Convolutional Transformer for Sensor data on Human Activity Recognition: Classification and Feature Extraction

Simone De Renzis[†], Davide Varotto[‡]

***Abstract*—The expanding use of wearable accelerometers has transformed Human Activity Recognition (HAR). These sensors, present in various devices, capture valuable movement data. The data from accelerometers have the possibility to enhance diverse fields including healthcare, sports, and human-computer interaction.**

The primary focus lies in exploring a novel approach to HAR using a model based on convolutional transformers, here named SCvT (Sensor Convolutional Transformer), combining the strengths of Convolutional Neural Networks (CNN) and the transformer architecture. Unlike standard methods usually employed in this domain, this model operates directly on raw sensory data, bypassing the need for data augmentation and signal processing techniques.

Additionally, an investigation into the significance of accelerometer positioning on activity prediction was conducted, showing insights into the varying contributions of different accelerometer positions. The study revealed that certain positions, such as the wrists, might contribute relatively less information for predictive tasks compared to other locations like hips and ankles.

The evaluation of the SCvT model across different window sizes demonstrated its robustness and consistent high performance, surpassing existing models in accuracy and automating the data processing pipeline, with performance exceeding 90% across all metrics.

***Index Terms*—Deep Learning, Human Activity Recognition, Sensors, Transformers, Signals**

I. Introduction

In recent years, the proliferation of wearable devices equipped with accelerometers has revolutionized the landscape of Human Activity Recognition (HAR). Accelerometers are sensors designed to measure acceleration forces. They work based on the principles of physics, primarily utilizing the force applied to a mass to generate an electrical signal proportional to the force experienced [1].

These sensors are commonly used in various devices, including smartphones, wearable gadgets, industrial equipment, and vehicles, to detect changes in speed, direction, and movement. Accelerometers have become an easy way to collect a vast amount of data, which can be used to monitor and diagnose activities and behavious. The analysis of this data can benefit in the fields of healthcare, as it can support continuous patient movement monitoring, aiding eldercare, rehabilitation, and conditions like Parkinson's disease. They are also used in sports and fitness, and human-computer interaction [2] [3].

In this study we focus on human activity recognition, the task of identifying the specific actions individuals are performing based on the recorded signals data. Numerous approaches have been explored to analyze this type of data, from traditional signal processing methods to more recent machine and deep learning techniques. While signal processing techniques may offer deeper insights and interpretability regarding data composition, they often fall short in performance compared to deep learning methods in classification tasks.

Most state-of-the-art deep learning methods on this domain rely on Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). We propose a deep learning model based on an adaptation of the vision transformer model, incorporating convolution: it will be called SCvT (Sensor Convolutional Transformer) from now on for brevity. This hybrid approach aims to exploit the superior performance of transformers while addressing their limitations related to large dataset requirements. We demonstrate the successful application of our approach directly on raw sensory data, bypassing the requirement for feature extraction methods such as FFT or DFT. Also, unlike many existing methodologies that rely on a sliding window technique to segment data, we aim at make it superfluous with this model. By eliminating specific pre-processing techniques, our process becomes more automatic, straightforward, and universally applicable.

Lastly, we introduce a method to examine the model's hidden layers to gain insights into feature importance. Our specific focus is to identify which accelerometers provide the most valuable information for analysis.

To summarize, our contributions involve a deep learning model that combines transformer and convolutional elements, eliminating the need for data segmentation and offering insights into feature importance. This advancement not only enhances performance but also facilitates real-time applications on less powerful devices.

This report is structured as follows. In Sec. II we describe the state of the art of the models and preprocessing techniques used for HAR, the data processing pipeline and the feature extraction technique are respectively presented in Sec. III and IV. The model is detailed in Section V and its performance evaluation is carried out in Sec. VI. Concluding remarks are provided in Sec. VII.

[†]Department Mathematics, University of Padova, email: {simone.derenzis}@studenti.unipd.it
[‡]Department Mathematics, University of Padova, email: {davide.varotto}@dei.unipd.it

II. Related Work

Considerable research has focused on analyzing sensory data from devices like accelerometers and gyroscopes for HAR, and numerous datasets exist.

Our study concentrates on distinguishing between three activities: walking, ascending stairs, and descending stairs. Existing approaches generally fall into two categories: one involves feature extraction based on signal processing theory, while the other utilizes deep learning models for automated feature extraction.

In the first approach, as demonstrated in [4], specific methods such as Short-time Fourier Transform (STFT), Discrete Wavelet Transform (DWT), and time domain feature extraction are employed. These techniques derive various features related to frequency, energy distribution, and statistical metrics from sensor data windows. The resulting features are then utilized with machine learning classifiers like Random Forests and Support Vector Machines (SVM) for activity classification.

While this process is valid, the classification outcomes do not achieve notably high accuracy. For instance, when combining data from all sensors, the accuracy for descending stairs falls between 70% and 74%, and for ascending stairs, it ranges between 67% and 71% [4]. We will focus on improving the classification performance on them.

