Online Human Activity Recognition Using Efficient Neural Architecture Search with Low Environmental Impact

Nassim Mokhtari[a], Alexis Nédélec[b], Marlène Gilles[c] and Pierre De Loor[d]

Lab-STICC (CNRS UMR 6285), ENIB,
Centre Européen de Réalité Virtuelle, Brest, France

Keywords: 3D Skeleton Data, Image Encoding, Online Human Activity Recognition, Deep Learning, Neural Architecture Search, FireFly Algorithm, NAS-BENCH-101, Efficiency Estimation, Energy Consumption.

Abstract: Human activity recognition using sensor data can be approached as a problem of classifying time series data. Deep learning models allow for great progress in this domain, but there are still some areas for improvement. In addition, the environmental impact of deep learning is a problem that must be addressed in today's machine learning studies.

In this research, we propose to automate deep learning model design for human activity recognition by using an existing training-free Neural Architecture Search method. By this way, we decrease the time consumed by classical NAS approaches (GPU based) by a factor of 470, and the energy consumed by a factor of 170.

Finally, We propose a new criterion to estimate the relevance of a deep learning model based on a balance between both performance and computational cost. This criterion allows to reduce the size of neural architectures by preserving its capacity to recognize human activities.

1 INTRODUCTION

Many techniques today use a deep learning approach applied to video or temporal data to recognize human activities. An activity is characterized by a spatial and temporal evolution of data, therefore one problem is managing this spatio-temporal evolution with a deep learning architecture.

Some solutions involve using recurrent architectures, such as LSTM, which are able to process sequences (Ordóñez and Roggen, 2016), or using convolutional neural networks to process the data (Li et al., 2018). Other approaches consist of encoding the time into an image and then use traditional image recognition architectures on this image (Liu et al., 2017b; Laraba et al., 2017; Ludl et al., 2019; Mokhtari et al., 2022b; Mokhtari et al., 2023). Whatever the approach, another challenge is to find the best architecture and encoding to optimize the recognition rate.

Furthermore, the number of possibilities to design the architecture is so great that a new challenge is to find it with algorithms named Neural Architecture Search (NAS) (Elsken et al., 2019; Wistuba et al., 2019). They usually test a lot of architectures and run for several days on a GPU before finding a good one. However, today, it is also important to take into account the cost of computational power as it is linked to energy costs and environmental concerns.

(Strubell et al., 2019) showed that AI (training and use of deep neural networks) has a significant impact on the environment through its energy consumption and carbon emission. According to (Schwartz et al., 2019), we should evaluate AI on the basis of its efficiency, alongside accuracy and related metrics. For a certain amount of accuracy, it is better to use the smallest architecture and to find it with less tests.

The aim of the work presented in this article is to find the most appropriate deep learning model for online recognition of human activity based on 3D skeleton data, while addressing the various problems listed above. To this end, two proposals are combined:

1. We applied a metaheuristic named "Improved Fire Fly Algorithm" to avoid the training of networks during a Neural Architecture Search, in order to design our deep learning model. By this way, we decrease the time consumed by classical NAS ap-

[a] https://orcid.org/0000-0002-9402-3638
[b] https://orcid.org/0000-0003-3970-004X
[c] https://orcid.org/0000-0003-1806-1672
[d] https://orcid.org/0000-0002-5415-5505

Mokhtari, N., Nédélec, A., Gilles, M. and De Loor, P.
Online Human Activity Recognition Using Efficient Neural Architecture Search with Low Environmental Impact.
DOI: 10.5220/0012314600003660

In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 2: VISAPP*, pages 357-365
ISBN: 978-989-758-679-8; ISSN: 2184-4321

proaches (GPU based) by a factor of 470, and the energy consumed by a factor of 170.

2. We finally propose a new criterion to estimate the relevance of a deep learning model based on a balance between both performance and computational cost. This criterion allows to reduce the size of neural architectures by preserving its capacity to recognize human activities.

These propositions are in line with current environmental concerns of humanity. We reduce the architecture search time by a factor of 476 compared to baseline techniques, and a factor of 114 compared to (Wang et al., 2023) proposition. We also reduce energy consumption and carbon footprint by a factor of 14 compared to (Wang et al., 2023) proposition. This article details the different algorithms and presents various results that show the relevance of our proposition. Moreover, our proposition could be extended to a lot of applications of deep learning.

