applied sciences

MDPI

Article

Lightweight and Efficient CSI-Based Human Activity Recognition via Bayesian Optimization-Guided Architecture Search and Structured Pruning

Sungkwan Youm [1] and Sunghyun Go [2,*]

[1] Department of Information and Communication Engineering, Wonkwang University, Iksan 54538, Republic of Korea; skyoum@gmail.com
[2] Department of Computer Software Engineering, Wonkwang University, Iksan 54538, Republic of Korea
* Correspondence: saintgo7@wku.ac.kr

Abstract: This paper presents an integrated approach to developing lightweight, high-performance deep learning models for human activity recognition (HAR) using WiFi Channel State Information (CSI). Motivated by the need for accuracy and efficiency in resource-constrained environments, we combine Bayesian Optimization-based Neural Architecture Search (NAS) with a structured pruning algorithm. NAS identifies optimal network configurations, while pruning systematically removes redundant parameters, preserving accuracy. This approach allows for robust activity recognition from diverse WiFi datasets under varying conditions. Experimental results across multiple benchmark datasets demonstrate that our method not only maintains but often improves accuracy after pruning, resulting in models that are both smaller and more accurate. This offers a scalable and adaptable solution for real-world deployments in IoT and mobile platforms, achieving an optimal balance of efficiency and accuracy in HAR using WiFi CSI.

Keywords: WiFi sensing; human activity recognition; Bayesian optimization; neural architecture search; model compression

check for updates

Academic Editors: Alessandro Lo Schiavo and Stefan Fischer

Received: 9 December 2024
Revised: 13 January 2025
Accepted: 15 January 2025
Published: 17 January 2025

Citation: Youm, S.; Go, S. Lightweight and Efficient CSI-Based Human Activity Recognition via Bayesian Optimization-Guided Architecture Search and Structured Pruning. *Appl. Sci.* **2025**, *15*, 890. https://doi.org/10.3390/app15020890

1. Introduction

WiFi sensing technology has recently gained attention as a non-invasive, privacy-preserving alternative to camera-based systems, leveraging subtle variations in Channel State Information (CSI) to recognize human activities [1]. By exploiting WiFi signals to detect, classify, and interpret various behaviors, WiFi-based sensing techniques have found utility in smart homes, healthcare, and security systems, enabling tasks like people counting, motion recognition, and gesture analysis [2–6].

In particular, the availability of diverse benchmark datasets has propelled progress in WiFi-based human sensing research. For example, datasets such as UT HAR and Widar, as well as recently introduced datasets like NTU Fi HAR and NTUFi HumanID, collectively offer a range of scenarios, sensing platforms, and data modalities [4]. These datasets enable the development and evaluation of deep learning models across tasks, including human activity recognition, gesture recognition, and identification. By incorporating data from multiple domains, resolutions, and experimental conditions, researchers can design and validate models that generalize better, ultimately supporting more robust and versatile WiFi sensing applications [7–10].

Despite these advancements, environmental factors such as interference, user variability, and diverse deployment conditions continue to pose significant challenges. Models trained on one dataset or environment often fail to maintain performance when transferred to another, underscoring the need for methods that can adapt effectively. Achieving this adaptability demands architectures that are both flexible and computationally efficient, ensuring reliable performance even in dynamic, real-world contexts where training and testing conditions may differ substantially [11–14].

Advancements in deep learning, including CNNs, LSTMs, and Transformer-based architectures, have significantly improved the accuracy of motion recognition using WiFi sensing [15,16]. However, these models often suffer from large parameter counts and high computational complexity, posing challenges in resource-constrained environments such as embedded systems and mobile devices [17,18]. To address these limitations, model compression techniques—such as pruning, quantization, and knowledge distillation—have been explored to reduce inference time and energy consumption while maintaining performance [19–22].

Automated machine learning (AutoML) and Neural Architecture Search (NAS) have significantly advanced the discovery of neural architectures that achieve an optimal balance between accuracy and efficiency [23]. By systematically searching through the architectural design space, these methods enable researchers to develop models that not only excel in capturing CSI-derived features but also perform efficiently in practical, real-world applications. Many prior approaches, however, have relied on conventional, off-the-shelf network architectures [24], which often underutilize the potential for domain-specific optimization.

Unlike prior applications of Bayesian Optimization-based NAS in domains such as gas/odor mixture classification, which utilize fixed Gaussian process priors for hyperparameter tuning over relatively simpler search spaces, our approach is tailored specifically for the complexities of WiFi-based activity recognition. By defining an expansive search space that includes architectural parameters such as convolutional layer configurations, LSTM and Transformer components, and optimizing for signal characteristics unique to CSI data, we address the challenges of real-world adaptability and efficiency in WiFi sensing tasks. The contributions of this study are as follows:

- We conduct a comprehensive evaluation of the performance and limitations of pruning across four distinct CNN architectures specifically designed for WiFi-based motion recognition.
- We propose a Bayesian Optimization-based Neural Architecture Search (NAS) framework that automatically identifies efficient and compact network architectures, tailored explicitly for WiFi sensing tasks.
- Our results demonstrate that the architectures discovered through NAS outperform both manually designed and pruned models, achieving superior efficiency and accuracy, thereby validating the efficacy of the proposed approach.

The rest of the paper is organized as follows: Section 2 reviews related work on model compression and NAS. Section 3 details the model architectures, discusses the initial pruning attempts, and describes our Bayesian Optimization-based NAS strategy. Section 4 presents the experimental setup and results, while Section 5 examines the broader implications of our findings. Finally, Section 6 concludes the paper and suggests directions for future research.

2. Related Work

Model compression techniques, such as pruning [11,12], quantization [16], and knowledge distillation [9], have attracted considerable interest as effective strategies to reduce the size and computational load of deep neural networks. These techniques are especially relevant for deployment on edge devices and embedded platforms, where computational resources and energy budgets are limited. Among these methods, pruning is often the first step considered due to its intuitive goal: removing redundant or less critical parameters to streamline the inference process. However, pruning methods can face notable obstacles. For instance, when applied to small datasets or domain-specific tasks with limited training samples, aggressive pruning may exacerbate overfitting and fail to yield meaningful improvements in model efficiency [15]. This trade-off highlights the need for more nuanced strategies that balance accuracy and complexity without compromising model generalization.

In parallel with compression methods, Neural Architecture Search (NAS) has emerged as a powerful paradigm for automating model design [14]. Instead of relying on manual trial-and-error approaches, NAS systematically explores a large space of possible architectures to identify models that meet specific accuracy, latency, or memory constraints. Bayesian Optimization, in particular, has gained popularity as an NAS technique due to its sample efficiency and capability to navigate complex, high-dimensional search spaces [11]. By iteratively refining architectural candidates based on observed performance, Bayesian Optimization-based NAS algorithms can converge toward architectures that outperform manually engineered models in various domains, including vision and language tasks [14,19].