In the context of deep learning methodologies, as detailed in the article at [5] two primary data formatting paradigms are discussed. The first approach treats each input dimension as an individual channel and applies 1D convolution on a temporal window. However, this method doesn't take into account the spatial relationships within or between the sensors. The second approach involves resizing the input data into a virtual 2D image and applying 2D convolution. This second approach leverages the spatial information present in the sensor data. The paper also discusses the extraction of additional features such as FFT and Wavelet Transformation (WT), which are then integrated as supplementary information into the dataset. To avoid the need for feature extraction, we opted for the 2D approach using raw sensor data, as this allows us to retain the maximum amount of information. By carefully designing filter shapes, we can effectively exploit the spatial information to achieve favorable results.

Deep learning has revolutionized the landscape of artificial intelligence and data processing, offering methods to handle complex data in various domains.

CNNa specialize in visual data analysis, excelling in image and video recognition tasks. They leverage convolutional layers to identify spatial patterns and hierarchical structures within images, enabling robust feature extraction and classification [6].

RNNs are focused on sequential data by maintaining memory across the sequence. Their recurrent connections allow feedback loops, enabling the network to retain information about previous inputs. This makes RNNs suitable for tasks such as natural language processing, time series analysis, and speech recognition.

Long Short-Term Memory (LSTM) networks adresses the vanishing gradient problem, a common issue in traditional RNNs. LSTMs include specialized memory cells that can store and access information over extended time periods, making them adept at capturing long-range dependencies in sequences.

Transformers, a relatively newer architecture, introduced a paradigm shift in natural language processing. They rely on self-attention mechanisms, enabling parallel computation and capturing relationships between words in a sentence or document. Transformers are highly efficient in tasks such as language translation, text summarization, sentiment analysis, and question-answering systems.

The distinctive feature of transformers lies in their ability to understand contextual relationships between different elements in the input sequence. They achieve this through attention mechanisms, which allow them to focus on relevant parts of the input, enabling a deeper understanding of the context and relationships between different elements.

In exploring solutions for the problem, the literature has applied a various number of models and architectures. Researchers have employed techniques ranging from CNNs [7] RNNs and LSTMs [8] to Transformer models [9]. The afromentioned work used vision-adapted Transformer architectures for addressing this challenge. However, it also needs some augmentation steps that may not be necessary when leveraging convolutional transformers.

Visual Transformers, inspired by the success of Transformers in natural language processing, adapt these architectures for image-based tasks. These models operate on a grid of image patches, treating them as tokens similar to words in natural language, allowing global context understanding [10]. On the other hand, Convolutional Transformers blend the strengths of both CNNs and Transformer architectures. They integrate convolutional layers, known for their spatial hierarchies and translational invariance, with the self-attention mechanism of Transformers. This fusion creates a model capable of capturing spatial relationships in visual data while also learning global dependencies, thus enhancing the ability to understand context and relationships in images [11].

In our work, we present and adapt this architecture to handle sensory data, demonstrating that employing convolutional approaches eliminates the need for additional data augmentation. This adaptation emphasizes the capability of convolutional methods in understanding and processing sensory information.

III. Processing Pipeline

The dataset comprises unprocessed data captured through accelerometers, measuring motion, collected from 32 individuals without health issues. This data was gathered during various activities such as outdoor walking, climbing stairs, and driving. The recording devices were placed on four specific

body locations: the left wrist, left hip, left ankle, and right ankle [4].

We focus on three specific activities: walking, ascending stairs, and descending stairs due to their similarities and the challenge in distinguishing between them. To process these activities, we transform the data into virtual 2D images and segment it into fixed-size time windows. These time windows serve as inputs for testing various architectures in our pipeline. The pipeline is illustrated in Fig. 1.

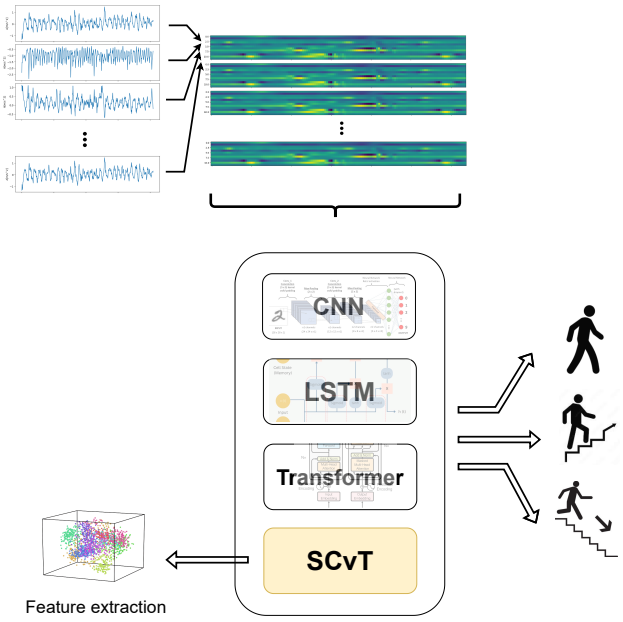

Fig. 1: Visualization of the main steps of the processing pipeline.