The remainder of the document is organized as follows: Section 2 introduces a synthesis of the various works carried out in the of field skeleton data representation, Neural Architecture Search, and neural network efficiency. In Section 3 we detail our proposed method for building our deep learning model, and our proposed method for evaluating neural network efficiency. Section 4, shows the data set that used for the experimental part of the work, addressed in Section 5. Finally, we discuss the results of this work as well as the possible developments to improve the proposed metric in Section 6.

2 RELATED WORKS

In this section we will review existing work related to our proposition to perform online human activity recognition in an efficient way.

2.1 Skeleton Based Human Activity Recognition

According to (Wang et al., 2019), Human Activity Recognition (HAR) falls into two categories: sensor-based (e.g., accelerometers) and video-based (e.g., 3D skeleton data from a Kinect). Successful HAR requires encoding data while handling spatial and temporal dependencies. In this study, we utilize skeletons obtained from an RGB-D camera, but this approach is applicable to various types of spatio-temporal data.

To encode skeletal data, some prior research, such as (Yan et al., 2018; Delamare et al., 2021) and (Chen et al., 2022) proposed to use graphs. This approach effectively addresses both spatial and temporal dependencies, as each node connects with its spatial neighbors (in accordance with skeletal structure) and its temporal counterparts (representing the previous and subsequent states of the joint). Typically, this representation is paired with Graph Convolutional Networks (GCN) (Yan et al., 2018; Delamare et al., 2021; Chen et al., 2022).

Conversely, several studies have employed images to represent skeletal data (Ludl et al., 2019; Laraba et al., 2017; Mokhtari et al., 2022b; Mokhtari et al., 2023). In this approach, each joint is encoded as a pixel within an image. To achieve this, the coordinates (X, Y, Z) of a joint are first normalized and then used to calculate the values for the (R, G, B) color channels. Figure 1 provides an illustration of skeletal data encoding using the Encoded Human Pose Image (EHPI) method introduced by (Ludl et al., 2019).

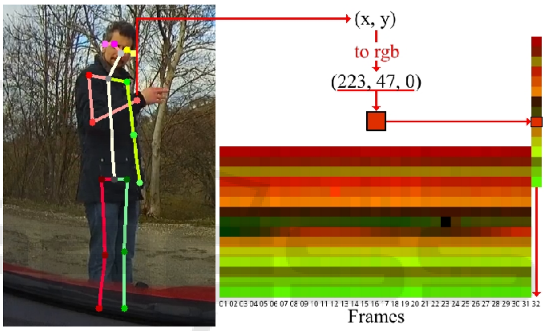

Figure 1: From skeletal joints to an Encoded Human Pose Image (EHPI) (Ludl et al., 2019).

In a related study, (Mokhtari et al., 2022b) introduced the Spatio-Temporal Image Encoding (STIE) technique to represent a sequence of human skeleton data as an image. To enhance the performance of Convolutional Neural Networks (CNNs) and address both spatial and temporal dependencies effectively, they proposed a method of reordering the skeleton data based on the human body's structure. This reordering allocates specific areas of the image to each body part, such as the legs (see figure 2). They also proposed to use the VGG16 model introduced in (Simonyan and Zisserman, 2014), this model was first trained on the *ImageNet Dataset*, then frozen and used as a feature extractor. As a result, the model achieved an accuracy of 86.81% on the Online Action Detection (OAD) dataset (Li et al., 2016).

In an effort to enhance the Spatio-Temporal Image Encoding (STIE) approach, (Mokhtari et al., 2023) introduced the concept of motion energy, originally presented by Liu et al. (Liu et al., 2017b). This concept emphasizes the impact of motion on generating color images by assigning weights to skeleton joints

Figure 2: Writing encoded using the STIE (Mokhtari et al., 2022b).

based on their motion. To create a more comprehensive and detailed representation of human actions, the motion energy image is integrated with STIE, resulting in a comprehensive view of the action (as illustrated in Figure 3). Additionally, the authors retrained the VGG16 model using the Online Action Detection (OAD) dataset. Consequently, this approach achieved an outstanding accuracy rate of 95.22%, establishing a new benchmark for the dataset.

Figure 3: Writing encoded using the ESTIE (Mokhtari et al., 2022b).