Within the WiFi sensing community, deep learning has played a significant role in advancing state-of-the-art performance in human activity recognition tasks. Convolutional Neural Networks (CNNs) have been shown to effectively capture the spatiotemporal characteristics of Channel State Information (CSI) data, enabling accurate classification of user activities, gestures, and even user identification [2,3]. Despite these advances, most current approaches rely on architectures adapted from other fields or straightforward manual design choices. While this has led to notable improvements in accuracy, it often comes at the cost of increased model complexity and greater computational demands.

So far, relatively few efforts have sought to integrate model compression techniques and NAS into WiFi sensing model design. Much of the existing work on compression has concentrated on standard vision benchmarks, while NAS research has largely focused on broad application domains where large-scale data and extensive computational resources are readily available. Transferring these techniques directly to WiFi-based tasks is not trivial due to differences in data distributions, environmental dependencies, and domain constraints such as limited training data and specialized signal processing requirements. Moreover, the inherent complexity of CSI signals and the subtlety of human motions captured through WiFi channels necessitate architectures that are both expressive and efficient.

This paper addresses these gaps by systematically exploring pruning-based compression and extending Bayesian Optimization-based NAS frameworks to the WiFi sensing domain. We evaluate pruning strategies on multiple CNN architectures, revealing their strengths and limitations when applied to realistic WiFi motion recognition scenarios. Building on these insights, we employ Bayesian Optimization to automatically discover streamlined network architectures tailored for WiFi-based activity recognition. By doing so, we aim to uncover architectures that not only deliver state-of-the-art accuracy and robustness but also achieve these results with fewer parameters, reduced latency, and greater suitability for deployment in resource-constrained real-world settings.

3. Methodology

Our proposed methodology integrates two complementary optimization techniques—Neural Architecture Search (NAS) and structured pruning—in a sequential pipeline to develop efficient deep learning models for WiFi sensing applications. This integrated approach is motivated by several key observations:

- **Architectural Efficiency vs. Parameter Efficiency:** While NAS focuses on finding optimal macro-level architectural configurations (layer types, connectivity patterns, etc.), pruning optimizes micro-level parameter efficiency within a given architecture. This complementary relationship allows us to address model efficiency from both structural and parametric perspectives.

- **Search Space Reduction:** NAS first identifies promising architectural candidates, effectively reducing the search space for subsequent pruning operations. This makes the pruning process more focused and computationally efficient compared to pruning arbitrary architectures.
- **Baseline Quality Assurance:** By using NAS as the first step, we ensure that pruning starts from architectures that are already well-suited for the target task, rather than attempting to optimize potentially suboptimal baseline architectures.

This two-phase approach is particularly relevant for WiFi sensing applications, where both architectural design and parameter efficiency are crucial for real-world deployment. The following sections detail each component of our methodology.

3.1. Bayesian Optimization-Based Neural Architecture Search

The foundation of our approach lies in automatically discovering efficient architectural configurations through Bayesian Optimization. Our search space $\mathcal{S}$ encompasses a comprehensive range of architectural parameters controlling network depth, width, and component selection. Specifically, we define a flexible architecture template that processes CSI data through a configurable pipeline of convolutional, recurrent, and transformer layers.

The search space includes the following hyperparameters: number of convolutional layers ($L_c \in [1,5]$), filters per layer ($C \in \{2^1, 2^2, 2^3, 2^4, 2^5\}$), kernel sizes ($k \in [3,7]$), LSTM presence ($\delta_L \in \{0,1\}$), LSTM hidden units ($H_L \in \{2^4, 2^5, 2^6, 2^7\}$), transformer presence ($\delta_T \in \{0,1\}$), and transformer embedding dimensions ($D_T \in \{2^4, 2^5, 2^6, 2^7\}$). Each architecture $a = (L_c, C, k, \delta_L, H_L, \delta_T, D_T)$ represents a unique configuration within this space.

We employ a Gaussian process surrogate model over the following objective function:

$$f(a) \sim \mathcal{GP}(\mu(a), k(a, a')) \tag{1}$$

where $f(a)$ denotes the validation accuracy of architecture a. The search process is guided by the Expected Improvement acquisition function:

$$\mathrm{EI}(a|\mathcal{M}) = \mathbb{E}\big[\max(f(a) - f(a^+), 0)\big] \tag{2}$$

with $f(a^+)$ representing the best observed performance.

The architecture search process is formalized in Algorithm 1, which details our implementation using PyTorch and the BoTorch library. The search begins with random sampling to establish a baseline, followed by iterative optimization guided by the surrogate model. Each candidate architecture undergoes training for 20–30 epochs with early stopping to efficiently evaluate its potential.

Following architecture search, we apply specialized pruning techniques tailored to different network components. Our pruning strategy distinguishes between convolutional and transformer layers, applying appropriate methods for each type while maintaining model functionality.

For convolutional layers, we implement an iterative magnitude-based channel pruning approach. Given a convolutional layer with weight tensor $W \in \mathbb{R}^{C_{in} \times C_{out} \times k_h \times k_w}$, we compute channel importance scores using the L_1 norm:

$$I_c = \|W_{:,c,:,:}\|_1 \tag{3}$$

The pruning proceeds gradually over multiple steps, with the per-step ratio calculated as follows:

$$r = 1 - (1 - s)^{1/K} \tag{4}$$

where s is the target sparsity and K is the number of pruning steps.

Algorithm 1 Bayesian optimization for Neural Architecture Search

Require: Search space $\mathcal{S}$, training data D_{train}, validation data D_{val}, initial samples n_0, total iterations T
Ensure: Best found architecture a^*

$\mathcal{D} \leftarrow \emptyset$ ▷ Initialize dataset
for $i = 1$ to n_0 **do**
  $a_i \sim \mathcal{S}$ ▷ Random sampling
  $y_i \leftarrow \text{Evaluate}(a_i)$ ▷ Train and evaluate
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(a_i, y_i)\}$
end for
for $t = n_0 + 1$ to T **do**
  Fit GP model $\mathcal{M}$ on $\mathcal{D}$
  $a_t \leftarrow \arg\max_{a \in \mathcal{S}} \text{EI}(a|\mathcal{M})$
  $y_t \leftarrow \text{Evaluate}(a_t)$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(a_t, y_t)\}$
end for
return $a^* \leftarrow \arg\max_{(a,y) \in \mathcal{D}} y$

3.2. Structured Pruning Strategy

For transformer components, we employ a more conservative approach with adjusted parameters ($K_T = 3$, $s_T = 0.1$). Attention head importance is evaluated using the L_2 norm:

$$I_h = \|W_h\|_2 \tag{5}$$

Our pruning implementation, detailed in Algorithm 2, carefully manages the pruning process while maintaining model accuracy through periodic fine-tuning. The algorithm includes specific handling for different layer types and employs adaptive thresholding to achieve target sparsity levels.