As for the SCvT model, it is composed by the following blocks:

- A convolutional layer, which is a convolutional neural network that extracts features from the input sensor data and converts them into activations. This component has two convolutional layers, each followed by a ReLU activation and a max pooling layer.
- A attention block layer, which is a stack of transformer blocks that encode the tokens into latent representations. Each attention block has a multi-head attention layer and a feed-forward network layer, both followed by layer normalization and dropout layers.
- A mlp head layer, which is a multi-layer perceptron that outputs the final predictions for the input sensor data. The mlp head layer has two dense layers, the first one with a gelu activation and the second one with a linear activation. The output dimension of the mlp head layer corresponds to the number of classes to be predicted.

The architecture will be explained in detail in Sec. V.

IV. Signals and Features

The study carried out in [4] involved 32 healthy adults (13 men, 19 women) between the ages of 23 and 52. These individuals wore 3-axial ActiGraph GT3X+ accelerometer devices at four different body locations (left wrist, left hip, left ankle, and right ankle) simultaneously. The accelerometers collected raw data at a sampling frequency of 100 Hz during activities such as outdoor walking, stair climbing (both descending and ascending), driving. To accurately mark the beginning and end times of various activities, participants were instructed to perform three claps both at the start and conclusion of each activity. Each data point is labeled with the corresponding activity type.

A. Data cleaning

As previously stated, our focus excludes the activity of driving, leading to the removal of all recorded data associated with this specific activity from the dataset. Additionally, all data concerning the clapping used to indicate recordings was eliminated.

B. Window sizes

Each accelerometer captures information across three axes: x, y, and z. With the use of four accelerometers, a total of 12 different variables are recorded for each moment of data collection. The data is transformed into a 2D image format, where each row represents an axis reading from an accelerometer. The resulting images are 12 units in height, with the length determined by the chosen window size.

In some experiments, only one accelerometer is considered to assess the sensor's position relevance. For these specific cases, the image's height is reduced to the 3 specific axis involved.

Regarding the window size, one of our contributions is evaluating its influence on the models. We implemented two window sizes: 256 and 2048, which roughly equate to 2.5 seconds and 20 seconds respectively. The choice of 256 is based on it being the minimum size required to involve at least two cycles of the activity. The selection of 2048 offers the longest possible sequence length without needing to pad specific (shorter) activities. Using a longer sequence enables the retention of maximal data. In both cases, there is no overlap allowed between consequent windows.

C. Train, validation and test sets

The dataset was divided into distinct subsets - training, validation, and test sets - following a specific approach. The division method involved assigning all data related to the three activities for each of the 32 participants into the same subset. This was aimed at preventing any inadvertent information leakage or influence across the subsets. The test set is distinct and includes data from 6 individuals, while the validation set comprises data from 4 individuals. The validation set changes for each of the 5 iterations conducted for every model. This was done to ensure more robust and varied measurements across the different runs.

D. Feature extraction with convolutional layers

The objective of this part of work is to enhance the interpretability of the model and gain valuable insights into the significance of various features. We refer here to the SCvT model as described in Sec. V. A key aspect of our research is to determine the accelerometers that contribute the most informative data regarding an individual's activities. In simpler terms, we aim to identify the optimal placement for an accelerometer to effectively recognize activities, thereby reducing the overall costs associated with using multiple recording devices.

To achieve this goal, we initially trained the model using a dataset that includes data from four accelerometers. Subsequently, we performed an analysis of the convolutional layers and the projection layer.

Fig. 2: Example of kernel fitlers of the first convolutional layer, visualized as images.

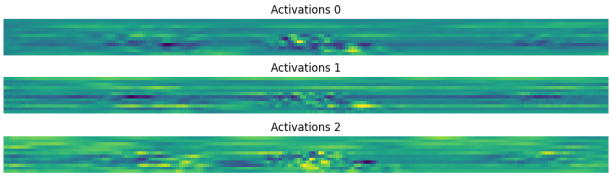

Fig. 3: Example of activation for the first convolutional layer.

Fig 2 presents a visual representation of kernel weights obtained during the model training phase. In the context of image data, an examination of these kernels can give valuable insights into the model's learning process. Each kernel is responsible for detecting specific visual patterns, including edges, textures, shapes, or more complex structures, depending on the network's architecture and training dataset. Over the course of training, the network fine-tunes these kernel values to gain features critical for accurate classification or detection tasks.

However, when dealing with time series numerical data, such as those found in accelerometer readings, the utility of inspecting kernels may be limited. This is because time series lack the graphical structures like edges or textures that are interpretable by human perception.

Certain insights can be obtained from activation maps: the kernel filters convolve with specific segments of the input image, extracting various features. The resulting feature maps reveal areas of greater activation, meaning where specific features are detected within the input.

