BALKAN JOURNAL OF ELECTRICAL & COMPUTER ENGINEERING, Vol. 13, No. 1, March 2025
Research Article

Development of a Smart Activity Recognition System with Transfer Learning Based Deep Learning Models for Elderly Care

Mehmet Ilyas Bayindir and Fahri Cihan Attila

***Abstract*—In recent years, smart healthcare services have become popular in recent scientific research trends. Under elderly care topic, fall detection and activity recognition of elderly person living alone in their house or in a nursing home are vitally important solutions. Because falls are primary cause of most of the injuries, traumas, need of care and even deaths. To find a solution to this issue, scientists are start to use Artificial Intelligence. In this study, an intelligent activity recognition and fall detection system based on Convolutional Neural Network was developed. To develop this system an original dataset was created. By the proposed system, time distributions and classes of the activities are observed. When a fall is detected, the system gives an alert and warns relevant persons. The performances of used different models were compared using the dataset we created. To evaluate the performance of the systems, accuracy, precision, recall and F1 score metrics was used. For the ResNet101, these metrics are obtained as 98.66%, 98.54%, 98.78%, 98.66% respectively, that is the best of all scores. This results show that trained ResNet101 system can be used to help elderly persons and can be integrated to the other IoT systems.**

Index Terms— **Deep Learning, Elderly Care, Fall Detection, Human Activity Recognition, Internet of Things.**

I. Introduction

THE HEALTH conditions of elderly individuals are fragile, and they have a challenge for self-care due to underlying illnesses. The population of these individuals is growing, for instance, in the UK, it was identified that there were 2376 individuals aged 75 and above living alone [1], [2]. As cities turn to smart solutions under increasing population pressures, ensuring the safety and well-being of these individuals becomes a battleground.

Mehmet İlyas Bayındır, is with Department of Ecoinformatics, Institute of Natural and Applied Sciences, Fırat University, Elazığ, Turkey, (e-mail: mbayindir@firat.edu.tr)

https://orcid.org/0000-0003-1999-014X

Fahri Cihan Attila, is with Department of Ecoinformatics, Institute of Natural and Applied Sciences, Fırat University, Elazığ, Turkey, (e-mail: fahricihanattila@gmail.com)

https://orcid.org/0009-0002-8680-2591

Manuscript received Oct 24, 2024; accepted Jan 31, 2025.
DOI: 10.17694/bajece.1572976

The conditions of the pandemic have emphasized the significance of minimum contact while monitoring quarantined elderly and/or sick individuals. Elderly people living alone or receiving healthcare in care facilities can be monitored by using artificial intelligence to analyze camera stream. If camera surveillance is provided for solitary or care-dependent elderly individuals, monitoring can be carried out by analyzing activities from video. When any falls occur, emergency alerts are triggered, supportation is provided to healthcare security and reducing the workload of caregivers. Employing deep learning solutions within live video monitoring processes will swiftly and effectively support the maintenance of a healthy state and facilitate easier caregiving.

As stated in many studies of the World Health Organization [3], falls are common in the initial stages of most of the injuries, traumas that cause need of care, and even deaths [4]. This is a danger case not only for the elderly but also for individuals in working life [5]. If the fall event is determined with early warning and intervened early, the individual can receive more effective healthcare and the cost of the health insurance system can be reduced. If scene tracking and human activity recognition solutions used in elderly care, a caregiver can serve more patients.

Internet of Things (IoT) technologies are especially used by smart city services. Deep Learning (DL) solutions, particularly Convolutional Neural Network (CNN) architectures, can effectively process large datasets to make predictions, forecasts, and insights without the need for preprocessing tasks like feature extraction [6] . When real-time videos are classified by a deep learning network trained on a comprehensive dataset, vital contributions can be provided for smart healthcare and elderly care.

Three main classifications can be made for assessing systems about falls detection and monitoring activities of humans, as regarding the types of data and sensors employed. As illustrated in Fig. 1, these classifications contain (I) systems depending on wearable sensors, (II) camera based systems, and (III) systems incorporating environmental sensors. For the first class, examples include wearable gyroscopes, smartwatches, wristbands, belt clips, motion detectors, and similar devices. The second category predominantly employs RGB cameras and depth cameras and infrared cameras can also be mentioned. Lastly, the third category entails acoustic sensors, pressure and floor sensors, presence detectors, inertial sensors, microphones etc. [7].

