Published by
NIGERIAN SOCIETY OF PHYSICAL SCIENCES
Available online @ https://asr.nsps.org.ng/index.php/asr

African Scientific Reports 5 (2026) 645

African Scientific Reports

Human activity recognition with a fine-tuned CLIP vision transformer and EfficientNet-B0 baseline

Aanjan Kumar[a], Midhunchakkaravarthy Janarthanan[a,*], Ganesh Khekare[b]

[a]*School of AI Computing and Multimedia, Lincoln University College, Kota Bharu, Kelantan, Malaysia*
[b]*School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India*

Abstract

Image-based human activity recognition (HAR) is a building block for camera-based monitoring of daily routines, but reported accuracies are often inflated by train–test leakage. This study evaluates 15-class HAR from single RGB images under a leakage-controlled protocol: 12,600 labelled images were grouped by perceptual-hash near-duplicate detection and split into training, validation and test sets (9,005/1,796/1,799) before any augmentation, and augmentation was applied to training batches only. A contrastive language–image pre-training (CLIP) vision transformer (ViT)-B/16 backbone fine-tuned with layer-wise learning-rate decay, exponential-moving-average (EMA) weights and label smoothing reached 91.44% test accuracy (macro-F1 91.45%, macro one-vs-rest area under the curve (AUC) 99.55%) and 91.53 ± 0.74% in five-fold group-stratified cross-validation. A lightweight EfficientNet-B0 baseline evaluated on the same split reached 80.93% accuracy with 4.03 M parameters and 0.77 billion floating-point operations (GFLOPs), compared with 85.81 M parameters and 35.13 GFLOPs for the ViT. On an NVIDIA Tesla T4 the ViT required 7.39 ms per image (batch size 1, FP16) and approximately 0.22 J per image at a measured whole-board GPU power of about 65.9 W. The results characterise generic activity recognition which could become one component of a future, clinically validated daily-routine monitoring system.

DOI: 10.46481/asr.2026.5.3.645

Keywords: Human activity recognition, CLIP, Vision transformer, EfficientNet, Data leakage

Article History:
Received: 26 August 2026
Received in revised form: 29 October 2026
Accepted for publication: 05 November 2026
Available online: 23 November 2026

© 2026 The Author(s). Published by the Nigerian Society of Physical Sciences under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

1. Introduction

Alzheimer's disease is a neurodegenerative disorder that causes daily routine activities of affected individuals to degenerate, leading to severe cognitive impairment. According to the Alzheimer's Association report, an estimated 7.4 million people aged 65 years and older are living with Alzheimer's disease. People with this disease face difficulties in performing daily tasks such as talking with family members, walking, sleeping, and using electronic gadgets. Healthcare professionals manually assess these activities using questionnaires and caregiver feedback. Clinical staging relies on validated instruments such as the Mini-Mental State examination (MMSE) and the Clinical Dementia Rating (CDR) together with imaging; manual observation of daily activities is

*Corresponding Author Tel. No.: +60-378-063-478.

Email addresses: aanjansrrcet.90@gmail.com (Aanjan Kumar), midhun@lincoln.edu.my (Midhunchakkaravarthy Janarthanan), khekare.123@gmail.com (Ganesh Khekare)

Table 1: Comparative analysis of existing techniques.

Reference	Methods	Dataset Type	Strength	Summary
Ref. [5]	Deep CNN+ Transformer	COCO; human pose estimation and running-pose measurement	Effective Pose detection	Few activity classes
Ref. [6]	Support Vector Machine	Clinical data; depression and quality-of-life prediction	Good accuracy on behaviour analysis	Features depend on Manual allocation
Ref. [7]	Ensemble CNN	3D MRI; AD diagnosis	Effective classification of disease progression	Very high computational cost
Ref. [8]	Evolutionary CNN	MRI; AD detection (federated learning)	Fast prediction of disease progression	Very high training time
Ref. [9]	Autoencoder + Graph Network	Resting-state fMRI; multi-label AD stage classification	Multiple outputs, including prediction of Alzheimer's disease stage	Limited real-time deployment due to high computational cost
Ref. [10]	Open Pose + Graph Neural Network	Video; workers' compliance with safety regulations	Human activity recognition through video analysis	Limited generalization

time-consuming and subjective. To overcome this limitation, an artificial intelligence (AI) model could assist in observing routine activities, but such a tool must first be shown to recognise activities reliably before it can be linked to any clinical outcome according to Ref. [1]

Camera-based recognition of everyday activities such as eating, drinking, sitting or sleeping is a prerequisite for any vision-based routine-monitoring tool, including tools intended for people with dementia. Before such a tool can be linked to disease severity, its activity-recognition component has to be shown to work under an evaluation that is free of data leakage, and its computational cost has to be measured in a reproducible way. Both requirements are frequently unmet: benchmark images are often duplicated or augmented before splitting, and all computational cost figures are often estimated rather than measured.

This research therefore studies 15-class human activity recognition from single RGB images on the public Human Action Recognition (HAR) dataset according to Ref. [2]. The dataset contains no longitudinal data or daily schedule. This study provides (i) a leakage-controlled evaluation protocol, (ii) a ViT architecture, initialized from the CLIP image encoder according to Ref. [3] and (iii) a lightweight EfficientNet-B0 baseline according to Ref. [4], together with measured parameters, GFLOPs, latency and GPU power for both models.

1.1. Contributions of this work

The primary contributions of this work are:

A leakage-controlled protocol for the 15-class HAR image dataset: perceptual-hash near-duplicate grouping, group-stratified train/validation/test splitting before augmentation, and online augmentation restricted to training batches.

A fine-tuned CLIP ViT-B/16 classifier (layer-wise learning-rate decay, EMA weights, label smoothing) that reaches 91.44% held-out accuracy and 91.53 ± 0.74% in five-fold group-stratified cross-validation.

An efficiency comparison with an EfficientNet-B0 baseline (80.93%, 0.77 GFLOPs) on the identical split, with parameters, GFLOPs and energy measured in a cloud virtualized environment.

A complete reproducibility package (hyperparameters, environment, per-class report, confusion matrix and fold results generated from the same test predictions) and an explicit statement of what generic HAR results can analyze the activity performed by the humans.

2. Literature review

Most existing deep learning and pose-based tracking models depend on neuroimaging-based analysis or on wearable sensor data. A comparative analysis of representative techniques is given in Table 1.

The literature review shows that most existing models rely on MRI-based neuroimaging data and involve computationally expensive deep learning models. Some models achieve reasonable accuracies but their limitations hinder real-time deployment. Some more recent studies on patient tracking are discussed below:

Zhou *et al.* Ref. [11] combined 3D CNN with a video Swin Transformer to predict neurodegenerative disease in Alzheimer's patients using the ADNI dataset by extracting the neuroimaging data, and achieved an AUC of 92.92%. Although the AUC of the video Swin Transformer is high, its real-time deployment is difficult because of its heavy architecture.

Mohan *et al.* Ref. [12] investigated an IoT based CCTV monitoring system that uses a 3D adaptive residual DenseNet to predict the neurodegenerative disease in Alzheimer's patients by monitoring their activities through CCTV data. The mean fold CSI value

obtained is 92.65% and the run time to detect the disease is 20.54 minutes. Although the fold value of the DenseNet is good, the prediction time per patient is excessively high.

Zhang *et al.* Ref. [13] employed a fused model for 3D pose estimation by using a temporal graph neural network. This study uses human joints as key points to map a pose from a long sequence, and the temporal dependency captures the relationship between human pose joints to estimate the pose. Synthetic human poses serve as the initial pre-processed data, and normalization and temporal mapping are then performed with a graph network. The outcome shows an optimized final pose feature. This work requires extensive training to capture temporal data, and its computationally expensive in-depth analysis makes it unreliable for real-time human pose estimation.

Other relevant work includes video-based deep models for Alzheimer's diagnosis, MRI-based lightweight models, attention-based detectors, and general deep-learning methods for medical imaging. For activity recognition itself, pose-based pipelines build on multi-person 2D pose estimation with part affinity fields and on hand and 3D body pose estimation, whereas single-image recognition of human activities with convolutional networks has been studied according to Basly *et al.* Ref. [14]. Large pretrained image encoders such as CLIP and Vision Transformers, and compact CNNs such as EfficientNet, are the two families compared in this work. Evidence that structured physical activity can benefit people with Alzheimer's disease motivates activity monitoring but does not by itself validate any recognition model.