Fig. 3 illustrates how the kernel activations manifest in the training data. All activations are cropped to a width of 200 for improved visualization. Fig. 4 showcases the same activations respectively before and after passing through the ReLU function, which effectively sets negative values to zero, making it easier to identify regions of increased activation (indicated by lighter-colored pixels). This helps in detecting the areas that provide more information for classification. It becomes evident through visual inspection that most significant features are concentrated in the lower channels of the activation map.

To further validate this observation, we analyze the activation shown in Fig. 5. This layer has undergone processing with a max-pooling layer using a stride of three. This pooling is performed to consolidate information from each axis of the accelerometers (x, y, z) and condense the information from each accelerometer into one row. The visualization underscores that the first row (corresponding to the first accelerometer) does not exhibit significant activation. This observation implies that the first accelerometer may not make a substantial contribution to the activity detection process.

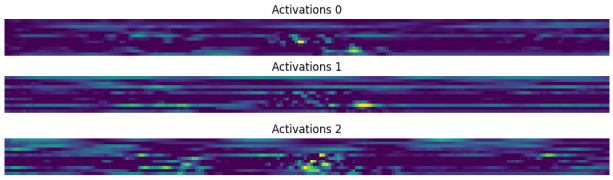

Fig. 4: Example of activation for the first convolutional layer, after ReLU application.

Fig. 5: Example of activation for the Max-Pooling layer.

To formalize this concept, we propose a way of combining and summing up all the activation values across different channels and spatial positions within the dataset. By collapsing the information along the data, channel, spatial height, and spatial width dimensions, the resultant summary provides an aggregated representation for each row in the dataset.

This aggregation process enables the extraction of row-specific summaries, revealing cumulative information from the activations across the entire dataset. Each resulting value in the summary represents the accumulation of activation details from multiple channels and spatial positions.

The aggregation of activations grouped by rows can be represented symbolically as:

$$\text{features}_{ij} = \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{m=1}^{M} \sum_{n=1}^{N} \text{activation}_{klmn}$$

Where:

- i represents the row index.
- j represents the column index (not grouped).
- K denotes the number of elements along the first dimension (data samples).
- L denotes the number of elements along the second dimension (spatial height: number of filters).

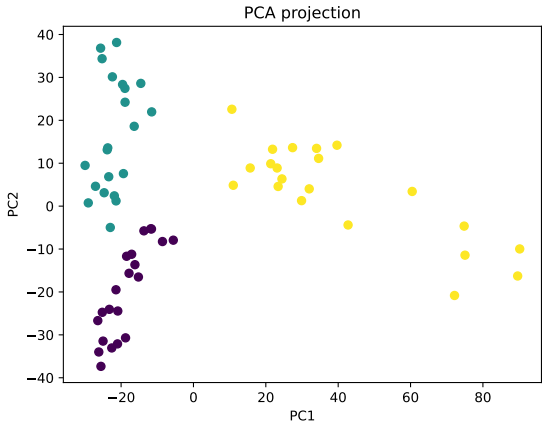

Fig. 6: Visualization of the projection layer, compressed with PCA.

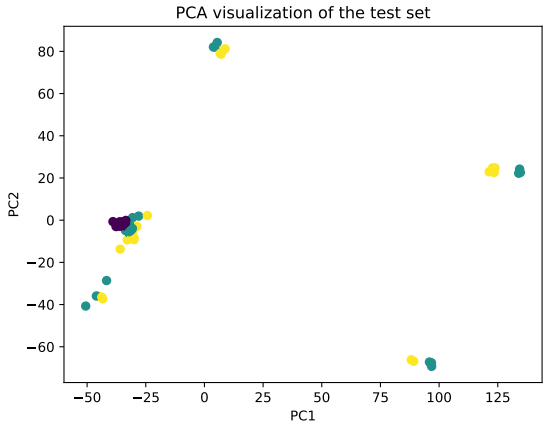

Fig. 7: Visualization of the raw dataset (test set), compressed with PCA.

- M denotes the number of elements along the third dimension (spatial width).
- N denotes the number of elements along the fourth dimension (number of channels).
- activation$_{klmn}$ signifies the activation value at the indices $k, l, m,$ and n within the layers tensor.

E. Feature extraction with projection layer

Another approach we used to gain insights into the feature involved an examination of the projection layer subsequent to the convolutional layer. During this stage, the final activations of the convolutional layer are projected into a reduced-dimensional space (specifically, 32 dimensions across 4 channels). Subsequently, these projections are fed into the attention block.

Post-training, this lower-dimensional space is intended to function as a summarizer and compressor of the information. The idea is that, if the training process has effectively engendered meaningful representations of the data, this layer should offer a concise yet meaningful characterization of our dataset.

The visualization in Fig. 6 shows these inner layer activations derived from the test set. These activations are further compressed into a 2D space using Principal Component Analysis (PCA). It is evident from the visualization that PCA organizes the data into three distinct clusters, each corresponding to one of the three classes. The visible separation between these clusters signifies that the network has learned a meaningful compression. As a point of comparison, Fig. 7 demonstrates the application of PCA to the raw test set, wherein no significant separation is observed. This serves as evidence supporting the assertion that the projection layer has indeed contributed to the meaningful characterization of the data.