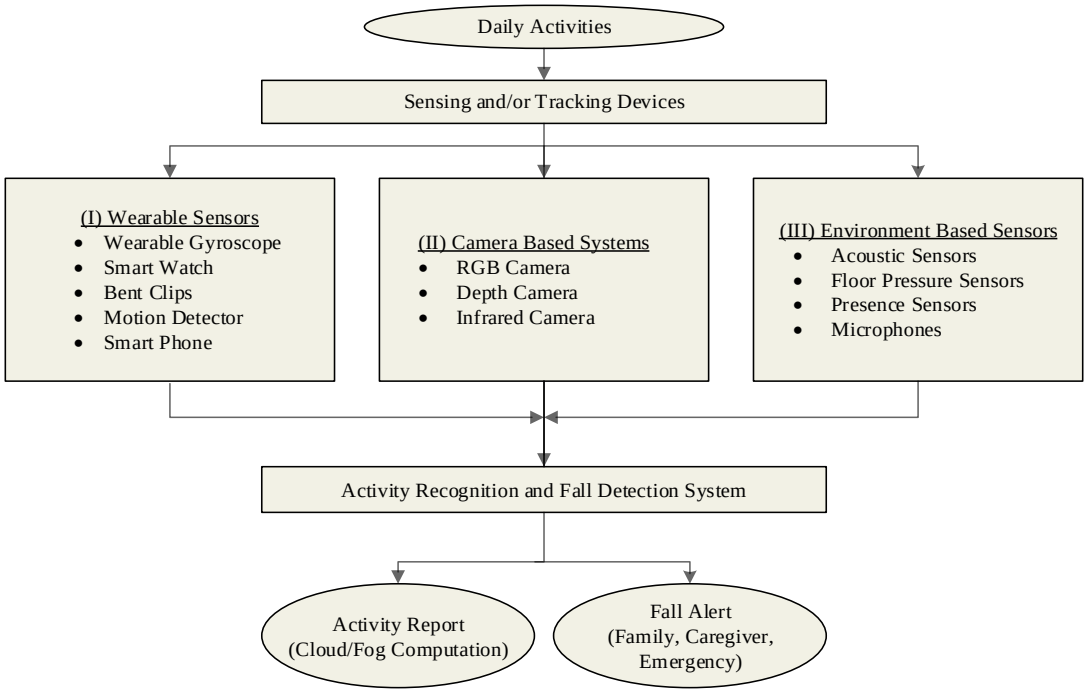

Fig. 1. Classification of fall detection approaches based on sensors

Perumal et al.[8] studied on fall detection systems developed for using in smart / non-smart home environment. They analyzed the most dangerous fall situation seen on wet floors. Different sensors, gateways and classification methods such as machine learning, rule-based, deep learning are investigated classification. They presented a scope especially for the researchers studying on the interoperability aspect of smart homes and activity recognition, especially fall detection systems in bathrooms/ toilets. Islam et al. [9] proposed a review of recent advances about fall detection systems based on deep learning. They are categorized as: CNN based systems, LSTM based systems, and Auto-encoder based systems. The most used method is CNN and 3D CNN or CNN with 10-fold cross-validation performed the best among the evaluated systems. Hardware technology have two categories: sensor based systems and vision based systems. The goal of the reviewed fall detection systems was to detect cases of elderly people falling using the best deep learning methods and to inform a nearby nurse or support/medical staff within a short time. Purwar, A. and Chawla [10] reviewed papers published in 2017-2023 on fall detection systems to protect elderly people. The papers contain both wearable, non-wearable systems and hybrid systems as regarding new technologies such as deep learning, computer vision, Internet of Things (IoT) and big data. The drawback of wearable sensors is that elderly people can forget to wear these devices. Zhou et al. [11] suggested using data from radar and optical cameras in a similar manner. Features extracted from these data by using Short-Time Fourier transform. They also used three different CNN models for feature extracting from radar signal. These are Alex-Net, a single-shot detector (SSD-Net) and a second SSD-Net. Maitre and Bouchard [12] introduced a new system to detect falls by using UWB radars and a CNN-LSTM architecture to use for indoor environment. They suggested that real-time fall detection performance should be studied in future works by using proper devices and communication protocol. Santos et al. [13] employed fog computing-enabled CNN to build a human fall detection system. They gathered training data from smartwatches and phones. The proposed CNN model was operated on a local fog device, capable of extracting appropriate features from the acquired data. Torti and colleagues [14] presented embedded software utilizing a Recurrent Neural Network (RNN) on the SensorTile wearable device, which incorporates a 3D accelerometer, gyroscope, magnetometer, and barometer. This software can detect the falls and then transmit alerts to a surveillance system via a wireless communication. Mauldin et al. [15] created an Android application. They used data from a smartwatch worn by the user. Their system is based on RNN model. This intelligent application is installed on a smartphone. In order to recognition instantaneously, computations are carried out directly on the smartphone. Another a CNN and LSTM architecture combination is also used for fall detection. The "ConvLSTM" model was developed by Nait Aicha et al. [16] by combining convolutional and recurrent models. Lie et al. [17] proposed the utilization of a Long Short-Term Memory (LSTM) classifier on human body joints for the characterization of various movements by using RGB sequence. Shojaei-Hashemi et al. [18] used transfer learning for human fall detection. This technique can identify human falls when they occur, extract features from camera depth maps, and send out an alert to notify families. Lu et al. [19] used 3D CNN and LSTM techniques. While the LSTM model is used to identify the interesting section of a frame, the 3D CNN is used to extract motion features from the temporal sequence of video sequences. Min et al. [20] introduced a deep CNN technique called Regions with CNN (R-CNN), which proves to be a more precise and expedient method for object detection in categorizing falling

incidents. The fundamental concept of this approach is to determine the distances between individuals and furniture within the scene. This method is implemented by extracting various characteristics like human body's center of mass, shape parameters, and velocity of motion. Adhikari et al. [21] presented a vision-based monitoring method to detect falls utilizing an CNN model. Additionally, they employed recurrent networks with time-sequential and mobile data to rapidly and accurately capture falls when they occur. Jiang et al. [22] proposed a real-time human fall detection method by using an infrared array sensors and a lightweight deep learning network. They used RetinexNet to enhance picture contrast and remove noise information in the dataset. Some improvements are made in the original YOLOv5 to reduce the complexity of model calculations and the calculation time. Their suggestion about future work is to improve the robustness of the model by adding additional images of various home scenarios to a larger dataset. Mobsite et al. [23] are introduced a system for activity recognition and fall detection with monocular depth and motion analysis. They used the Xception network to reduce the false alerts caused by non-human motion. Zi et al. [24] proposed a vision-based fall detection system. The proposed method integrates dual illumination estimation to the YOLOv7 + Deep SORT tracking algorithm to enhance fall detection performance under suboptimal lighting conditions. The performance of the proposed method is validated on two fall detection datasets, namely, Le2i FDD and UR-Fall datasets. In a recent study, Khekan et al. [25] are investigated fall detection applications based on deep learning especially for YOLOv1-YOLOv8. They provide a benchmark for future studies based on YOLO for human fall detection. The literature studies considered in this study generally used processed datasets. However, in our study, the proposed system's performance was tested on videos which are obtained from realized experimental setup.

In this study, an intelligent activity recognition and fall detection system based on Convolutional Neural Network was developed. To develop this system an original dataset was created. It is aimed to observe time distributions of the daily activities by the proposed system. Five class of activity are defined to recognize by the system. When the system detects a fall, the system gives an alert and warns caregivers or insurance services. These activities are classified by analyzing video frames captured by a general purpose camera. Transfer learned successful ten different CNN models employed as classifiers in this system. These models trained by created dataset. At the end of training process, proposed system is separately tested as regarding employed different CNN models. To evaluate the performance of the systems, accuracy, precision, recall and F1 score metrics was used. The performances of proposed systems were compared using the dataset we created. The best of all scores are obtained for the ResNet101. This results show that trained ResNet101 system can be used to help elderly persons and can be integrated to IoT systems.

The remainder of the paper is organized as follows: In Section 2, the proposed method and its theoretical background are explained. Section 3 provides a detailed discussion of the dataset used in the study, the experimental setup and evaluation metrics, as well as the experimental results. Finally, a general summary of the study is presented in Section 4, the conclusion.

II. MATERIAL AND METHODS

A. The Created Dataset

This study is aimed to develop an smart system based on a deep learning network that can operate in real-time through simple general-purpose cameras for elderly care in smart healthcare services. In order to train the network, an original dataset was created. To avoid bandwidth overload when used in cloud computing, the video format is based on the 320x240 QVGA basic resolution. The dataset is created from videos captured while subjects of different ages and genders perform natural movements in a setting resembling an elderly care/resting room. Six types of classifications commonly accepted in the literature are identified: Empty room, standing, sitting, sleeping, bending, and falling. The used labels and the number of frames in the dataset are listed in Table 1 sorted by numbers. Fig. 2 shows sample images from the dataset with six different classes.

B. Transfer Learning

Transfer learning involves reusing a pre-trained primary deep learning network, which has been successfully trained on a large dataset [26], by retraining it for a specific classification purpose. When there is not enough large data available, the transfer learning process can be highly beneficial. In this process, by default, the weights of the convolutional layers before the fully connected layer in the primary network structure, which goes from general to specific, are kept untouched, and training with the new dataset is started from this stage.

TABLE I.
LABELS AND NUMBERS OF DATA IN THE DATA SET

Class	Number of Images
Standing	5349
Empty room	2022
Falling	2835
Bending	2328
Sitting	3765
Sleeping	1173
TOTAL	17472

Fig. 2. Sample images of the categories in the data set

These weights can be frozen or re-adjusted during the training process. If the new dataset is not comprehensive enough and the weights are frozen, the problem of overfitting may be encountered. In this study, ten different deep learning architectures based on transfer learning are developed by the ImageNet dataset. These include ResNet50-ResNet101 [27], Xception [28], InceptionV3 [29], MobileNet [30], MobileNetV2 [31], DenseNet121-DenseNet201 [32] , EfficientNetB0 [33] and ConvNeXtTiny [34] methods.

1. Residual Network (ResNet): ResNet is a family of deep convolutional neural network architectures introduced by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in their 2015 paper titled 'Deep Residual Learning for Image Recognition' [27]. ResNet is known for its innovation in addressing the vanishing gradient problem, which frequently arises in very deep neural networks. The core idea behind ResNet is the use of residual connections, or skip connections, which allow information to flow more easily through the network. These skip connections essentially skip one or more layers, making it possible to train very deep networks. There are various variants of the ResNet architecture, primarily differing in depth, meaning the number of layers they contain. These include ResNet50, ResNet101, and ResNet152, described as follows:

ResNet50: A relatively shallow variant of the ResNet architecture. It has a total of 50 layers, including convolutional layers, batch normalization, and fully connected layers. The '50' refers to the total number of layers in the network. This model is widely used for a variety of computer vision tasks and is suitable for scenarios where computational capacity is limited.