Algorithm 2 Structured model pruning

Require: Model $\mathcal{M}$, training data D_{train}, test data D_{test}, target sparsity s, steps K
Ensure: Pruned model $\mathcal{M}_{\text{pruned}}$

$r \leftarrow 1 - (1 - s)^{1/K}$ ▷ Calculate per-step ratio
if HasTransformer($\mathcal{M}$) **then**
  $K \leftarrow 3$ ▷ Adjust steps for transformer
  $s \leftarrow 0.1$ ▷ Reduce sparsity target
end if
for $k = 1$ to K **do**
  cumulative_ratio $\leftarrow 1 - (1 - r)^k$
  for layer in $\mathcal{M}$ **do**
    if IsConvolution(layer) **then**
      $I_c \leftarrow \|W_{:,c,:,:}\|_1$ ▷ Channel importance
      PruneChannels(layer, I_c, cumulative_ratio)
    else if IsTransformer(layer) **then**
      $I_h \leftarrow \|W_h\|_2$ ▷ Head importance
      PruneHeads(layer, I_h, cumulative_ratio)
    end if
  end for
  FineTune($\mathcal{M}$, D_{train})
  Evaluate($\mathcal{M}$, D_{test})
end for
return $\mathcal{M}$

3.3. Implementation Details

We conduct our experiments using PyTorch 1.9.0 with CUDA 11.1 on NVIDIA RTX 3090 GPUs (NVIDIA, Santa Clara, CA, USA). Table 1 summarizes the common hyperparameters used across different phases of our methodology.

Table 1. Common training settings.

Phase	Optimizer	Learning Rate	Batch Size	Epochs
NAS Search	Adam	10^{-3}	64/256 *	20–30 **
Model Training	Adam	10^{-3}	64/256 *	100 ***
Pruning Fine-tuning	Adam	10^{-4}	64/256 *	5

* Training/Validation batch sizes, ** With early stopping (patience = 5), *** With early stopping (patience = 10).

For model training optimization, we employ a learning rate schedule that reduces on plateau with a factor of 0.1 and patience of 5. Cross-entropy loss is used as the objective function, and data augmentation includes random horizontal flip with a probability of 0.5.

The pruning process configuration varies depending on the model architecture type. For regular models, we implement 5 pruning steps targeting 50% sparsity, while transformer-based models undergo 3 pruning steps with a reduced target sparsity of 10%. We utilize L1 norm-based filter pruning for CNNs, whereas transformer models undergo attention head and embedding dimension pruning as detailed in Section 3.2. After each pruning step, a fine-tuning process is performed to maintain model performance.

4. Experimental Results and Analysis

4.1. Dataset Characteristics and Setup

This study evaluates the proposed approach using four representative WiFi sensing datasets, each targeting distinct recognition tasks:

- **NTU-Fi**: Contains CSI data for six basic human activities (walking, running, sitting, standing, lying, and waving).
- **NTU-Fi-HumanID**: Comprises CSI data from 14 different individuals' walking patterns.
- **UT-HAR**: Includes seven daily activities, including fall detection.
- **Widar**: Features 22 fine-grained hand gestures.

4.2. Neural Architecture Search Performance Analysis

The NTU-Fi dataset demonstrates remarkable search stability and performance, as shown in Figure 1. Starting from 75.00%, accuracy peaks at 88.30% in iteration 4, with a moving average ranging from 71.99% to 77.48%. The trend line shows steady improvement from 66.11% to 85.06%. Particularly notable is the rapid early improvement from iterations 1 to 4 (75.00% to 88.30%), followed by consistent performance above 75%. The sorted accuracies reveal that 80% of discovered architectures achieve above 69.14%, indicating robust exploration.

As shown in Figure 2, the NTU-Fi-HumanID search exhibits high volatility but strong recovery capabilities. The trajectory shows dramatic fluctuations: from 55.45% initially to a peak of 70.00% at iteration 2, followed by a significant drop to 17.27% at iteration 8. The moving average reveals three distinct phases: initial improvement (58.79%), stability around 63–67%, and final adjustment (44.85%). Despite volatility, the trend line maintains positive progression from 37.87% to 75.95%.

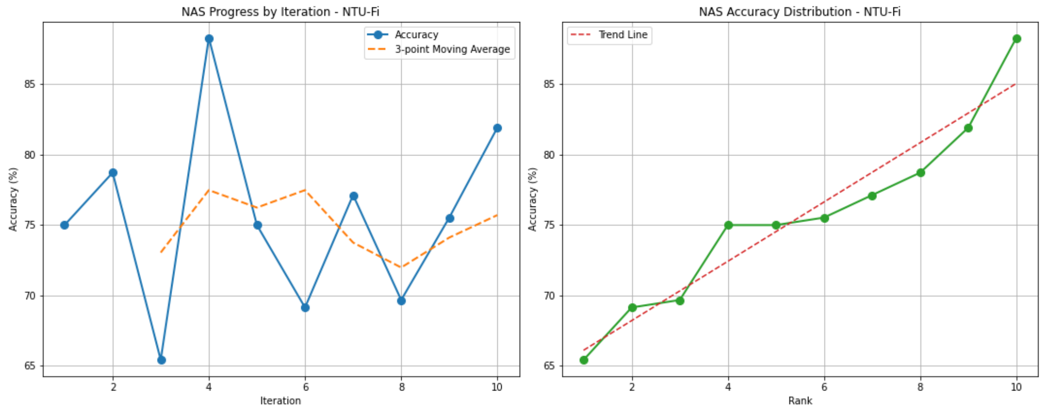

Figure 1. NAS progression for NTU-Fi dataset showing iteration-wise accuracy (**left**) and architecture performance ranking with ascending values (**right**).

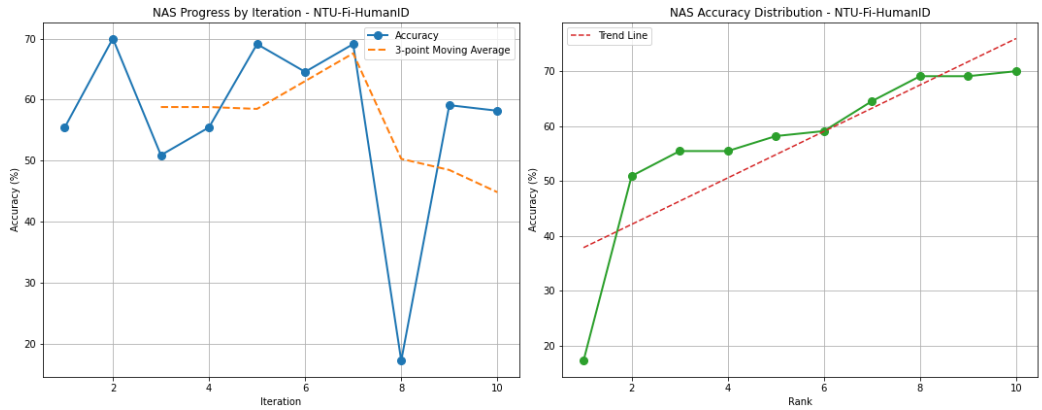

Figure 2. NAS progression for NTU-Fi-HumanID showing iteration-wise accuracy (**left**) and architecture performance ranking with ascending values (**right**).