Real-time human action recognition faces a significant challenge in identifying the start and end of actions within a continuous data stream. To address this challenge, prior research has suggested utilizing a sliding window approach, which allows the model to be trained on the continuous stream of data. This method has been proposed as a solution to effectively handle this issue in real-time HAR (Liu et al., 2019; Weng et al., 2017; Delamare et al., 2021; Mokhtari et al., 2022b; Mokhtari et al., 2023).

In this work, we are interested to find the most appropriate deep learning model to the OAD dataset, by using ESTIE proposed by (Mokhtari et al., 2023) to encode the skeletal data.

2.2 Neural Architecture Search

One of the challenges in Deep Learning is to design the architecture of the network automatically due to the large number of hyperparameters that can be used and the many possible configurations of the network that these hyperparameters allow.

For this purpose, several works have been interested in the use of metaheuristics. For example, (Sun et al., 2020) used a Genetic Algorithm to automatically design a convolutional neural network by using the trained network accuracy as a fitness function. (Carvalho et al., 2010) proposed a fitness function based on both train and test errors that was used with VNS, SA, GEO and GA algorithms. (Strumberger et al., 2019) preferred to use a FireFly algorithm to design their CNN, where the used fitness function was based on the error computed on the test set. These studies showed that metaheuristics are suitable for neural architecture search, since they are well known for solving combinatorial problems, such as finding a good neural network organisation among several possibilities offered by the search space. However, they can also be slow due to the need to train each architecture to evaluate its quality. (Wang et al., 2023) proposed a multi-objective evolutionary algorithm NAS, by setting weighted f1-score, floating-point operations per second (FLOPs) and the number of parameters as objectives, to speed up search by a factor of 4.17. It is therefore useful to be able to evaluate the quality of an architecture without training it, to save time and computational resources.

(Mellor et al., 2021) proposed a method to evaluate a neural network without previous training by identifying a binary indicator (0 for inactive unit, and 1 for active unit), focusing only on the rectified linear units (ReLU) of an untrained network (randomly initialized weights). The intuition behind their approach is that the more similar the binary codes associated with two inputs are, the more challenging it is for the network to learn how to discriminate these inputs. A mini-batch of data $X = [x_1, x_2, ..., x_n]$ is mapped through a neural network (composed of ReLU and several other units type) to get binary codes $C = [c_1, c_2, ...c_n]$, where each c_i (obtained from ReLU outputs only) refers to the binary code of x_i. To evaluate the neural network, (Mellor et al., 2021) compute the log of the determinant of a matrix Kh, where each component is calculated using the Hamming distance (Eq (1)). The higher the score, the better the network.

$$K_h[i,j] = N_A - Hamming\_distance(c_i, c_j) \quad (1)$$

where N_A is the number of rectified linear units (ReLU).

(Mokhtari et al., 2022a) proposed another way to evaluate a neural network without training. Based on the work of (Mellor et al., 2021), they proposed to use binary codes for calculating an intra-class distance (ICD) for evaluating the ability of an untrained neural network to distinguish data. They propose that the ICD would be more interesting to assess this dispersion of representation, since it is used to assess the quality of clustering, where the objective is to produce distinct clusters. This ICD metric was used as the fitness function of an improved version of the firefly algorithm (IFA) which uses genetic operators (selection, crossover and mutation) to be more robust to local optimum. Algorithm 1 illustrates this proposition. Note that the choice of the optimal solution from the "candidates" list is determined based on the models' performance after training. As the ICD metric serves as an approximation of model quality rather than an exact measurement, maintaining a candidate list enhances the likelihood of identifying a promising architecture (Mokhtari et al., 2022a).

Algorithm 1: Improved FireFly Algorithm.

```
Randomly generate the population
Define MaxChances
chances = MaxChances
candidates = [ ]
$LocalBest$ = NULL
while not Stopping criteria do
    Running an iteration of FireFly
    $Best_t$ = current population's best solution
    if $LocalBest = NULL$ then
        $LocalBest = Best_t$
    else
        if $fitness(Best_t) \geq fitness(Best_{t-1})$ then
            $LocalBest = Best_t$
        else
            chances - -
        end
    end
    if $chances = 0$ then
        add $LocalBest$ to candidates
        $LocalBest$ = NULL
        Perform an iteration of the Genetic Algorithm
        chances = MaxChances
    end
end
Determining the best solution from the candidates list
```

This method outperform existing NAS without training techniques, and obtained results that are close to those obtained by the NAS with training. Consequently, in this work, we focus on the use of the Improved Firefly Algorithm proposed by (Mokhtari et al., 2022a), to build a model dedicated to human activity recognition.