In order to assess the relative contributions of each accelerometer, a K-means clustering method was employed to cluster the (32x4) vectors in an unsupervised manner. Given that each of the four vectors encapsulates information from a distinct accelerometer, we individually presented each of the four 32-dimensional vectors to the K-means algorithm. The underlying idea is that the ease with which K-means accurately clusters these vectors with respect to the ground truth label indicates the significance of each accelerometer's contribution. Detailed results from this analysis will be presented in Sec. VI.

V. Learning Framework

The following section presents the main characteristics of the various models put at test.

A. SCvT

1) The `AttentionBlock` class forms the core building block of the Transformer architecture. It includes:
   - Multi-head Self-Attention Mechanism: Utilizes the `tf.keras.layers.MultiHeadAttention` layer with 4 heads and 32 units to enable the model to focus on different parts of the input sequence simultaneously.
   - **Feed-Forward Neural Network (FFN)**: Implemented as a sequence of dense layers with 128 units and GELU activation.
   - **Normalization and Dropout**: Incorporates layer normalization with an epsilon value of 1×10^{-6} and dropout of 0.1 to enhance the model's learning and generalization capabilities.
2) The `SCvT` class represents the overall model architecture tailored for sensor data processing. It contains the following components:
   - **Positional and Class Embeddings**: Learnable embeddings of shape $(1, 5, 32)$ for positional information and $(1, 1, 32)$ for class-specific information to provide context to the input data.
   - **Projection**: Employs a dense layer to transform input activations into a format compatible with the Transformer model using 32 units.
   - **Convolution Block**: Utilizes a set of convolutional layers (`ConvolutionBlock` class) to extract and

process activations from the input data. The first convolutional layer uses 50 filters with a kernel size of 3×24 and no biases. The subsequent ReLU activation is followed by max-pooling with a pool size of 12 and strides of 3. The next convolutional layer employs 25 filters with a kernel size of 4×24 and no biases.

- **Transformer Layers**: Stacks multiple instances of the `AttentionBlock` class (in our case 2) using the specified hyperparameters.
- **MLP Head**: Concludes the model with a multi-layer perceptron (MLP) composed of two dense layers. The first dense layer has 128 units and uses the GELU activation, while the final layer produces predictions based on the number of output classes with a softmax function.

3) The `ConvolutionBlock` class is responsible for extracting and converting the input data using convolutional operations. This class helps in breaking down the input into smaller, more manageable segments for subsequent processing within the Transformer architecture.

This design aims to effectively process and extract meaningful patterns from sensor data, combining the strengths of both convolutional and self-attention mechanisms in a Transformer-based model. A full visualization of the architecture is provided in Fig. 8

B. CNN

The CNN model is designed as follows:

- **Input Layer**: Expects grayscale images with dimensions 12×2048.
- **Convolutional Layers**: Two layers; the first with 512 12×12 filters, the second with 256 4×4 filters. Both use ReLU activation.
- **Pooling**: Utilizes max-pooling with a 12×12 pool size to reduce spatial dimensions.
- **Dropout Layers**: Incorporated after each convolutional layer to prevent overfitting, with a dropout rate of 0.5.
- **Flattening Layer**: Converts convolutional outputs into a one-dimensional array for dense layers.
- **Dense Layers**: Comprises a 128-neuron layer with ReLU activation and a final 3-neuron layer using softmax activation for multi-class classification.

C. LSTM

The LSTM model is designed as follows:

- **LSTM Layers**:
  - First LSTM: 1024 units, input shape 12×256, returning sequences.
  - Second LSTM: 512 units.
- **Dense Layer**: 3 neurons, softmax activation for multi-class classification.

D. Transformer

This architecture represents a standard implementation of the Transformer model. The hyperparameters utilized in this configuration follows.

The multi-head attention mechanism employs a head size of 256, the number of attention heads utilized is 4, enabling the model to perform multi-faceted analysis on the input sequences. The dimensionality of the feed-forward layers within the Transformer blocks is set at 4. The architecture includes 4 stacked transformer blocks to facilitate learning complex patterns in the data. The subsequent feed-forward network after the transformer blocks consists of a single layer with 256 units. A dropout rate of 0.25 is applied within the transformer encoder blocks to prevent overfitting during training, while a dropout rate of 0.4 is employed within the feed-forward layers.

This is a standard implementation of transformers and serves as a baseline to confront the SCvT one.