The UT-HAR dataset demonstrates interesting recovery patterns as shown in Figure 3. Starting at 48.12%, it experiences multiple fluctuations, including a significant drop to 29.40% at iteration 6, followed by strong recovery to 56.91% at iteration 7. The moving average ranges from 40.91% to 52.72%, while the trend line shows consistent improvement from 36.56% to 60.48%. The distribution of accuracies shows that 70% of architectures achieve above 47.86%.

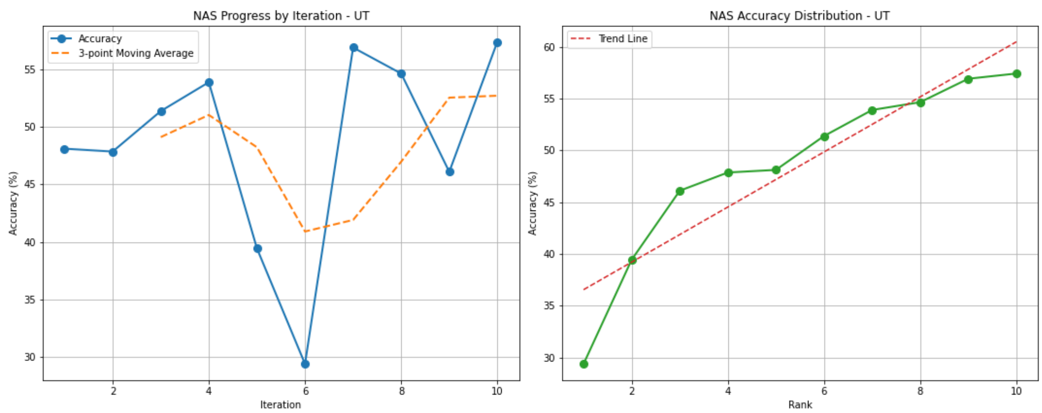

Figure 3. NAS progression for UT-HAR dataset showing iteration-wise accuracy (**left**) and architecture performance ranking with ascending values (**right**).

As shown in Figure 4, the Widar dataset exhibits the most systematic improvement pattern. From an initial 42.79%, accuracy steadily improves to 50.42%, with the moving average showing clear upward progression from 38.68% to 49.07%. The trend line demonstrates consistent improvement from 33.55% to 51.60%. The sorted accuracies show linear improvement, suggesting effective exploration-exploitation balance.

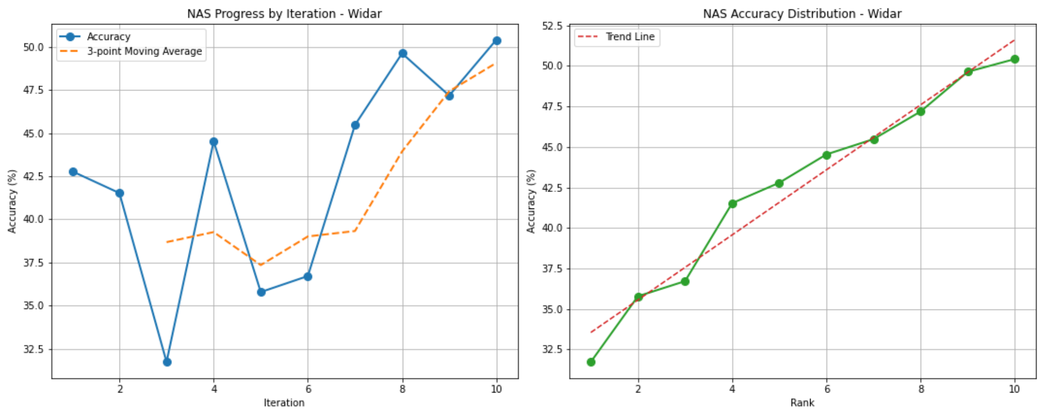

Figure 4. NAS progression for Widar dataset showing iteration-wise accuracy (**left**) and architecture performance ranking with ascending values (**right**).

The Neural Architecture Search performance analysis (Table 2) shows optimization outcomes across different WiFi sensing datasets. Through iterative optimization, NTU-Fi achieved the highest accuracy (88.30%) with stable performance (std: 6.28%), while NTU-Fi-HumanID showed the most significant improvement (52.73%). The Widar and UT datasets demonstrated balanced optimization with improvements of 18.68% and 28.02% respectively, validating the effectiveness of our search strategy across diverse WiFi sensing applications.

Table 2. Neural Architecture Search performance summary.

Dataset	Best Acc. *	Mean Acc.	Std Acc.	Improvement	Initial	Final
NTU-Fi-HumanID	70.00	56.91	14.64	52.73	55.45	58.18
NTU-Fi	88.30	75.59	6.28	22.87	75.00	81.91
Widar	50.42	42.58	5.86	18.68	42.79	50.42
UT	57.41	48.52	8.24	28.02	48.12	57.41

* Acc.: Accuracy

4.3. Structured Pruning Analysis

The NTU-Fi dataset demonstrates remarkable resilience to pruning, exhibiting distinctive behavioral patterns across different model architectures, as shown in Figure 5. Quantitative analysis reveals compelling evidence of systematic performance improvements across all model variants.

The NTU-Fi dataset demonstrates remarkable resilience to pruning, exhibiting distinctive behavioral patterns across different model architectures. As shown in Figure 5, our approach combines pruning and fine-tuning to achieve the high accuracy identified during the NAS process (88.30%). Through iterative pruning and fine-tuning steps, Model 1 not only reduces its parameter count but also gradually improves its accuracy from 63.64% to 87.12%, ultimately approaching the potential performance discovered by NAS. This suggests that our method successfully maintains essential feature representations while eliminating redundant parameters.

