Scaling Activity Recognition Using Channel State Information Through Convolutional Neural Networks and Transfer Learning

Jeroen Klein Brinke
j.kleinbrinke@utwente.nl
University of Twente
Enschede, Netherlands

Nirvana Meratnia
n.meratnia@utwente.nl
University of Twente
Enschede, Netherlands

ABSTRACT

Unobtrusive sensing is receiving much attention in recent years, as it is less obtrusive and more privacy-aware compared to other monitoring technologies. Human activity recognition is one of the fields in which unobtrusive sensing is heavily researched,, as this is especially important in health care. In this regard, investigating WiFi signals, and more specifically 802.11n channel state information, is one of the more prominent research fields. However, there is a challenge in scaling it up. Transfer learning is rarely applied, and when applied, it is done on filtered/modified data or extracted features. This paper focuses on two aspects. First, convolutional networks are used across multiple participants, days and activities and analysis is done based on these results. Secondly, it looks into the possibility of applying transfer learning based on raw channel state information over multiple participants and activities over multiple days. Results show channel state information is accurate for single participants (F_1-score of 0.90), but sensitive to different participants and fluctuating WiFi signals over days (F_1-score of 0.25-0.35). Furthermore, results show both clustering and transfer learning can be applied to increase the performance to 0.80 when using minimal resources and retraining.

CCS CONCEPTS

• **Computer systems organization** → Sensor networks; • **Human-centered computing** → Ubiquitous and mobile computing; • **Computing methodologies** → Cross-validation.

KEYWORDS

datasets, channel state information, human activity recognition, remote sensing, deep learning, convolutional neural networks, transfer learning

ACM Reference Format:
Jeroen Klein Brinke and Nirvana Meratnia. 2019. Scaling Activity Recognition Using Channel State Information Through Convolutional Neural Networks and Transfer Learning. In *First International Workshop on Challenges in Artificial Intelligence and MachineLearning (AIChallengeIoT'19), November 10–13, 2019, New York, NY, USA*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3363347.3363362

AIChallengeIoT'19, November 10–13, 2019, New York, NY, USA

ACM ISBN 978-1-4503-7013-4/19/11…$15.00
https://doi.org/10.1145/3363347.3363362

1 INTRODUCTION

Monitoring the world unobtrusively is increasingly desirable and possible, due to evolving technologies enabling smaller and smarter ways to observe the environment. It allows a safer, healthier and more comfortable life(style) as they allow continuous monitoring of activities, physiological and mental state. These pervasive systems are also often applied in other fields, such as monitoring animal behaviour or structural degradation.

Audiovisual techniques (based on cameras and microphones) and wireless sensor networks (in- and on-body sensors) are established solutions for the aforementioned continuous monitoring challenges. An advantage of audiovisual techniques over wireless sensor networks is that collected data is often easily interpretable by humans, especially images. This also causes the biggest downside: due to being easily interpretable images, there are privacy concerns. This is where the wireless sensor networks are superior: they add a layer of anonymity, as data is less easily interpretable and often requires (complex) algorithms or machine learning to interpret. While wireless sensor systems are less obtrusive when it comes to privacy, they are more physically obtrusive compared to audiovisual techniques. This is due to the required sensor being located in or on the body.

An interesting field that recently found more traction is remote sensing. Remote sensing measures the effects of an activity or event on the environment, rather than the activity or event itself (like wireless sensor networks). There is thus no need for in- nor on-body sensors. An increasingly popular technique for remote sensing is analyzing radio waves, especially channel state information [15] (CSI). It takes advantage of the multipath effect in wireless networks and gives insight in the propagation of packets between the transmitter and receiver over different subcarriers and antenna pairs.

Health care is an important societal pillar, struggling with increasing demands and fewer funds. Therefore, research often looks into applying remote sensing in this field. It has been shown that indoor localization [6, 16, 18], measuring physiological signals [11, 13, 17, 20], human identification [2, 12], and general human activity recognition/gesture detection [1, 7, 21, 22] are achievable by using CSI. The performances of such systems is comparable to the existing wearable wireless sensor systems. However, the user-friendliness and privacy-awareness of remote systems is higher due to providing a physically unobtrusive alternative to these wearables.

Machine learning or artificial intelligence is often used to interpret CSI as it is not easily interpretable by humans. Deep learning is applied more often due to increasingly more powerful hardware. Especially convolutional neural networks (CNNs) have proven to be effective in interpreting CSI [1, 6, 12, 16, 19, 22]. A CNN takes an

input for which the spatial and structural information is important (usually an image) and filters it through multiple convolutional or supportive layers (pooling or dropout) in order to extract features and ultimately classify or predict this information based on these features.