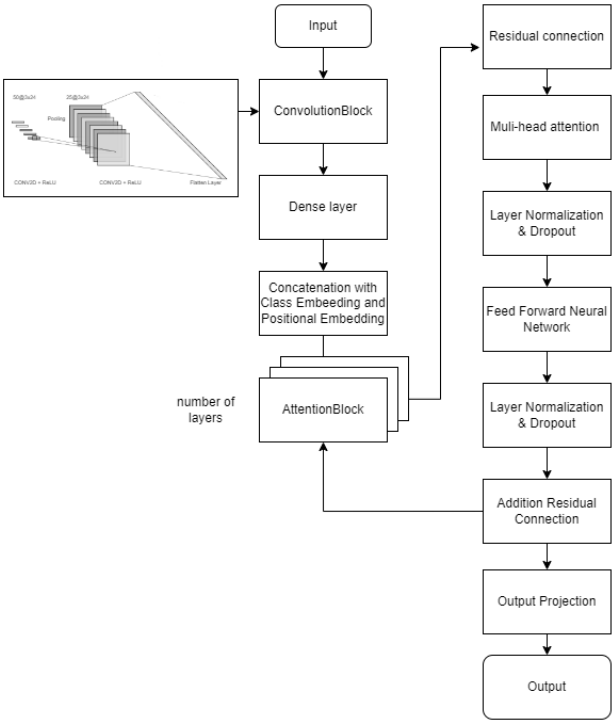

Fig. 8: Visualization of the SCvT architecture.

VI. Results

A. Windows sizes

A comprehensive comparison detailing the performance across various window sizes has been documented in Tab. 1. The reported precision, recall, and F1 scores is related to each model evaluated under the window sizes of 256 and 2048. These metrics represent averages derived from five distinct training sessions conducted with different validation sets and were assessed on the respective validation set. This method

of cross-validation has been employed due to the dataset's limited size, as a technique to assess model performance robustness and ability to generalize. We can see how the SCvT model greatly outperforms all the other models both on both windows sizes. We note how smaller sliding windows benefits the other models, which otherwise struggled with the 2048 one. Our model handles both sizes with almost identical performance, gaining over 90% of scores on all the metrics. Performances on the standard Transformer are low, probably due to the limited amount of data available to exploit the standard implementation.

TABLE 1: Models metrics on windows sizes, performance on the validation set. Precision, Recall and F1 are weighted averages.

	256			2048		
	Prec	Recall	F1	Prec	Recall	F1
CNN	0.7500	0.6217	0.5845	0.5870	0.6348	0.5860
LSTM	0.7438	0.6722	0.6603	0.4548	0.4538	0.4230
Transformer	0.4686	0.4537	0.4456	0.3659	0.3650	0.3629
SCvT	**0.9344**	**0.9298**	**0.9296**	**0.9198**	**0.9047**	**0.9014**

Presented in Fig. 9 and 10 is a boxplot visualization showcasing the F1 scores and their respective variances across the five runs for each model at windows sizes of 256 and 2048 respectively. It allows us to observe that the variations in F1 scores are similar across all models. This comparative analysis serves to confirm that when the models are run multiple times, they do not exhibit significantly different results, particularly due to the influence of a specific, less favorable validation dataset sample.

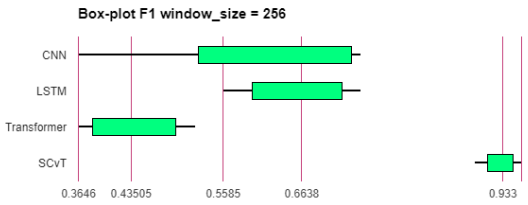

Fig. 9: Box-plot with F1 score related models trained with window size = 256.

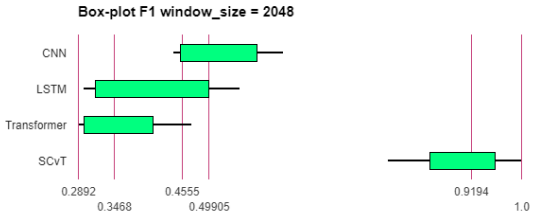

Fig. 10: Box-plot with F1 score related models trained with window size = 2048.

Tab. 2 comprises a table illustrating the performance of each model across the two window sizes on the test set. This evaluation represents a single run, maintaining the test set as a non variable entity. Notably, the models were retrained using the entire training and validation set combined to maximize the utilization of available data. Thanks to this, we see their performances raising, but maintaining the same pattern as seen for the validation set. This further confirms the high accuracy of the SCvT model, which scores over 98% on all the metrics. On the 2048 windows size, they even reach perfect results. These results also outperforms the one reached in [4] which stands between 70% and 74% when combining all sensors.

TABLE 2: Models metrics on windows sizes, performance on the test set. Precision, Recall and F1 are weighted averages.

	256			2048		
	Prec	Recall	F1	Prec	Recall	F1
CNN	0.8756	0.8210	0.8094	0.8944	0.8888	0.8893
LSTM	0.8341	0.8192	0.8170	0.5480	0.5555	0.4648
Transformer	0.5643	0.5245	0.5179	0.4932	0.4761	0.4617
SCvT	**0.9844**	**0.9842**	**0.9842**	**1.0000**	**1.0000**	**1.0000**

B. Division per class

We are also interested in addressing the difficulty associated with classifying each specific class. Table 3 illustrates that the "walking" class obtains a higher accuracy in its detection. This class stands out as it differs from the other activities, which, on the other hand, shows greater similarity in the movements executed by the individual.