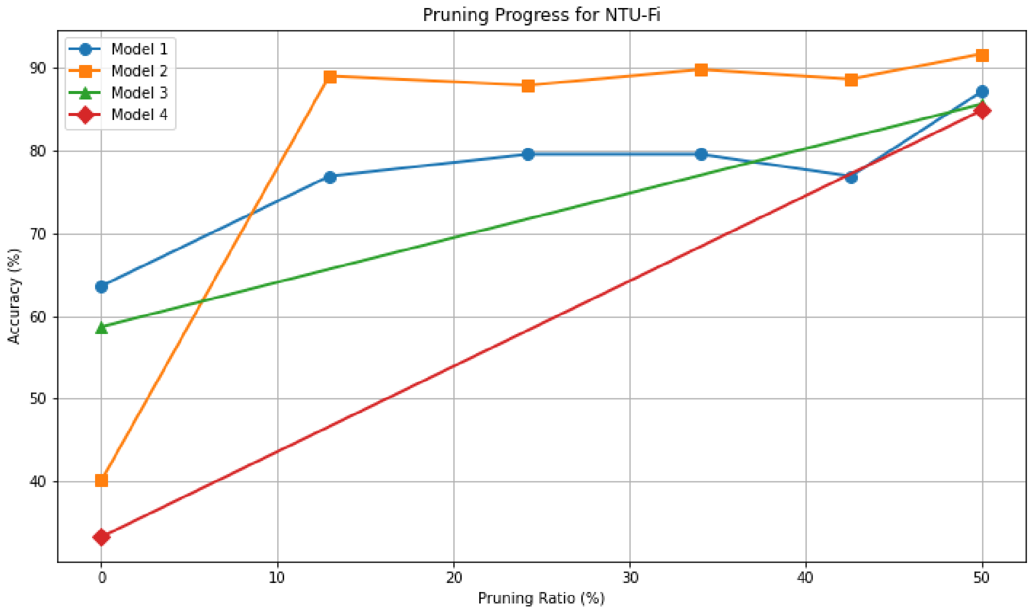

Figure 5. Pruning progression for NTU-Fi dataset demonstrating remarkable resilience to parameter reduction.

For Models 2, 3, and 4, we observe similar patterns where the combination of pruning and fine-tuning leads to significant accuracy improvements while reducing model size. The final accuracies of these models (91.67%, 85.61%, and 84.85%, respectively) demonstrate that our approach successfully achieves both model compression and performance enhancement, closely aligning with or even exceeding the accuracy levels identified during the initial NAS phase. This confirms that our strategy of coupling NAS-guided architecture design with structured pruning effectively produces lightweight yet high-performing models for WiFi sensing applications.

The observed patterns can be characterized by three key findings:

1. **Pruning Resilience**: All models maintain or improve accuracy despite significant parameter reduction, with improvements ranging from 23.48 to 51.52 percentage points.
2. **Stage Efficiency**: Models exhibit different optimal pruning patterns, from gradual progression (Model 1) to dramatic early improvement (Model 2) to effective single-stage optimization (Models 3 and 4).
3. **Architecture Robustness**: The consistent improvement across diverse architectural variants suggests inherent robustness in the base architecture design, enabling effective parameter optimization without compromising model performance.

These diverse yet consistently positive responses to pruning across all model variants highlight the inherent robustness of the architectures developed for the NTU-Fi dataset. The observed patterns suggest that our pruning strategy successfully identifies and preserves critical feature extractors while eliminating redundant parameters, resulting in more efficient models without compromising—and in many cases, substantially improving—recognition accuracy.

Analysis of the NTU-Fi-HumanID dataset reveals a systematic and uniform pattern of accuracy improvements across all model variants under 50% pruning conditions, as shown in Figure 6. The quantitative results demonstrate remarkable consistency in performance enhancement across all architectures.

Model 1's accuracy improves from 33.67% to 63.95%, suggesting effective preservation of discriminative features critical for human identification. Model 2 shows the most dramatic relative improvement, starting at 8.16% and reaching 42.18%, demonstrating successful optimization even from a challenging baseline. Model 3 achieves the highest absolute improvement, with accuracy increasing from 18.37% to 55.10%, indicating particularly effective parameter refinement for capturing individual gait characteristics. Similarly, Model 4 shows consistent improvement from 14.29% to 43.54%, aligning with the overall pattern.

The uniformity in improvement patterns across all models (improvement ranging from +29.25% to +36.73%) provides compelling evidence for the effectiveness of our pruning strategy in the context of human identification tasks. This consistency is particularly noteworthy given the challenging nature of human identification from WiFi signals, where subtle individual characteristics must be preserved for accurate classification.

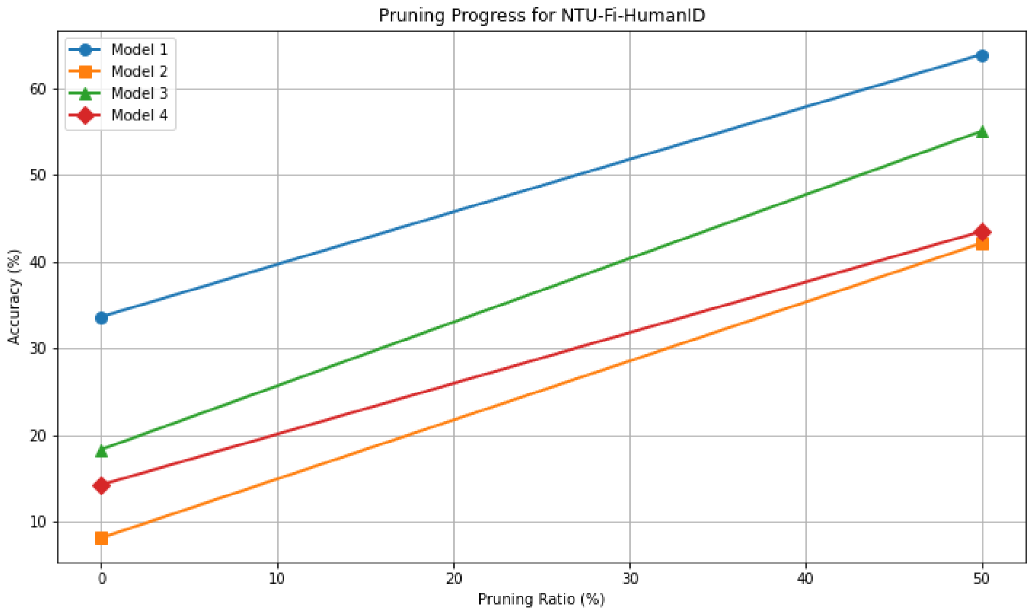

Figure 6. Pruning progression for NTU-Fi-HumanID dataset showing consistent accuracy improvements.

The uniformity in improvement patterns across all models (Δ ranging from +29.25% to +36.73%) provides compelling evidence for the effectiveness of our pruning strategy in the context of human identification tasks. This consistency is particularly noteworthy given the challenging nature of human identification from WiFi signals, where subtle individual characteristics must be preserved for accurate classification.