However, training these CNNs is usually a time- and energy-consuming task, as they are often contain millions of parameters to train. An alternative to retraining is using existing networks and applying transfer learning. With transfer learning most of the network is viewed as a black box and only the last few layers are retrained. A few deep convolutional neural networks exist that can be used for transfer learning, such as AlexNet [10] and VGG [14]. These are usually trained on millions of images and only require the last layer(s) to be retrained. Bu *et al.* [1] used the aforementioned VGG-16 and VGG-19 for feature extraction. Another way to implement is by data synthesis: creating custom data to train networks for new situations [21]. This often requires retraining the entire network, although it is also possible to only retrain the later layers.

1.1 Challenges and contribution

To the authors' best knowledge, little to no research has gone into the stability and scalability of raw CSI over the same and different participants over multiple days. While performances are usually shown for all activities, little to no differentiation is made between participants or days to investigate effects on CSI.

Most research in human activity recognition and CSI focuses on the classification, rather than cross-validation. It focuses often on different participants and environments, but the machine learning techniques are often retrained for each participant and/or environment and then compared. Leave-x(-subject)-out cross-validation is often not considered, let alone the stability of channel state information over different days and participants. The main challenge lies in scaling existing solutions up, without requiring the entire neural network to be retrained.

Filtering, extracting features and training entire CNNs is both time- and energy-consuming. It is important that as less filtering, feature extracting and (re)training is needed for a real-time, easy-to-use and scalable solution. Other research focuses mainly on either filtering or feature extraction, which in IoT applications cost considerable energy. Furthermore, pooling layers in convolutional neural networks provide some basic filtering (such as either smoothing out signals or focusing on the maxima). Being able to adapt CNNs quickly and without filtering or feature extraction is therefore desirable.

The contributions of this paper are to:

- Show the effects of different days and participants on the stability of channel state information by comparing F_1-scores
- Show how transfer learning can be applied to raw channel state information to improve performances with minimal time and resources

2 RELATED WORK

CNNs have been proven to be a useful tool when dealing with CSI and human activity recognition. Hsieh *et al.* [6] applied a multi-layer perceptron and a 1-dimensional CNN for localization by dividing a rectangular room into two dimensional blocks where each block is a class. Accuracy is reported to be high (90% and up) when using CSI and excluding RSSI. Wang *et al.* [16] applied a residual neural network (ResNet [5]) consisting of multiple ResNet layers to a dual-task CNN for both activity recognition and indoor localization for 6 activities at 16 indoor locations. An accuracy was reported of 88% and 95% was reported for activity recognition and indoor localization, respectively. Wang *et al.* [18] estimated angle-of-arrival information from the extracted phase of the CSI. These were converted to images and classified using a CNN for localization.

A combination between CNNs and CSI can be found in the papers of Tang *et al.* [21] and Bu *et al.* [1]. However, there is a main difference between both. Tang *et al.* trained a *roaming* model for new environments based on synthesised data. This synthesised data was generated from extracted features available from existing data. Statistical analysis is applied to evaluate the data and provide consistent data. Based on this analysis, a certain time for walking and a fixed number of activities need to be performed. The difference here is that this paper deals CNNs and a less controlled environment (unknown number of activities performed). Furthermore, it removes the use for statistical analysis or any form of filtering and feature extraction.

Bu *et al.* used an existing CNN, namely VGG-16/19 [14]. First, the gathered CSI was denoised and converted into grayscale images. These grayscale images are used to . This paper is different as it treats the CSI differently. Here, it is considered as structured and spatial dependent data, rather than a grayscale image. This reduces preprocessing time. Furthermore, this paper considered raw data, instead of denoised data, and it considers a multitude of participants over multiple days.

It is important to stress that this paper does not attempt to perform accurately, but rather to explore and evaluate other solutions requiring less filtering, feature extraction and retraining.

