Exploring Human Activity Recognition with Convolutional Neural Network and Distilled Networks

Justin Branstetter **Zixi Liu** **Vasyl Shcherbatyuk**

Thayer School of Engineering, Dartmouth College

{justin.l.branstetter.th, zixi.liu.th, vasyl.shcherbatyuk.th}@dartmouth.edu

Abstract

Human Activity Recognition (HAR) has applications in various domains such as medicine, personal health, sports, security, and virtual reality. With the ubiquitous presence of smartphones equipped with sensors, the inertial signal data are invaluable for capturing contextual information about human activities. In this work, we investigated the application of Convolutional Neural Network (CNNs) to a HAR dataset comprising 30 individuals performing daily activities while carrying smartphones on their waist. Our primary focus lies in leveraging CNNs for sensor-based HAR tasks and addressing the practical challenges associated with deploying CNN models on edge devices. Through comprehensive analysis, we're able to reach a high accuracy of 95.9% on the HAR tasks with our CNN model. Model compression techniques such as knowledge distillation allowed us to significantly reduce model size, memory footprint, and inference time in a real-world setting.

1 Introduction

Human Activity Recognition (HAR) has many applications in areas such as medicine, personal health, sports, security, and virtual reality. With smartphones being ubiquitous and equipped with a variety of sensors, they provide valuable contextual information about human activity. In this work, we focused on utilizing Convolutional Neural Networks (CNNs) for sensor-based HAR tasks. Our motivation stems from several factors: CNNs' capability to automatically learn and extract features from raw sensor data, their proficiency in identifying local connectivity patterns, and their effectiveness in processing multichannel data from various sensors. This approach eliminates the need for manual feature engineering, which can be time-consuming and domain-specific.

Our study addresses the practical challenges of deploying CNN models on edge devices such as smartphones. These challenges include resource constraints, the need for real-time processing, and ensuring robust performance in diverse environments. We tackle these concerns by examining model size and complexity, computation power consumption, memory footprint, and inference latency.

Through rigorous preprocessing and a detailed analysis of the dataset, we aim to create an efficient and accurate model for human activity recognition. By leveraging advanced techniques like model compression and knowledge distillation, we were able to develop models that are not only high-performing but also optimized for deployment on resource-constrained devices.

2 Related Work

Extensive research has been conducted on sensor-based Human Activity Recognition. The dataset we focused on in this work is the "Human Activity Recognition Using Smartphones" archived on the UCI website. This Human Action Recognition database is built from experiments with a group of 30 volunteers. Each person performed six activities (Walking, Walking Upstairs, Walking Downstairs, Sitting, Standing, Laying) wearing a smartphone (Samsung Galaxy S II) on the waist. The sensor signals (accelerometer and gyroscope) were then pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap (128 readings/window) as illustrated in Fig 1 [4].

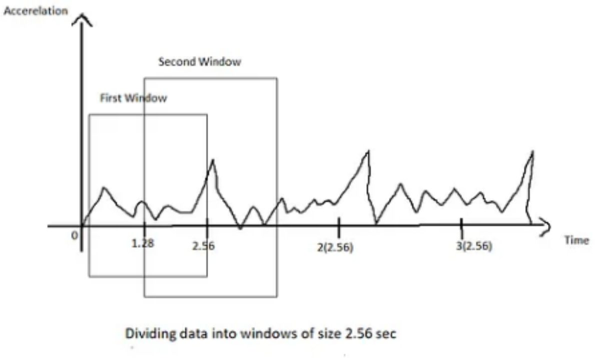

Fig.1. Illustration of Sensor Signal Processing

In early studies, traditional algorithms such as SVM were applied coupled with handcrafted features by domain experts and reached a high accuracy of 96% on this HAR dataset [2]. However, this method relied heavily on manual feature engineering, which requires domain expertise and the process is time-consuming. We thought this paper is valuable as a source of understanding the data collection for the HAR dataset and served as our baseline model to compare model performance.