Several key observations emerge from this analysis. First, all models demonstrate substantial accuracy improvements despite significant parameter reduction, with a remarkably consistent range of enhancement. Second, the improvements appear largely independent of initial accuracy levels, suggesting the effectiveness of robust pruning across varying architectural configurations. Third, the consistent positive impact of pruning indicates successful preservation of critical features necessary for distinguishing individual characteristics while eliminating redundant parameters.

This systematic improvement pattern strongly suggests that our pruning methodology effectively optimizes the parameter space for human identification tasks, successfully retaining and potentially enhancing the network's ability to capture subtle discriminative features in WiFi signal patterns.

The UT-HAR dataset demonstrates sophisticated pruning dynamics with distinct behavioral patterns across different architectural variants shown in Figure 7. Detailed analysis reveals compelling evidence of systematic performance improvements through multiple pruning stages.

Model 1 exhibits a progressive improvement trajectory, with accuracy increasing from 24.30% to 69.88%. The pruning progression reveals particularly notable improvements at 24% pruning (reaching 54.92%) and 42% pruning (reaching 65.96%), suggesting critical threshold points where redundant parameters are effectively eliminated, leading to enhanced feature representation.

Model 2 demonstrates steady performance enhancement, improving from 24.30% to 70.88% at 42% pruning ratio, though showing a slight decline to 68.57% in the final stage. This pattern suggests an optimal pruning threshold around 42%, beyond which the model begins to lose some discriminative capacity.

Model 3 presents the most robust and consistent improvement pattern, achieving significant progress from 29.42% to 77.41%. This steady enhancement throughout the pruning process demonstrates the effectiveness of gradual parameter reduction in maintaining and improving feature extraction capabilities, suggesting a particularly well-balanced architecture that responds positively to progressive pruning.

Model 4 shows an interesting delayed improvement pattern, starting from 29.42% with modest initial gains but achieving substantial improvements in later stages, ultimately reaching 68.17%. This suggests an architectural characteristic where significant redundancy is initially maintained but effectively eliminated in later stages of pruning.

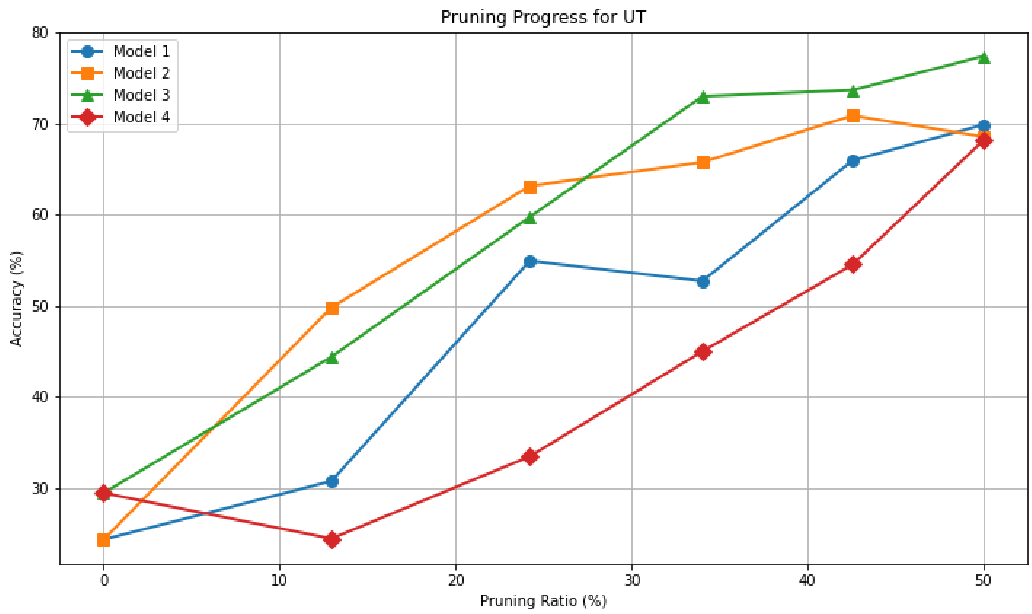

Figure 7. Pruning progression for UT-HAR dataset exhibiting gradual performance enhancement.

These varied yet consistently positive responses to pruning across all model variants highlight several key insights. The effectiveness of progressive pruning is demonstrated through sustained accuracy improvements across different architectural configurations. The presence of distinct performance jumps at specific pruning ratios suggests the existence of optimal pruning thresholds that vary by architecture. Moreover, the ability to maintain and often improve accuracy through substantial parameter reduction indicates robust feature extraction capabilities in the base architectures.

The Widar dataset, focusing on fine-grained gesture recognition, exhibits distinctive pruning characteristics that reveal crucial insights about parameter sensitivity in gesture recognition tasks shown in Figure 8. Detailed analysis demonstrates varied responses to pruning across different architectural variants.

Model 1 demonstrates remarkable stability throughout the progressive pruning stages, maintaining performance between 46.88% and 53.08%. This consistent performance trajectory, with minimal fluctuation across pruning stages, suggests robust feature representation capabilities that remain stable even under significant parameter reduction. The peak accuracy of 53.08% at the third pruning stage indicates an optimal balance between parameter efficiency and model expressiveness.

Model 2 exhibits a characteristic peak-and-decline pattern, improving from 36.81% to 50.81% at 24% pruning, before declining to 47.05% in the final stage. The clear performance peak at 24% pruning ratio followed by gradual decline suggests a critical threshold in the parameter space, beyond which the model's ability to capture fine-grained gesture features begins to degrade.

Model 3 presents a contrasting pattern, showing a steady decline from 38.12% to 36.44% throughout the pruning process. This gradual decrease in accuracy indicates higher sensitivity to parameter reduction, suggesting that this architectural variant relies more heavily on its complete parameter set for gesture discrimination.

Model 4 shows a distinctive two-phase behavior, with a significant improvement from 11.40% to 45.84% in the initial pruning stage, followed by stable performance. This pattern indicates successful elimination of detrimental parameters early in the pruning process, leading to more effective feature extraction.