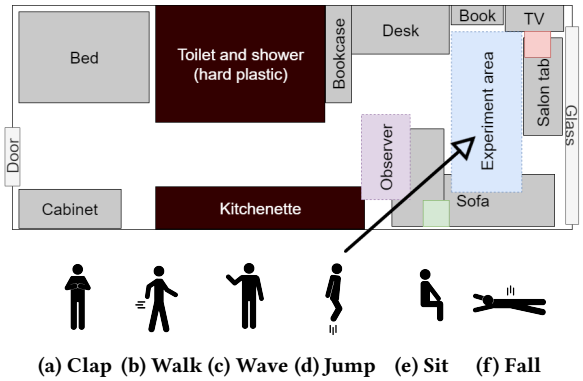


Figure 1: Layout of the experiment studio, including visualization of performed activities

3 DATA ACQUISITION

In order to produce a dataset that is reminiscent of day-to-day living, an actual living area was used (Figure 1). A mini-PC and an access point (TP-LINK AC1750) were used to collect CSI. The

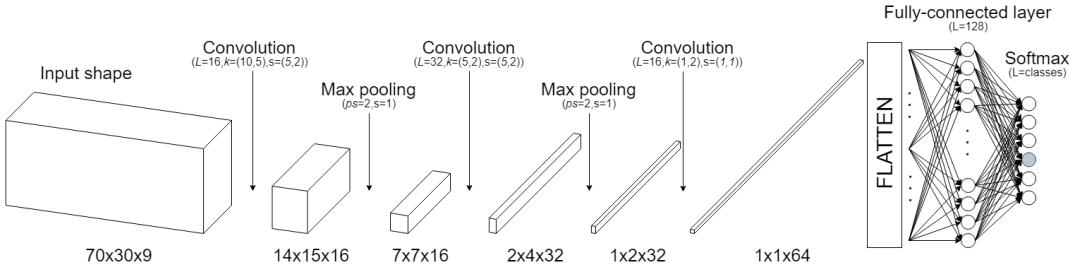


Figure 2: Configuration of the CNN, where L is the amount of layers, k kernel size, ps pool size and s strides

distance between the mini-PC and access poin (AP) was approximately 2.5 meters and the height difference approximately a meter. The mini-PC was equipped with the Intel Ultimate Wi-Fi Link 5300 NIC in order to run the Linux CSI Tool [4]. The mini-PC and AP were connected over a 802.11n 2.4 GHz network (*client mode*). The mini PC pinged the AP and the CSI of the response was recorded. Furthermore, packets were transmitted using 48 Mbps using 3x3 MIMO and 64QAM(1/2).

From the received packets, the CSI is extracted. This is shaped in a $N_t x N_r x N_s$ matrix, with N_t being the number of transmitters, N_r being the number of receivers and N_s being the number of subcarriers. For this research, the dimensions of the matrix were thus $3x3x30$. With a rate of 20 Hz and a sampling time of 5 seconds, this means that a total of 100 packets per trial per activity were recorded (not accounting for any packet loss or corrupted files).

Data was collected from 9 different participants over 6 multiple days. In total, 16 experiments were conducted: 9 experiments with 9 different participants over 3 days (denoted as $d \in \{1, 2, 3\}$) and 6 experiments with 2 different participants over 3 days (denoted as $d \in \{6, 7, 8\}$). These participants had strongly different characteristics, which cannot be shared due to privacy concerns. The 9 participants for $d \in \{1, 2, 3\}$ were asked to perform 6 activities freely, meaning they were allowed to change the way they performed these activities and move freely in the experiment area (Figure 1a-f). The 2 participants for $d \in \{6, 7, 8\}$ were asked to perform the 5 activities (jumping excluded due to health concerns) on a fixed location and in a similar pattern, by watching the recorded activities of $d = 6$ on a screen.

The dataset and metadata are available at the 4TU.ResearchData under the *CC BY-NC-SA* license with the DOI *10.4121/uuid:42bffa4c-113c-46eb-84a1-c87b6a31a99f* and contains 407978 data points spread over 6 days, 9 different participants and 6 activities [9].

4 METHODOLOGY

4.1 Preprocessing

No preprocessing in terms of filtering or feature extraction was done. This is due to the fact that a CNN was chosen and by performing preprocessing, the number of features it can learn from is potentially reduced. The focus of current research is mostly on preprocessing in order to decrease training time and/or increase accuracy, but this adds time preparing the data while pooling layers can account for basic filtering.

Shaping the data for the input layer ($l_{in} = (w, h.d)$) of the CNN is also part of preprocessing, with w being the width, h being the height and d being the depth. As mentioned before, the CSI is shaped as a $3x3x30$ matrix. It is expected that there are 100 packets per trace. However, this is not the case due to packet loss and corrupted files. Instead, an input length of 70 packets was considered, as more than 98% of all frames had more than a 100 packets. Therefore, the input of a single trial can be seen as a 4-dimensional matrix with size $70x3x3x30$.

As mentioned before, l_{in} requires a 3-dimensional matrix. Therefore, the final input matrix was shaped to be $70x30x9$. where 70 is the trace length, 30 the number of subcarriers and 9 the different antenna pairs. This essentially means that equally indexed subcarriers of different antenna pairs are adjacent to each other.