Kong and Fu Ref. [15] surveyed vision based human action recognition and prediction. According to Sun *et al.* Ref. [16] reviewed action recognition from RGB, skeleton, event, acceleration and Wi-Fi-data and concluded that each modality has its own strengths. According to Zheng *et al.* Ref. [17] surveyed deep learning based human pose estimation, including a part affinity field methods such as open pose and motion recognition in video has also been addressed with feature tracking. According to Han *et al.* Ref. [18] and Khan *et al.* Ref. [19] surveyed vision transformer, which transfer well to downstream image tasks. According to Patterson *et al.* Ref. [20] examined the energy use of machine learning and argued that it can be reduced substantially. According to Ebrahimi *et al.* Ref. [21] applied convolutional neural network to MRI images from Alzheimer disease classification with the use of neuroimaging-based methodology.

According to Kapoor *et al.* Ref. [22] showed that data leakage is a widespread cause of irreproducible results in machine learning research and according to Yagis *et al.* Ref. [23] showed that slice level splitting inflates accuracy in brain MRI classification. According to Adegoke-Elijah *et al.* Ref. [24] applied an LSTM autoencoder to ECG anomaly detection, according to Keji *et al.* Ref. [25] Converted tabular data to images and classified them with transfer learned VCG and ResNet models, and according to Mohammed *et al.* Ref. [26] integrated multivariate convolutional neural network with dynamic conditional correlation model. These findings motivate the group-level splitting and used in this study; image data of everyday activities are used here as a non-clinical first step towards a later clinical endpoint.

2.1. Research gap summary

Three gaps follow from the review. First, reported HAR accuracies are rarely accompanied by a leakage-controlled protocol (near-duplicate handling, splitting before augmentation, group-level cross-validation). Second, activity-recognition accuracy has been linked to a routine activity analysis, so recognition results can be presented as measures of missed activity detection. Third, efficiency claims (GFLOPs, energy) are often given without the hardware, tool and workload that produced them. This study addresses the first and third gaps on a generic HAR dataset and leaves the second to future work with a clinically characterized cohort.

3. Materials and methods

The objective of this study is to measure, under a leakage-controlled protocol, how accurately single RGB images of everyday activities can be classified into 15-classes, and what computational cost the classifiers require. Two classifiers are compared on the same data split: a fine-tuned CLIP ViT-B/16 (proposed) and an effective lightweight EfficientNet-B0 baseline as a part of the evaluated system. The dataset used in this study, https://www.kaggle.com/datasets/meetnagadia/human-action-recognition-har-dataset is a human action recognition dataset.

3.1. Material

3.1.1. Dataset description

The Human Action Recognition (HAR) dataset was released as DPhi Data Sprint 76 and is hosted on Kaggle (Version 1, 328.18 MB). Its labelled training folder holds 12,600 JPEG images listed in Training_set.csv (columns filename and label), balanced across 15-classes with 840 images per class: calling, clapping, cycling, dancing, drinking, eating, fighting, hugging, laughing, listening to music, running, sitting, sleeping, texting and using laptop. Each image shows one activity category. The data consist of images only; there are no subject identifiers. No synthetic images were added; 12,600 is the number of original labelled images. Table 2 illustrates the dataset parameters.

The 12,600 images were divided at the level of near-duplicate groups into training, validation and test sets before any augmentation (Table 3). No duplicate group appears in more than one subset.

Table 2: Dataset parameters.

Attribute	Description
Dataset Name	Human Action Recognition (HAR) dataset, DPhi Data Sprint 76/Kaggle, Version 1
Activity Range	Number of classes: 15 (840 images each)
Data Model	Single RGB still images (JPEG), one activity label per image; resized to 224 x 224 x 3
Original labelled images	12,600
Unlabelled test images	5,400
Near-duplicate groups	12,438 groups from 12,600 images (166 near-duplicate pairs, Hamming distance (4)

Table 3: Sample counts per subset.

Subset	Images	Role
Original labelled images	12,600	All images in Training_set.csv
Training	9,005 (71.5%)	Weight updates; online augmentation applied
Validation	1,796 (14.3%)	Epoch-wise model selection (EMA weights)
Test	1,799 (14.3%)	Single final evaluation; per-class supports in Table 6
Duplicate groups shared across subsets	0	Verified programmatically

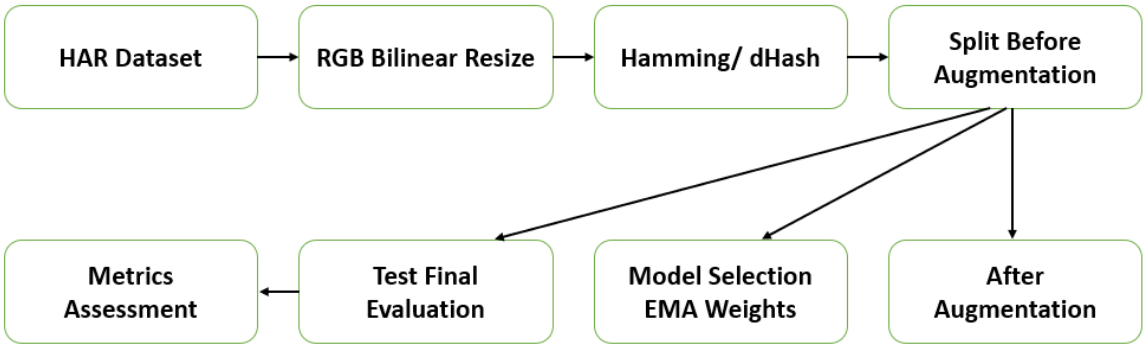

Figure 1: Data handling pipeline.

3.2. Methods

The evaluation proceeds through the following steps, as shown in Figure 1 and detailed in Algorithm 1.

1. Image loading and resizing: every labelled image is converted to RGB and resized to 224 x 224 pixels (bilinear).
2. A 64-bit difference hash (dHash) is computed for every image; two images are linked if their Hamming distance is 4, and connected components define groups.
3. Stratified Group K-Fold separates test, validation and training images so that classes are balanced and no group crosses subsets.
4. Training with online augmentation of training batches only, model selection on the validation set, and one final evaluation on the test set.
5. Evaluation: accuracy, macro- and weighted-average precision, recall, and F1, macro one-vs-rest AUC, confusion matrix and classification report from the same test predictions, followed by five-fold group-stratified cross-validation, and measurement of parameters, GFLOPs, latency and power.

3.2.1. Data preprocessing

Each 224 x 224 RGB image is scaled to [0,1] and then standardized per channel. Scaling is the min-max form with (xmin =0 and xmax= 255) applied to image pixels only Eq. (1); standardization uses the mean μc and standard deviation σc of the pretrained backbone as expressed in Eq. (2). The CLIP backbone uses the CLIP mean and standard deviation supplied with its pretrained configuration, and the EfficientNet-B0 baseline uses the ImageNet values.

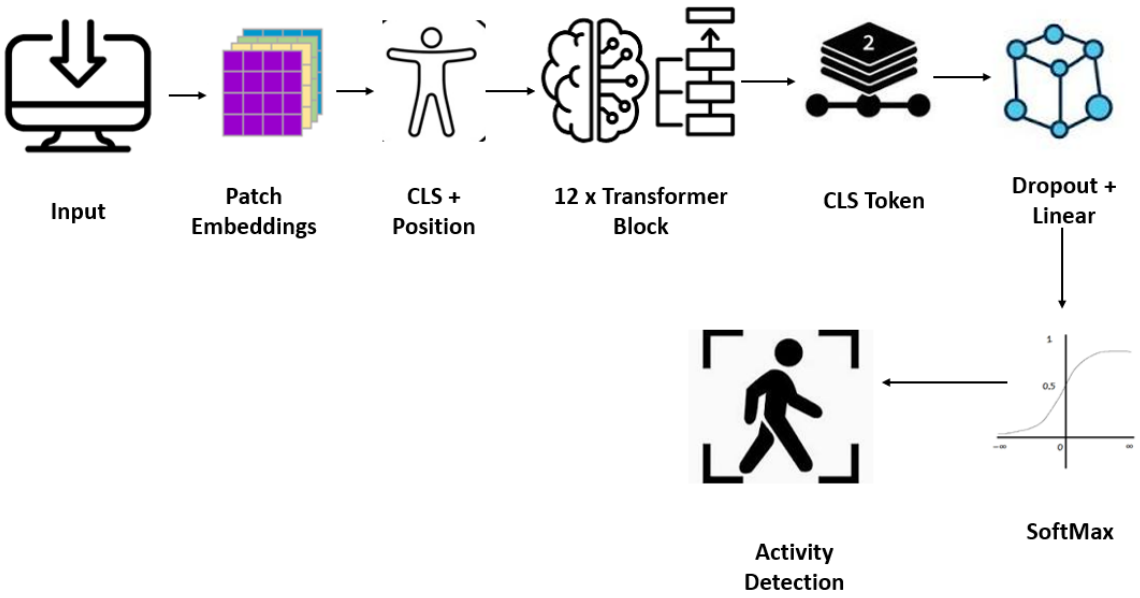

Figure 2: Workflow of the proposed activity classification system.