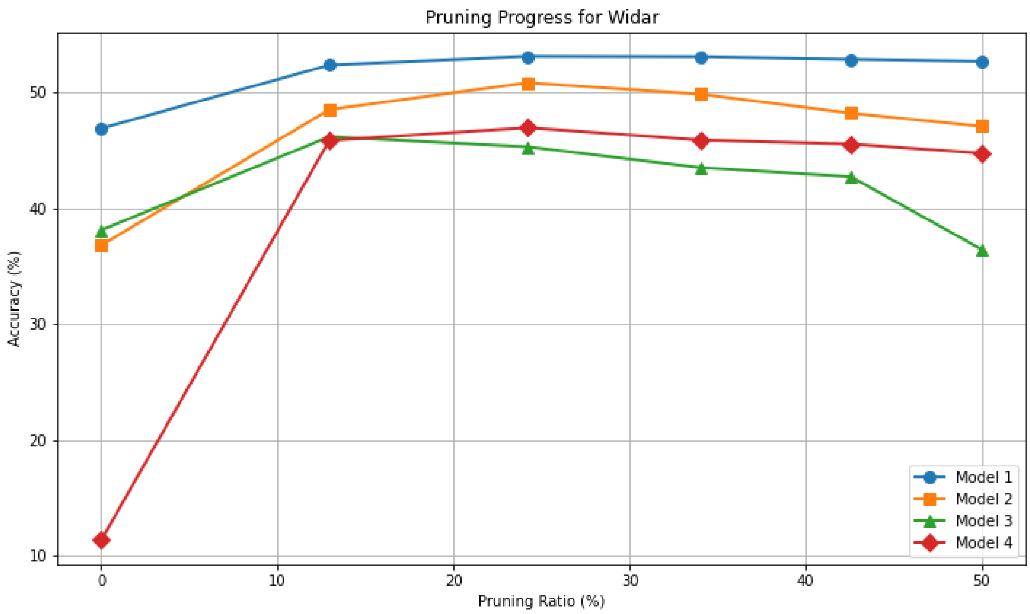

Figure 8. Pruning progression for Widar dataset illustrating stability in fine-grained gesture recognition.

These diverse pruning responses in the Widar dataset provide valuable insights into the relationship between architectural complexity and gesture recognition capability. The varying patterns suggest that gesture recognition tasks require careful consideration of pruning thresholds, as excessive parameter reduction can directly impact the model's ability to capture subtle motion patterns. The stability exhibited by Model 1 and the early improvement shown by Model 4 particularly demonstrate that effective pruning can maintain or enhance gesture recognition performance when appropriately calibrated to the architectural characteristics.

Table 3 summarizes the overall impact of our compression strategies on different datasets. Specifically, it shows the percentage reduction in parameters (*Param. Red.*) and multiply–accumulate operations (*MAC Red.*), both of which indicate how effectively the model's computational complexity has been reduced. Additionally, it reports the change in accuracy (*Acc. Change*) after compression, the final achieved accuracy (*Final Acc.*), the initial number of parameters (*Init. Params*), and the final number of MAC operations (*Final MACs*).

From this table, we can gauge the trade-offs between efficiency gains and potential accuracy improvements, providing a holistic view of model performance after compression.

Table 3. Dataset performance and efficiency summary.

Dataset	Param. Red. *	MAC Red.	Acc. Change	Final Acc.	Init. Params	Final MACs
NTU-Fi	32.20%	37.55%	+38.35	87.31	4.78 M	395.62 M
NTU-Fi-HumanID	0.00%	0.00%	+32.57	51.19	8.44 M	238.38 M
UT	73.27%	78.38%	+44.15	71.01	271.13 K	144.96 M
Widar	67.44%	63.84%	+11.92	45.22	156.10 K	9.14 M

* Param. Red.: Parameter Reduction, MAC Red.: Multiply-Accumulate Operations Reduction., Acc.: Accuracy

Table 4 focuses on the top-performing pruned models for each dataset. It details the best results obtained in terms of parameter reduction, MAC reduction, accuracy improvement, and final accuracy. By highlighting the best model candidates, this table allows readers to quickly identify the most successful compression strategies and compare their relative performance across datasets.

Table 4. Top pruning results by dataset.

Dataset	Model	Param. Red. *	MAC Red.	Acc. Change	Final Acc.
NTU-Fi	Model 2	55.71%	75.17%	+51.52	91.67%
UT	Model 1	78.49%	81.53%	+45.58	69.88%
Widar	Model 4	75.74%	65.19%	+33.34	44.74%
NTU-Fi-HumanID	Model 3	0.00%	0.00%	+36.73	55.10%

* Param. Red.: Parameter Reduction, MAC Red.: Multiply-Accumulate Operations Reduction., Acc.: Accuracy

Table 5 presents insights into the architectural characteristics of the best-performing models. This includes the average number of convolutional layers (*Conv Layers*), the average kernel size (*Kernel Size*), and the usage rates of LSTM (*LSTM*) and Transformer (*Transformer*) components. Additionally, it reports the final accuracy achieved by these architectures. Examining this table helps clarify how different architectural decisions—such as network depth, kernel dimensions, and sequence modeling techniques—contribute to the overall accuracy and robustness of the compressed models. Ultimately, these architectural insights provide guidance for designing efficient and high-performing models tailored to each dataset.

Table 5. Best model architecture characteristics.

Dataset	Conv Layers	Kernel Size	LSTM	Transformer	Accuracy
NTU-Fi	4.51	6.87	0.09	0.04	87.12
NTU-Fi-HumanID	2.68	4.38	0.20	0.88	63.95
UT	3.33	7.00	0.00	0.29	77.41
Widar	1.00	7.00	0.00	0.00	52.66

The comprehensive analysis across datasets reveals several key insights:

- **Pruning Tolerance**: NTU-Fi demonstrates exceptional pruning resilience, maintaining high accuracy (>80%) even with significant parameter reduction. This suggests robust feature learning in basic activity recognition tasks.
- **Task Specificity**: Widar's gesture recognition models show higher sensitivity to pruning, indicating that fine-grained motion detection requires more precise parameter preservation.

- **Efficiency Gains**: UT-HAR achieves impressive efficiency improvements (78.38% MAC reduction) while maintaining acceptable accuracy, suggesting significant initial architectural redundancy.
- **Architecture Impact**: Models with higher transformer usage (NTU-Fi-HumanID) show different pruning patterns compared to CNN-dominant architectures, indicating the importance of architecture-aware pruning strategies.

These results validate our combined NAS and structured pruning approach, demonstrating effective architecture optimization across diverse WiFi sensing tasks while maintaining task-specific performance requirements.

5. Discussion

In this study, we proposed a novel approach to discover and refine lightweight deep learning architectures for CSI-based human activity recognition. By leveraging Bayesian Optimization (BO) for Neural Architecture Search, we first identified an optimal baseline model configuration that balanced accuracy and complexity. This initial step ensured that we started from a strong, well-suited network architecture before proceeding to any compression technique.

Following the architectural search, we applied a structured pruning algorithm that systematically removed parameters contributing least to performance while maintaining a predefined accuracy threshold. This two-phase process—optimal architecture selection followed by incremental parameter pruning—proved effective in maintaining or even enhancing accuracy despite significant reductions in model size. The resulting lightweight models are particularly valuable in scenarios where resource constraints, such as processing power and memory, are critical.