ResNet101: A deeper variant of ResNet compared to ResNet50. It has a total of 101 layers, providing more depth and, in some cases, better performance. Deeper networks can capture more complex features and tend to perform better on challenging tasks, provided sufficient computational resources are available.

ResNet152: Even deeper than ResNet101, with a total of 152 layers. It is the deepest among the three variants mentioned. The increased depth allows ResNet152 to capture more complex and fine-grained features from input data. However, this also requires more computational resources and takes longer to train.

All these ResNet variants produce highly effective results in tasks like image classification, object detection, and other computer vision tasks. They are often used as a foundation for transfer learning, where pre-trained models on large image datasets are fine-tuned for specific applications. Researchers can take pre-trained ResNet models and adapt them to various image-related tasks, saving time and resources compared to training from scratch. The choice between ResNet50, ResNet101, and ResNet152 depends on the specific requirements of the task, available computational resources, and the desired level of model depth.

2. Xception: Xception is a deep convolutional neural network architecture introduced by François Chollet, the creator of the Keras deep learning library, in a research paper published in 2017 titled 'Deep Learning with Depthwise Separable Convolutions.' The name 'Xception' stands for 'Extreme Inception,' indicating its relationship with the Inception neural network architecture family [28]. Xception is particularly known for its innovative approach to convolutions, especially through the use of depthwise separable convolutions. Unlike traditional convolutions, which apply a single convolutional filter to the entire input image, Xception employs the depthwise separable convolution process. This process breaks down the traditional convolution into two steps: depthwise convolution and pointwise convolution. In depthwise convolution, a separate convolution operation is applied to each channel of the input image independently. This significantly reduces computational cost compared to traditional convolutions, where the same filter is applied to all input channels. After depthwise convolution, pointwise convolution, also known as 1x1 convolution, is applied to combine the output channels. This step helps capture complex patterns by allowing linear combinations of the output channels. The use of depthwise separable convolutions in Xception enables more efficient use of model parameters and computational resources. This architectural choice results in a highly impressive and efficient deep learning model. Xception also uses skip connections, similar to ResNet architectures, to mitigate the vanishing gradient problem.

The Xception neural network is an ESA architecture with a depth of 71 layers. The Xception architecture contains a total of 36 convolutional layers, which form the core of the feature extraction process of the network. These convolutional layers are grouped into 14 modules, making up a total of 36 layers. Except for the first and last modules, all modules have linear residual connections surrounding them. The Xception architecture consists of a linear stack of depthwise separable convolutional layers with residual connections. The Xception model achieved 94.5% accuracy on ImageNet, a dataset with over 14 million images across more than 1000 classes.

*3. InceptionV3:*InceptionV3 is a deep convolutional neural network architecture introduced as part of the Inception family, specifically in the 2015 paper "Rethinking the Inception Architecture for Computer Vision" by Google researchers [29]. It is a refined and more efficient version of its predecessors, such as InceptionV1 and InceptionV2, and is known for its effective balance between computational cost and accuracy in large-scale image classification tasks. One of the key innovations in InceptionV3 is the use of factorized convolutions, where larger convolutional filters are broken down into smaller, more efficient operations (for example, a 5x5 filter is factorized into a series of 3x3 convolutions). This reduces computational demands while preserving high performance. InceptionV3 also introduces a more efficient method for grid size reduction, replacing pooling layers with strided convolutions, which allows for better information flow in deeper layers while minimizing computational load. The model includes auxiliary classifiers during training, acting as regularizers to help the model converge faster and prevent overfitting by adding additional supervision to the middle layers. Asymmetric convolutions further boost efficiency by decomposing large convolutions, such as a 7x7, into smaller

operations like a 1x7 convolution followed by a 7x1 convolution, improving feature extraction with fewer parameters. Extensive use of batch normalization in InceptionV3 enhances training stability and accelerates convergence, while reducing overfitting risks. The model achieved remarkable success on the ImageNet dataset, with a top-5 error rate of 3.46%, making it one of the top-performing architectures for image classification. InceptionV3 is widely used in various computer vision tasks such as image classification, object detection, and image generation. It is also popular in transfer learning, where pre-trained weights are fine-tuned for specific applications. Overall, InceptionV3 stands out for its computational efficiency and accuracy, making it highly suitable for both research and real-world applications in computer vision.

4. MobileNet: MobileNet is a family of deep convolutional neural network architectures specifically designed for mobile and embedded vision applications, aimed at providing high performance with low computational cost. Introduced by Google researchers, MobileNet employs depthwise separable convolutions, which separate the convolutional operation into two stages: depthwise convolution, where individual filters are applied to each input channel, and pointwise convolution, which combines the outputs. This innovative approach significantly reduces the number of parameters and computational complexity compared to traditional convolutions. Additionally, MobileNet incorporates width and resolution multipliers that allow for fine-tuning the model's size and resource usage according to specific application requirements. The architecture is optimized for real-time performance on devices with limited processing power, making it suitable for tasks such as image classification, object detection, and face recognition. MobileNet models are also widely used in transfer learning, as pre-trained versions can be easily adapted for various specific tasks, ensuring a good balance between accuracy and efficiency in diverse mobile and embedded environments [30].

*5. MobileNetV2:*MobileNetV2 is an advanced version of the MobileNet architecture designed to further improve efficiency and performance for mobile and embedded vision applications. One of its key innovations is the introduction of inverted residuals and linear bottlenecks, which enhance feature extraction and reduce information loss. Inverted residuals allow for a lightweight convolutional operation where the input is first expanded into a higher-dimensional space using a pointwise convolution, followed by a depthwise separable convolution, and then reduced back to a lower-dimensional space with another pointwise convolution. This process helps in capturing more complex features while maintaining a small model size. MobileNetV2 also utilizes linear activation in the bottleneck layers instead of ReLU, which helps preserve information, especially in deeper layers. The architecture comprises an initial standard convolution layer, followed by a series of inverted residual blocks that alternate depthwise separable convolutions and pointwise convolutions, all concluded with a global average pooling layer and a fully connected layer for classification. This structure not only enhances the model's ability to learn intricate patterns but also maintains high efficiency, making MobileNetV2 highly suitable for real-time applications in constrained environments [31], [35].

6.DenseNet121-DenseNet201: DenseNet (Densely Connected Convolutional Networks) is a deep learning architecture that emphasizes feature reuse and efficient gradient flow through its unique connectivity pattern. Unlike traditional convolutional networks that connect each layer to its preceding layer, DenseNet connects each layer to every other layer in a feed-forward manner. This means that each layer receives feature maps from all preceding layers, promoting the flow of information and gradients throughout the network, which helps alleviate the vanishing gradient problem and reduces the number of parameters required. Key features of DenseNet include dense connections, which improve feature propagation, and bottleneck layers, which reduce dimensionality before applying convolution, thereby enhancing computational efficiency [32] .

DenseNet121: This model consists of 121 layers and is organized into four dense blocks, each followed by a transition layer that performs down sampling. The first block has a convolution layer followed by a batch normalization and ReLU activation. Each dense block consists of multiple convolutional layers where each layer takes input from all previous layers, culminating in a global average pooling layer and a fully connected layer for classification [32] [36] .

