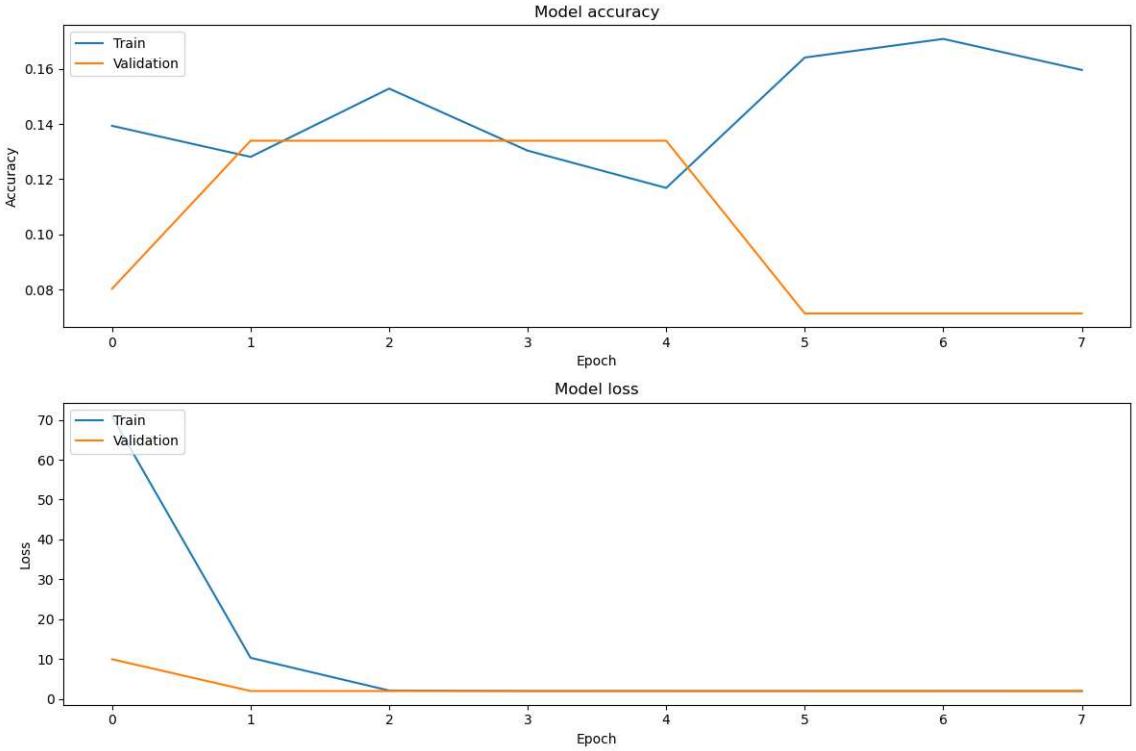


FIGURE 4.10: Plotted loss and accuracy of pretrained ResNet model.

.

were made. The model commenced with a notably high training loss of 70.8669 during its initial epoch, but by the second epoch, there was a precipitous drop to 10.2807. This swift reduction hints at the model beginning with an unfavorable initialization and undergoing considerable weight adjustments in the initial stages.

Post this rapid decline, the model's progress seemed to stagnate. By the third epoch and beyond, both the training and validation loss plateaued around a value of 1.945. This stagnation, coupled with a persistently low accuracy that lingered between 11% to 17% for the training set, indicates that the model is only marginally better than random guessing for this 7-class classification problem.

Perhaps most alarmingly, the model's precision and recall metrics, starting from the third epoch, plummeted to zero for both training and validation sets. This implies a stark inability to correctly predict any positive samples, a severe shortcoming for any classification model. Moreover, validation loss showcased a gradual yet consistent increase, raising concerns about the model's potential overfitting, even though its overall performance remained unsatisfactory.

These observations collectively suggest that the model is facing significant challenges: when using pre-trained weights, it's crucial to note that ResNet50 weights are commonly trained on the ImageNet dataset. This means that the features it has

learned from ImageNet may not be as pertinent to choseen for the projecty specific dataset, especially in comparison to a simpler CNN model that was either trained from scratch. Futhermore, in terms of model capacity, ResNet50 possesses a more profound architecture, giving it a high capacity to understand and represent complex data patterns. This depth is undoubtedly beneficial for intricate datasets. However, for ours simpler or smaller datasets, such a high model capacity can inadvertently become a disadvantage, potentially leading to overfitting.

Concluding, we can see that no matter what we do to improve performance, ResNet models seem not be suitable for this data type.

4.5.3 Dense Network

For the classification of images in our dataset, we designed and implemented a feed-forward neural network, often referred to as a Dense or Fully Connected Neural Network.

The **Input Layer** is tasked with accepting the reshaped one-dimensional image array. The dimensionality corresponds to the flattened version of our image data.

The architecture incorporates two **Hidden Layers**. The first hidden layer encompasses 64 neurons, utilizing the Rectified Linear Unit (ReLU) as its activation function. The choice of ReLU offers advantages in terms of computational efficiency and convergence. Subsequent to the first, a second hidden layer is incorporated with 32 neurons, also leveraging the ReLU activation function. The purpose of multiple hidden layers is to learn and capture intricate patterns and nuances from the image data.

The **Output Layer** functions as the decision layer of the network. It comprises neurons equal to the number of distinct classes present in the dataset. Given its role in multi-class classification, the softmax activation function is employed. The softmax ensures that the output values are in the range $(0, 1)$ and their sum equals 1, effectively providing a probability distribution over the classes.

The model's structure is linear, meaning each layer's output is the subsequent layer's input, ensuring a seamless flow of information. The utilization of dense layers enables the network to learn and establish intricate connections, thereby making decisions based on complex patterns discerned during the training process.

TABLE 4.6: Results of the Dense Neural Network model.

Metric	Value
Test accuracy	0.5267
Test precision	0.5714
Test recall	0.5000
Test F1 Score	0.5333

Considering training logs of the model, the model appears to be learning and improving over the epochs, with mild signs of overfitting. It would be helpful to continue training and monitor for any signs of stagnation or increasing overfitting. Adjustments to hyperparameters or the architecture might be considered depending on the results of further training epochs.

The spatial characteristics of our data and task might align better with the inherent biases of a CNN compared to a Dense Network.