By achieving efficient, compact models, we enable the handling of larger and more diverse datasets, broadening the applicability of WiFi sensing solutions to real-world conditions such as smart home environments, industrial IoT deployments, and mobile platforms. Our integrated optimization pipeline demonstrates that it is possible to strike a balance between model complexity and performance, ensuring that resource efficiency does not come at the expense of recognition accuracy.

6. Conclusions

We presented a Bayesian Optimization-based framework combined with pruning techniques to develop efficient and accurate models for CSI-based human activity recognition. Our experimental results confirmed that these approaches effectively identify architectures that are both small in size and high in accuracy, making them ideal candidates for practical deployments where computational resources are limited.

By employing BO to navigate the architectural search space, we uncovered architectures that outperform manually designed baselines, and through pruning, we further reduced their complexity without sacrificing performance. This integrated method offers a systematic way to find the most suitable model among numerous candidates, ensuring that even under real-world constraints, the chosen model operates optimally.

Looking ahead, we anticipate that this methodology can be extended to various other datasets and tasks in the wireless sensing domain, facilitating the adaptation of our approach to different conditions and requirements. Ultimately, this strategy paves the way for more efficient, reliable, and scalable WiFi-based sensing applications capable of handling increasingly complex scenarios and larger-scale data.

Author Contributions: Conceptualization, S.G. and S.Y.; methodology, S.G.; software, S.G.; validation, S.G. and S.Y.; formal analysis, S.G.; investigation, S.G.; resources, S.G.; data curation, S.G.;

writing—original draft preparation, S.G.; writing—review and editing, S.Y.; visualization, S.G.; supervision, S.Y.; project administration, S.Y.; funding acquisition, S.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Wonkwang University in 2023.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable

Data Availability Statement: The data presented in this study are available upon reasonable request from the corresponding author.

Acknowledgments: The authors thank the anonymous reviewers and editors for their insightful comments and suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, W.; Liu, A.X.; Shahzad, M.; Ling, K.; Lu, S. Understanding and Modeling of WiFi Signal Based Human Activity Recognition. In Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, Paris, France, 7–11 September 2015; pp. 65–76.
2. Zou, H.; Zhou, Y.; Yang, J.; Jiang, H.; Xie, L.; Spanos, C.J. DeepSense: Device-Free Human Activity Recognition via Autoencoder Long-Term Recurrent Convolutional Network. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
3. Yang, J.; Zou, H.; Zhou, Y.; Xie, L. Learning Gestures From WiFi: A Siamese Recurrent Convolutional Architecture. *IEEE Internet Things J.* **2019**, *6*, 10763–10772. [CrossRef]
4. Yang, J.; Hu, F.; Yu, W.; Zhang, J.; Li, S.; Zhu, M.; Lin, Y.; Xie, L. SenseFi: A Library and Benchmark on Deep-Learning-Empowered WiFi Human Sensing. *Patterns* **2023**, *4*, 100703. [CrossRef] [PubMed]
5. Ma, Y.; Zhou, G.; Wang, S. WiFi Sensing with Channel State Information: A Survey. *ACM Comput. Surv.* **2019**, *52*, 46. [CrossRef]
6. Zou, H.; Zhou, Y.; Yang, J.; Gu, W.; Xie, L.; Spanos, C. WiFi-Based Human Identification via Convex Tensor Shapelet Learning. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
7. Yousefi, S.; Narui, H.; Dayal, S.; Ermon, S.; Valaee, S. A Survey on Behavior Recognition Using WiFi Channel State Information. *IEEE Commun. Mag.* **2017**, *55*, 98–104. [CrossRef]
8. Wang, Z.; Guo, B.; Yu, Z.; Zhou, X. A Survey on Human Behavior Recognition Using Channel State Information. *IEEE Access* **2019**, *7*, 155986–156024. [CrossRef]
9. Zhang, J.; Tang, Z.; Li, M.; Fang, D.; Nurmi, P.; Wang, Z. CrossSense: Towards Cross-Site and Large-Scale WiFi Sensing. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, New Delhi, India, 29 October–2 November 2018; pp. 305–320.
10. Wang, X.; Gao, L.; Mao, S. CSI Phase Fingerprinting for Indoor Localization with a Deep Learning Approach. *IEEE Internet Things J.* **2016**, *3*, 1113–1123. [CrossRef]
11. He, Y.; Lin, J.; Liu, Z.; Wang, H.; Li, L.J.; Han, S. AMC: AutoML for Model Compression and Acceleration on Mobile Devices. In *Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer: Cham, Switzerland, 2018; pp. 815–832.
12. He, Y.; Zhang, X.; Sun, J. Channel Pruning for Accelerating Very Deep Neural Networks. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 1389–1397.
13. Han, S.; Pool, J.; Tran, J.; Dally, W. Learning both Weights and Connections for Efficient Neural Network. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 1135–1143.
14. Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; Zhang, C. Learning Efficient Convolutional Networks through Network Slimming. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2755–2763.
15. Cheng, Y.; Wang, D.; Zhou, P.; Zhang, T. A Survey of Model Compression and Acceleration for Deep Neural Networks. *IEEE Signal Process. Mag.* **2018**, *35*, 126–136. [CrossRef]
16. Han, S.; Mao, H.; Dally, W.J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In Proceedings of the International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.
17. Sze, V.; Chen, Y.H.; Yang, T.J.; Emer, J.S. Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proc. IEEE* **2017**, *105*, 2295–2329. [CrossRef]
18. Yang, J.; Chen, X.; Zou, H.; Wang, D.; Xu, Q.; Xie, L. EfficientFi: Toward Large-Scale Lightweight WiFi Sensing via CSI Compression. *IEEE Internet Things J.* **2022**, *9*, 13086–13095. [CrossRef]

19. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
20. Molchanov, P.; Tyree, S.; Karras, T.; Aila, T.; Kautz, J. Pruning Convolutional Neural Networks for Resource Efficient Inference. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
21. Pasandi, M.M.; Hajabdollahi, M.; Karimi, N.; Samavi, S. Modeling of Pruning Techniques for Deep Neural Networks Simplification. In Proceedings of the International Conference on Digital Image Processing, Shanghai, China, 12–14 May 2019; pp. 1–5.
22. Chen, J.; Chen, S.; Pan, S.J. Storage Efficient and Dynamic Flexible Runtime Channel Pruning via Deep Reinforcement Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 3790–3804.
23. Luo, R.; Tian, F.; Qin, T.; Chen, E.; Liu, T.-Y. Neural Architecture Optimization. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 3–8 December 2018; pp. 7827–7838.
24. Song, M.; Lou, L.; Chen, X.; Zhao, X.; Hong, Y.; Zhang, S.; He, W. Wi-LADL: A Wireless-Based Lightweight Attention Deep Learning Method for Human–Vehicle Recognition. *IEEE Sens. J.* **2023**, *23*, 2803–2814. [CrossRef]