There are some notable works that explored deep learning models for HAR. For example, Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) models are used for HAR and reached an high accuracy of 97.89% [1]. Both CNNs and LSTMs are capable of automatically feature learning from raw sensor data. CNNs excel at recognizing local spatial relationships, whereas LSTMs can effectively model long-range dependencies in sequential data. Researchers have combined them in a hybrid architectures to leverage their complementary strengths.

Existing studies also acknowledged the practical challenges in deploying deep learning models to wearable devices. For example, a CNN model was pruned and quantized with relatively low memory and computational overhead in [6]. Specifically, the quantized and pruned CNN was able to reach a comparable accuracy of 95.89% on the HAR dataset but with a model size of 30.76 KB, which is almost one fifth the size of their baseline CNN model.

In our work, we strove to extend our research to more advanced model compression techniques such as knowledge distillation on deep neural networks and investigate not only model size, but also memory footprint and inference speed to further improve on the application of deep learning on HAR.

3 Methodology

3.1 Problem Formulation

In this section, we first propose a baseline Convolutional Neural Network (CNN) for sensor-based human activity recognition (HAR) tasks. The motivation was explored from multiple angles:

1. **Feature Extraction**: CNNs are highly effective in automatically learning and extracting features from raw sensor data. This eliminates the need for manual feature engineering, which can be time-consuming and domain-specific.
2. **Local Connectivity Patterns**: CNNs excels at learning local connectivity patterns, focusing on small regions of data at a time. This is especially useful for HAR tasks where local patterns such as sequences of movements are crucial for identifying activities.

3. **Multichannel Data Processing**: The sensor signals data from UCI comprise 9 dimensions from accelerometers and gyroscopes. CNNs can handle multichannel inputs effectively, learning interdependencies between different sensors.

We formulated the sensor-based human activity recognition task as follows: Let X_i =

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ \vdots & & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

where X_i is an observation from the experiment sensor dataset. We denote m = {1,2,...,128} to represent the 128 readings from signal data and n = {1,2,...,9} to stand for the 9 dimensions of inertial signals. For each observation X_i in the training data, our goal is to train a classifier to predict its corresponding activity label Y_i. Then we validate the model's performance with unseen observations in the test set to assess the generalization of the model.

While extensive research has been conducted in leveraging CNNs for HAR (1), practical challenges remain when we aim to deploy CNN models on edge devices such as smartphones. These challenges stem from the resource constraints of edge devices, the need for real-time processing, and the requirement for robust and reliable performance in diverse environments. Our study addresses these concerns from the aspects of (i) model size and complexity, (ii) computation power consumption and memory footprint, and (iii) inference latency.

3.2 Convolutional Neural Networks

The application of Convolutional Neural Networks (CNNs) for sensor-based human action recognition has developed significantly over time. Sensor data, typically in the form of time-series, was reshaped into formats suitable for CNNs, such as 2D representations or multi-channel sequences. With the growing interest in deploying models on edge devices (e.g., wearables, smartphones), researchers developed lightweight CNN architectures [1] and model compression techniques like pruning and quantization were applied to reduce model size and computational requirements [6].

In this work, we designed the convolutional layers to capture the spatial and temporal patterns from the inertial signal data representing human movements. By applying filters to extract local features from input sensor data, we gradually increased the number of filters in subsequent layers (e.g. 64, 128, 256, 512) to capture more complex patterns. After each convolutional layer, we also leveraged batch normalizations to enable stable gradients and prevent vanishing or exploding gradients. ReLU activation function is then applied to learn complex temporal patterns more efficiently with its non-linear nature. The sparse activation from ReLU also helps in capturing the salient aspects of human actions while ignoring irrelevant or noisy information. To help the model generalize better, we applied Adaptive Pooling that connects to a fully connected layer in later stage so that the model can adapt to input sensor data of varying lengths in a real-world setting. Figure 2 illustrates the CNN model architecture we designed.