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}, \tag{1}$$

$$\widetilde{x}_c = \frac{X'_c - \mu_c}{\sigma_c}, \quad c \in \{R, G, B\}. \tag{2}$$

Augmentation is performed on the GPU, independently for every sample, on training batches only: random horizontal flip (p=0.5), random zoom with scale 0.65–1.0 (0.70–1.0 for the baseline), random rotation, random translation, and brightness and contrast scaling in 0.8–1.2. Validation and test images are never augmented; the only test-time operation is averaging the softmax outputs of an image and its horizontal flip. Because augmented images are generated on the fly from training images only, no augmented copy can reach the validation or test subsets, and no augmented 'dataset size' exists.

$$\begin{aligned} \overline{p}(x) &= \frac{1}{2}[\text{softmax}(f(x)) \\ &\quad + \text{softmax}(f(\text{flip}(x)))], \\ \widehat{y} &= \underset{i}{\text{argmax}}\, \overline{p}_i(x). \end{aligned} \tag{3}$$

At test time the class probabilities are the average of the softmax output for an image and for its horizontal flip, as expressed in Eq. (3); the predicted class is the arg max of this averaged vector.

4. Proposed methodology

Two image classifiers are evaluated. The proposed classifier is a CLIP ViT-B/16 image encoder fine-tuned end-to-end with a new 15-way linear head; the lightweight baseline is EfficientNet-B0 fine-tuned from ImageNet weights. Both receive RGB images, not pose key points. Implementations are taken from the timm library. The data flow of the proposed classifier is given in Figure 2.

4.1. CLIP ViT-B/16 classifier model

The ViT-B/16 encoder divides the 224 x 224 image into 196 non-overlapping 16 x 16 patches, linearly embeds each patch to 768 dimensions, prepends a class (CLS) token and adds learned position embeddings, giving a B x 197 x 768 sequence. Twelve Transformer blocks (12 attention heads, MLP hidden size 3,072) process the sequence; the final (CLS) representation (B x 768) passes through dropout (0.1) and a linear layer (768 15) to give class logits ri, which are converted to probabilities by the softmax function Eq. (4). The encoder is initialised from the OpenAI CLIP image-tower weights distributed with timm (vit_base_patch16_clip_224.openai) and all 85.81 M parameters are fine-tuned.

$$p_i = \frac{\exp(r_i)}{\sum_{j=1}^{k} \exp(r_j)} . \tag{4}$$

Training minimises the label-smoothed cross-entropy with smoothing factor, where y is the true class and 1 is the indicator function, as expressed in Eq. (5).

$$\begin{aligned} L &= -\sum_{i=1}^{k} q_i \log p_i, \\ q_i &= (1-\varepsilon)\,\mathbf{1}[i=y] + \frac{\varepsilon}{K} . \end{aligned} \tag{5}$$

Parameters are updated with AdamW, the decoupled-weight-decay form of Adam, where g is the gradient, learning rate n, and are the decay rates, is a small constant and λ is the weight decay. The first and second moment estimates m and v and their bias corrected values are computed as in Eq. (6) and (7), and the update is given by Eq. (8). In Eq. (8) the weight decay is decoupled from the adaptive gradient step and is subtracted, scaled by learning rate, together with adaptive update =0.05 for weight matrices and 0 for bias and normalization parameters, and scheduled learning rate of the parameter group. This is the update implemented by torch. AdamW.

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t, \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 . \end{aligned} \tag{6}$$

$$\widehat{M}_t = \frac{m_t}{1-\beta_1^t}, \qquad \widehat{v}_t = \frac{v_t}{1-\beta_2^t} . \tag{7}$$

$$\Theta_t = \Theta_{t-1} - n_t \left(\frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \varepsilon} \right) + \lambda \Theta_{t-1} . \tag{8}$$

Layer-wise learning rate decay assigns the learning rate n1 to transformer block l in Eq. (9) with L=12 blocks and =0.75, where embeddings take l =0, and the new head uses 5 x n0. The learning rate follows a one epoch linear warm-up and a cosine decay to a floor of 1% of its base value, as expressed in Eq. (9).

$$n_l = n_0 \cdot \gamma^{L+1-l} . \tag{9}$$

An exponential moving average (EMA) of the weights, defined in Eq. (10) as, is maintained for the ViT only and is used for its validation, model selection and testing.

$$\begin{aligned} \overline{\theta}_t &= \delta_t \overline{\theta}_t - 1 + (1-\delta_t)\Theta_t, \\ \delta_t &= \min\left(0.998, \frac{1+t}{10+t}\right) . \end{aligned} \tag{10}$$

4.2. EfficientNet-B0 baseline model

EfficientNet-B0 applies compound scaling across width, depth and resolution to improve efficiency over traditional convolutional neural network models. EfficientNet-B0 uses scaling coefficients for depth d, width w and input resolution r. It is built from a special block, MBConv, the mobile inverted bottleneck convolution. Each MBConv layer consists of a 1 x 1 expansion convolution, a depthwise convolution with same padding, a squeeze and excitation block, a 1 x 1 projection convolution and an activation function, followed by an output layer for the final classification of the 15 activity classes. EfficientNet-B0 is used here as a standard ImageNet-pretrained image classifier taking RGB images (B x 3 x 224 x 224) directly.

The 1,280- dimensional pooled feature is mapped by dropout (0.3) and a linear layer to 15 logits, and the softmax of Eq. (4) gives class probabilities. It is trained with the same label-smoothed cross-entropy Eq. (5) using AdamW Eq. (6)-(8) with λ=0.01, no layer-wise decay and no EMA with a one-cycle learning-rate schedule (Table 4).

Role of padding and convolutional layers, In MBConv block the 1 x 1 convolutions mix channel at every spatial position, whereas the depth-wise k x k convolutional filter each channel spatially with k=3 at a cost proportional to k2 per channel instead of k2 times the number of input channels. A stride-1 convolution with same padding k/2 pixel keeps the spatial size unchanged which allows the residual connection to be added to the block input, whereas stride-2 convolutions halve the feature map size at the stage transitions of the network; without padding the feature map would shrink at every layer, discarding border information and breaking the skip connection. For a 224 x 224 input the five stride-2 stages leave a 7 x 7 x 1280 feature map before pooling.

Lowering the input resolution reduces the convolution cost roughly in proportion to r2, and replacing 5 x 5 by 3 x 3 depthwise kernels reduce cost, reducing the width multiplier shrinks all layers; FP16 and channel last memory layout reduce latency on supported hardware. These are general design levers; none of them was varied in this research, which uses the standard EfficientNet-B0 configuration at 224 x 224 input, each change is evaluated on leakage-controlled split to attain efficient detection accuracy.

Global average pooling (GAP) versus a learned head. After the last convolution, EfficientNet-B0 applies GAP, which averages every channel of the feature map F into a single value, and a linear layer maps the pooled vector to 15 logits, as given in Eq. (11).

$$r = W \cdot \mathrm{GAP}(F) + b,$$
$$\mathrm{GAP}(F)_c = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} F_{c,h,w}\,. \tag{11}$$

Here W ∈ R15x1280 and b ∈ R15. GAP has no trainable parameters, so the head contains 1280 x 15 + 15 = 19,215 parameters. A learned alternative that flattens the 7 x 7 x 1280 map into one fully connected layer would need 62,720 x 15 +15 = 940,815 parameters, and an attention based learned pooling layer would add further parameters. A smaller head has fewer free weights to fit on 9,005 training images, is invariant to where in the image the discriminative evidence lies, and gives gradients that are averaged over all spatial positions; these properties are expected to favour stable convergence and lower overfitting. The ViT classifier by contrast aggregates information through the CLS token, which is itself a learned, attention-based pooling over all patch tokens. The classifier output is one probability vector per image expressed in Eq. (4), mapping it to a daily routine requires expected activities, an observation window and a count rule, and is presented as a design only mapping in algorithm 2.