CNNs leverage the spatial organization of data, especially in scenarios like images where adjacent pixels often share semantic significance. On the other hand,

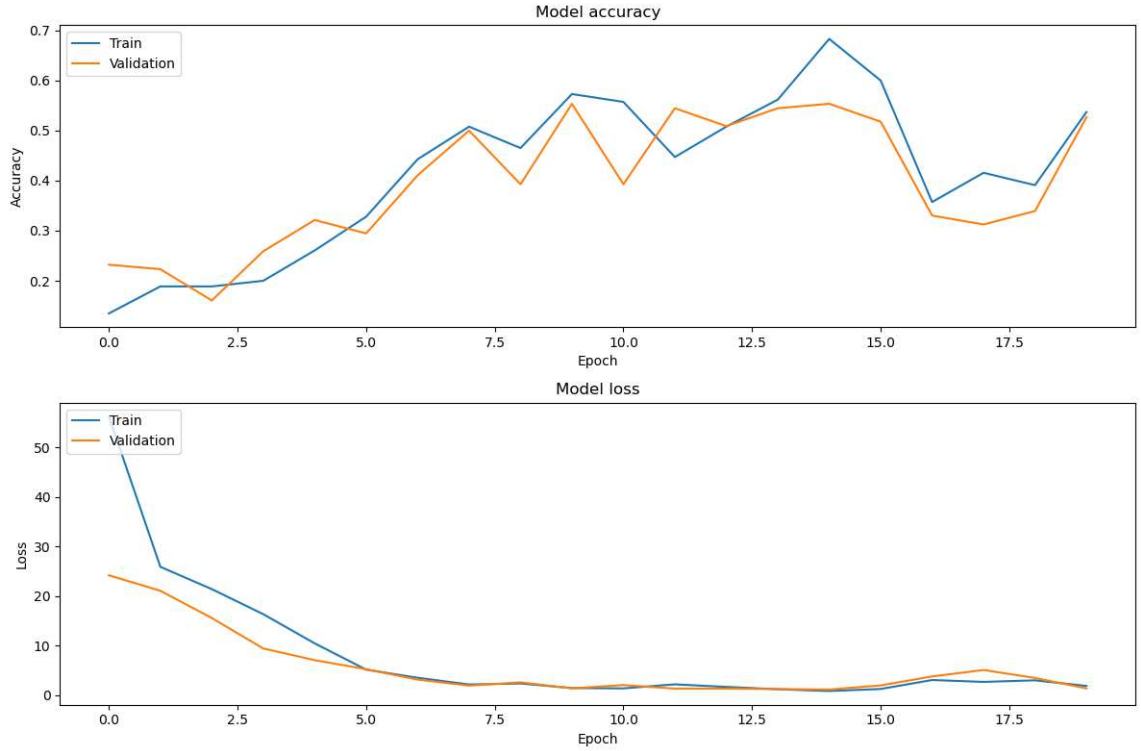


FIGURE 4.11: Plotted loss and accuracy of DenseNet model.

.

Dense networks don't consider this spatial relationship, potentially leading to inefficiencies when the data has inherent spatial or temporal patterns.

4.5.4 InceptionV3

The model is constructed on the backbone of the robust InceptionV3 architecture, which is initialized with weights from the renowned ImageNet dataset.

Base Model (InceptionV3): The model incorporates the InceptionV3 architecture but discards its top layer. This flexibility allows for modifications tailored to the task at hand. Furthermore, the input is auto-adjusted based on the first image from the dataset, ensuring synchronization with the data's specific dimensions.

Global Average Pooling Layer: Positioned after the InceptionV3 foundation, this layer compresses spatial dimensions and focuses on extracting significant features.

Dense Layer: This fully-connected layer houses 1024 units and is powered by the ReLU activation function. It's meticulously crafted to understand and derive feature nuances from the preceding pooling stage.

Dropout Layer: With a dropout rate of 0.6, this layer plays a crucial role in regularization. By sporadically nullifying certain input units during training updates, it provides a robust defense against overfitting.

Output Dense Layer: Marking the model's conclusion, this layer's unit count mirrors the total class number, standing at seven. The application of the Softmax activation ensures that its outputs lie between [0,1], effectively illustrating class probabilities.

The training logs presented give an insightful overview of a deep learning model's performance over multiple epochs. Initially, during the first epoch, the model started

TABLE 4.7: Results of the Inception Neural Network model.

Metric	Value
Test accuracy	0.8125
Test precision	0.8484
Test recall	0.7500
F1 Score	0.7962

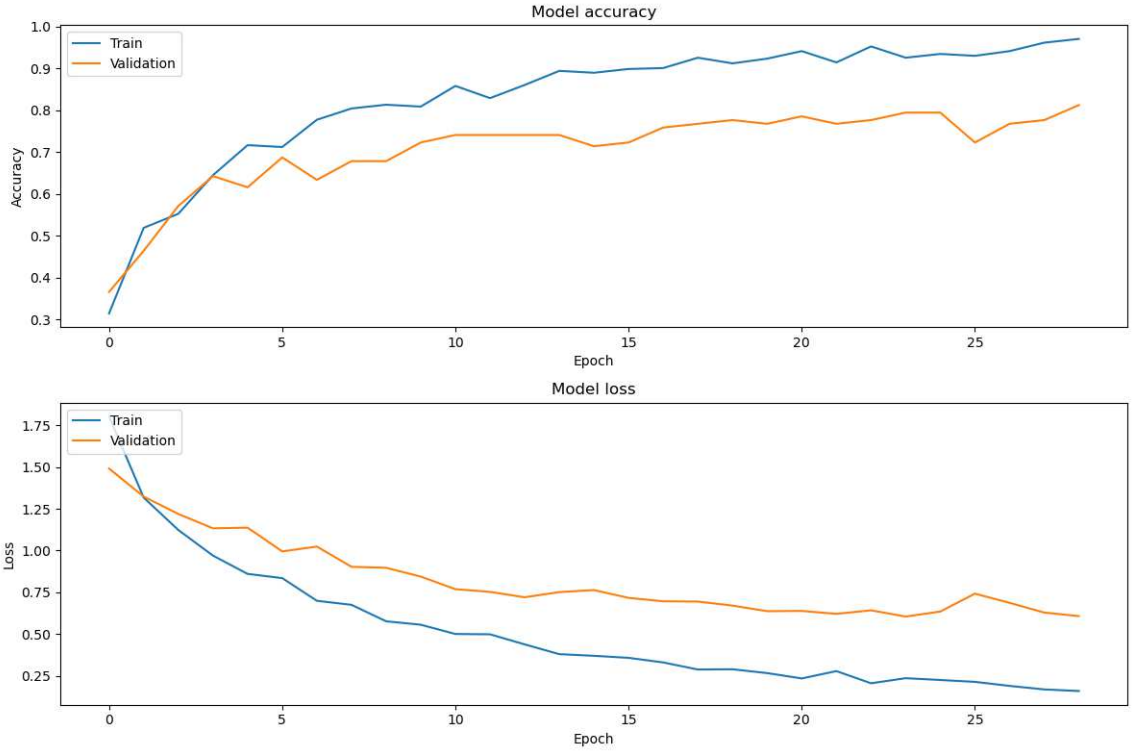


FIGURE 4.12: Plotted loss and accuracy of InceptionV3 model.

with a modest accuracy of 31.46% on the training set, which was unsurprising as most models begin their training with limited understanding. However, a promising aspect was that the precision was already at 66.67%, though the recall remained low at 6.74

As the training progressed, a consistent trend of improvement was evident. By the 10th epoch, the training accuracy had climbed to 80.90%, and both precision and recall showcased significant progress, reaching 90.36% and 73.71% respectively. The validation accuracy also demonstrated substantial growth, achieving 72.32%, indicating the model was generalizing well and not just memorizing the training set.