TABLE 3: The three activities classed are compared on how well they are classified by the models. Precision and Recall are class specific.

	Walking		Ascending		Descending	
	Prec	Recall	Prec	Recall	Prec	Recall
CNN	0.6525	0.6952	0.5510	0.7618	0.5576	0.4475
LSTM	0.3653	0.6093	0.5698	0.4189	0.4294	0.3999
Transformer	0.4192	0.3999	0.3515	0.3713	0.3270	0.3237
SCvT	0.8924	1.0000	0.8845	0.8412	0.9824	0.8730

In Fig. 11 is a confusion matrix providing a relative analysis concerning the SCvT model and its classification performance across the three distinct classes. We can see how most of the errors are between the classes ascending and descending stairs.

C. Accelerometer position

A key objective of this study was to investigate the influence of accelerometer positioning on activity prediction.

Tab. 4 shows the results derived from the methodology described in Sec. IV-D for extracting feature importance from the convolutional layers. These results show the significance and impact of various features extracted from the convolutional layers, providing insights into their respective importance for the predictive task. We can see that the values are similar for the accelerometers located in hips, left and right ankles. The accelerometer placed on the wrist has a markedly lower value: as the sum expresses the activation of the convolutional layers over the accelerometer, this result suggests that wrist sensor doesn't bring as much information as the others.

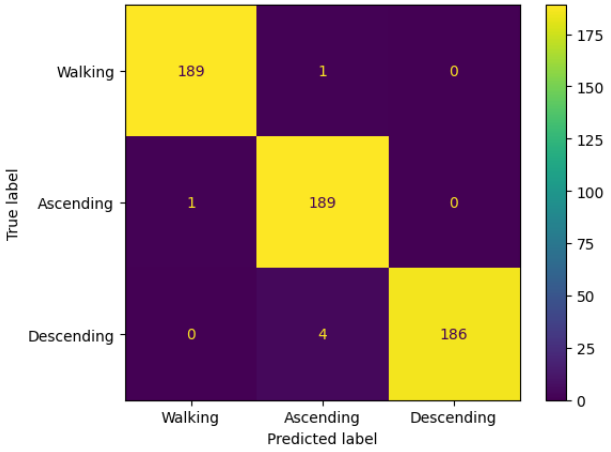

Fig. 11: Confusion Matrix SCvT model with window˙size = 256.

TABLE 4: Feature extraction from convolutional layer

	Cumulated sum
Wrist	623685.4
Hips	**734796.25**
Left Ankle	**735297.6**
Right Anlke	708355.94

Table 5 showcases the K-means analysis on (32x4) accelerometer vectors, aiming to assess each accelerometer's contribution. By individually clustering the 32-dimensional vectors from distinct accelerometers, it evaluates their effectiveness in accurate clustering based on accuracy. The table presents how well K-means clustered the data, revealing the varying significance of each accelerometer in data representation and modeling. Again from this result emerges that the wrists accelerator performs slightly worse than the others.

TABLE 5: Feature extraction from projection layer with K-means

	Accuracy
Wrist	0.9523
Hips	**0.9682**
Left Ankle	**0.9682**
Right Anlke	**0.9682**
All	**0.9682**

To confirm the findings, the SCvT model underwent training using datasets uniquely containing data from individual accelerometers. The varying performance of the model concerning each accelerometer provided insights into the informational contribution of each accelerometer. A higher performance observed for a specific accelerometer shows its greater significance in the predictive task. The results are in line with what observed with the feature extraction methods, seeing the wrist accelerator bringing the less information, while the others, particularly the left ankle one that reaches perfect accuracy, are by themselves enough to provide high quality classification results.

TABLE 6: SCvT results when trained on single accelerometer data. 2048 window size. Precision, Recall and F1 are weighted averages.

	Prec	Recall	F1
Wrist	0.6166	0.6349	0.6173
Hips	0.9420	0.9365	0.9346
Left Ankle	**1.0000**	**1.0000**	**1.0000**
Right Anlke	0.9689	0.9682	0.9682

D. Hyperparameters

In Table 7, we present several examples with different hyperparameters experimented with in the SCvT model. These instances served as illustrative examples that informed the final decision regarding their selection. The number of layers and the inclusion of the max pooling layer emerged as important factors. Conversely, the length of the dense projection layer did not significantly impact performance. Although modifications were made to other aspects of the architecture, such as the number of attention blocks and heads, these adjustments did not provide noticeable differences in performance. The exploration of these hyperparameters was conducted manually, without employing grid search or other optimization techniques.