Algorithm 1. Leakage-controlled training and evaluation of the HAR image classifiers.

Input: Labelled images $\{(I_n, y_n)\}$, $n = 1, \ldots, 12{,}600$; $K = 15$ classes; hyperparameters in Table 4 (ViT: $E = 8$, $B = 32$, EMA on; EfficientNet-B0: $E = 15$, $B = 64$, EMA off).

Output: Trained classifier, test metrics, confusion matrix, per-class report, and fold results.

1. Load each I_n as RGB, resize to 224×224, and store as uint8.
2. Compute the 64-bit dHash h_n for every image. Link images n and m if their Hamming distance is 4. Let g_n denote the connected-component group index.
3. Set (trainval, test) using Stratified Group K-fold (seven folds; labels y, groups g), taking one fold as the test set. Set (train, val) using Stratified Group K-fold (six folds) on trainval. Assert that no group appears in more than one subset.
4. Initialise the backbone from pretrained weights and replace the head with a K-way linear layer. For ViT only, copy the weights to the EMA model.
5. For epochs $1, \ldots, E$, augment training mini-batches only, normalise (Eqs. (1)–(2)), compute logits and loss (Eq. (5)), clip gradient norm to 1.0, update AdamW (Eqs. (6)–(9)), and update EMA (Eq. (10)).
6. After each epoch, compute validation accuracy without augmentation or flip averaging. Retain the best EMA weights (ViT) or best current weights (EfficientNet-B0).
7. Load the best weights. For each test image x, average softmax predictions for x and its horizontal flip as described in Eq. (3); predict the class from the largest probability.
8. From one set of test predictions, calculate accuracy, macro- and weighted-average precision, recall, F1, macro one-vs-rest AUC, confusion matrix, and classification report (Eq. (12)).
9. For cross-validation (ViT only), repeat the split, training, and test evaluation using five group-stratified folds, four epochs per fold, and augmentation only inside training folds.

5. Experimentation, results and analysis

The experimental procedure involves several steps: the experimental environment, testing and training, and classification analysis, which evaluates the performance of the classifiers using different metrics.

5.1. Experimental setup

The experimental setup for training and testing uses an existing public dataset the Human Action Recognition (HAR) dataset. All training, evaluation and efficiency measurements were run on a single platform: a Kaggle notebook with one NVIDIA Tesla T4 GPU (16 GB), PyTorch and timm, mixed-precision (FP16 autocast) arithmetic, and CPU-side image decoding with PIL. All results in this paper were produced on this platform. Training of the ViT took about 99 s per epoch (about 13 min for eight epochs). Table 4 lists the hyperparameters.

Both classifiers are fine-tuned on the training subset only. Augmentation is applied on the fly to training batches (Section 3.2.1), so no augmented image exists outside the training loop.

Table 4: Hyperparameters and environment.

Hyperparameter	CLIP ViT-b/16 (proposed)	EfficientNet-B0 (baseline)
Pre-trained weights	OpenAI CLIP (timm: vit_base_patch16_clip_224.openai)	ImageNet (timm: efficientnet_b0)
Input size / channels	224 x 224 x 3 RGB	224 x 224 x 3 RGB
Optimizer	AdamW	AdamW
Peak learning rate	1 x 10-4	6 x 10-4
Layer-wise LR decay	0.75	None
Schedule	1-epoch linear warm-up + cosine (floor 1%)	One-cycle (15% warm-up)
Weight decay	0.05 (not in bias/norm)	0.01
Batch size	32	64
Epochs	8 (4 per fold in cross-validation)	15
Loss	Cross-entropy, label smoothing 0.1	Cross-entropy, label smoothing 0.1
Regularisation	Drop-path 0.1, dropout 0.1, grad-clip 1.0	Dropout 0.3
EMA of weights	Decay 0.998	None
Precision	FP16 autocast (AMP)	FP16 autocast (AMP)
Augmentation (train only)	Horizontal flip, zoom 0.65–1.0, rotation, translation, brightness/contrast 0.8 -1.2	As left, zoom 0.7–1.0
Test-time augmentation	Horizontal-flip averaging	Horizontal-flip averaging
Model selection	Best validation accuracy (EMA weights)	Best validation accuracy
Validation loss	Unsmoothed cross entropy (EMA Weights)	Unsmoothed cross entropy
Split / seed	Stratified Group K-fold, 9,005/1,796/1,799; seed 42	Same split and seed
Hardware	NVIDIA Tesla T4 (Kaggle)	NVIDIA Tesla T4 (Kaggle)

5.2. Experimental results

All results in this subsection come from one held-out test set of 1,799 images (Table 3) that was not used for training, augmentation, or model selection. The fine-tuned CLIP ViT-B/16 classified 1,645 of the 1,799 test images correctly, an accuracy of 91.44% (balanced accuracy 91.54%). Macro-averaged precision, recall and F1 were 91.43%,91.54% and 91.45%; weighted precision, recall and F1 were 91.44%, 91.44% and 91.41%; and the macro one-vs-rest AUC was 0.9955. With n=1,799 the binomial 95% confidence interval of the accuracy is approximately ± 1.3 percentage points (about 90.2–92.7%). The EfficientNet-B0 baseline reached 80.93% accuracy (macro-F1 80.92%, macro-AUC 0.9791) on the same test images.

Per-class results are given in Table 6 and Figure 4. The best recognized classes were cycling (F1 0.992), eating (0.955), and sleeping (0.954), whereas texting (0.812), calling (0.838) and sitting (0.849) were the hardest, followed by listening to music (0.899) and using laptop (0.908). These weaker classes are visually similar, hand- and object-centred or posture-defined activities, and the off-diagonal cells of the confusion matrix (Figure 3) show which class pairs are exchanged.

A second training run with the identical seed and configuration produced 92.50% test accuracy, which illustrates the roughly one-percentage-point run-to-run variation caused by non-deterministic GPU kernels; the first run (91.44%), whose predictions underlie Table 5 and Figure 3, and the cross-validated mean (Section 5.3) are reported as the primary results.

Consistency of the reported values. Every method evaluated here (the ViT, the EfficientNet-B0 baseline and each cross-validation fold) receives the same input type, a single RGB image, and predicts the same 15-classes; no action sequences are processed, all methods therefore follow the same processing steps. All numbers derive from one set of 1,799 held-out predictions, and Table 5 lists each accuracy value together with its evaluation set and role. Generalisation is supported by group-level splitting and by the agreement of the validation accuracy (90.92%), the fixed-split test accuracy (91.44%) and the cross-validated accuracy (91.53 ± 0.74%), which differ by less than one percentage point.

5.3. Performance metrics

The performance metrics play a crucial role in analysing the effectiveness and quality of the classifiers. Moreover, these metrics are essential for validating the performance of the classifiers on a commonly held-out set. The predictive performance of the model is analysed through metrics like accuracy, precision, recall, and F1 score. These are reported per class (Table 6, Figure 4) and as macro and weighted averages, so that the 15-class problem is summarised unambiguously.

5.3.1. Model testing and training

Accuracy is defined as the number of correctly identified test images out of all test images. The ViT was trained for eight epochs. Training loss (label-smoothed) fell from 1.671 to 0.688, training accuracy on augmented batches rose from 57.6% to 95.3%, and