By the mid-20s in terms of epochs, the training accuracy had already surpassed the 90% mark, highlighting the model's enhanced capability. In particular, by the 24th epoch, the training accuracy peaked at 95.28% with an impressive precision of 96.71% and recall of 92.36%. The validation set also saw commendable results, with accuracy reaching as high as 79.46% during the same epoch.

However, there were a few fluctuations in the validation metrics, especially in the later epochs, suggesting possible overfitting. For instance, by the 27th epoch, while the training accuracy was at a robust 94.16%, the validation accuracy dropped to 76.79%.

In summary, the model showcased a commendable performance throughout its training journey. Starting from humble beginnings in its initial epoch, the model rapidly learned and adapted, achieving over 90% accuracy in the training set by the mid-phase of its training. The validation set results also indicated good generalization capabilities, though some inconsistencies hinted at potential overfitting in the later stages. Overall, with its high precision and recall rates, the model has certainly demonstrated efficacy and potential, and with some fine-tuning, it could achieve even better generalization on unseen data.

4.5.5 Xception

The model's configuration begins with the establishment of key parameters. These parameters serve as the backbone for training and include the number of target categories (`num_classes`), which is set to seven, indicating that there are seven distinct labels or types within the dataset. Additionally, the model is intended to undergo training for a total of 50 cycles or `num_epochs`. During each of these cycles, a set of 32 images, designated by the `batch_size`, will be fed into the model.

At the heart of this image classification system is the Xception architecture. This choice is grounded in Xception's well-established efficacy for tasks involving image categorization. In this configuration, the Xception model is not used in its entirety. Rather, it's initialized using pre-trained weights derived from the extensive ImageNet dataset. Notably, the uppermost layers, which traditionally comprise fully connected layers, are omitted from this model.

Subsequent to setting up the Xception base, the model undergoes further customization. This begins with the introduction of a `GlobalAveragePooling2D` layer, an addition aimed at streamlining the spatial dimensions of the data. Following this, a dense layer equipped with 1,024 neurons is added. This layer, activated by the ReLU function, plays a pivotal role in facilitating the model's capacity for intricate feature detection. Recognizing the potential for models to overfit, or become excessively tailored, to the training data, a dropout layer is also incorporated. Set at a rate of 0.6, this layer occasionally turns off a portion of its neurons during training, ensuring that the model remains generalized. The final custom layer is designed specifically for the classification task. It employs a dense structure with seven units, corresponding to the seven possible classes. This layer, activated by the softmax function, outputs the probability distribution over these classes.

Training a model is not a static process, and the chosen architecture embraces this dynamism. Initially, all the layers that are inherent to the Xception model are set in a frozen state. This tactic, frequently seen in transfer learning, ensures that during the training process, only the weights of the newly introduced layers are adjusted. As for the model's compilation, it leans on the Adam optimizer—a method known for its efficiency. The model's performance, in terms of its predictions, is evaluated against the ground truth using the categorical cross-entropy loss metric. Additionally, to gain a nuanced understanding of the model's efficacy, accuracy, precision, and recall are monitored throughout.

Lastly, the training process is augmented with a couple of key enhancements. The `ReduceLROnPlateau` callback is a proactive feature. If it observes that the validation loss isn't showing signs of improvement, it automatically reduces the learning rate. This adjustment can facilitate better convergence during training. In tandem, the `EarlyStopping` callback acts as a safeguard against unnecessary computation. If the model's validation loss doesn't show any positive shifts over a span of five

epochs, this feature halts the training. This not only saves computational resources but also prevents the model from potential overfitting due to excessive training.

TABLE 4.8: Results of the Xception Neural Network model.

Metric	Value
Test accuracy	0.8125
Test precision	0.8484
Test recall	0.7500
Test F1 Score	0.7962

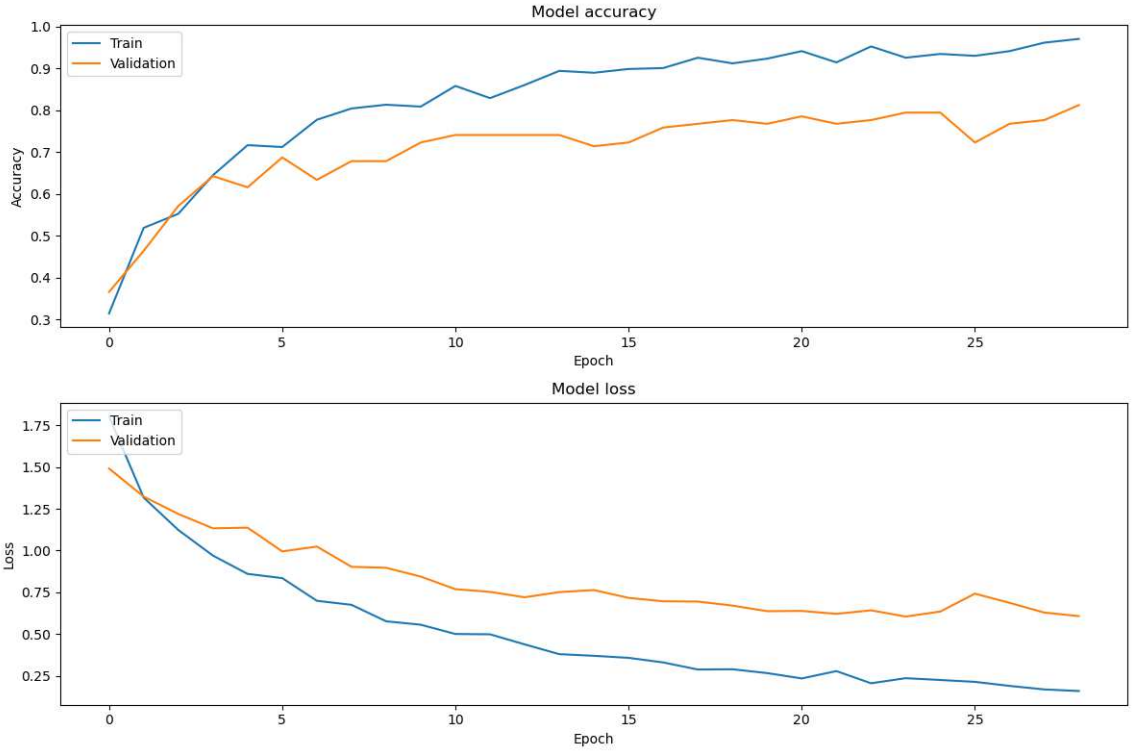


FIGURE 4.13: Plotted loss and accuracy of Xception model.