4.2 Convolutional neural network

Several deep CNNs are available (such as VGG [14] and AlexNet [10]) for transfer learning. These networks are often trained on a large and diverse dataset on powerful hardware for days or even weeks, making them a powerful tool to use in object detection or image processing. However, in this research the choice was made to design a custom CNN. The main reason is that the images these neural networks are trained on are significantly different compared to signals: these networks are usually trained on images containing objects or even art, but not so much on an abstract representation of signals. For example, an image can have any width or height, but will have an input depth of either 1 or 3 (grayscale and RGB), whereas the input layer in this research has a depth of 9. Another reasons is that it is desired to have a fully customizable CNN that can be trained from scratch multiple times with different parameters to compare, something not easily achievable by the aforementioned larger networks.

4.2.1 Configuration. The configuration of the used CNN can be found in Figure 2. The activation function used after pooling was the *tanh* activation and batch normalization was used between every pooling and convolution. For each convolution except the last, *same* padding was applied. The last convolutional layer was done with *valid* padding, in order to get a 1x1xl output. The number of epochs for training was 1000 for random initialization and 600 for transfer learning. A learning rate of $1 * 10^{-3}$ and batch size of 16 were found to be optimal (trade-off between accuracy, loss and training time) during tuning of the CNN. The optimization used was Adam [8].

4.2.2 Weight initialization and training. For the training, validation and testing set, a split of 65/20/15 was made, respectively. The convolutional network was trained for each possible combination of participant and days (including clustering) a total of 10 times using the Glorot uniform initializer [3]. This resulted in several thousand different sets of weights. Out of these, the optimal weights per classification or cross-validation task were used.

4.3 "Plain" classification and cross-validation

This means no adjustments are made to the weights after training: either the network is completely trained from scratch with randomly initialized weights, or an unmodified model is being used for cross-validation. Three categories are identified, for each of which the F_1-score is calculated based on the resulting confusion matrices. For all categories, all possible classifications and cross-validations were made. A distinction was made between $d \in \{1, 2, 3\}$ and $d \in \{6, 7, 8\}$, except for individual classifications (which included all participants over all days).

4.3.1 Individual classification. The different sets are all from the same distribution, namely a specific individual (no overlap between the sets). Individual classifications are used as ground truth, as these classifications compared to later classifications for cross-participant and days. An example of individual classification is training, validating and testing on the same participant (5-fold cross-validation, 10 repetition).

4.3.2 Cross-participant validation. Using the optimized weights found for each participant in the previous section, the activities of all other participant are classified. This is done within the same day and across different days, but a distinction is made between these two in Section 5.1.2. An example of cross-participant validation on the same day is training and validating on $p = 1$ and then trying to classify $p = 2$ for $d = 1$, whereas an example of cross-participant on a different day is training and testing on $p = 1$ on $d = 1$ and then testing on $p = 1$ on $d = 2$. Subsets of the data were used for testing (5-fold, 10 repetitions).

4.3.3 Clustered classification and cross-validation. Clustered participants are used for training. These newly trained models are used to classify and validate the activities of each involved participant. Clustering also comes with a distinction: excluding (*leave-one-subject-out cross-validation*) and including (5-fold cross-validation) training samples from the validated participant. From these classifications and (cross-)validations, the same accuracy metrics as before are analyzed. An example of included-clustered is training and testing on $p = (1, 2, 3)$ and classify $p = 1$ on $d = 1$, whereas an example of excluded-clustering is testing on $p = (1, 2)$ and classify $p = 3$ on $d = 1$.

4.4 Transfer learning

Transfer learning is often applied to decrease training time, while using fewer resources. It takes the weights of a trained CNN (with y hidden layers) which has already been trained on a comparable dataset and view the first n hidden layers as a black box (thus $n < y$). These n hidden layers are frozen and the weights of the final $y - n$ layers are trained using the new dataset (usually the fully-connected layers). The assumption here is that the first n hidden layers extract features (such as edges) which are comparable between the datasets. The $y - n$ final layers combine these basic features to detect more complex features.

For this research, the fully-trained networks were retrained with 20% of the data from the new participant, evenly spread across different activities to not create a bias. A look is taken at the effect of retraining different layers. The evaluation metric here are averaged the F_1-scores, together with the number of epochs and training time. Transfer learning is applied over all appropriate combinations, with a distinction for $d \in \{1, 2, 3\}$ and $d \in \{6, 7, 8\}$.