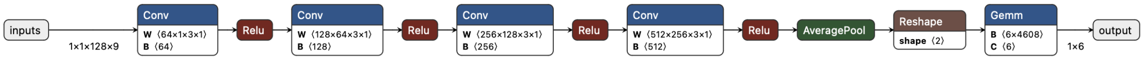

Fig.2. Illustration of CNN Model Architecture

3.3 Model Compression

Smartphones have limited computational resources, including CPU, GPU, and memory. Recently, lightweight kilobyte-sized machine learning models that can run directly on smartphones has been a popular research area. In this work, we explored multiple advanced techniques for model compression, including knowledge distillation and quantization to combat the practical challenges in deploying CNNs to edge devices.

Knowledge Distillation. For model compression, we transfer the knowledge from the teacher model, a complex CNN architecture, to a smaller, shallower student model. The shallow neural network use fewer layers and parameters than the teacher model, making it computationally more efficient and faster to train and run, which is beneficial for deployment on resource-constrained devices. In our work, we designed a simple feedforward neural network and trained the shallow neural network with Knowledge Distillation (KD) loss, which is a combination of the Kullback-Leibler(KL)-Divergence loss and the regular Cross Entropy loss.

Specifically, we leveraged the KD loss definition from [6] as follows:

$L_{KD} = \alpha T^2 * KL(Q_S, Q_T) + (1 - \alpha) * CrossEntropy(Q_S, y_{true})$.

where Q_S and Q_T are the softened probabilities by applying the softmax function with temperature T from the student and teacher model respectively. T is the temperature hyperparameter that controls the smoothness of the probability distribution over classes and α is another hyperparameter that tunes the weighted average between two components of the loss.

As a result, we're able to train a shallower student model with significantly smaller model size, faster inference speed, and less memory footprint.

Quantization Quantization is another popular approach to decrease model size and improve computational efficiency. For example, our CNN model was trained suing 32-bit floating-point precision (FP32). In our experiment, we reduced this precision to 8-bit integer (INT8) post-training. This reduction decreases memory footprint and accelerates inference by leveraging hardware accelerators that are optimized for lower-precision arithmetic.

4 Experiments

4.1 Exploratory Data Analysis

We first checked the class imbalance for both training data and test data. This is because imbalanced datasets may lead to biased decision boundaries. This can impact the interpretability of our model's predictions, especially in HAR applications where all activities are equally important. As shown in the Figure 3.1 below, we have almost same number of reading from different volunteers in the training set. This means there are no significant differences in reading contribution among the volunteers.

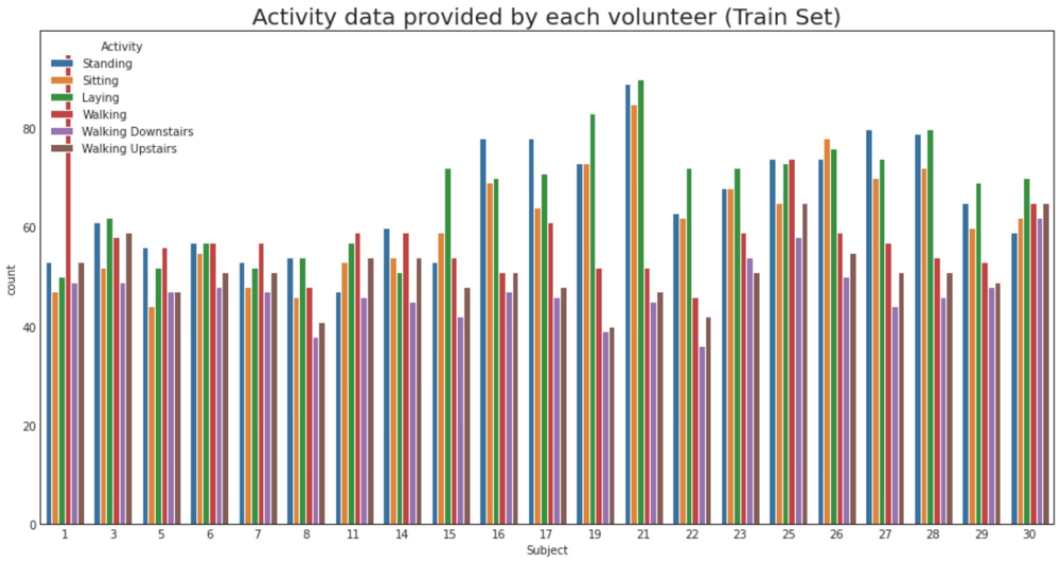

Fig.3.1. Activity Data Provided by Each Volunteer (Training Set)

Similarly in the test set, we have almost same number of reading from each different volunteer.