4.5.6 MobileNet

The foundational model we use is **MobileNet**, known for its efficiency in tasks related to mobile and embedded vision. It has been pretrained on the **ImageNet** dataset. For our purposes, when initializing the MobileNet, its top layers are excluded by setting `include_top` to `False`. The rationale behind this is that the ImageNet's original classification layer, tailored for 1,000 categories, is not congruent with our specific task, which has just 7 categories (i.e., `num_classes = 7`).

To make this model fit our needs, we've integrated the following layers:

1. A `GlobalAveragePooling2D` layer, responsible for reducing the spatial dimensions.
2. A fully-connected `Dense` layer, having 1,024 units and using the `ReLU` activation function.
3. A `dropout` layer with a dropout rate of 0.6, aimed at mitigating overfitting.

4. Lastly, for classification into one of the 7 categories, a `Dense` layer with 7 units combined with a `softmax` activation is added.

By fusing the base **MobileNet** and the newly added layers, we get a unified, optimized model for our task.

TABLE 4.9: Initial results of the MobileNet model.

Metric	Value
Test accuracy	0.6875
Test precision	0.7064
Test recall	0.6875
Test F1 Score	0.6968

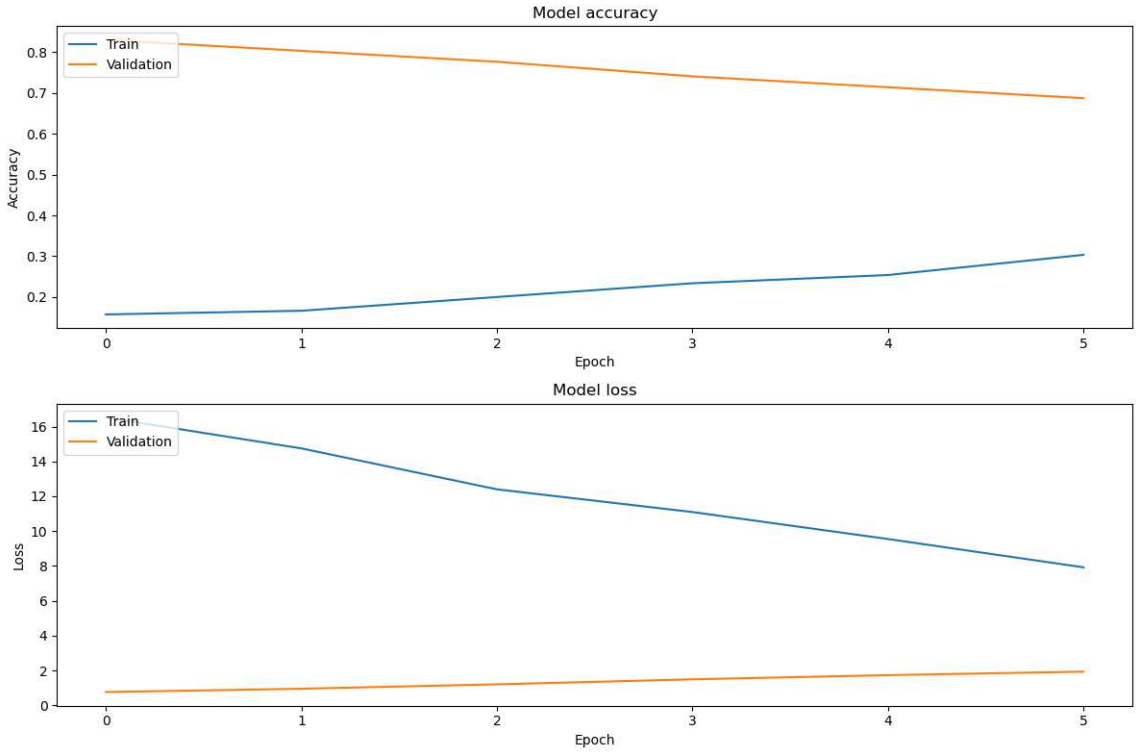


FIGURE 4.14: Plotted loss and accuracy of initial MobileNet model.

.

After refining the model further, we attained the peak accuracy for this particular model by transitioning from the Adam optimizer to RMSprop. The training logs provide insight into the behavior of a model trained using the TensorFlow framework. The logs have flagged a potential issue with the 'input shape' parameter—it doesn't align with standard sizes like 128, 160, 192, or 224. However, despite this discrepancy, the model defaults to using weights for an input shape of (224, 224).

Regarding the training structure, the model undergoes 50 epochs, and each epoch encompasses 14 steps. This pattern implies that the dataset's size, divided by the selected batch size, results in 14 batches for every epoch.

As the training progresses, we observe a decrease in the loss metric, signifying positive growth. Simultaneously, other metrics, such as accuracy, precision, and recall, show a general upward trend. Notably, the model's peak accuracy reaches approximately 94.61% by the 28th epoch.

TABLE 4.10: Updated results of the MobileNet model.