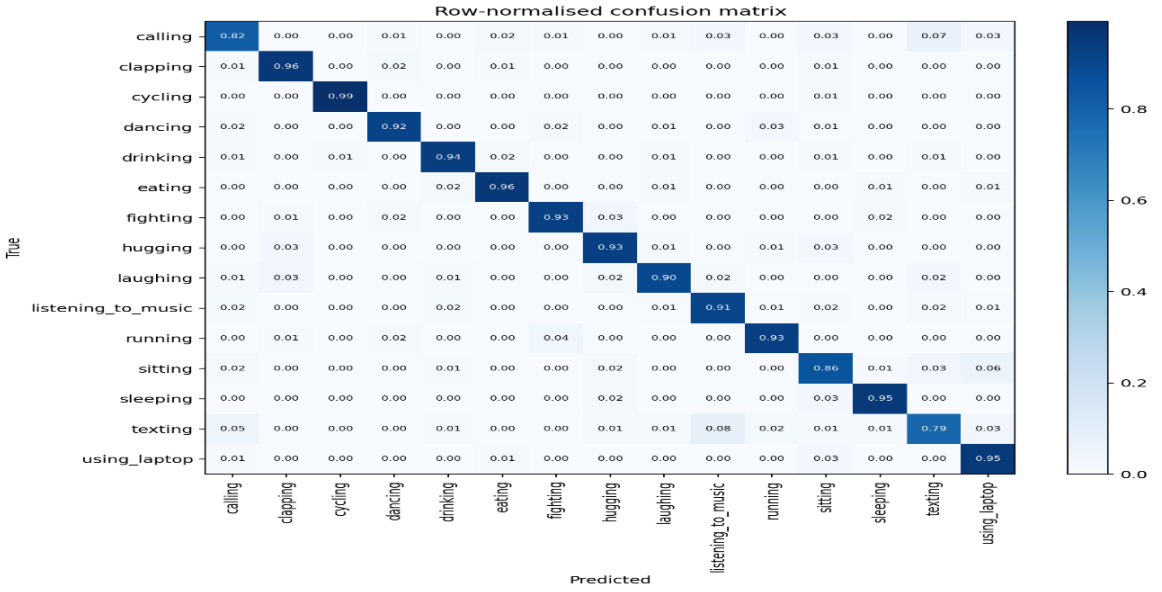

Figure 3: Multiclass confusion matrix showing classification performance across the 15 activity classes.

Table 5: Reconciliation of the accuracy values reported in this study.

Quantity	Value	Evaluation set	Role
CLIP ViT-B/16, test accuracy, first run	91.44%	1,799 held-out test images	Primary result (Table 6 and Table 9, Figure 3–Figure 4, Abstract)
Same configuration and seed, repeat run	92.50%	Same 1,799 test images	Run-to-run variation only
Best validation accuracy (epoch 7)	90.92%	1,796 validation images	Model selection
five-fold group-stratified cross-validation	91.53 ± 0.74%	Five test folds of 2,513-2,530 images	Generalisation across splits (Table 7)
EfficientNet-B0 baseline, test accuracy	80.93%	Same 1,799 test images	Baseline (Table 9)

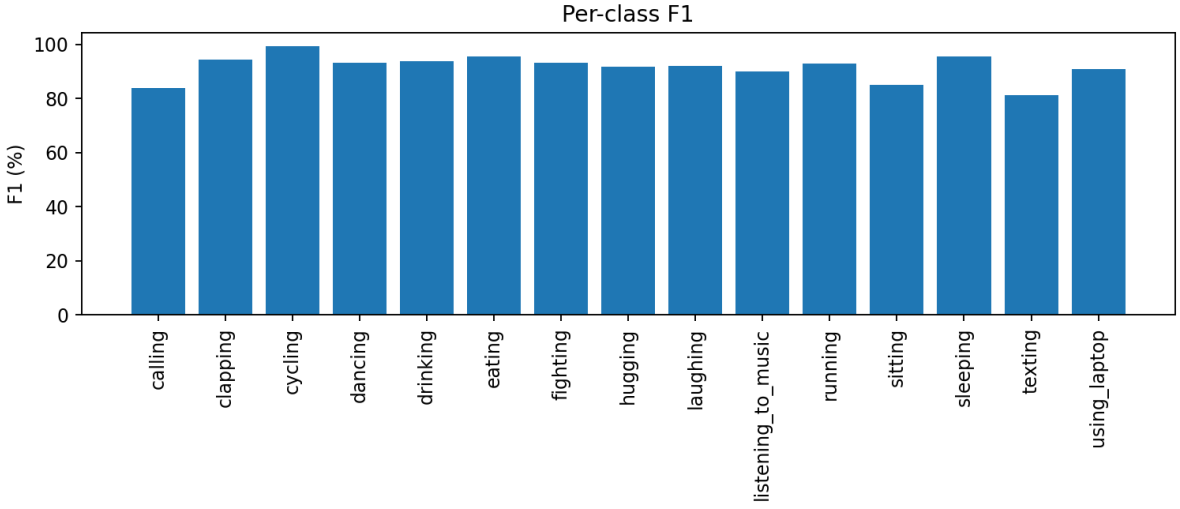

Figure 4: Per-class F1 score of the CLIP ViT-B/16 classifier on the held-out test set.

Table 6: Per-class results on the held-out test set (n = 1,799), CLIP ViT-B/16.

Class	Precision	Recall	F1-score	Support
Calling	0.8596	0.8167	0.8376	120
Clapping	0.9280	0.9587	0.9431	121
Cycling	0.9918	0.9918	0.9918	122
Dancing	0.9457	0.9173	0.9313	133
Drinking	0.9391	0.9391	0.9391	115
Eating	0.9508	0.9587	0.9547	121
Fighting	0.9365	0.9291	0.9328	127
Hugging	0.9068	0.9304	0.9185	115
Laughing	0.9426	0.8984	0.9200	128
Listening to music	0.8889	0.9091	0.8989	132
Running	0.9238	0.9327	0.9282	104
Sitting	0.8417	0.8559	0.8487	118
Sleeping	0.9541	0.9541	0.9541	109
Texting	0.8390	0.7857	0.8115	126
Using laptop	0.8655	0.9537	0.9075	108
Macro average	0.9143	0.9154	0.9145	1799
Weighted average	0.9144	0.9144	0.9141	1799
Accuracy			0.9144	1799

validation accuracy of the EMA weights rose from 88.8% to a maximum of 90.9% (epoch 7) and was 90.7% at epoch 8 (Figure 5). The validation loss in Figure 5 (about 0.36–0.40) is plotted as the standard cross entropy of the EMA weights without label smoothing and is therefore not comparable with the label smoothed training loss, for K = 15 and ε=0.1 the smallest attainable label smoothed cross entropy is –[(1-ε+ε/K) ln(1-ε+ε/K)+(K-1) (ε/K) ln(ε/K)] 0.556, so a label-smoothed loss below this value is impossible. The 4–5 percentage-point gap between training and validation accuracy indicates mild overfitting; it is not evidence that the model 'avoids overfitting'. Validation-set accuracy (90.9%) and test accuracy (91.44%) agree within the sampling error expected for subsets of about 1,800 images.

For a K-class problem, accuracy is computed directly from the confusion matrix C Eq. (12), and per-class and averaged metrics follow Eq. (13) and (14). Figure 5 plots the accuracy and loss of the proposed classifier.

$$\text{Accuracy} = \frac{\text{correct predictions}}{\text{all predictions}}. \tag{12}$$

$$P_c = \frac{TP_c}{TP_c + FP_c}, \quad R_c = \frac{TP_c}{TP_c + FN_c}, \quad F1_c = \frac{2P_cR_c}{P_c + R_c}. \tag{13}$$

$$\text{Macro-F1} = \frac{1}{K}\sum_c F1_c, \qquad \text{Weighted-F1} = \sum_c \left(\frac{n_c}{N}\right) F1_c. \tag{14}$$

Here TP, FP and FN are counted as one-vs-rest for class c, nc is the number of the test images of class c and N is the total number of test images. The macro average weights all classes equally, the weighted average weights each class by nc, and micro averaged precision, recall and F1 equal accuracy for single label multiclass data. The AUC reported is the macro average of one-vs-rest AUCs.

5.3.2. Comparative experiments

The robustness of the results is examined with group-stratified K-fold cross-validation, and the computational complexity of the models is measured with named tools.