2.3 Neural Network Efficiency

There has been a lot of discussion and debate recently about our energy consumption and the impact it has on the environment.

Recent studies reported by (Strubell et al., 2019) have shown that performing a neural architecture search (including training) for a big transformer emits about 626,155 lbs of CO_2, which is five times more than a car in its lifetime (126,000 lbs).

According to (Schwartz et al., 2019), the amount of computation needed for deep learning research has been growing rapidly, with a doubling rate of every few months. This has resulted in a significant increase in computation, estimated to be 300,000 times higher in 2018 than it was in 2012. This leads the author to propose the evaluation of the neural network efficiency through several indices such as carbon emission, electricity consumption or floating point operations (FPO). (Schwartz et al., 2019) proposed to use the following formula (Eq. 2) to estimate the cost of obtaining a result (R) using according to the cost of executing (E) the model on a single example, the size of the dataset (D) and the number of hyperparameters (H).

$$cost(R) = E \times D \times H \tag{2}$$

This proposed metric illustrates three quantities that are important factors in the cost of generating a result, but it ignores other factors such as the number of training epochs.

In this work, we propose to use a new metric (that goes beyond the proposition of (Schwartz et al., 2019)) to evaluate a deep neural network according to its performance and efficiency including its training.

3 PROPOSED METHOD

In this section, we will outline our proposals to find the most appropriate deep learning model for online human activity recognition based on skeletons. To do that, we propose to apply the Improved FireFly Algorithm proposed by (Mokhtari et al., 2022a) to build a model dedicated to the Online Action Detection dataset. This dataset will be first encoded using the Enhanced Spatio-Temporal Image Encoding (ESTIE) proposed by (Mokhtari et al., 2023).

We also outline our proposal to evaluate a model according to its performance and efficiency by using a benefit cost ratio (BCR). This metric will be used to compare our model to the state-of-the-art methods.

3.1 Deep Learning Model

We use the Improved FireFly Algorithm for which the fitness function of each architecture is evaluated without training according to the work of (Mokhtari et al., 2022a). This method explores the NAS-Bench-101 search space which is of 423,624 neural networks (CNN) (Ying et al., 2019), using the data from the OAD dataset presented in Section 4, that will be encoded using the ESTIE method.

In the NAS-BENCH-101, all the networks share the same pattern, which is composed of one or several stacks, each one includes one or several cells. The networks are different in the « module » (cell), which is represented by directed acyclic graphs (up to 7 vertices and 9 edges). The valid operations at each vertex are 3x3 convolution, 1x1 convolution, and 3x3 max-pooling (Ying et al., 2019).

Figure 4 shows an example of a network architecture from the NAS-BENCH-101 composed of 3 stacks, each one including 3 cells.

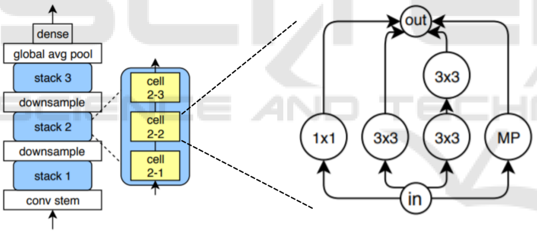

Figure 4: Network architecture in the NAS-BENCH-101 (Ying et al., 2019): Left part is the skeleton shared by all models, middle part is a stack of cells and the right part is an example of a cell (module) (Mokhtari et al., 2022a).

We explored various combinations of the number of stacks and cells, ranging from 1 to 5 for each, a total of 25 network pattern organizations.

3.2 Performance and Efficiency Evaluation

To take into account the training cost of the networks, we propose to modify the formula of Schwartz et al. (Schwartz et al., 2019) (Eq. 2) by introducing the number of epoch used during the training. This training cost can be estimated as follow:

$$training\_cost = E_p \times D \times H \tag{3}$$

where E_p is the number of epochs, D is the size of the training dataset and H, the number of parameters.

This training cost can be used to evaluate the effectiveness of a neural network, in the same way as the total number of floating point operations (FPO) required to generate a result.