Metric	Value
Test accuracy	0.7503
Test precision	0.7614
Test recall	0.7410
Test F1 Score	0.7511

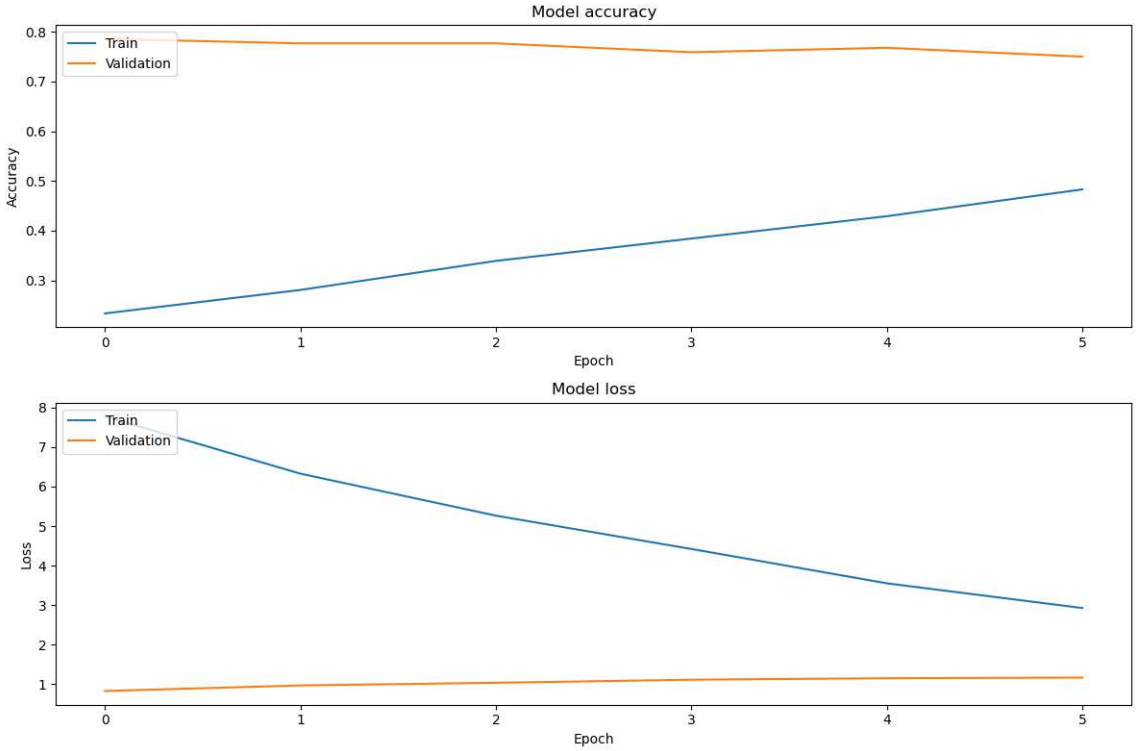


FIGURE 4.15: Plotted loss and accuracy of final MobileNet model.

When we shift our focus to the model's performance on the validation set, the validation accuracy reaches its zenith at about 83.04% during the 25th epoch. But to gain a full picture, it's crucial to evaluate all 50 epochs. The fluctuations in validation metrics could suggest overfitting or the model's sensitivity to specific batches of validation data. This potential inconsistency may be alleviated using methods like dropout, dataset augmentation, or regularization.

In conclusion, while the training data results appear promising.

4.6 Source Code

Source code can be found in this GitHub repository CSImageRecognition.

Chapter 5

Evaluation

This chapter describes the empirical evaluation using real-world dataset, as well as report and analyze the experimental results.

5.1 Results Evaluation

Leveraging WiFi Channel State Information (CSI) for human movement recognition represents a cutting-edge confluence of wireless communication and machine learning disciplines. Human movements can induce perturbations in WiFi signals, which manifest as distinct patterns. Encoding these patterns as images and subjecting them to image recognition models presents a novel paradigm for movement detection and classification.

Prominent Performers:

- **InceptionV3, Xception, and CNN**:
  - Empirical results suggest that these models exhibit efficacy in discerning intricate patterns in CSI-derived images. Their superior accuracy, complemented by a harmonized precision-recall equilibrium, positions them as front-runners for more rigorous, real-world evaluations.
  - The commensurate performance of InceptionV3 and Xception is consistent with existing literature, given their design philosophy tailored for sophisticated image recognition challenges.

Moderate Performer:

- **MobileNet**:
  - This model, recognized for its architectural parsimony, produces commendable results. In scenarios mandating computational frugality, such as edge devices or real-time processing, MobileNet emerges as a pragmatic choice, albeit with a nuanced decrement in performance.

Suboptimal Performers:

- **DenseNet**:
  - Its median performance warrants introspection. While DenseNet has been acclaimed for its performance in myriad image recognition endeavors, its modest scores in the present context might indicate a necessity for architectural fine-tuning or an assessment of its applicability for this unique application.
- **ResNet**:

– The pronounced underperformance of ResNet is an aberration from its documented success in image classification. Potential catalysts for such discrepancies could encompass:
  1. Selection of non-optimal hyperparameters or training regimen.
  2. Adoption of a ResNet variant incongruent with the intrinsic complexity of the CSI patterns.
  3. Pre-processing or data augmentation anomalies during CSI image generation.

TABLE 5.1: Consolidated Results of the Machine Learning Models.

Metric	CNN	ResNet	DenseNet	InceptionV3	Xception	MobileNet
Test accuracy	0.8125	0.0714	0.5268	0.8125	0.8125	0.7500
Test precision	0.8108	0.0	0.5714	0.8485	0.8485	0.7615
Test recall	0.8036	0.0	0.500	0.75	0.7500	0.7411
Test F1 Score	0.8072	0.0	0.5333	0.7962	0.7962	0.7511

For researchers and practitioners at the intersection of WiFi CSI and human movement recognition:

1. **Operational Deployment**: The empirical findings advocate for the consideration of InceptionV3 and Xception for pilot deployments or further iterative research.

2. **Edge-Centric Applications**: For environments constrained by computational resources, the efficiency of MobileNet underscores its potential relevance.

3. **Further Investigation**: ResNet's unanticipated results underscore the exigency for a meticulous audit of the employed variant, training protocols, and data preprocessing steps.

4. **Architectural Relevance**: Given the distinctive nature of CSI patterns, there is a plausible merit in investigating bespoke model architectures or domain-adaptive modifications to extant ones, aimed at enhancing congruence with the task intricacy.

5.2 Comparison With Benchmark

We'll conduct a comparative analysis using the benchmark SenseFi, as detailed by Yang et al. [35]. To ensure consistency and credibility in our comparison, all benchmark tests were performed utilizing the source code provided in the original paper.

SenseFi stands as a prominent benchmark and model zoo platform tailored for deep learning applications specific to WiFi CSI (Channel State Information) sensing. Within its arsenal, SenseFi offers a suite of renowned deep learning models optimized for WiFi sensing challenges.

Distinctive Attributes of the SenseFi Platform:

- Examination of the adaptability of prominent deep learning models, which have their roots in computer vision and natural language processing, for the realm of WiFi sensing.