DenseNet201: Similar to DenseNet121, this model comprises 201 layers and follows the same architecture principles with more layers in each dense block, allowing it to capture more complex features. The additional layers enable deeper representations while maintaining the benefits of dense connectivity. Like DenseNet121, it concludes with a global average pooling layer and a fully connected layer, enhancing its performance in image classification tasks [32], [37].

Overall, DenseNet's innovative approach to layer connectivity significantly improves feature extraction and model efficiency, making it highly effective for various computer vision applications.

7. EfficientNetB0: EfficientNetB0 is the foundational model of the EfficientNet family, designed to balance accuracy and efficiency in image classification tasks. It introduces a novel compound scaling method, which systematically scales up the model's depth, width, and input resolution in a way that optimizes performance without excessive computational cost. This approach allows EfficientNetB0 to outperform many previous architectures while using significantly fewer parameters. Key features of EfficientNetB0 include its use of depthwise separable convolutions, which reduce the number of parameters and computational load by separating the spatial and channel-wise convolutions. Additionally, it employs the swish activation function, which provides better performance compared to traditional activation functions by enabling smoother gradients and improving convergence during training [33].

EfficientNetB0 comprises several essential layers structured to maximize efficiency and feature extraction. The model begins with a stem block that consists of a standard

convolutional layer. This layer captures initial features and reduces the spatial dimensions of the input image. The core of EfficientNetB0 consists of multiple MBConv blocks (Mobile Inverted Bottleneck Convolution). Each MBConv block utilizes depthwise separable convolutions, where a depthwise convolution is applied first to each channel, followed by a pointwise convolution to mix the outputs. This arrangement allows the model to efficiently capture both local and global features while keeping the computational requirements low. The blocks are arranged in a hierarchical fashion, with the number of filters and layers increasing as the network progresses, which enhances its capacity to learn complex patterns. Following the MBConv blocks, the architecture includes a global average pooling layer that aggregates the features across the spatial dimensions, effectively reducing the output to a single vector per feature map. This step significantly decreases the model's parameter count while retaining essential information. Finally, a fully connected layer processes the output from the global average pooling layer, producing class probabilities through a softmax activation function, making it suitable for multi-class classification tasks [33] .

In summary, EfficientNetB0 is a pioneering model that effectively balances performance and efficiency through its innovative architecture, making it highly suitable for various applications in computer vision. Its combination of depthwise separable convolutions, compound scaling, and an efficient layer structure allows it to achieve superior accuracy with reduced resource requirements.

8. ConvNeXtTiny: ConvNeXtTiny is a compact yet powerful convolutional neural network architecture that combines the principles of traditional convolutional networks with modern design elements inspired by transformer architectures. Key features of ConvNeXtTiny include its emphasis on simplicity and efficiency while maintaining high performance across various computer vision tasks. It employs a streamlined design that focuses on improved normalization techniques, such as layer normalization, and a novel architecture that utilizes depthwise separable convolutions to enhance computational efficiency. This model benefits from fewer parameters compared to larger models while still achieving competitive accuracy, making it ideal for resource-constrained environments [34].

The ConvNeXtTiny architecture consists of several key layers structured to optimize feature extraction and efficiency. The model begins with a stem block that processes the input image through a standard convolutional layer, reducing the spatial dimensions while extracting initial features. The core of ConvNeXtTiny consists of multiple ConvNeXt blocks, which are designed to efficiently learn hierarchical representations. Each block incorporates depthwise convolutions followed by pointwise convolutions, allowing for efficient computation. These blocks also feature layer normalization and an innovative activation function, such as GELU (Gaussian Error Linear Unit), which helps to model complex relationships within the data. After processing through the ConvNeXt blocks, the model uses a global average pooling layer that aggregates the features across the spatial dimensions, transforming the feature maps into a compact representation. The architecture concludes with a fully connected layer that outputs the final class probabilities through a softmax activation function, enabling the model to perform multi-class classification effectively. ConvNeXtTiny is a modern convolutional network that integrates elements from both traditional CNNs and transformer architectures to create a compact, efficient, and high-performing model for various vision tasks. Its thoughtful architecture, characterized by depthwise separable convolutions and improved normalization techniques, enables it to achieve strong performance while maintaining a lightweight footprint, making it suitable for deployment in applications with limited computational resources [34], [38] .

C. *Proposed Method*

The schema of the proposed method using pre-trained models for transfer learning is as shown in Fig.3. As shown in Fig.3, the input images are of size 320x240x3. Using these input images, training was carried out separately with ten different pre-trained network models based on transfer learning. The pre-trained network models used in this study are as follows: ResNet50, ResNet101, Xception, InceptionV3, MobileNet, MobileNetV2, EfficientNetB0, DenseNet121, DenseNet201, and ConvNeXtTiny.

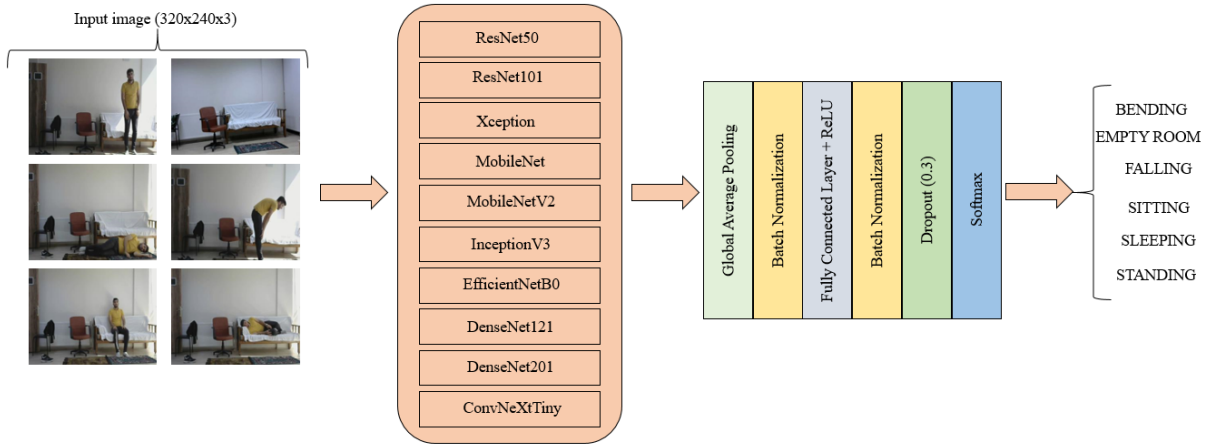

Fig. 3. The proposed method

After each pre-trained model, the output feature map is subjected to global average pooling (GAP), batch normalization, a fully connected layer with 128 neurons, batch normalization, and a dropout layer with a dropout rate of 0.3.