5 RESULTS AND DISCUSSION

Figure 3 shows the results of all classifications and cross-validations for different categories. This is elaborated in section 5.1. The number in brackets is the total number of classifications the boxplots are based on, excluding the number of repetitions (10). The top and bottom rows show information for $d \in \{1, 2, 3\}$ and $d \in \{6, 7, 8\}$, respectively.

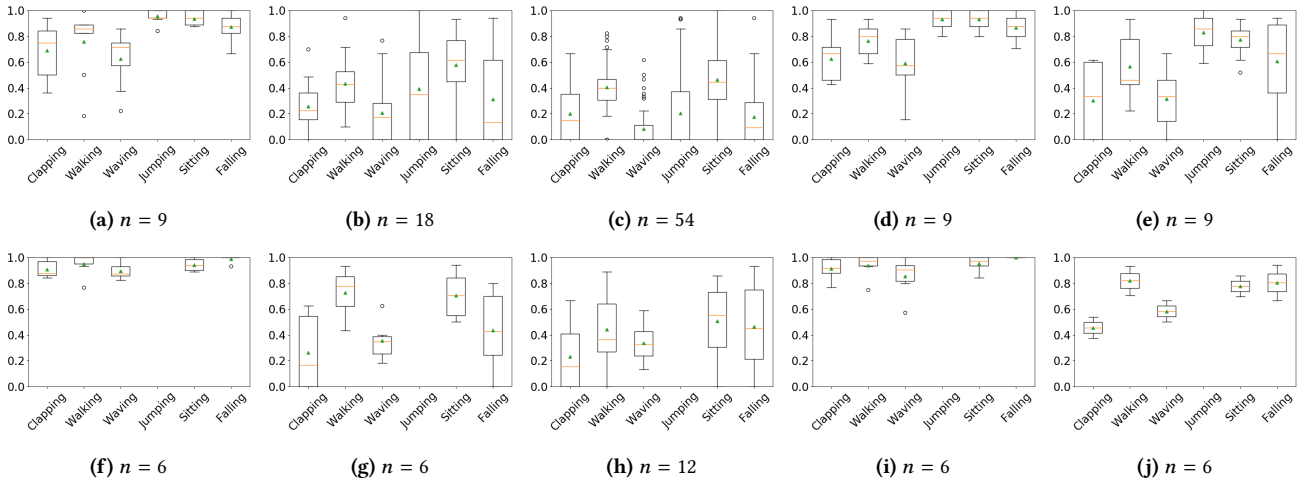
5.1 "Plain" classification and cross-validation

5.1.1 Individual classifications. For classification of individuals, the F_1-scores are high (Figure 3a,f). Higher averages are found for $d \in \{6, 7, 8\}$ (0.9303) compared to $d \in \{1, 2, 3\}$ (0.7917). This could be due to the later days having one less class to classify, but retraining the first three days with only 5 classes resulted in comparable results. More likely this is caused by participants performing the activities more similar to the other days. The lowest scores are recorded for $d = 2$, likely because participants were allowed the most freedom when it came to walking around and performing activities. For all participants, clapping and waving are the hardest to differentiate, likely as both are smaller, less distinctive movements with the forearms.

For cross-participant validation (Figure 3b,c,g,h), the F_1-scores drop to 0.2998 for $d \in \{1, 2, 3\}$ and to 0.4023 for $d \in \{6, 7, 8\}$ on average for all combinations. However, a differentiation can be made between cross-participant validation on the same day and across different days.

5.1.2 Cross-participants validation. Training across different participants on the same day lowered the F_1-score to an average of 0.3483 and 0.4879 for $d \in \{1, 2, 3\}$ (Figure 3b) and $d \in \{6, 7, 8\}$ (Figure 3g), respectively. It can be seen that jumping and falling are very challenging classes to classify, with F_1-scores ranging the entire $0 - 100\%$ range. On the other hand, walking, clapping and sitting are more stable and thus smaller boxplots. Walking and sitting score highly compared to the other classes ($0.4 - 0.6$ on average), which is likely due to the similar motion regardless of the day and participant: sitting is completely passive and walking is done in a same manner for all humans. Clapping and waving score lower compared to the other classes ($0.2 - 0.3$ on average). This is likely as they are minor movements and very similar to each other, especially between different participants.