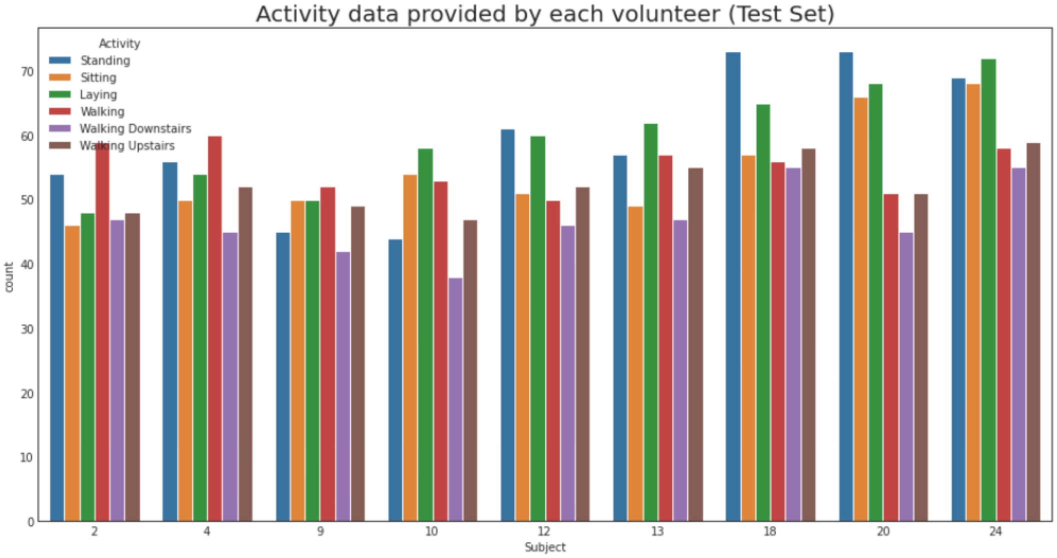

Fig.3.2. Activity Data Provided by Each Volunteer (Test Set)

We also wanted to check whether the raw sensor data makes sense for human movements in 3-D projection. So we sample some human activities from the inertial signals and plot the body acceleration along X, Y, Z direction.