GAP is a technique commonly used in CNNs to reduce the spatial dimensions of feature maps before the final classification layer (softmax). Instead of using fully connected layers, GAP computes the average of all values in the feature map to obtain a single value for each feature map. The advantages of using GAP are as follows: (i) By reducing the number of parameters in the network, GAP helps to prevent overfitting, which is especially important when working with a limited amount of training data. It also acts as a regularization function. (ii) GAP is more efficient in terms of computation compared to fully connected layers, which require a large number of parameters and computation resources, particularly when dealing with high-resolution feature maps. GAP significantly reduces computational complexity. Additionally, GAP is simple to implement and does not require learning any additional parameters, which can simplify the training process. (iii) GAP requires less memory compared to fully connected layers, making it a good choice for resource-constrained applications such as mobile or embedded devices. (iv) Replacing fully connected layers with GAP layers is a common practice when fine-tuning pre-trained models, allowing the model to be adapted to a different task while preserving the knowledge learned during pre-training. When batch normalization is used in pre-trained network models, (i) it helps maintain the stability and performance of the model during fine-tuning on new tasks or datasets. (ii) It accelerates the training process by reducing the number of training iterations required for model convergence, leading to faster training. Thus, batch normalization is used to speed up and simplify the training process. The advantages of using a dropout layer in the proposed model are as follows: Dropout serves as an effective regularization technique to prevent overfitting. By randomly deactivating a portion of the neurons during training, it helps the model generalize better to new tasks or datasets. This randomness caused by dropout reduces the dependency on certain learned features, making the model more adaptable and robust during fine-tuning. As a result, dropout can improve the model's ability to generalize across different data distributions and enhance its performance in transfer learning scenarios. After the GAP, batch normalization, fully connected and dropout layers, the resulting feature map is passed to the input of the softmax classifier for classification. Softmax is used as the final layer in the neural network for classification tasks. The softmax classifier takes a vector of arbitrary real-valued scores as input and converts them into a probability distribution over multiple classes. Softmax transforms the input scores into positive values by exponentiating them and then normalizes these values by dividing them by the sum of all the exponentiated scores. This normalization ensures that the resulting values are between 0 and 1 and that their sum equals 1, representing probabilities. The class with the highest probability is then predicted as the output class.

III. Experimental Results and Discussion

A. Experimental Setup

In the experimental studies, the Python programming language was utilized. All Python code was implemented on the Kaggle platform. The GPU P100 was used as the execution environment for training, validating, and testing the transfer learning models on the Kaggle platform. The hyperparameters used are as follows: a batch size of 64, a learning rate of 0.001, and a total of 50 epochs. The Adam optimization method was employed. The input image dimensions were set to 320x240x3. Sparse Categorical Crossentropy was used as the loss function during the training of the transfer learning models. Sparse Categorical Crossentropy is a highly advantageous loss function in multi-class classification tasks, particularly when target labels are represented as integers rather than one-hot encoded vectors. One of its primary benefits is reduced memory usage, as it eliminates the need to create a one-hot encoding of the target labels, which can be particularly advantageous in scenarios with a large number of classes, where one-hot encoding would require significant additional memory. This efficiency simplifies the data preprocessing pipeline, allowing for direct use of integer labels, which streamlines the workflow and reduces the potential for errors during data transformation. Furthermore, Sparse Categorical Crossentropy enhances interpretability, as it maintains the original labeling scheme without necessitating any transformation into binary formats, making it easier for practitioners to understand and debug model predictions. This loss function is also well-suited for applications with large class sets, as it effectively manages extensive label spaces without the overhead associated with one-hot encoding, ultimately leading to improved computational performance. Overall, Sparse Categorical Crossentropy offers a memory-efficient, simplified, and intuitive approach to handling multi-class classification tasks, making it a popular choice in modern deep learning frameworks. Training-testing separation was set to 70-30% in all models. In other words, a total of 17472 images were used, with 12231 images for training and 5241 images for testing.

B. Performance Metrics of Classification

Confusion matrix allows comparison of predicted classifications versus true classifications (Fig. 4). At the end of the training process, a confusion matrix is created with the classification results obtained from the test data. Performance evaluation is carried out using standard formulations based on this matrix. In the confusion matrix, the correctly predicted classifications of the trained network on the test data (not used in training) are defined as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP is the number of inputs that are actually positive and predicted as positive, FN is the number of inputs that are actually positive but predicted as negative, TN is the number of inputs that are actually negative and predicted as negative, and FP is the number of inputs that are actually negative but predicted as positive. After creating the confusion matrix, the TP value is found as the sum of the numbers in the diagonal cells.

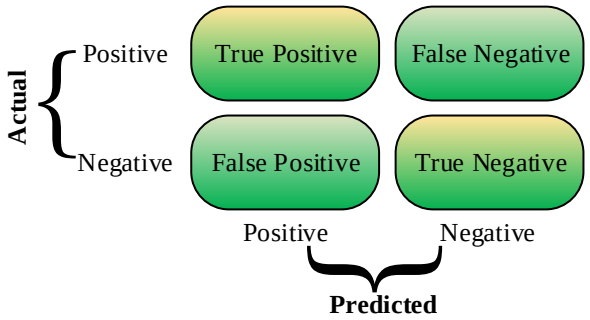

Fig. 4. Basic confusion matrix structure

TABLE II.
PERFORMANCE METRICS AND THEIR EXPRESSIONS

Performance metric	Formula
Sensivity (Recall)	SEN = TP / (TP + FN)
Precision	PRE = TP / (TP + FP)
Accuracy	ACC = (TP + TN) / (P + N)
F1 Score	F1 = 2TP / (2TP + FP + FN)

The TN value is the sum of the numbers in cells other than the row and column for that classification. The FP value is the sum of the numbers along the row, excluding the diagonal. The FN value is the sum of the numbers along the column, excluding the diagonal. Then, performance metrics such as sensitivity or recall, precision or positive predictive values, accuracy, and F1 Score can be calculated. Formulations of these metrics presented in Table 2.

C. Results of Experimental Studies

The confusion matrices obtained for all models are given in Fig.5 and Fig.6. The confusion matrix for ResNet50 shows strong classification performance, with minimal misclassifications across categories. The majority of instances are correctly classified, which is reflected in its high accuracy and balanced precision, recall, and F1-score values (Accuracy: 98.28%, F1: 98.34%). The confusion matrix for ResNet101 indicates even fewer misclassifications compared to ResNet50, suggesting that this deeper architecture is able to distinguish between classes more effectively. This is supported by its higher recall (98.78%) and accuracy (98.66%). The Xception model's confusion matrix shows slightly more misclassifications compared to ResNet models. Its lower precision (97.37%) could be due to false positives in certain classes, affecting its ability to maintain a balance between precision and recall (F1: 97.70%). The confusion matrices for MobileNet and MobileNetV2 show more noticeable errors. MobileNet's accuracy (96.70%) and MobileNetV2's lower accuracy (95.92%) suggest higher rates of misclassification. InceptionV3's confusion matrix reflects its relatively good performance, with accuracy of 97.99%. EfficientNetB0's confusion matrix is expected to show minimal misclassifications, reflecting its high accuracy (98.32%) and recall (98.60%). This model is effective at capturing relevant features while maintaining high precision. The confusion matrices for DenseNet121 and DenseNet201 are expected to be similar, with both models showing high accuracy and minimal errors. DenseNet121 performs slightly better, as indicated by its higher F1-score (98.25%) and accuracy (98.17%). ConvNeXtTiny's confusion matrix is among the best model, with few errors across all categories, reflected in its high accuracy (98.35%) and strong balance across precision (98.17%), recall (98.46%), and F1-score (98.31%). Overall, the confusion matrices indicate that deeper models like ResNet101, EfficientNetB0, and ConvNeXtTiny provide superior classification performance with fewer errors. Shallower or lighter models such as MobileNet and MobileNetV2 show more misclassifications, likely due to their reduced capacity to capture complex patterns.