Since it is already possible to evaluate the performance of a deep learning model using metrics like accuracy or F1-Score, we propose to combine these performance indices with an estimation of the neural network efficiency using a benefit cost ratio (BCR) as follow:

$$BCR = \frac{P}{C} \tag{4}$$

where P refers to a performance metric which can be the test accuracy, the F1-Score, etc., and the C refers to the network efficiency estimation through the training cost (Eq. 3) or the FPO.

The use of such a metric can allow us to evaluate a neural network in terms of its performance and efficiency, in order to choose a model that performs well but does not consume too much energy.

4 THE OAD DATASET

The proposed method is evaluated on the Online Action Detection dataset (OAD) collected from a Kinect v2, including 25 joints because it provides unsegmented online sequences of skeleton data (Li et al., 2016). It includes 59 long sequences and 10 actions, including drinking, eating, writing, opening a cupboard, washing hands, opening a microwave, sweeping, gargling, throwing trash, and wiping. The training is done on 30 sequences, and tested on 20 sequences. The remaining 9 sequences are ignored in our work, since they are used for the evaluation of the running speed (Li et al., 2016).

Several studies like (Mokhtari et al., 2023; Mokhtari et al., 2022b) and (Delamare et al., 2021) proposed to use a sliding window 40 frames, to segment this dataset, to allow real-time recognition without having to identify the start or end of the action.

5 RESULTS AND DISCUSSION

In this part of the work, we present our experimental results, obtained on the OAD dataset. On the one hand, a comparison between the different results obtained from the neural architecture search using the Improved Firefly Algorithm (IFA) for the OAD dataset is done. Then, the best model obtained from the IFA will be compared to the state-of-the-art models according to the accuracy metric. Finally, we compare our model to the best known model from the state-of-the-art according to our proposed BCR that takes into account the efficiency of the models.

5.1 Neural Architecture Search

In this part of the experiment, we run the IFA method on the NAS-BENCH-101 in order to find the most suitable network for the OAD dataset.

We experimented various combinations of the number of stacks and cells, ranging from 1 to 5 for each. On each combination, the IFA runs for 100 generations, using a population of 20 solutions. All executions were done on a DELL precision 5760, with an Intel(R) Core(TM) i7-11850H as CPU, and 32Gb of RAM.

Table 1 summarises the search time in second for each run, where we can notice that the search time is ranging from 1328 seconds (22 minutes), to 12991 (3 hours and 36 minutes).

Table 1: Search time in seconds for each run of the Improved Firefly Algorithm on NAS-BENCH-101.

	#Cells				
#Stacks	**1**	**2**	**3**	**4**	**5**
1	1328	2875	2719	5690	9728
2	2885	5993	3223	11404	15481
3	2710	7670	7062	8081	8716
4	2931	6685	10765	12991	10059
5	**2989**	5688	7130	12162	10504

Each resulting architecture is trained for 100 epochs on an NVIDIA RTX A3000 Laptop GPU, for 5 times on the segmented OAD dataset using the ESTIE method. Table 2 summarizes the mean test accuracy and the standard deviation (std) obtained by each mode on the OAD dataset, while Table 3 contains the best test accuracy (on a test set not used for the learning phase) for each network.

According to the results presented in Table 2, we notice that the best mean accuracy is obtained by a network architecture composed of 5 stacks, where each one includes only one cell, with a mean accuracy of 93.58% and a standard deviation of 0.5. Followed by the model a smaller architecture, composed of 2 stacks and one cell by stack only, with 93.24% as mean accuracy and 0.45 as standard deviation.

Table 3 confirms the previous observation, by showing that the best model, reaching 94.47% of test accuracy is composed of 5 stacks, each one including only one cell, followed by the one composed of 2 stacks and one cell by stack, reaching 93.77%. It can also be seen from the two tables that the performance of the models decreases when the network is larger than 5 stacks and one cell.

For the rest of the experiment, we choose to keep both well performing models, since the first one (5 stacks, 1 cell) is getting the best accuracy, and the second one (2 stacks, 1 cell) offers an interesting result while being more than 2 time smaller. We will refer to the model composed of 5 stacks and 1 cell as IFA$_{5.1}$ and the one composed of 2 stacks and 1 cell as IFA$_{2.1}$.