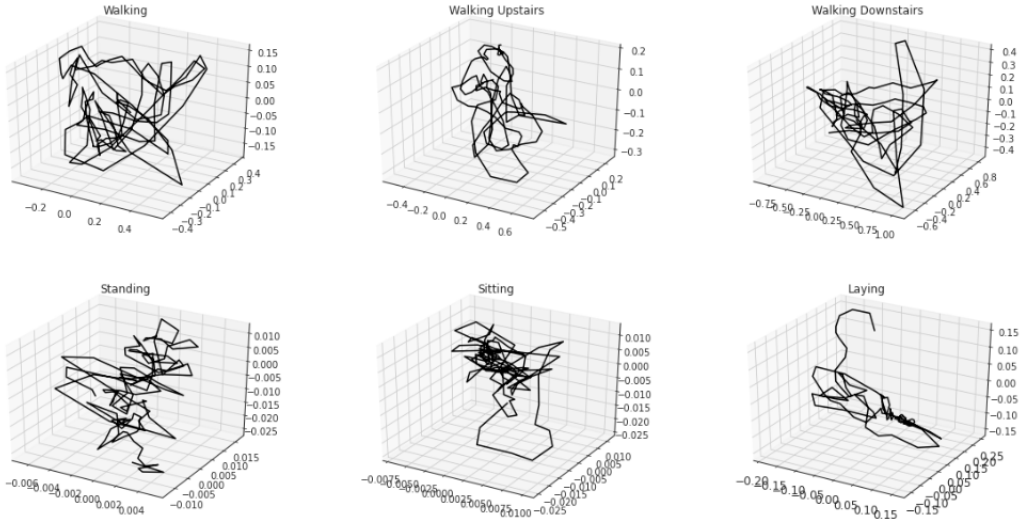

Fig.3. 3-D Projection of Human Movements from Inertial Signals Data

4.2 Data Preprocessing

Data preprocessing is a critical step to ensure the quality and integrity of the data before modeling. In our work, we mainly focused on following steps for data cleaning:

- Removing Duplicates: Duplicate records need to be identified and removed to prevent skewing the data. By running through the duplicate check, we concluded that there are neither duplicated features nor observations in our dataset.
- Filtering Outliers: We tried to identify outliers for each feature in our dataset, which are data points that deviate at least 3 standard deviations away from the mean. But it's important to note that we believe outliers can be legitimate data points in our task because variation

exists in human movements among different volunteers. We were able to identify outliers in all 9 features and plot feature distributions, but we determined to not remove or impute outliers at this stage.

- Handling Missing Data: Usually, depending on the missing data mechanism, we take different approaches to handle the missingness. However, there's no missingness detected in the underlying data.
- Standardizing Data: We have checked that the feature distribution for all 9 features are approximately normal distribution so we did not perform any transformations.

As a result, we're able to conclude that after running through the Data Cleaning pipeline for both training and test set, the datasets we curated are ready for training and evaluation.

4.3 Experiment Setting

Train/Test Data Split: In this work, train and test data are split in the original UCI-HAR dataset to ensure reproducibility of the experiment results. Specifically, 7352 observations of activities from 21 subjects and their corresponding inertial signals are contained in the training data; 2947 observations from the other 9 subjects are contained in the test set to evaluate the model's performance. This evaluation set is crucial to assess how well the model generalizes to unseen data. We then report the accuracy metrics and confusion metrics to conclude experiment results.

Hyperparameter Tuning: Hyperparameter tuning is a crucial step in training CNNs to achieve optimal performance. After a few rounds of hyperparameter searching, we decided to use a reasonable learning rate of 0.0005 and leveraged a learning rate scheduler to adjust the learning rate during training. This is because learning rate schedulers helps in achieving faster convergence initially and more precise convergence in the later stages. We also used the Adam algorithm for optimization with a relatively large weight decay of 0.001 that set as a form of regularization to penalize large weights in the model. This method is especially useful to reduce the risk of overfitting. We then trained the CNN model with 20 epochs and stopped when the validation set performance stagnated as shown in Figure 4.