Group-stratified K-fold cross-validation. The fixed split of Section 3 is complemented by five-fold cross-validation that is stratified by class and grouped by near-duplicate group, so that no group (and therefore no near-identical image) appears in both the training and test portions of any fold. In each fold, one fifth 20% of the groups from the test fold, and of the remaining 80% the training pool, one-sixth is held out for validation and model selection; augmentation is applied inside the training folds only. To limit the run time, each fold was trained for four epochs instead of 8(cross-validation for ViT). The fold sizes and results are given in Table 7

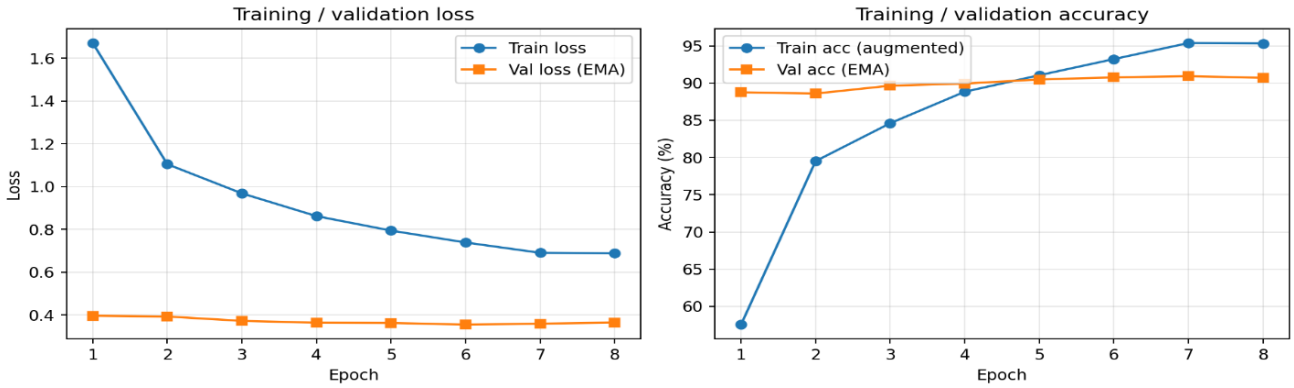

Figure 5: Training and validation loss and accuracy (EMA weights) of the proposed classifier over eight epochs.

Table 7: Five-fold group-stratified cross-validation, CLIP ViT-B/16 (four epochs per fold).

Fold	Train	Validation	Test	Accuracy (%)	Macro-F1 (%)
1	8,401	1,678	2,521	90.36	90.31
2	8,394	1,676	2,530	91.98	91.94
3	8,405	1,677	2,518	91.58	91.43
4	8,398	1,684	2,518	92.30	92.34
5	8,398	1,689	2,513	91.44	91.49
Mean ± SD				91.53 ±0.74	91.50 ±0.76

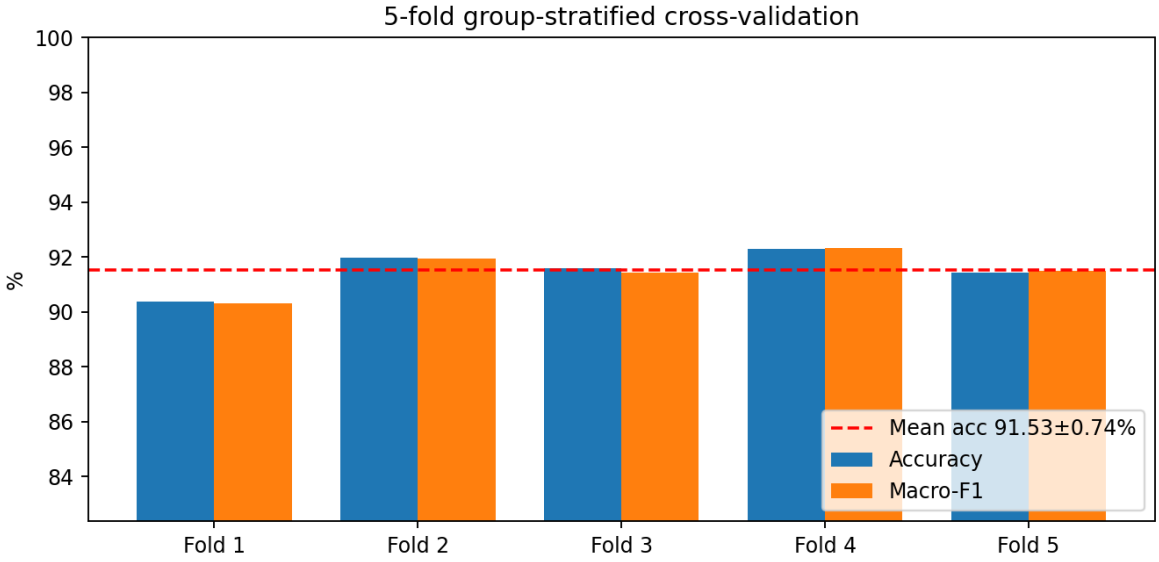

Figure 6: K-fold cross-validation results ($K = 5$) of the proposed model.

and Figure 6; the cross-validated accuracy is 91.53 ± 0.74% (mean ± sample standard deviation over five-folds) and the macro-F1 is 91.50 ± 0.76%, in agreement with the fixed-split result (91.44%).

Computational complexity and measured cost. Parameters were counted from the instantiated networks. FLOPs were counted with the PyTorch FlopCounterMode profiler for one 224 x 224 RGB image (batch size 1) with fused attention disabled so that attention matrix multiplications are counted; the profiler counts a multiply-accumulate (MAC) as two floating-point operations, so GFLOPs = 2 x GMACs. All values are therefore GFLOPs per image of the complete classifier. The 'trainable parameter' count of the saved EMA copy is 0 only because gradients are disabled on that copy; all parameters of the fine-tuned network are trainable.

The fine-tuned CLIP ViT-B/16 has 85.81 M parameters and requires 35.13 GFLOPs (17.56 GMACs) per image. The EfficientNet-B0 baseline has 4.03 M parameters and requires 0.77 GFLOPs (0.38 GMACs). Table 8 lists the parameters and computational

Table 8: Parameters and computational complexity (224 × 224 input, batch size 1, 15-class head).

Model	Parameters (M)	GFLOPs (2 x MAC)	GMACs
MobileNetV3-Large	4.22	0.43	0.22
EfficientNet-B0 (baseline)	4.03	0.77	0.38
ResNet-50	23.54	8.17	4.09
ConvNeXt-Tiny	27.83	8.91	4.45
CLIP ViT-B/16	85.81	35.13	17.56

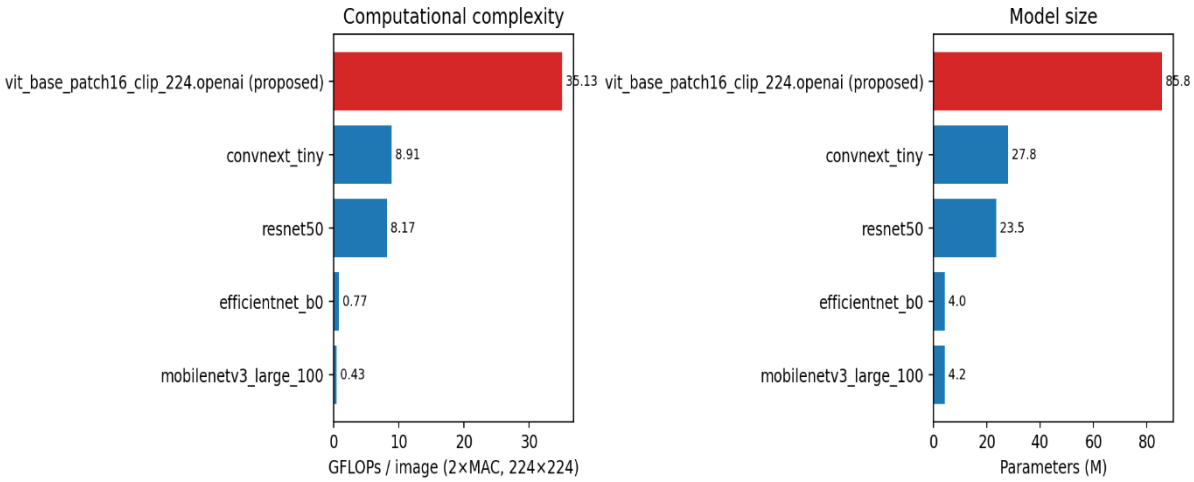

Figure 7: Number of model elements (parameters) used by the proposed system.

complexity of the evaluated models and Figure 7 illustrates the computational complexity and model size of the proposed model compared with baseline models. No accuracy results are reported for MobileNet V3, ResNet 50 and ConvNeXt-Tiny are included as complexity references only.

.

Latency, throughput and power were measured in FP16 on the Tesla T4. Latency is the mean of 100 forward passes at batch size 1 after 10 warm-up passes, with CUDA synchronization (ViT 7.39 ms; EfficientNet-B0 9.22 ms). Throughput was 335 images/s for the ViT (batch 32) and 1,556 images/s for EfficientNet-B0 (batch 64). Board power was sampled with nvidia-smi every 100 ms while the benchmark ran: 65.85 W for the ViT and 65.77 W for EfficientNet-B0. Energy per image was computed as mean power x elapsed time / number of images processed, giving 0.222 J per image (ViT, batch 32) and 0.046 J (EfficientNet-B0, batch 64). The elapsed time included the warm-up passes, so these energies are slightly overestimated from the measured power and throughput the steady state values are 65.85 W/ 335 images 0.197 J (ViT) and 65.77 W/ 1,556 images 0.042 J (EfficientNet-B0), so the reported values are higher by about 13% and 9%, and the ViT to EfficientNet-B0 energy ratio is about 4.7 rather than 4.8.