- Incorporation of recognized public datasets such as UT-HAR and Widar, while pioneering with the introduction of NTU-Fi HAR and Human-ID datasets.
- Venture into the domain of transfer learning, emphasizing the cross-transfer of knowledge among diverse sensing tasks.
- Delve into unsupervised learning, focusing on the art of feature extraction devoid of data annotations.
- Facilitation of open-access benchmarking scripts, marking a trailblazing effort in establishing benchmarks for WiFi sensing deep learning endeavors.

For the purpose of this discourse, our focus narrows down to the UT-HAR dataset, aligning with its utilization in the thesis project. Additionally, the benchmark envelops a spectrum of models, enumerating MLP, LeNet, ResNet18, ResNet50, ResNet101, RNN, GRU, LSTM, BiLSTM, CNN+GRU, and ViT. To streamline our comparative analysis, the ensuing discussions will pivot around the mean values of the metrics, serving as emblematic outcomes.

Tables 5.2 and 5.3 depict the replicated results from the benchmark.

TABLE 5.2: Results of the Machine Learning Model MLP for the benchmark.

Model	Metric	Average Value
MLP	Test accuracy	0.98
	Test precision	0.98
	Test recall	0.98
	Test F1 Score	0.98

TABLE 5.3: Results of the Machine Learning Model BiLSTM for the benchmark.

Model	Metric	Average Value
BiLSTM	Test accuracy	0.81
	Test precision	0.82
	Test recall	0.81
	Test F1 Score	0.81

As shown in Figures 5.1, 5.2, and 5.3, we have a consistent theme in the comparative performance between the benchmark models and our experimental model. Each figure represents a different metric, from accuracy to precision and recall.

In Figure 5.1, it is clear that our model's accuracy matches closely with the BiLSTM configuration. This trend is similarly observed in the precision graph shown in Figure 5.2. However, a marginal difference can be discerned in the recall in Figure 5.3, with our model slightly trailing the BiLSTM's performance.

Despite these nuanced variations, the salient takeaway is the competitive stance of our model. While it does not outshine the benchmark's scores, its parallelism with the BiLSTM across different metrics speaks to its viability. Such results signal that, with continued refinement and potential tweaks, our experimental model may indeed bridge the narrow performance gaps and possibly rival the benchmarks in future iterations.

The bar chart provides a visual representation comparing the accuracy scores of two distinct models. Among these, one is derived from a recognized benchmark, and the other embodies our experimental results.

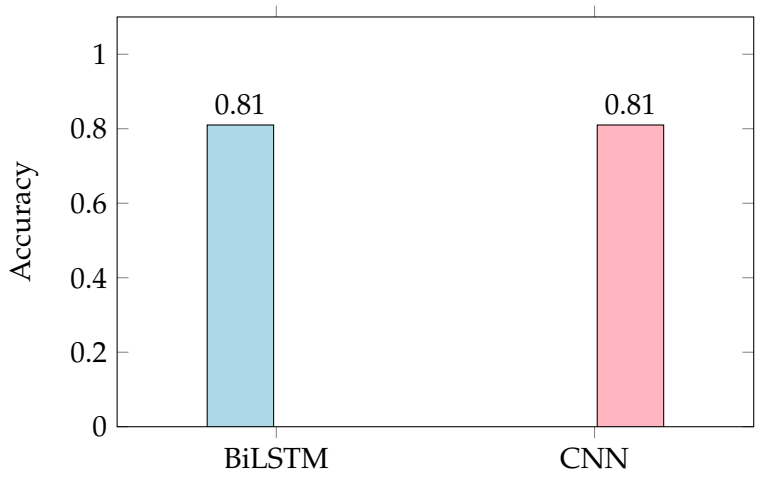


FIGURE 5.1: Bar chart showing accuracy of benchmark/our model.

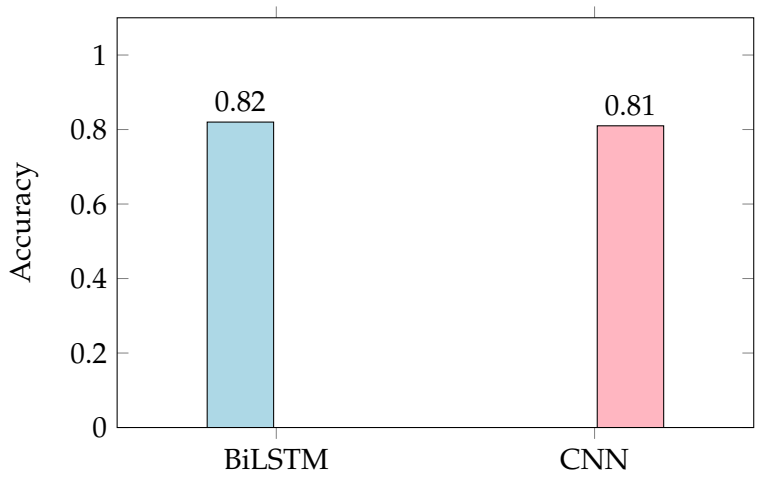


FIGURE 5.2: Bar chart showing precision of benchmark/our model.

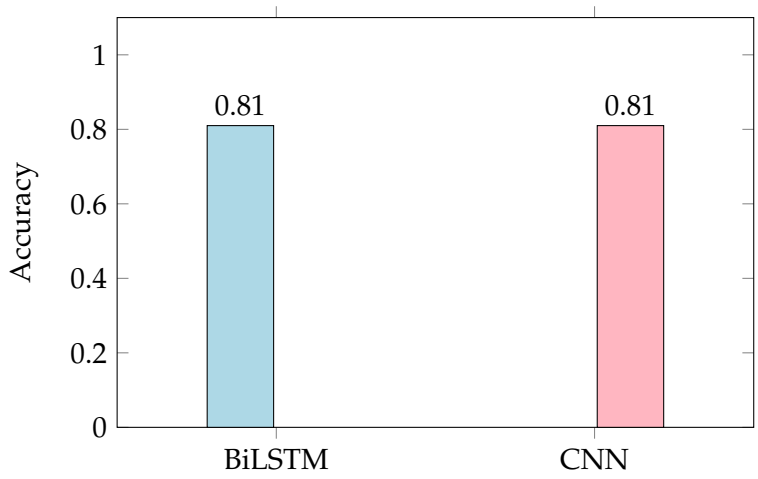


FIGURE 5.3: Bar chart showing recall of benchmark/our model.

The benchmark model, delineated in blue, is the BiLSTM configuration, which achieves a commendable accuracy of 0.81.