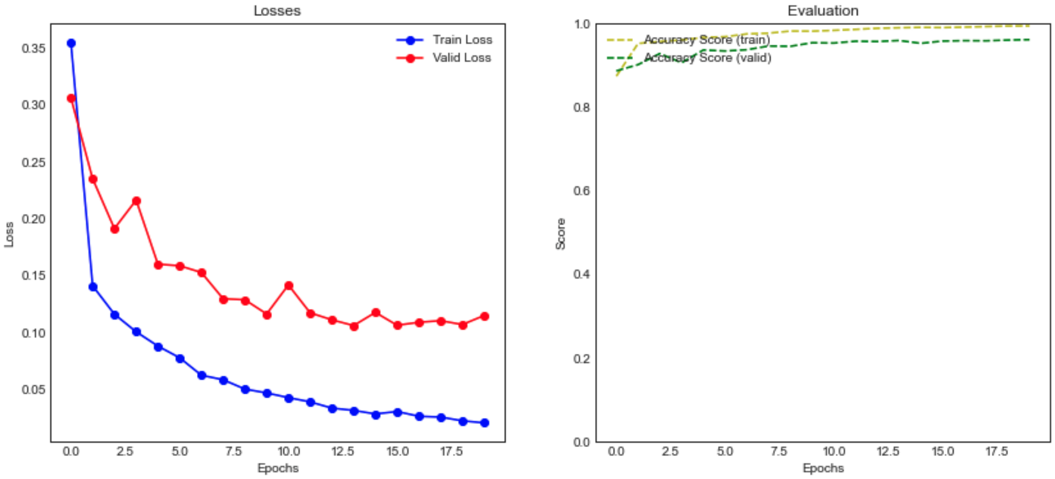

Fig.4. Illustration of Training and Validation Loss and Accuracy

4.4 Results and Discussion

This section discusses the results of all experiments and compares them to baselines (summarized in Table 1) established in prior works. The CNN model used in this study produced a high accuracy of 95.9% on human activity recognition tasks, which is comparable to the accuracy of 96% produced from the original paper [2]. However, the CNN model was trained on the raw sensor data and did not rely on handcrafted features as in the SVM model. In our own SVM model, without handcrafted features using grid search for hyperparameter tuning we were able to achieve 90% F1 Macro score.

It's worth noting that we also trained the CNN model with quantization and knowledge distillation. We observed that when we ran model inference for CNN model with quantization, the model performance did not degrade too much from the complete CNN model, reach an high accuracy of 95.8%. However, for the student model trained with knowledge distillation, the model accuracy degraded to 88.9%. After we further applied quantization to the student model, its accuracy remains at 88.9%, which could be due to following reasons: (i) Mismatched Model Architecture: The architecture of the student model may be too shallow and simple for distillation from the teacher model, so it struggled to capture the knowledge transferred during distillation. (ii) Inappropriate Hyperparameters: Hyperparameters such as the temperature parameter used in distillation might not be tuned properly, which could lead to poor convergence during training and degraded performance. But the student model is still worth considering when we further explore the practical challenges when deploying the models to edge.

Table 1. A Comparison of Model Accuracy and F1 (Macro) on the Evaluation Data

Model Description	Eval Accuracy	Eval F1 (Macro)
SVM Model from [2]	96%	
SVM Model w/o Feature Eng	90%	90%
CNN Model	95.9%	95.9%
CNN Model w/ Quantization	95.8%	95.9%
KD Model	88.9%	88.8%
KD Model w/ Quantization	88.9%	88.8%

In addition, we systematically evaluated several key factors for edge AI applications such as the model size on disk, memory footprint during inference and the inference speed in a real-world setting. As shown in Table 2, we identifed that the student model was able to significantly reduce the model size, memory footprint and inference time from the CNN model. This demonstrates a trade-off between these practical concerns and model performance, but there's still areas for improvement for the knowledge distillation model as mentioned above.

Table 2. A Comparison of Model Size, Memory Footprint and Inference Speed

Model Description	Model Size	Memory Footprint	Inference Speed
CNN Model	2.20 Mb	219.40 Mb	693.5 ms
CNN Model w/ Quantization	2.12 Mb	218.84 Mb	449.7 ms
KD Model	0.66 Mb	146.97 Mb	396.3 ms
KD Model w/ Quantization	0.17 Mb	2.53 Mb	352.2 ms

4.5 Model Diagonostics

We further inspected the confusion matrix for the CNN model and Knowledge Distilled model to understand the model performance across different classes. For example, we observed that out of the six activities, the CNN model has the highest accuracy of 100% on "Laying" and lowest accuracy of 87% on "Sitting". Specifically, 13% of the Sitting movements are misclassified as Standing and Walking Upstairs. This result is quite consistent with the original study from [2] where Sitting activity also had lowest recall equal to 88%. In the Knowledge Distilled model, we also observed the same trend where the model accuracy on the Sitting activity degraded to 75%. For all other activities, this model still reached a relatively high accuracy at around 90%.