These are whole-GPU board values on a data-centre T4 and are not edge-device measurements. Batch-1 latency on a GPU is dominated by kernel-launch overhead, so the lower batch-1 latency of the ViT does not indicate that it is cheaper to run; throughput and energy per image reflect the roughly 46-fold difference in FLOPs between the models as more informative measures.

6. Comparative result analysis

This section compares the proposed CLIP ViT-B/16 classifier with the EfficientNet-B0 baseline under the same protocol. On this dataset, the proposed model classifies the 15 generic activities more accurately than the baseline.

6.1. Same-split comparison and contributions

This section compares the two classifiers evaluated in this study on the same dataset, the same grouped split and the same test images (Table 9).

Table 9 shows that, on identical data, fine-tuning the CLIP ViT-B/16 raised test accuracy from 80.93% to 91.44% (+10.5 percentage points) and macro-F1 from 80.92% to 91.45%. The cost of this gain is about 21 times more parameters (85.81 M vs 4.03 M), about 46 times more FLOPs (35.13 vs 0.77 GFLOPs) and about 4.8 times more energy per image at the same board power.

Table 9: Same-split comparison of the two classifiers (held-out test set, n = 1,799).

Metric	EfficientNet-B0 (baseline)	CLIP ViT-B/16 (proposed)	ViT relative to EfficientNet-B0
Test accuracy (%)	80.93	91.44	+10.51 points (+13.0% relative)
Balanced accuracy (%)	81.14	91.54	+10.40 points
Macro-F1 (%)	80.92	91.45	+10.53 points
Weighted-F1 (%)	80.88	91.41	+10.53 points
Macro AUC (one-vs-rest)	0.9791	0.9955	+0.0164
Best validation accuracy (%)	80.29	90.92	+10.63 points
Parameters (M)	4.03	85.81	x 21.3
GFLOPs per image (2 x MAC)	0.77	35.13	x 46
Latency, batch 1, FP16 (ms)	9.22	7.39	-19.8%
Throughput (images/s)	1,556 (batch 64)	335 (batch 32)	-78.5%
Mean GPU board power (W)	65.77	65.85	+0.1%
Energy per image (J)	0.046 (batch 64)	0.222 (batch 32)	x 4.8

Table 10: Contribution of this study.

Aspect	This study
Evaluation protocol	One dataset, one grouped split, plus five-fold group-stratified CV
Leakage control	dHash near-duplicate groups; split before augmentation; 0 groups shared across subsets
Metrics	Accuracy, macro/weighted P/R/F1, macro-AUC from one test prediction set
Statistical reporting	95% CI of about ± 1.3 points; CV mean ± SD (91.53 ± 0.74%)
Efficiency	Measured parameters, GFLOPs (2 x MAC), latency, throughput, power (W) and energy (J per image)
Accuracy-cost trade-off	+ 10.51 points for x 46 FLOPs and x 4.8 energy (Table 9)
Scope of claims	Generic image-based activity recognition only

Table 11: Inference energy measurement protocol.

Item	Specification
Device	NVIDIA Tesla T4 GPU in a Kaggle notebook (same device for training and inference); host CPU excluded
Measurement tool	Nvidia-smi power draw (on-board sensor), polled every 100 ms by a background thread while the benchmark runs
Workload	Forward passes of the full classifier on random 224 x 224 x 3 tensors (no disk or decoding): ViT, 80 passes at batch 32; EfficientNet-B0, 150 passes at batch 64; FP16 autocast
Quantities	Mean power P (W); elapsed time T (s); images processed N; energy per image Eimg (J)
Reported values	ViT: 65.85 W, 0.222 J per image; EfficientNet-B0: 65.77 W, 0.046 J per image
Derived	Elapsed time T implied by Eimg x N / P versus N / Throughput: ViT, N=80 x 32 = 2,560 images, T≈ 8.6 s versus 7.6 s; EfficientNet-B0, N=150 x 64 = 9,600 images, T≈ 6.7 s versus 6.2 s.

Table 9 gives, next to every metric, the relative difference of the ViT with respect to the EfficientNet-B0 baseline, so that the gain (+10.51 accuracy points, a 55.1% relative reduction of the test error) and its cost (about 46 times the FLOPs and 4.8 times the energy per image) are both visible. Table 10 summarises the contribution of this study.

6.2. *Energy analysis*

Energy per image was estimated from sampled GPU board power telemetry rather than from a software power model; the measurement protocol is specified in Table 11.

A software simulation is only as valid as the power model behind it and must be validated against hardware, whereas board-level telemetry is a measurement made by the device itself and needs no model. The measured value is also physically plausible: the Tesla T4 is a 70 W board, so a mean of about 65.9 W under sustained load is credible. Because P is nearly identical for both models (65.85 W and 65.77 W), the 4.8-fold difference in Eimg results from the longer time the ViT needs per image rather than from a higher power draw.

6.3. Discussion

Fine-tuning the CLIP ViT-B/16 raised held-out accuracy from 80.93% to 91.44% (+10.51 points, +13.0% relative) and reduced the test error from 19.07% to 8.56% (-55.1% relative). Every one of the 15-classes improved in F1-score. The largest gains were for sitting (+17.4 points, from 67.5% to 84.9%), listening to music (+14.4), fighting (+13.4) and clapping (+12.8), and the smallest for cycling (+1.6, already 97.6%). By contrast texting (F1 81.2%), calling (83.8%) and sitting (84.9%) stay 14-18 points below cycling (99.2%). These are small-object, hand-centred or posture-defined activities, so higher resolution, object-aware cues or additional training data are the most promising levers. The gain in accuracy comes at a cost of 45.7 times the GFLOPs, 21.3 times the parameters and 4.8 times the energy per image, i.e. about 0.017 J per image for each additional accuracy point over the baseline (0.176 J/10.51 points).

Stability: the cross-validated accuracy (91.53 ± 0.74%) differs from the fixed-split test accuracy by only 0.09 percentage points, the standard deviation is 0.8% of the mean, and the 95% interval on the test set is about ± 1.3 percentage points; a repeat run changed the test accuracy by +1.06 percentage points (92.50%), so differences of about one point should not be interpreted. Training accuracy on augmented batches reached 95.35% against 90.7% validation accuracy at the last epoch (a gap of 4.6 points), showing mild overfitting that EMA weights, label smoothing and drop-path only partly control. On this data the pretrained CLIP encoder is clearly the more accurate classifier, whereas EfficientNet-B0 is the only one of the two that fits a low-compute budget; neither result says anything about Alzheimer's disease, because the dataset contains no patients. The limitations that qualify these percentages are listed in Section 6.4.

6.4. Limitations

Generic activity recognition only. The dataset contains healthy-appearing, web-sourced images of everyday activities. No Alzheimer's patients, diagnoses, clinical severity labels (MMSE, CDR), longitudinal outcomes or ethics-approved patient recordings are involved; the results therefore say nothing about disease severity, progression, behavioural change or clinical decision support. No detection of unperformed activities. The classifier labels what is visible in one image. Detecting forgotten or missed routine activities needs an expected daily schedule, observation windows, a missing-activity criterion and validation against caregiver or clinical records. Residual leakage, Near-duplicates were detected with a 64-bit difference hash at Hamming distance 4 (166 pairs); visually similar images from the same web source or the same person that are not near-identical at this threshold may still straddle subsets, which would make the accuracies optimistic. Efficiency on target hardware. Power and latency were measured on a data-centre GPU and not measured using edge devices. Energy was estimated from sampled board power telemetry in a shared cloud notebook, the two models were run at different batch sizes, and cross-validation and runs are performed for the ViT model.