5.1.1 Search Time Improvement

The design of IFA$_{2.1}$ required to explore at least 2020 models: 20 by epoch over 100 epochs, in addition to the first population composed of 20 architectures. Evaluating a model using the training-free approach costs only 1 second, for this reason our IFA exploration took 2885 seconds (48 minutes) to generate this model (IFA$_{2.1}$). In the case of classical NAS (that involves training the model to evaluate it), we estimate the required time to perform the same task to be 147 days 7 hours 15 minutes, considering a training over 100 epochs on the Intel(R) Core(TM) i7-11850H which lasts around 6300 seconds (1h45m). Furthermore, we estimate the required time to perform this exploration to be 15 days 21 hours and 48 minutes, considering a training over 100 epochs on the NVIDIA RTX A3000 Laptop GPU which lasts around 680 seconds (11m20s).

These estimations confirm that using NAS with a training-free evaluation is more efficient than classic NAS which use train-based evaluation over 100 epochs, since the IFA is being more than 4400 times faster compared to the version on CPU, and 476 times faster than the GPU version, hence, our proposition improves (Wang et al., 2023) proposition by a factor of 114.

5.1.2 Energy Consumption and Carbon Footprint Improvement

We estimated the carbon footprint resulting from these explorations, and their energy consumption, using Green Algorithms Calculator proposed by (Lannelongue et al., 2021). We found out that the IFA consumes about 298 Wh, while the consumption of the CPU and GPU versions are estimated to be respectively 15.55 MWh and 50.547 KWh. In terms of carbon footprint, the IFA produce approximately 15.26g CO$_{2e}$, while the CPU and GPU versions have a carbon footprint around 797.46 kg CO$_{2e}$ and 2.6 kg CO$_{2e}$. From the above estimations, we notice that training-free NAS are less energy consuming and less polluting by around 52 000 times compared to the

Table 2: Mean test accuracy (from 5 runs) and standard deviation(std) on the OAD dataset, obtained by the result of the Improved Firefly Algorithm search on NAS-BENCH-101, for each combinition.

	#Cells				
#Stack	**1**	**2**	**3**	**4**	**5**
1	91.24%±0.59	92.72%±0.55	91.8%±0.53	92.23%±0.87	91.43%±0.33
2	93.24%±0.45	93.04%±0.66	91.9%±0.59	90.91%±0.68	91.72%±0.45
3	92.43%±0.32	92.06%±0.83	91.94%±0.2	90.99%±0.85	90.8%±1.16
4	91.74%±0.43	91.88%±0.42	91.76%±0.52	93.03%±0.39	89.0%±3.55
5	**93.58%±0.5**	90.9%±0.37	89.98%±0.62	88.3%±3.6	90.07%±0.7

Table 3: Test accuracy on the OAD dataset obtained by the best result of the Improved Firefly Algorithm search on NAS-BENCH-101, for each combination.

	#Cells				
#Stack	**1**	**2**	**3**	**4**	**5**
1	91.91%	93.55%	92.62%	93.04%	91.79%
2	93.77%	93.74%	92.89%	91.96%	92.42%
3	92.92%	93.58%	92.31%	91.84%	92.16%
4	92.28%	92.37%	92.44%	93.49%	92.27%
5	**94.47%**	91.35%	90.83%	91.35%	90.7%

training NAS (over 100 epochs) based on CPU, and around 170 times compared to the training NAS based on GPU.

5.2 Comparison with State-of-Art Methods

We compared our obtained models (IFA$_{2.1}$ and IFA$_{5.1}$) to several existing works on the OAD dataset. Table 4 summarizes the obtained results. We notice that the two models performed well, since they both outperforms state-of-the-art methods, except the ESTIE from (Mokhtari et al., 2023), which obtained 95.22% on the OAD dataset by using a VGG16 model, while IFA$_{5.1}$ obtained 94.47%.

Table 4: Comparison with related works on the OAD dataset according to accuracy.

Method	Authors	Accuracy
JCR-RNN	(Li et al., 2016)	78.8%
ST-LSTM	(Liu et al., 2018)	77.5 %
Attention Net	(Liu et al., 2017a)	78.3%
FSNet	(Liu et al., 2019)	81.3 %
SSNet	(Liu et al., 2019)	82.8%
STIE	(Mokhtari et al., 2022b)	86.81 %
ESTIE	(Mokhtari et al., 2023)	**95.22%**
IFA$_{2.1}$	our proposition	93.77%
IFA$_{5.1}$	our proposition	94.47%

5.3 Efficiency Evaluation

In this part of the work, we will compare our obtained result from the IFA, to the VGG16 model from (Mokhtari et al., 2023).