For cross-participants validation on different days, we can see the same 10% difference between $d \in \{1, 2, 3\}$ (Figure 3c) and $d \in \{6, 7, 8\}$ (Figure 3h): 0.2512 and 0.3832, respectively. Overall, the same implications can be found here as for classification on the



(a) $n = 9$ (b) $n = 18$ (c) $n = 54$ (d) $n = 9$ (e) $n = 9$

(f) $n = 6$ (g) $n = 6$ (h) $n = 12$ (i) $n = 6$ (j) $n = 6$

Figure 3: Boxplot of F_1-scores over 10 trials for individual (a,f), cross-participant on same day (b,g), cross-participants over different days (c,h), classified clustering over all participants on a day (d,i) for $d \in \{1, 2, 3\}$ (top) and for $d \in \{6, 7, 8\}$ (bottom); (e) shows excluded-clustering for same days for $d \in \{1, 2, 3\}$ and (j) shows training for participants on $d \in \{6, 7\}$ and cross-validating $d = 7, 8$ (no overlap). n denotes the number of classifications.

same day: clapping and waving are the hardest to classify (lowest F_1-scores), whereas walking and sitting have the highest F_1-scores.

In both cases, there is a difference of 10% between cross-validation of participants on the same and different days, likely caused by i) a combination of training for one less class and ii) it being the same participants performing the same activities. While preliminary research showed the same performance for 5 and 6 activities, Figure 3b indicates that jumping is a very challenging activity to classify, compared to the other activities. It is likely that this contributes to the lower accuracy among the two.

5.1.3 Clustered classification and cross-validation. A differentiation can be made for two cases in clustered participants: classification and cross-validation. Figure 3d,i show classification (thus including the tested participant) for $d \in \{1, 2, 3\}$ (d) and $d \in \{6, 7, 8\}$ (i). An increase can be seen compared to cross-participant validation, but the average performance (0.7817) is still lower than individual classification. This is likely due to the additional participants added to training acting as adding noise for any given classification. Once again, it can be seen that clapping and waving are the classes with the lowest F_1-scores. Interestingly, jumping and falling both perform well. This is likely due to every person falling and jumping differently: it is hard to differentiate these similar activities from only one given participant to another, but including training samples of each participant makes this doable. As expected, $d \in \{6, 7, 8\}$ once again performs better (0.9180), likely due to the participants being the same.

Figure 3e shows F_1-scores for cross-validating participants on the same day with excluded-clustering ($d \in \{1, 2, 3\}$). The average F_1-score is 0.5901, which is lower than the classified clustering (Figure 3d), which is to be expected as the participant is excluded. However, the performance is better than training on a single participant on the same day and classifying activities of other participants (Figure 3b). This implies that when training with more data from different participants, performances can be increased of participants excluded from the training set. However, it is likely that the opposite holds true, as well: adding more participants could also lower the accuracy eventually, due to there being too many different ways of performing an activity.

5.1.4 Same participant over different days. Figure 3j shows the F_1-scores of training for $p = 1, 2$ on $d = 6, 7$ and then classifying the activities of this same participant on $d = 8$. The average F_1-score over all activities is 0.7143, which is lower than the clustered data for the same day (Figure 3i), but higher than cross-participant on both the same day Figure 3g and different days Figure 3h.

Also, the boxplots are smaller compared to the aforementioned situations, meaning that the F_1-scores are clustered more. This implies that a correlation can be seen across the same participants over different days, which is a lesser case for different participants over different days Figure 3e. The lower F_1-score compared to individual classification (Figure 3f) shows there are significant fluctuations on the CSI over time.

5.2 Transfer learning

Figure 4a-c shows the effect of retraining different layers of the CNN for different classification tasks (cross-participant on the same day, cross-participant over different days and same participant over different days), whereas 4d shows the average time needed to retrain layers. Note that what is shown is not the accuracy, but rather the F_1-scores after training for a certain number of epochs.

First, the retraining of different amount of layers is considered. It is important to note that layers are counted from the back and only the fully-connected and convolutional layers are included. So, *1 Conv + FC* means the fully-connected and last convolutional layer are retrained. As this network only has 3 convolutional layers, *3 Conv + FC* means the entire network was retrained. *Regular* means

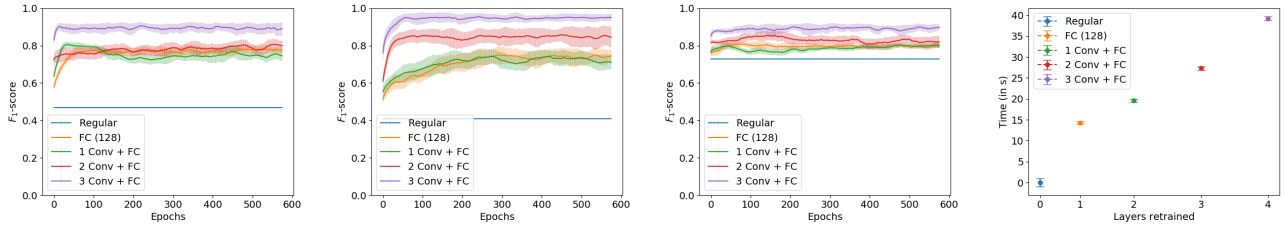


Figure 4: Graph showing effect of transfer learning on F_1-scores over 600 epochs for (a) cross-participant, same day; (b) cross-participant, different day; (c) same participant, different days; (d) shows average calculation time for retraining.

no layers were retrained and regular classification was performed, which explains the straight line.