7. Conclusion and future scope

In this research, a fine-tuned CLIP ViT-B image classifier and a lightweight EfficientNet-B0 baseline, both operating on single RGB images, were evaluated for 15-class human activity recognition. This study evaluated 15-class image-based human activity recognition on the public HAR dataset under a leakage-controlled protocol in which near-duplicate groups were kept within one subset and augmentation was applied to training batches only. A fine-tuned CLIP ViT-B/16 reached 91.44% accuracy and 91.45% macro-F1 on 1,799 held-out images and 91.53 ± 0.74% in five-fold group-stratified cross-validation, whereas an EfficientNet-B0 baseline reached 80.93% at 0.77 GFLOPs. The ViT needs 85.81 M parameters, and 35.13 GFLOPs per image and drew about 65.9 W of board power on the T4 GPU. The ViT is therefore the most accurate but also the more computationally demanding classifier, whereas EfficientNet-B0 fits a low compute budget. These results characterize generic image-based activity recognition and may inform a future non-intrusive monitoring solution for family members and caregivers.

Future work follows from the measured performance. (i)Accuracy-cost gap: the ViT gains 10.51 accuracy points over EfficientNet-B0 at 46 times the FLOPs; knowledge distillation from the ViT into EfficientNet-B0, quantisation and pruning should be evaluated with the aim of closing at least half of this gap (about 86% accuracy) below 1 GFLOPs, with latency and power measured on the intended edge device using an external power meter. (ii) Reliability: repeated runs with at least five seeds, stricter duplicate detection and external datasets are needed to bound the roughly one-point run-to-run variation and to confirm generalization. (iii) Clinical relevance: with ethics approval and informed consent, collect an Alzheimer-specific cohort with diagnoses and MMSE/CDR endpoints, validate routine mapping against caregiver records, and extend to video with temporal models and, where useful, a pose library with a fully specified key point schema.

Data availability

The data supporting the findings of this article are available in the Human Action Recognition (HAR) dataset at https://www.kaggle.com/datasets/meetnagadia/human-action-recognition-har-dataset.

Declaration of competing interest

The authors declare no conflict of interest, financial or otherwise.

Funding

No funding was provided for the preparation of this manuscript.

Acknowledgment

The authors thank Lincoln University for supporting this research.

References

[1] Alzheimer's Association Report, "2026 Alzheimer's disease facts and figures", Alzheimer's & Dementia **22** (2026) e71345. https://doi.org/10.1002/alz.71345.
[2] M. Nagadia, "Human Action Recognition (HAR) dataset", Kaggle. (Accessed 19 October 2026). Available online: https://archive.ics.uci.edu/dataset/226/opportunity+activity+recognition
[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger & I. Sutskever, "Learning transferable visual models from natural language supervision", International Conference on Machine Learning, 2021, pp. 8748–8763. https://doi.org/10.48550/arXiv.2103.00020.
[4] M. Tan & Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks", International Conference on Machine Learning, 2019, pp. 6105–6114. https://doi.org/10.48550/arXiv.1905.11946.
[5] Q. Wu, Y. Wu, Y. Zhang & L. Zhang, "A local–global estimator based on large kernel CNN and transformer for human pose estimation and running pose measurement", IEEE Transactions on Instrumentation and Measurement **71** (2022) 5019612. https://doi.org/10.1109/TIM.2022.3200438.
[6] M. Habib, Z. Wang, S. Qiu, H. Zhao & A. S. Murthy, "Machine learning based healthcare system for investigating the association between depression and quality of life", IEEE Journal of Biomedical and Health Informatics **26** (2022) 2008. https://doi.org/10.1109/JBHI.2022.3140433.
[7] A. Gamal, M. Elattar & S. Selim, "Automatic early diagnosis of Alzheimer's disease using 3D deep ensemble approach", IEEE Access **10** (2022) 115974. https://doi.org/10.1109/ACCESS.2022.3218621.
[8] A. Lakhan, T.-M. Grønli, G. Muhammad & P. Tiwari, "EDCNNS: Federated learning enabled evolutionary deep convolutional neural network for Alzheimer disease detection", Applied Soft Computing **147** (2023) 110804. https://doi.org/10.1016/j.asoc.2023.110804.
[9] A. Alorf & M. U. G. Khan, "Multi-label classification of Alzheimer's disease stages from resting-state fMRI-based correlation connectivity data and deep learning", Computers in Biology and Medicine **151** (2022) 106240. https://doi.org/10.1016/j.compbiomed.2022.106240.
[10] B. Lee, S. Hong & H. Kim, "Determination of workers' compliance to safety regulations using a spatio-temporal graph convolution network", Advanced Engineering Informatics **56** (2023) 101942. https://doi.org/10.1016/j.aei.2023.101942.
[11] J. Zhou, Y. Wei, X. Li, W. Zhou, R. Tao, Y. Hua & H. Liu, "A deep learning model for early diagnosis of Alzheimer's disease combined with 3D CNN and video Swin transformer", Scientific Reports **15** (2025) 23311. https://doi.org/10.1038/s41598-025-05568-y.
[12] R. Mohan, P. Borugadda, N. R. Lavuri & S. Venugopal, "An IoT-assisted Alzheimer's disease patient monitoring system using adaptive deep learning models with recommendation about patient abnormality", Journal on Advances in Signal Processing **2026** (2026) 26. https://doi.org/10.1186/s13634-026-01301-7.
[13] L. Zhang, F. Lu, K. Zhou, X.-D. Zhou & Y. Shi, "Hierarchical spatial-temporal adaptive graph fusion for monocular 3D human pose estimation", IEEE Signal Processing Letters **31** (2023) 61. https://doi.org/10.1109/LSP.2023.3339060.
[14] H. Basly, W. Ouarda, F. E. Sayadi, B. Ouni & A. M. Alimi, "Lahar-CNN: Human activity recognition from one image using convolutional neural network learning approach", International Journal of Biometrics **13** (2021) 385. https://www.inderscienceonline.com/doi/abs/10.1504/IJBM.2021.117855.
[15] Y. Kong & Y. Fu, "Human action recognition and prediction: A survey", International Journal of Computer Vision **130** (2022) 1366. https://doi.org/10.1007/s11263-022-01594-9.
[16] Z. Sun, Q. Ke, H. Rahmani, M. Bennamoun, G. Wang & J. Liu, "Human action recognition from various data modalities: A review", IEEE Transactions on Pattern Analysis and Machine Intelligence **45** (2023) 3200. https://doi.org/10.1109/TPAMI.2022.3183112.
[17] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz & M. Shah, "Deep learning-based human pose estimation: A survey", ACM Computing Surveys **56** (2023) 1. https://doi.org/10.1145/3603618.
[18] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang & D. Tao, "A survey on vision transformer", IEEE Transactions on Pattern Analysis and Machine Intelligence **45** (2023) 87. https://doi.org/10.1109/TPAMI.2022.3152247.
[19] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan & M. Shah, "Transformers in vision: A survey", ACM Computing Surveys **54** (2022) 200. https://doi.org/10.1145/3505244.
[20] D. Patterson, J. Gonzalez, U. Hölzle, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. R. So, M. Texier & J. Dean, "The carbon footprint of machine learning training will plateau, then shrink", Computer **55** (2022) 18. https://ieeexplore.ieee.org/document/9810097
[21] A. Ebrahimi, S. Luo & Alzheimer's Disease Neuroimaging Initiative, "Convolutional neural networks for Alzheimer's disease detection on MRI images", Journal of Medical Imaging **8** (2021) 024503. https://doi.org/10.1117/1.JMI.8.2.024503.
[22] S. Kapoor & A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science", Patterns **4** (2023) 100804. https://doi.org/10.1016/j.patter.2023.100804.
[23] E. Yagis, S. W. Atnafu, A. García Seco de Herrera, C. Marzi, R. Scheda, M. Giannelli, C. Tessa, L. Citi & S. Diciotti, "Effect of data leakage in brain MRI classification using 2D convolutional neural networks", Scientific Reports **11** (2021) 22544. https://doi.org/10.1038/s41598-021-01681-w.
[24] A. Adegoke-Elijah, T. O. Ojewumi & K. O. Jimoh, "ECG anomaly detection: A deep learning perspective with LSTM encoders", African Scientific Reports **4** (2025) 359. https://doi.org/10.46481/asr.2025.4.3.359.
[25] A. A. Keji, O. Fakeye, N. N. Onochie & O. Sangotoki, "Predicting long-term deposit customers using convolutional neural network and data conversion technique", African Scientific Reports **3** (2024) 192. https://doi.org/10.46481/asr.2024.3.3.192.
[26] A. E. A. Mohammed, H. Mwambi & B. Omolo, "Integrative approaches of multivariate convolutional neural networks with a dynamic conditional correlation model for forecasting stock market volatility", Recent Advances in Natural Sciences **3** (2025) 197. https://doi.org/10.61298/rans.2025.3.2.197.