The VGG16 model contains 18,939,722 parameters, the IFA$_{5.1}$ model is composed from 3,385,018 parameters, while the IFA$_{2.1}$ is comprises 63,866 parameters. All of them were trained for 100 epochs, on the OAD dataset that includes 11935 samples. From this data, we can compute, for each model, the training cost (TC) and floating point operations (FPO), which will be combined with the test accuracy (suffixed *acc*) to compute the BCR$^{tc}_{acc}$ and BCR$^{fpo}_{acc}$ respectively .

From Table 5 we can notice that the most efficient model is the IFA$_{2.1}$ according to the training cost and the FPO score. This model also obtained the best ratio for both BCR$^{tc}_{acc}$ and BCR$^{fpo}_{acc}$. The less efficient is the VGG16 model, which is also the worst in terms of BCR$^{tc}_{acc}$ and BCR$^{fpo}_{acc}$.

Note that, in terms of performance, the VGG16 is only 1.45% better than the IFA$_{2.1}$, while being almost 300 larger in terms of size, with nearly 19 millions parameters against 64 thousands parameters. In addition, the IFA$_{2.1}$ requires only 998KB of storage space, allowing it to be easily embedded when needed.

Moreover, training the VGG16 on the NVIDIA RTX A3000 Laptop GPU takes 1915 seconds (2.8 times more than IFA$_{2.1}$), and according to Green Algorithms Calculator proposed by (Lannelongue et al., 2021) it consumes around 4.22 kWh of energy and released about 216.43 g of CO$_{2e}$, meaning that the IFA$_{2.1}$ is 14 times less energy consuming and less polluting compared to VGG16.

This result confirms the observation of (Schwartz et al., 2019) that there is an urgent need to evaluate the efficiency of a neural network with the same importance as its performance, because increasing the ac-

Table 5: Efficiency comparaison between VGG16, $IFA_{2.1}$ and $IFA_{5.1}$.

Model	**Test Accuracy**	**Training Cost**	$\mathbf{BCR}^{tc}_{acc}$	**Floating Point Operations**	$\mathbf{BCR}^{fpo}_{acc}$
VGG16	95.22%	2.26e+13	4.21e-12	2.52e+09	3.79e-08
$IFA_{2.1}$	93.77%	7.62e+10	1.23e-09	9.95e+07	9.42e-07
$IFA_{5.1}$	94.47%	4.04e+12	2.34e-11	2.30e+08	4.10e-07

curacy by 1.45% with a 300 times larger network that consumes 14 times more energy is not such a good deal.

6 CONCLUSION

In this research, we apply the Improved Firefly Algorithm, a training-free neural architecture search technique, to automate the model design and find the most suitable neural network for the chosen dataset. This method produced interesting results for on the Online Action Data (OAD) dataset in 48 minutes on a single CPU, improving the baseline NAS (using training over 100 epochs) by a factor of 476 (GPU based) and a factor of 4400 (CPU based), while improving (Wang et al., 2023) proposition by a factor of 114. this method is also 170 times less power consuming and polluting than training-based NAS using GPU.

While achieving high performance, it is important to take into account the efficiency of neural networks, which are growing exponentially. With this in consideration, we propose the benefit cost ratio (BCR), a metric to evaluate the quality of a neural network in terms of its performance, but also its cost.

Experimentation on the Online Action Detection dataset showed that using the IFA provides a little lower performing model (93.77% of accuracy 95.22% from the state-of-the-art) but allows reducing the computation cost in terms of time by a factor of 2.8, and 14 times in terms of energy consumption and pollution, by producing a neural network that is 300 times smaller than the VGG16 model.

As a future work, we consider using the BCR as the fitness of the Improved FireFly algortihm, to bring the aspect of efficiency into the architecture search.

ACKNOWLEDGEMENTS

This work has been carried out within the French-Canadian project DOMAID which is funded by the National Agency for Research (ANR-20-CE26-0014-01) and the FRQSC

REFERENCES

Carvalho, A., Ramos, F., and Chaves, A. (2010). Metaheuristics for the feedforward artificial neural network (ann) architecture optimization problem. *Neural Computing and Applications*, 20.