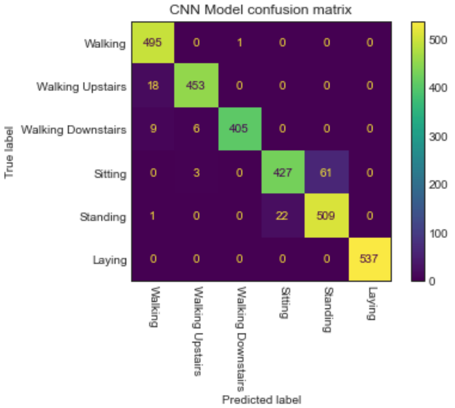

Fig. 5.1. CNN Model Confusion Matrix

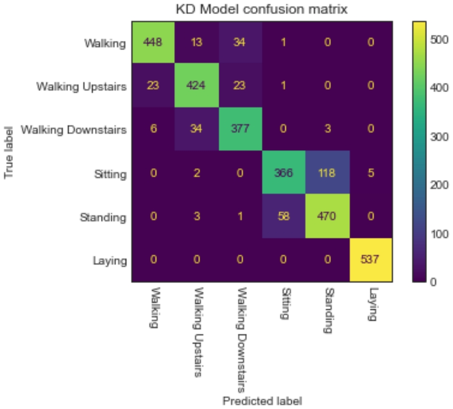

Fig. 5.2. KD Model Confusion Matrix

5 Conclusion

In conclusion, our work demonstrated the effectiveness of CNNs in sensor-based human activity recognition tasks. Our model achieved high accuracy rates of 95.9% across various activities without manual feature engineering, underscoring the potential of deep learning approaches for edge AI.

While our findings offer valuable insights into the feasibility of CNN-based activity recognition, several practical challenges were identified, including resource constraints and inference latencies while running deep neural networks on edge devices. We extended our work to exploring several model compression techniques and achieved promising results: even though model accuracy degraded a little for the knowledge distilled model and model quantized post-training, we're able to significantly reduce the model size, memory footprint during inference as well as the inference time from the original CNN model.

Future research could explore alternative CNN architectures, fine-tune the student model architecture during distillation and use quantization aware model training to further address the practical concerns for model deployment in real-world settings. We would also like to test the generazability of the models we built with other datasets such as the WISDM: Wireless Sensor Data Mining as well. By advancing our understanding of human activity recognition, we can contribute to the improvement of accuracy, scalability, and applicability of HAR systems for various domains including healthcare, fitness tracking, and smart environments.

References

[1] Ankita, S. Rani, H. Babbar, S. Coleman, A. Singh, and H. M. Aljahdali, "An efficient and lightweight deep learning model for human activity recognition using smartphones," *Sensors (Basel, Switzerland)*, vol. 21, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235423075

[2] D. Garcia-Gonzalez, D. Rivero, E. Fernández-Blanco, and M. R. Luaces, "A public domain dataset for real-life human activity recognition using smartphone sensors," *Sensors (Basel, Switzerland)*, vol. 20, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:215793300

[3] H. Li, "Exploring knowledge distillation of deep neural networks for efficient hardware solutions," *Stanford CS230 Report*, 2018. [Online]. Available: https://cs230.stanford.edu/files_winter_2018/projects/6940224.pdf

[4] R. Mohammad, "Human activity recognition(har)," *Medium Blog*, 2019. [Online]. Available: https://medium.com/@rubeen.786.mr/human-activity-recognition-har-db5c1432cd98

[5] J. Reyes-Ortiz, D. Anguita, A. Ghio, L. Oneto, and X. Parra, "Human Activity Recognition Using Smartphones," UCI Machine Learning Repository, 2012, DOI: https://doi.org/10.24432/C54S4K.

[6] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang, and J. Liu, “Human action recognition from various data modalities: A review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–20, 2022. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2022.3183112

[7] M.-K. Yi, W.-K. Lee, and S. O. Hwang, “A human activity recognition method based on lightweight feature extraction combined with pruned and quantized cnn for wearable device,” *IEEE Transactions on Consumer Electronics*, vol. 69, no. 3, pp. 657–670, 2023.