In Table 3, we compare the performance of various deep learning models based on transfer learning for the classification task. The models compared include ResNet50, ResNet101, Xception, MobileNet, MobileNetV2, InceptionV3, EfficientNetB0, DenseNet121, DenseNet201, and ConvNeXtTiny. The classification performance metrics used are accuracy, precision, recall, and F1-score, as shown in Table 3. ResNet50 achieved a high classification accuracy of 98.28%, with precision and recall values of 98.42% and 98.27%, respectively. The F1-score, which balances precision and recall, is 98.34%. ResNet101 performed slightly better, with an accuracy of 98.66%, precision of 98.54%, recall of 98.78%, and an F1-score of 98.66%. The higher recall in ResNet101 indicates better detection of relevant instances compared to ResNet50, which slightly improves overall performance. The Xception model demonstrated slightly lower performance with an accuracy of 97.73%. Precision (97.37%) and recall (98.08%) values were slightly unbalanced, leading to an F1-score of 97.70%. The model still performs effectively but falls short compared to some of the other models, particularly in terms of precision. MobileNet achieved an accuracy of 96.70%, with precision and recall values of 96.63% and 96.81%, respectively, resulting in an F1-score of 96.72%. MobileNetV2, however, showed slightly lower accuracy (95.92%), with similar precision (96.12%) and recall (95.77%). The F1-score of MobileNetV2 was 95.93%, suggesting that this model has slightly lower performance compared to MobileNet. Despite their lightweight architecture, MobileNet-based models deliver decent performance but lag behind the deeper networks in this study. InceptionV3 provided a balanced performance with an accuracy of 97.99%, precision of 97.92%, and recall of 98.07%, resulting in an F1-score of 97.99%. The high recall value reflects the model's ability to correctly classify most of the relevant instances, positioning it competitively among the other models. EfficientNetB0 performed well across all metrics, achieving an accuracy of 98.32%, precision of 98.23%, recall of 98.60%, and an F1-score of 98.39%. Its high recall indicates that the model effectively captured relevant instances, while maintaining a strong balance with precision, making it one of the best-performing models. DenseNet121 and DenseNet201 performed similarly, with DenseNet121 having a slightly higher accuracy (98.17%) compared to DenseNet201 (98.13%). The F1-scores for both models are also close, with 98.25% for DenseNet121 and 98.16% for DenseNet201. DenseNet121 exhibited a marginally better balance between precision (97.97%) and recall (98.56%), indicating that it might generalize slightly better than DenseNet201.

ConvNeXtTiny achieved one of the highest performances among all models, with an accuracy of 98.35%, precision of 98.17%, recall of 98.46%, and an F1-score of 98.31%. The model's strong results across all metrics demonstrate its potential in delivering accurate and robust classifications. Overall, the results show that ResNet101, EfficientNetB0, and ConvNeXtTiny consistently outperform other models in terms of accuracy, precision, recall, and F1-score. ResNet101, in particular, achieved the highest recall, accuracy, precision and F1-score, making it the most reliable model for this task. EfficientNetB0 also stands out due to its efficient architecture and high performance across all metrics. ConvNeXtTiny, with its strong balance of metrics, proves to be a competitive model as well. Models like Xception and MobileNet series, while still effective, lag behind the top-performing models in terms of classification accuracy.

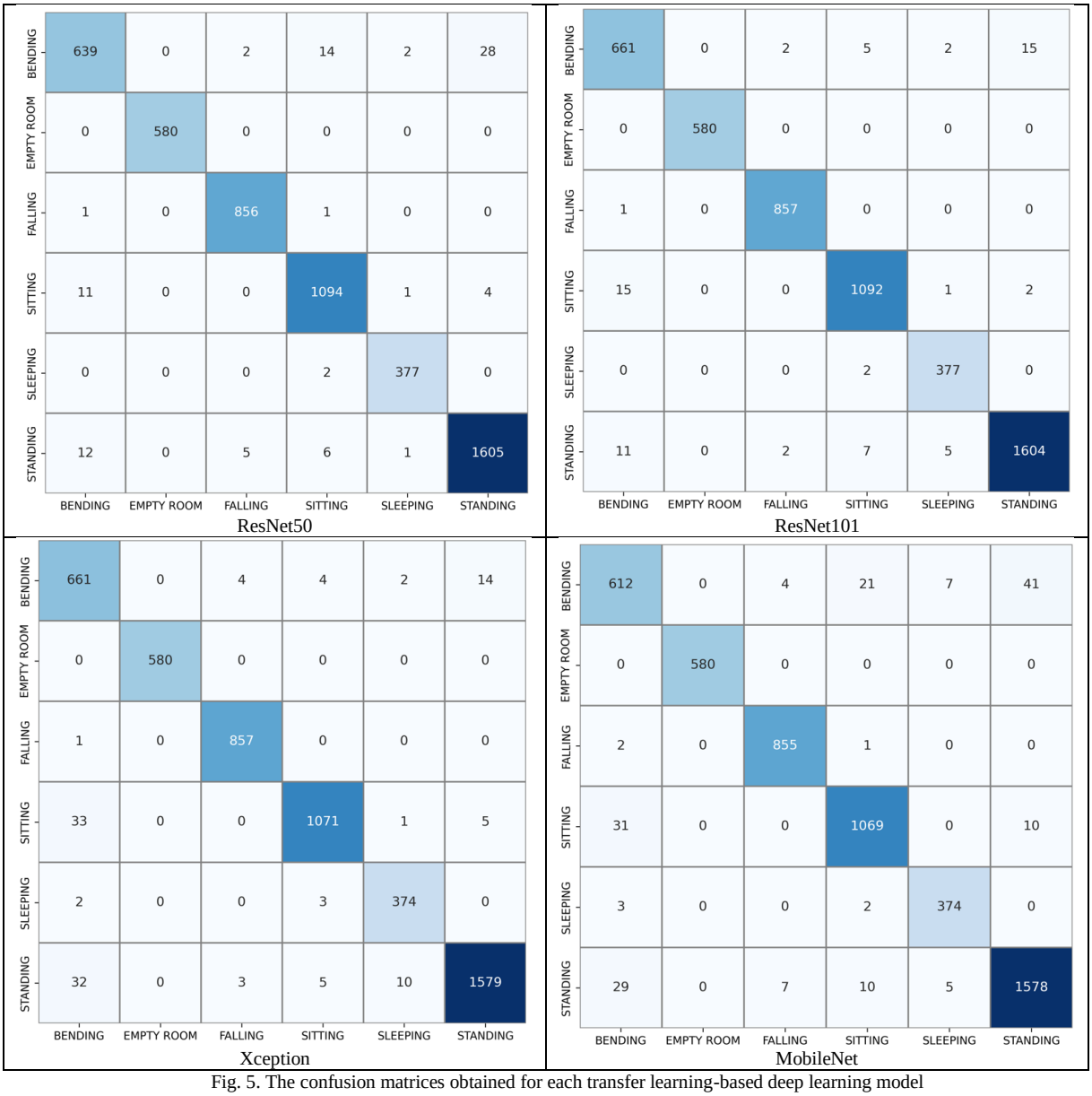

Fig. 5. The confusion matrices obtained for each transfer learning-based deep learning model