Our own model's representation is the CNN bar, shaded in red, and it mirrors the

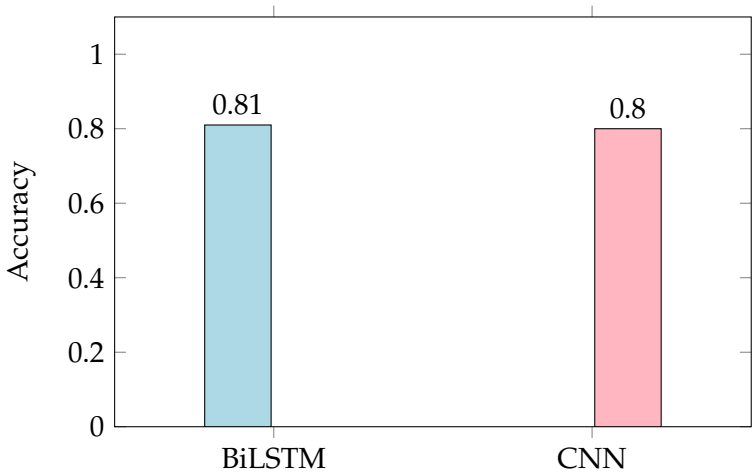


FIGURE 5.4: Bar chart showing F1-score of benchmark/our model.

BiLSTM's performance with an accuracy of 0.81. While our result does not surpass the benchmark, the congruent accuracy with the BiLSTM model is a testament to its potential. This implies that, notwithstanding its experimental nature, our model showcases compatibility and promise when juxtaposed against renowned benchmarks. This augurs well, hinting that with ensuing refinements, our model could potentially elevate its performance trajectory in subsequent iterations.

Turning CSI data into images presents both opportunities and challenges. On the bright side, it offers a pathway to harness advanced image processing techniques, which might not be accessible with raw data. However, there's a tangible risk of losing critical information during this transformation, which could have contributed to the slight dip in performance compared to SenseFi.

Moving forward, the possibility of blending elements from both methodologies or refining the image conversion process could lead to improved outcomes. In essence, while the traditional methods currently have the upper hand, the image-based approach carves out a promising avenue worth further exploration and refinement.

Chapter 6

Conclusions

In this thesis, we explored human movement recognition using visualized Channel State Information (CSI) WiFi readings. Pitting traditional methods against a novel approach, our investigation presented a rich tapestry of results and implications.

The innovative technique of representing CSI data as three-channeled images, despite its nascent stage, showcases a profound shift from conventional methodologies. Achieving an accuracy of 0.81, this approach does fall behind the benchmark set by SenseFi, which boasts an accuracy of 0.92 using traditional machine learning techniques on raw CSI readings.

Beyond the numerical comparison, the merits of the image-based method are manifold. By transposing CSI data into an image format, we tap into the expansive world of convolutional neural networks (CNNs) and image processing techniques. This transformation not only broadens the scope of techniques applicable but also potentially enables data augmentation specific to image data, enhancing the diversity and robustness of datasets. Yet, innovation also brings challenges, such as ensuring minimal information loss during data transformation and the intricacies of navigating vast neural network architectures.

On the other hand, the SenseFi benchmark, with its impressive accuracy, reaffirms the robustness of raw CSI readings and the maturity of traditional methodologies. Raw CSI readings, inherently, seem to possess features conducive to prediction. However, it is pertinent to consider potential constraints with SenseFi, especially in the backdrop of evolving data complexities and the need for adaptability in rapidly shifting data landscapes.

Concerning the future work, several exciting avenues emerge. The idea of developing a hybrid model, combining the strengths of both the novel and traditional techniques, is particularly promising. Furthermore, optimizing the process of converting raw CSI data into images to retain intrinsic data qualities can elevate the performance of the innovative approach. We also see untapped potential in employing pre-trained models on extensive image datasets and collaborating with experts from the broader fields of computer vision and signal processing.

In summary, while the SenseFi benchmark underscores the power of traditional methodologies, our innovative image-based approach marks a paradigm shift in CSI data analysis for human movement recognition. By opening doors to the realms of convolutional neural networks and advanced image processing, it beckons a future where sophisticated movement recognition becomes even more nuanced and precise.

Bibliography

[1] Fadel Adib and Dina Katabi. "See through walls with WiFi!" In: vol. 43. Aug. 2013, pp. 75–86. ISBN: 978-1-4503-2056-6. DOI: 10.1145/2534169.2486039.

[2] Peter Almers et al. "Survey of Channel and Radio Propagation Models for Wireless MIMO Systems." In: *EURASIP J. Wireless Comm. and Networking* 2007 (Jan. 2007).

[3] Mohammad Abu Alsheikh et al. *Deep Activity Recognition Models with Triaxial Accelerometers*. 2016. arXiv: 1511.04664 [cs.LG].

[4] Laith Alzubaidi et al. "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions". In: *Journal of Big Data* 8 (Mar. 2021). DOI: 10.1186/s40537-021-00444-8.

[5] Davide Anguita et al. "A Public Domain Dataset for Human Activity Recognition using Smartphones". In: Jan. 2013.

[6] François Chollet. *Xception: Deep Learning with Depthwise Separable Convolutions*. 2017. arXiv: 1610.02357 [cs.CV].

[7] Jianyang Ding and Yong Wang. "WiFi CSI-Based Human Activity Recognition Using Deep Recurrent Neural Network". In: *IEEE Access* 7 (2019), pp. 174257–174269. DOI: 10.1109/ACCESS.2019.2956952.

[8] Xue Ding et al. "Wi-Fi-Based Location-Independent Human Activity Recognition via Meta Learning". In: *Sensors* 21.8 (2021). ISSN: 1424-8220. DOI: 10.3390/s21082654. URL: https://www.mdpi.com/1424-8220/21/8/2654.

[9] Parisa Fard Moshiri et al. "A CSI-Based Human Activity Recognition Using Deep Learning". In: *Sensors* 21.21 (2021). ISSN: 1424-8220. DOI: 10.3390/s21217225. URL: https://www.mdpi.com/1424-8220/21/21/7225.

[10] Xu Feng, Khuong An Nguyen, and Zhiyuan Luo. "A survey of deep learning approaches for WiFi-based indoor positioning". In: *Journal of Information and Telecommunication* 6.2 (2022), pp. 163–216. DOI: 10.1080/24751839.2021.1975425. eprint: https://doi.org/10.1080/24751839.2021.1975425. URL: https://doi.org/10.1080/24751839.2021.1975425.

[11] Glenn Forbes, Stewart Massie, and Susan Craw. "WiFi-based Human Activity Recognition using Raspberry Pi". In: *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*. 2020, pp. 722–730. DOI: 10.1109/ICTAI50040.2020.00115.