Understandably, retraining the entire network (*3 Conv + FC*) achieves the highest performance. Interesting to note is that in some cases the performance is higher when training on a pretrained network than one initialized with random weights. This is even achieved after only 200 epochs, whereas a randomly initialized network requires at least 600 epochs to stabilize. This is explainable by the fact that the weights of the pretrained network are already based on the same type of structural data, which in this case is a 3-dimensional matrix containing raw CSI. This does imply that in order to achieve better performances, new networks should be trained on an existing network for better results.

However, it is more interesting to look at the partially retrained networks. In all cases, retraining the last convolutional layer together with the fully-connected layer (*1 Conv + FC*) results in the lowest increase in performance. This is likely the case as changing only the last convolutional layer results in confusion: the first two layers extract features, which suddenly match less with the newly trained third layer. The performance is still improved, likely as the final layer is still more fitted towards the new participant. In most cases, retraining two convolutional layers (*2 Conv + FC*) results in similar performance as retraining only one. Different participants across different days being the exception, as this one clearly outperforms retraining the fully connected layer (with and without the last convolutional layer).

The most interesting case to consider is retraining just the fully-connected layer: it can be seen that this increases the accuracy significantly when using raw CSI. Especially in the case of cross-participants on both the same and different days, for which the improvement is 25−30% when considering only the fully-connected layer (*FC(128)*) is retrained - which is in most cases comparable to both retraining the last two convolutional layers with the fully-connected layer. However, different participants on different days hold a lower performance. This is likely due to a combination of different participants and a different day both affecting the CSI significantly.

In all cases, the performance increase caps out after approximately 200 epochs (for two cases even after 100). This means that a higher accuracy can be achieved using up to three times less epochs. The time denoted in 4d is based on 600 epochs and would likely be 1.5 to 2 times less for the optimum of 100-200 epochs.

6 CONCLUSION AND FUTURE WORK

First, this paper shows the stability of raw CSI over multiple participants and days. Several different types of activities were performed, some closely related and some completely different. It is shown that the classification difficulty of different activities remain consistent over time: activities that are harder to distinguish remain so regardless of the participant and different days. This implies that it is not based on the participant, but rather on the activity. The overall accuracy decreases over different participants and days, showing that raw CSI is not transferable due to different characteristics of participants and fluctuations in the WiFi signal over days.

Secondly, it is shown working with raw CSI and CNNs for human activity recognition over different participants and days seems promising when applying transfer learning. For individual classification without transfer learning the F_1-scores are on average 85-90%. This can be increased to 90-95% when applying transfer learning. However, the biggest increase comes when applying transfer learning to different participants and same participants over a time period. When applying no transfer learning, F_1-scores were as low as 25-30%. However, with transfer learning this was improved to 80-85% by retraining only the last fully-connected layer.

This research also shows that it is beneficial to either retrain the entire network or only the last fully-connected layer. This is important, as it allows for solutions which do not require denoising, specific feature extraction, or data synthesis. Combining the fact that no preprocessing is needed and only the fully-connected layer needs to be retrained with minimum effort, this allows smaller and more energy-efficient processing of raw CSI on IoT devices.

However, there are some limitations and suggestions for future research. Future research should focus on tests with more fully-connected layers, as these are the more flexible layers. Convolutional layers tend to find the important features, whereas fully-connected layers combine these to the output. More fully-connected layers could potentially result in more flexibility and a higher increase in performance.

A limitation to CNNs is that their input size is fixed, so only frames of a specific length can be analysed. This may not be trouble some for solutions which record specific data, but it is potentially limiting to solutions that only analyze registered events (e.g. in edge intelligence).

REFERENCES

[1] Q. Bu, G. Yang, J. Feng, and X. Ming. 2018. Wi-Fi Based Gesture Recognition Using Deep Transfer Learning. In *2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. 590–595.