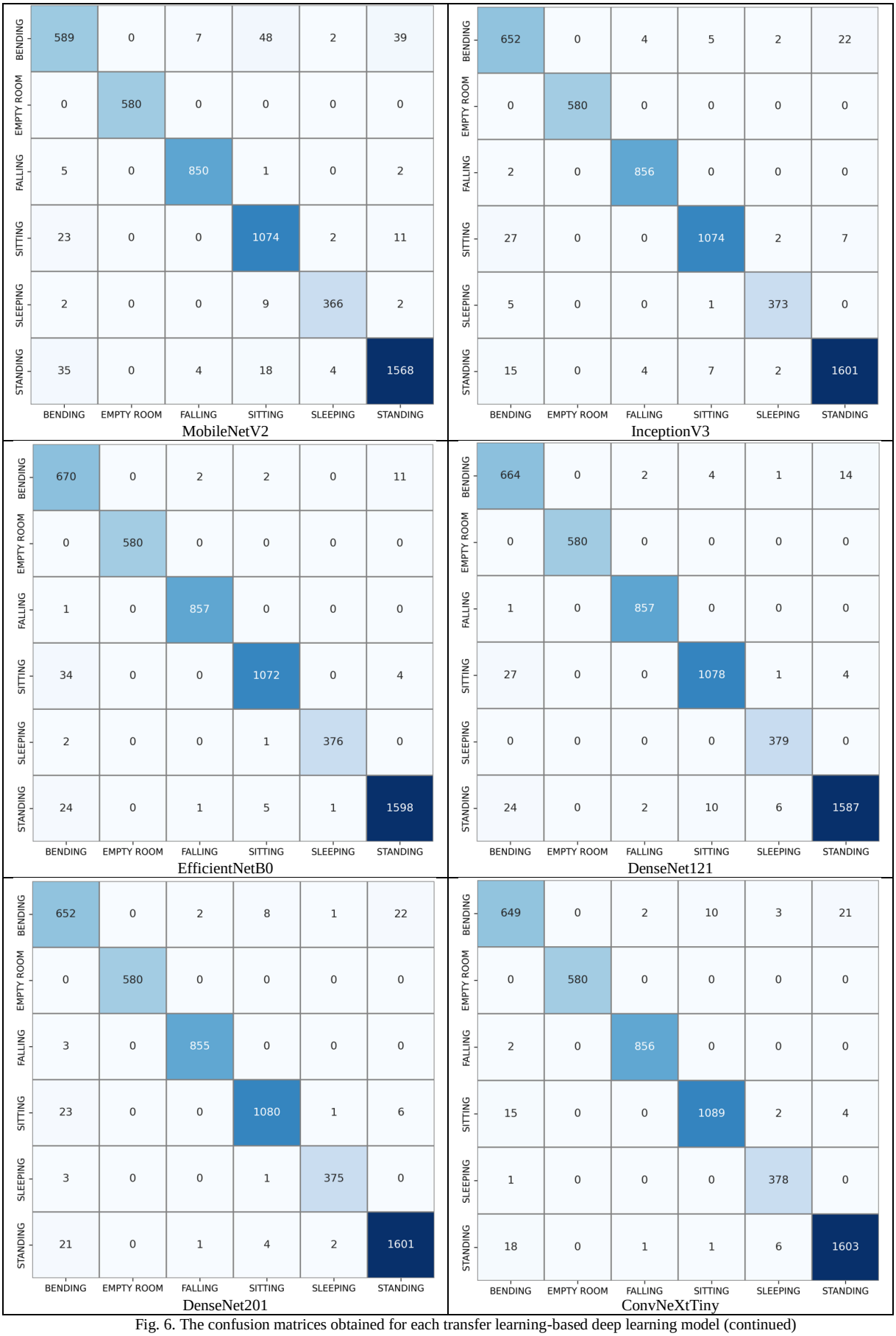

Fig. 6. The confusion matrices obtained for each transfer learning-based deep learning model (continued)

TABLE III.
CLASSIFICATION RESULTS FOR EACH MODEL (%)

Model	Accuracy	Precision	Recall	F1-score
ResNet50	98.28	98.42	98.27	98.34
ResNet101	**98.66**	**98.54**	**98.78**	**98.66**
Xception	97.73	97.37	98.08	97.70
MobileNet	96.70	96.63	96.81	96.72
MobileNetV2	95.92	96.12	95.77	95.93
InceptionV3	97.99	97.92	98.07	97.99
EfficientNetB0	98.32	98.23	98.60	98.39
DenseNet121	98.17	97.97	98.56	98.25
DenseNet201	98.13	98.09	98.23	98.16
ConvNeXtTiny	98.35	98.17	98.46	98.31

IV. CONCLUSION

A smart system for activity recognizing and fall detection has been developed for use as an elderly care application within the scope of smart healthcare services. This intelligent activity recognition and fall detection system is based on CNN type of deep learning. The system is aimed to make smart healthcare services more efficient through big data analytics. An original dataset is created with frame snapshots selected from videos recorded in an experimental environment. With the purpose of real-time video processing, this CNN based system used for reporting the duration of specific activities and whether a person in a room has fallen during live video capture.

The transfer learned networks are expected to yield higher performances as they have been previously trained on a very large dataset and have more advanced architectures. Ten successful CNN model used for transfer learning: ResNet50, ResNet101, Xception, InceptionV3, MobileNet, MobileNetV2, DenseNet121-DenseNet201, EfficientNetB0 and ConvNeXtTiny. The training performance metrics of the networks regarding test data are found very successful. These CNN models employed as classifiers in this system. After training process using by created dataset, proposed system is separately tested for each employed different CNN models. To evaluate the performance of the systems, accuracy, precision, recall and F1 score metrics was used. For the ResNet101, these metrics are obtained as 98.66%, 98.54%, 98.78%, 98.66% respectively, that is the best of all scores. This results show that trained ResNet101 system can be used to help elderly persons and can be integrated to the other IoT systems.

Application performance of the designed networks can be raised by training with a larger dataset. In an extended dataset that includes different features, it is suggested to add falling and semi-falling situations as additional classification categories. Additionally, to enhance practical performance, the addition of a logical error elimination process or a Long Short-Term Memory (LSTM) classifier for the flow of successive activities is recommended to distinguish between intentional sitting and unintentional falling events.

Funding and Ethics Permission Disclosure: This study was supported by the Scientific Research Projects Coordination Unit of Fırat University (FÜBAP) with project number ADEP.22.06.

The study was conducted with the approval of the Ethics Committee for Social and Human Sciences Research at Fırat University, at 03/03/2023 with approval number 9.

Data availability: The dataset created in this study and experiment videos are available in the GitHub - fahricihan/FALLDETECTION: FALL DETECTION MATLAB

REFERENCES

[1] S. B. Atitallah, M. Driss, W. Boulila, and H. B. Ghézala, "Leveraging Deep Learning and IoT big data analytics to support the smart cities development: Review and future directions," Comput. Sci. Rev., vol. 38, p. 100303, Nov. 2020, doi: 10.1016/j.cosrev.2020.100303.

[2] "Statista, UK: people living alone, 2019, https://www.statista.com/statistics/281616/people-living-alone-in-the-united-kingdom-uk-by- age-and-gender/ (Accessed 5 December 2023).".

[3] "World Health Organization, Falls." [Online]. Available: World Health Organization https://www.who.int/news-room/fact-sheets/detail/falls

[4] "Global Pharma News & Resources." [Online]. Available: https://www.pharmiweb.com/press-release/2020-05-05/fall-detection-system-market-to-register-cagr-4-growth-in-revenue-during-the-forecast-period-2019-t

[5] Bureau of Labor Statistics (2023) https://data.bls.gov/timeseries/FWU00X4XXXXX8EN00

[6] M. A. Khan, F. Algarni, and M. T. Quasim, Smart Cities: A Data Analytics Perspective. Springer, 2021.