Chen, S., Xu, K., Jiang, X., and Sun, T. (2022). Pyramid spatial-temporal graph transformer for skeleton-based action recognition. *Applied Sciences*, 12(18):9229.

Delamare, M., Laville, C., Cabani, A., and Chafouk, H. (2021). Graph convolutional networks skeleton-based action recognition for continuous data stream: A sliding window approach. In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP,*, pages 427–435. INSTICC, SciTePress.

Elsken, T., Metzen, J. H., and Hutter, F. (2019). Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21.

Lannelongue, L., Grealey, J., and Inouye, M. (2021). Green algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, 8(12):2100707.

Laraba, S., Brahimi, M., Tilmanne, J., and Dutoit, T. (2017). 3d skeleton-based action recognition by representing motion capture sequences as 2d-rgb images. *Computer Animation and Virtual Worlds*, 28.

Li, C., Zhong, Q., Xie, D., and Pu, S. (2018). Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 786–792. International Joint Conferences on Artificial Intelligence Organization.

Li, Y., Lan, C., Xing, J., Zeng, W., Yuan, C., and Liu, J. (2016). Online human action detection using joint classification-regression recurrent neural networks. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 203–220, Cham. Springer International Publishing.

Liu, J., Shahroudy, A., Wang, G., Duan, L.-Y., and Kot, A. (2019). Skeleton-based online action prediction using scale selection network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP.

Liu, J., Shahroudy, A., Xu, D., Kot, A., and Wang, G. (2018). Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:3007–3021.

Liu, J., Wang, G., Hu, P., Duan, L.-Y., and Kot, A. C. (2017a). Global context-aware attention lstm networks for 3d action recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3671–3680.

Liu, M., Liu, H., and Chen, C. (2017b). Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362.

Ludl, D., Gulde, T., and Curio, C. (2019). Simple yet efficient real-time pose-based action recognition. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 581–588.

Mellor, J., Turner, J., Storkey, A., and Crowley, E. J. (2021). Neural architecture search without training. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7588–7598. PMLR.

Mokhtari, N., Fer, V., Nédélec, A., Gilles, M., and De Loor, P. (2023). Enhanced spatio-temporal image encoding for online human activity recognition. In *International Conference on Machine Learning and Applications (ICMLA) 2023*, page to appear.

Mokhtari, N., Nédélec, A., Gilles, M., and De Loor, P. (2022a). Improving neural architecture search by mixing a firefly algorithm with a training free evaluation. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Mokhtari, N., Nédélec, A., and Loor, P. D. (2022b). Human activity recognition: A spatio-temporal image encoding of 3d skeleton data for online action detection. In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP,*, pages 448–455. INSTICC, SciTePress.

Ordóñez, F. J. and Roggen, D. (2016). Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1).

Schwartz, R., Dodge, J., Smith, N., and Etzioni, O. (2019). Green ai. *Communications of the ACM*, 63:54 – 63.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Strumberger, I., Tuba, E., Bacanin, N., Zivkovic, M., Beko, M., and Tuba, M. (2019). Designing convolutional neural network architecture by the firefly algorithm. In *2019 International Young Engineers Forum (YEF-ECE)*, pages 59–65.

Sun, Y., Xue, B., Zhang, M., Yen, G. G., and Lv, J. (2020). Automatically designing CNN architectures using the genetic algorithm for image classification. *IEEE Transactions on Cybernetics*, 50(9):3840–3854.

Wang, J., Chen, Y., Hao, S., Peng, X., and Hu, L. (2019). Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters*, 119:3–11.

Wang, X., He, M., Yang, L., Wang, H., and Zhong, Y. (2023). Human activity recognition based on an efficient neural architecture search framework using evolutionary multi-objective surrogate-assisted algorithms. *Electronics*, 12(1).

Weng, J., Weng, C., and Yuan, J. (2017). Spatio-temporal naive-bayes nearest-neighbor (st-nbnn) for skeleton-based action recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wistuba, M., Rawat, A., and Pedapati, T. (2019). A survey on neural architecture search.

Yan, S., Xiong, Y., and Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition.

Ying, C., Klein, A., Christiansen, E., Real, E., Murphy, K., and Hutter, F. (2019). NAS-bench-101: Towards reproducible neural architecture search. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7105–7114. PMLR.