TABLE 7: SCvT hyperparameter search. For convolutional and max-pooling layer, the notation presents the filter dimension [x,y] along with the stride [m,n]

Conv_1	MaxPool	Conv_2	Proj_layer	F1
[3,12],[1,8]	-	-	32	0.7386
[3,24],[1,1]	-	-	32	0.8110
[3,24],[1,1]	-	[4,24],[1,1]	32	0.8596
[3,24],[1,1]	[12,12],[3,3]	[4,24],[1,1]	128	0.9674
[3,24],[1,1]	[12,12],[3,3]	[4,24],[1,1]	64	0.9862
[3,24],[1,1]	**[12,12],[3,3]**	**[4,24],[1,1]**	**32**	**1.0000**

VII. Concluding Remarks

The study systematically investigated and compared various models for activity recognition based on accelerometer data. The evaluation was conducted across different window sizes: the results highlighted the superiority of the SCvT model, showing its robustness in handling different window sizes. Notably, the SCvT model exhibited remarkable consistency, achieving over 90% on all evaluation metrics for both window sizes, outperforming other models such as CNN, LSTM, and Transformer. It maintained its exceptional performance, exceeding 98% across all metrics on the test set, a notably higher accuracy compared to existing studies [4]. By alleviating the need to worry about selecting window sizes, along with other preprocessing methods and data augmentation (as referenced in [4]), this significantly enhances the automation of the entire pipeline.

The investigation into the influence of accelerometer positioning on activity prediction revealed interesting insights. The evaluation of feature importance from convolutional layers suggested that the wrist accelerometer might contribute comparatively less information for the predictive task compared to other positions. K-means clustering further

supported this observation, indicating slightly lower accuracy for the wrists' accelerometer. Additionally, training the SCvT model with datasets unique to each accelerometer reinforced the importance of accelerometer positioning in the predictive task, with individual performances varying across accelerometers. This experiment confirmed the importance of certain positions, such as hips and ankles, in achieving higher predictive performance. These results are in line with [4] and can help as a way to reduce the number of required accelerometers to promote a wide scale use of the sensors' data analysis techniques.

Future research could explore how well the suggested architecture and methods perform when applied to more diverse and larger datasets. These datasets could include a wider variety of activities, involve more people, and cover different settings. The goal would be to test whether the model remains effective when dealing with more complex datasets. Additionally, there's a possibility to investigate creating smaller, more efficient models that can run on low-powered devices like watches and wearable bands. This is important considering that transformer models can be quite demanding on memory and resources. The aim would be to find a balance between model size and performance, making them suitable for use on devices with limited resources.

Project observation

This project provided a practical test for applying the concepts learned throughout the course. It involved technical skills in experimenting with different architectures in a real-world scenario, allowing for the exploration of new ideas and getting original outcomes. Beyond technical abilities, it also tested the capability to read and comprehend scientific literature in the field of machine learning, a task involving going through numerous findings and studies.

Composing the paper was a test of addressing the main aspects of the work while effectively presenting them with limited space. A clear and concise writing style was important for making sure that the reader understand the work without unnecessary elaboration.

The primary challenge encountered was coming up with an original idea in the domain of sensor accelerometry analysis, a field that has been explored extensively with various techniques and architectures. Although acknowledging that this work might not represent a groundbreaking scientific discovery, we found it valuable and interesting due to the exploration of newer models in addressing this established problem and offering distinctive perspectives.

REFERENCES

[1] C. Yang and Y. Hsu, "A review of accelerometry-based wearable motion detectors for physical activity monitoring," *Sensors*, vol. 10, pp. 7772–7788, Aug. 2010.

[2] M. S. J. H. N. W. G. T. H. V. Z. C. C. Marta Karas, Jiawei Bai and J. K. Urbanek, "Accelerometry data in health research: challenges and opportunities," *Stat Biosci*, vol. 10, pp. 210–237, July 2019.

[3] L. Sánchez-Reyes, J. Rodríguez-Reséndiz, G. N. Avecilla-Ramírez, M. García-Gomar, and J. Robles-Ocampo, "Impact of EEG parameters detecting dementia diseases: A systematic review," *IEEE Access*, vol. 9, pp. 78060–78074, May 2021.

[4] A. S. L. X. C. A. H. J. Fadel WF, Urbanek JK, "Differentiating between walking and stair climbing using raw accelerometry data," *Stat Biosci*, vol. 2, pp. 334–354, Nov. 2019.

[5] T. T. Alemayoh, J. H. Lee, and S. Okamoto, "New sensor data structuring for deeper feature extraction in human activity recognition," *Sensors*, vol. 21, p. 2814, Mar. 2021.

[6] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Q. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, p. 53, Mar. 2021.

[7] L. Rosafalco, A. Manzoni, S. Mariani, and A. Corigliano, "Fully convolutional networks for structural health monitoring through multivariate time series classification," *Adv. Model. Simul. Eng. Sci.*, vol. 7, no. 1, p. 38, 2020.

[8] T. Zebin, M. Sperrin, N. Peek, and A. J. Casson, "Human activity recognition from inertial sensor time-series using batch normalized deep LSTM recurrent networks," in *40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2018, Honolulu, HI, USA, July 18-21, 2018*, July 2018.

[9] I. D. Luptáková, M. Kubovcík, and J. Pospíchal, "Wearable sensor-based human activity recognition with transformer model," *Sensors*, vol. 22, p. 1911, May 2022.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020.

[11] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," *CoRR*, vol. abs/2103.15808, Mar. 2021.