[7] N. Zerrouki, F. Harrou, Y. Sun, A. Z. A. Djafer, and H. Amrane, "A Survey on Recent Advances in Fall Detection Systems Using Machine Learning Formalisms," in 2022 7th International Conference on Frontiers of Signal Processing (ICFSP), IEEE, 2022, pp. 35–39.

[8] R. E, T. Perumal, and S. K, "A review on fall detection systems in bathrooms: challenges and opportunities," Multimed. Tools Appl., Jan. 2024, doi: 10.1007/s11042-023-18088-6.

[9] Md. M. Islam et al., "Deep Learning Based Systems Developed for Fall Detection: A Review," IEEE Access, vol. 8, pp. 166117–166137, 2020, doi: 10.1109/ACCESS.2020.3021943.

[10] A. Purwar and I. Chawla, "A systematic review on fall detection systems for elderly healthcare," Multimed. Tools Appl., vol. 83, no. 14, pp. 43277–43302, Oct. 2023, doi: 10.1007/s11042-023-17190-z.

[11] X. Zhou, L.-C. Qian, P.-J. You, Z.-G. Ding, and Y.-Q. Han, "Fall detection using convolutional neural network with multi-sensor fusion," in 2018 IEEE international conference on Multimedia & Expo Workshops (ICMEW), IEEE, 2018, pp. 1–5.

[12] J. Maitre, K. Bouchard, and S. Gaboury, "Fall Detection With UWB Radars and CNN-LSTM Architecture," IEEE J. Biomed. Health Inform., vol. 25, no. 4, pp. 1273–1283, Apr. 2021, doi: 10.1109/JBHI.2020.3027967.

[13] G. L. Santos, P. T. Endo, K. H. de C. Monteiro, E. da S. Rocha, I. Silva, and T. Lynn, "Accelerometer-based human fall detection using convolutional neural networks," Sensors, vol. 19, no. 7, p. 1644, 2019.

[14] E. Torti et al., "Embedded real-time fall detection with deep learning on wearable devices," in 2018 21st euromicro conference on digital system design (DSD), IEEE, 2018, pp. 405–412.
[15] T. R. Mauldin, M. E. Canby, V. Metsis, A. H. Ngu, and C. C. Rivera, "SmartFall: A smartwatch-based fall detection system using deep learning," Sensors, vol. 18, no. 10, p. 3363, 2018.
[16] A. Nait Aicha, G. Englebienne, K. S. Van Schooten, M. Pijnappels, and B. Kröse, "Deep learning to predict falls in older adults based on daily-life trunk accelerometry," Sensors, vol. 18, no. 5, p. 1654, 2018.
[17] W.-N. Lie, A. T. Le, and G.-H. Lin, "Human fall-down event detection based on 2D skeletons and deep learning approach," in 2018 International workshop on advanced image technology (IWAIT), IEEE, 2018, pp. 1–4.
[18] A. Shojaei-Hashemi, P. Nasiopoulos, J. J. Little, and M. T. Pourazad, "Video-based human fall detection in smart homes using deep learning," in 2018 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, 2018, pp. 1–5.
[19] N. Lu, Y. Wu, L. Feng, and J. Song, "Deep learning for fall detection: Three-dimensional CNN combined with LSTM on video kinematic data," IEEE J. Biomed. Health Inform., vol. 23, no. 1, pp. 314–323, 2018.
[20] W. Min, H. Cui, H. Rao, Z. Li, and L. Yao, "Detection of human falls on furniture using scene analysis based on deep learning and activity characteristics," IEEE Access, vol. 6, pp. 9324–9335, 2018.
[21] K. Adhikari, H. Bouchachia, and H. Nait-Charif, "Deep learning based fall detection using simplified human posture," Int J Comput Syst Eng, vol. 13, no. 5, pp. 255–260, 2019.
[22] Y. Jiang, T. Gong, L. He, S. Yan, X. Wu, and J. Liu, "Fall detection on embedded platform using infrared array sensor for healthcare applications," Neural Comput. Appl., vol. 36, no. 9, pp. 5093–5108, Mar. 2024, doi: 10.1007/s00521-023-09334-x.
[23] S. Mobsite, N. Alaoui, M. Boulmalf, and M. Ghogho, "Activity Classification and Fall Detection Using Monocular Depth and Motion Analysis," IEEE Access, vol. 12, pp. 1525–1541, 2024, doi: 10.1109/ACCESS.2023.3348413.
[24] X. Zi, K. Chaturvedi, A. Braytee, J. Li, and M. Prasad, "Detecting Human Falls in Poor Lighting: Object Detection and Tracking Approach for Indoor Safety," Electronics, vol. 12, no. 5, p. 1259, Mar. 2023, doi: 10.3390/electronics12051259.
[25] A. R. Khekan, H. S. Aghdasi, and P. Salehpour, "The Impact of YOLO Algorithms Within Fall Detection Application: A Review," IEEE Access, vol. 13, pp. 6793–6809, 2025, doi: 10.1109/ACCESS.2024.3496823.
[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.
[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[28] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1800–1807. doi: 10.1109/CVPR.2017.195.
[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.
[30] A. G. Howard et al., "MobileNets: efficient convolutional neural networks for mobile vision applications (2017)," ArXiv Prepr. ArXiv170404861, vol. 126, 2017.
[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.
[32] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
[33] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in International conference on machine learning, PMLR, 2019, pp. 6105–6114.
[34] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 11976–11986.
[35] H. Fırat and H. Üzen, "Detection of Pneumonia Using A Hybrid Approach Consisting of MobileNetV2 and Squeeze-and-Excitation Network," Türk Doğa Ve Fen Derg., vol. 13, no. 1, pp. 54–61, 2024.
[36] A. Bello, S.-C. Ng, and M.-F. Leung, "Skin Cancer Classification Using Fine-Tuned Transfer Learning of DENSENET-121," Appl. Sci., vol. 14, no. 17, p. 7707, 2024.
[37] Ü. Hüseyin and H. FIRAT, "ÖZNİTELİK ENTEGRASYONUNA DAYALI ESA MİMARİSİ KULLANILARAK ENDOSKOPİK GÖRÜNTÜLERİN SINIFLANDIRILMASI," Kahramanmaraş Sütçü İmam Üniversitesi Mühendis. Bilim. Derg., vol. 27, no. 1, pp. 121–132, 2024.
[38] A. R. KP and S. Gowrishankar, "Convnext-based mango leaf disease detection: Differentiating pathogens and pests for improved accuracy," Int. J. Adv. Comput. Sci. Appl., vol. 14, no. 6, 2023.

BIOGRAPHIES

Mehmet İlyas Bayındır received the B.S., the M.S. and the Ph.D. degrees in electrical and electronics engineering from Fırat University, Elazığ, Turkey, in 1992, 1996, 2003 respectively. He worked in Şanlıurfa Vocational High School in Harran University as lecturer in 1993-1998 years. He has been worked in Technical Sciences Vocational High School as assistant professor since 1998. He continues his research studies at the Department of Ecoinformatics, Institute of Natural and Applied Sciences, Fırat University, Elazığ, Türkiye.

Fahri Cihan Attila was born in Elazığ in 1998. He received his bachelor's degree from Fırat University, Department of Electrical and Electronics Engineering in 2016. He received Master's degree from Ecoinformatics department in the Institute of Natural and Applied Sciences at Fırat University in 2023. In 2021, he worked as a Telecommunications engineer at Geoplus Solutions Engineering Inc in Elazığ. He works as a Senior Fixed Network Operations and Infrastructure Planning Specialist at Turkcell Communication Services. He specialized in developing and optimizing telecommunications infrastructures.