[12] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[13] Andrew Howard et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". In: (Apr. 2017).

[14] A. Khalili et al. *Wi-Fi Sensing: Applications and Challenges*. Jan. 2019.

[15] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[16] Chien-Cheng Lee and Zhongjian Gao. "Sign Language Recognition Using Two-Stream Convolutional Neural Networks with Wi-Fi Signals". In: *Applied Sciences* 10.24 (2020). ISSN: 2076-3417. DOI: 10.3390/app10249005. URL: https://www.mdpi.com/2076-3417/10/24/9005.

[17] Chenning Li, Manni Liu, and Zhichao Cao. "WiHF: Gesture and User Recognition with WiFi". In: *IEEE Transactions on Mobile Computing* PP (July 2020), pp. 1–1. DOI: 10.1109/TMC.2020.3009561.

[18] Mohan Liyanage et al. "Indoor people density sensing using Wi-Fi and channel state information". In: *Advances in Modelling and Analysis A* 61 (Mar. 2018), pp. 37–47. DOI: 10.18280/ama_b.610107.

[19] Masoud Mahdianpari et al. "Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery". In: *Remote Sensing* 10 (July 2018), p. 1119. DOI: 10.3390/rs10071119.

[20] Ronald Poppe. "Poppe, R.: A Survey on Vision-based Human Action Recognition. Image and Vision Computing 28(6), 976-990". In: *Image Vision Comput.* 28 (June 2010), pp. 976–990. DOI: 10.1016/j.imavis.2009.11.014.

[21] Malikeh Pour Ebrahim, Majid Sarvi, and Mehmet Yuce. "A Doppler Radar System for Sensing Physiological Parameters in Walking and Standing Positions". In: *Sensors* 17 (Mar. 2017), p. 485. DOI: 10.3390/s17030485.

[22] Jorge-L Reyes-Ortiz et al. "Transition-Aware Human Activity Recognition Using Smartphones". In: *Neurocomputing* 171 (Aug. 2015). DOI: 10.1016/j.neucom.2015.07.085.

[23] Jihoon Ryoo et al. "BARNET: Towards Activity Recognition Using Passive Backscattering Tag-to-Tag Network". In: June 2018, pp. 414–427. DOI: 10.1145/3210240.3210336.

[24] Souvik Sen et al. "You Are Facing the Mona Lisa: Spot Localization Using PHY Layer Information". In: *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*. MobiSys '12. Low Wood Bay, Lake District, UK: Association for Computing Machinery, 2012, 183–196. ISBN: 9781450313018. DOI: 10.1145/2307636.2307654. URL: https://doi.org/10.1145/2307636.2307654.

[25] F.B. Serkin and N.A. Vazhenin. "USRP platform for communication systems research". In: *2013 15th International Conference on Transparent Optical Networks (ICTON)*. 2013, pp. 1–4. DOI: 10.1109/ICTON.2013.6602738.

[26] Christian Szegedy et al. *Rethinking the Inception Architecture for Computer Vision*. 2015. arXiv: 1512.00567 [cs.CV].

[27] Sheng Tan et al. "Commodity WiFi Sensing in Ten Years: Status, Challenges, and Opportunities". In: *IEEE Internet of Things Journal* 9.18 (2022), pp. 17832–17843. DOI: 10.1109/JIOT.2022.3164569.

[28] Hao Wang et al. "Human respiration detection with commodity wifi devices: do user location and body orientation matter?" In: Sept. 2016, pp. 25–36. DOI: 10.1145/2971648.2971744.

[29] Wei Wang et al. "Understanding and Modeling of WiFi Signal Based Human Activity Recognition". In: Sept. 2015. DOI: 10.1145/2789168.2790093.

[30] Wei Wang et al. "Understanding and Modeling of WiFi Signal Based Human Activity Recognition". In: *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. MobiCom '15. Paris, France: Association for Computing Machinery, 2015, 65–76. ISBN: 9781450336192. DOI: 10.1145/2789168.2790093. URL: https://doi.org/10.1145/2789168.2790093.

[31] Yan Wang et al. "E-Eyes: Device-Free Location-Oriented Activity Identification Using Fine-Grained WiFi Signatures". In: *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*. MobiCom '14. Maui, Hawaii, USA: Association for Computing Machinery, 2014, 617–628. ISBN: 9781450327831. DOI: 10.1145/2639108.2639143. URL: https://doi.org/10.1145/2639108.2639143.

[32] Yuxi Wang, Kaishun Wu, and Lionel M. Ni. "WiFall: Device-Free Fall Detection by Wireless Networks". In: *IEEE Transactions on Mobile Computing* 16.2 (2017), pp. 581–594. DOI: 10.1109/TMC.2016.2557792.

[33] Xuangou Wu et al. "TW-See: Human Activity Recognition Through the Wall with Commodity Wi-Fi Devices". In: *IEEE Transactions on Vehicular Technology* PP (Oct. 2018), pp. 1–1. DOI: 10.1109/TVT.2018.2878754.

[34] Dan Yang et al. "Passive Infrared (PIR)-Based Indoor Position Tracking for Smart Homes Using Accessibility Maps and A-Star Algorithm". In: *Sensors* 18.2 (2018). ISSN: 1424-8220. DOI: 10.3390/s18020332. URL: https://www.mdpi.com/1424-8220/18/2/332.

[35] Jianfei Yang et al. *SenseFi: A Library and Benchmark on Deep-Learning-Empowered WiFi Human Sensing*. 2023. arXiv: 2207.07859 [cs.LG].

[36] Siamak Yousefi et al. "A Survey on Behavior Recognition Using WiFi Channel State Information". In: *IEEE Communications Magazine* 55.10 (2017), pp. 98–104. DOI: 10.1109/MCOM.2017.1700082.

[37] Siamak Yousefi et al. "A Survey on Behavior Recognition Using WiFi Channel State Information". In: *IEEE Communications Magazine* 55 (Oct. 2017), pp. 98–104. DOI: 10.1109/MCOM.2017.1700082.

[38] Daqing Zhang et al. *Practical Issues and Challenges in CSI-based Integrated Sensing and Communication*. Mar. 2022.

[39] Andrii Zhuravchak, Oleg Kapshii, and Evangelos Pournaras. "Human Activity Recognition based on Wi-Fi CSI Data -A Deep Neural Network Approach". In: *Procedia Computer Science* 198 (2022). 12th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 11th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare, pp. 59–66. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2021.12.211. URL: https://www.sciencedirect.com/science/article/pii/S1877050921024509.