[2] J. Choi, W. Lee, J. Lee, J. Lee, and S. Kim. 2017. Deep Learning Based NLOS Identification with Commodity WLAN Devices. *IEEE Transactions on Vehicular Technology* (2017). www.scopus.com Article in Press.

[3] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks.. In *AISTATS (JMLR Proceedings)*, Yee Whye Teh and D. Mike Titterington (Eds.), Vol. 9. JMLR.org, 249–256. http://dblp.uni-trier.de/db/journals/jmlr/jmlrp9.html#GlorotB10

[4] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2011. Tool Release: Gathering 802.11n Traces with Channel State Information. *Computer Communication Review* 41 (01 2011), 53. https://doi.org/10.1145/1925861.1925870

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). arXiv:1512.03385 http://arxiv.org/abs/1512.03385

[6] C.-H. Hsieh, J.-Y. Chen, and B.-H. Nien. 2019. Deep Learning-Based Indoor Localization Using Received Signal Strength and Channel State Information. *IEEE Access* 7 (2019), 33256–33267. https://doi.org/10.1109/ACCESS.2019.2903487

[7] Wenjun Jiang, Dimitrios Koutsonikolas, Wenyao Xu, Lu Su, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, and Xin Ma. 2018. Towards Environment Independent Device Free Human Activity Recognition. 289–304. https://doi.org/10.1145/3241539.3241548

[8] Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* (12 2014).

[9] J. Klein Brinke and N. Meratnia. 2019. Dataset: Channel state information for different activities, participants and days. In *DATA'19 '19: Proceedings of the Second Workshop on Data Acquisition To Analysis, November 10, 2019, New York, NY, USA*, ACM (Ed.). https://doi.org/10.1145/3359427.3361913

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[11] J. Liu, Y. Chen, Y. Wang, X. Chen, J. Cheng, and J. Yang. 2018. Monitoring Vital Signs and Postures During Sleep Using WiFi Signals. *IEEE Internet of Things Journal* 5, 3 (June 2018), 2071–2084. https://doi.org/10.1109/JIOT.2018.2822818

[12] A. Pokkunuru, K. Jakkala, A. Bhuyan, P. Wang, and Z. Sun. 2018. Neuralwave: Gait-based user identification through commodity WiFi and deep learning. *Proceedings: IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society* (2018), 758–765. https://doi.org/10.1109/IECON.2018.8591820

[13] Jiacheng Shang and Jie Wu. 2016. Fine-grained vital signs estimation using commercial wi-fi devices. 30–32. https://doi.org/10.1145/2987354.2987360

[14] K. Simonyan and A. Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[15] C. Wang, S. Chen, Y. Yang, F. Hu, F. Liu, and J. Wu. 2018. Literature review on wireless sensing-Wi-Fi signal-based recognition of human activities. *Tsinghua Science and Technology* 23, 2 (2018), 203–222. www.scopus.com

[16] F. Wang, J. Feng, Y. Zhao, X. Zhang, S. Zhang, and J. Han. 2019. Joint activity recognition and indoor localization with WiFi fingerprints. *IEEE Access* 7 (2019), 80058–80068. https://doi.org/10.1109/ACCESS.2019.2923743

[17] J. Wang, X. Zhang, Q. Gao, H. Yue, and H. Wang. 2017. Device-Free Wireless Localization and Activity Recognition: A Deep Learning Approach. *IEEE Transactions on Vehicular Technology* 66, 7 (2017), 6258–6267. www.scopus.com

[18] X. Wang, X. Wang, and S. Mao. 2018. Deep Convolutional Neural Networks for Indoor Localization with CSI Images. *IEEE Transactions on Network Science and Engineering* (2018). https://doi.org/10.1109/TNSE.2018.2871165

[19] X. Wang, X. Wang, and S. Mao. 2018. Deep Convolutional Neural Networks for Indoor Localization with CSI Images. *IEEE Transactions on Network Science and Engineering* (2018). www.scopus.com Article in Press.

[20] Jin Zhang, Weitao Xu, Wen Hu, and Salil Kanhere. 2018. WiCare: Towards In-Situ Breath Monitoring. https://doi.org/10.4108/eai.7-11-2017.2274069

[21] Tang Z. Li M. Fang D. Nurmi P. Wang Z. Zhang, J. 2018. CrossSense: Towards cross-site and large-scale WiFi sensing. *MobiCom 2018*, 305–320.

[22] Q. Zhou, J. Xing, W. Chen, X. Zhang, and Q. Yang. 2018. From signal to image: Enabling fine-grained gesture recognition with commercial wi-fi devices. *Sensors (Switzerland)* 18, 9 (2018). https://doi.org/10.3390/s18093142