


A Novel GA-based Approach to Automatically Generate ConvLSTM Architectures for Human Activity Recognition


Sarah Khater

(Computer Engineering Department, Faculty of Engineering, Cairo University, Cairo, Egypt
 <https://orcid.org/0000-0002-8213-0022>, srashad@cu.edu.eg)

Magda B. Fayek

(Computer Engineering Department, Faculty of Engineering, Cairo University, Cairo, Egypt
magdafayek@eng.cu.edu.eg)

Mayada Hadhoud

(Computer Engineering Department, Faculty of Engineering, Cairo University, Cairo, Egypt, and
School of Computational Sciences and Artificial Intelligence (CSAI), Zewail City of Science and
Technology, Cairo, Egypt
 <https://orcid.org/0000-0002-2215-5594>, mayada.hadhoud@eng.cu.edu.eg)

Abstract: Human activity recognition (HAR) is a challenging computer vision problem that requires recognizing and categorizing human actions using spatiotemporal data. In recent years, ConvLSTM has shown distinctive advances in manipulating spatiotemporal data. ConvLSTM-based architectures, as any deep learning architecture, require deciding on many hyperparameters apart from trainable weights. State-of-the-art designs for general purpose datasets already exist, but specific purpose applications require architecture designs that perform well on application-dependent datasets. The design of such architectures requires either many trials and errors, which consume time and resources, or an experienced architect. Neural architecture search (NAS) methods have been introduced to automate the design process and address the challenge of relying on expert knowledge when creating neural architectures. NAS enables rapid prototyping and experimentation, reducing the time spent on trial and error in manual design. One of the leading approaches in NAS is Genetic Algorithm (GA), which plays a significant role in optimizing neural architectures. In this paper, a novel GA-based approach is proposed to automatically design ConvLSTM-based architectures from scratch for HAR applications. Our approach is based on multi-objective GA that maximizes recognition accuracy and minimizes the number of trainable parameters and overfitting measure. The experiments are held on KTH, Weizmann, and UCF Sports datasets. The best classification accuracies from the generated models are 97.92%, 96.77%, and 94.87% for KTH, Weizmann, and UCF Sports datasets, respectively. The experimental results show that the automatically generated models with the proposed approach outperform some of the state-of-the-art manually designed ConvLSTM-based architectures with percentages up to 9.92%, 5.77% and 23.64% for KTH, Weizmann, and UCF Sports, respectively. We also compared our approach with other NAS approaches. Our approach is found to outperform some of the introduced approaches with percentages approximately 2%, 11%, and 4% for KTH, Weizmann, and UCF Sports, respectively.

Keywords: ConvLSTM, HAR, KTH, Multi-objective fitness, NAS

Categories: I.2.6, I.2.10, I.4.0

DOI: 10.3897/jucs.131543

1 Introduction

Human activity recognition (HAR) is the problem of recognizing the actions done by one person or a group of people. HAR based on spatiotemporal data is a crucial task in video understanding [Wang and Schmid 2013]. It is a core component in many real-life applications, including surveillance systems [Jahlan and Elrefaei 2021], medical applications [Kaya and Kuncan 2022], industrial processes [Kumar et al. 2024], and entertainment applications [Heravi et al 2024]. Solving the HAR problem faces many challenges like data acquisition, input data representation and preprocessing, complexity of actions, and techniques to manipulate the data [Arshad et al. 2022]. Deep learning approaches are widely used in solving the HAR problem.

While new architectures continuously emerge, ConvLSTM-based architectures have distinctive performance in learning and analyzing spatiotemporal data [Ding et al. 2023, Yuan et al. 2018, Wang et al. 2018, Lin et al. 2020]. Their ability to capture both spatial and temporal dependencies simultaneously makes them valuable tools. However, designing these architectures requires deciding on many parameters, so it needs human experience and a lot of computation time [Jaafr et al. 2019]. Neural architecture search (NAS) approaches are proposed to automate the design process and overcome the experience problem in designing neural architectures. NAS allows rapid prototyping and experimentation which make up for trial-and-error time required by manual design [Elsken et al. 2019]. Among these approaches is Genetic Algorithm (GA). GA is a prominent method in NAS [Elsken et al. 2019].

ConvLSTM-based architectures are particularly important in solving HAR [White et al. 2023]. Based on existing ConvLSTM generation approaches, the generation methods primarily focus on tuning and adjusting parameters of existing architectures [Vrskova et al. 2021], integrating notable existing architectures [Houreh et al. 2021], or creating architectures for a specific type of operation and then stacking these types together [Jahlan and Elrefaei 2021]. We were motivated to propose a solution that generates a ConvLSTM-based architecture from scratch while optimizing specific parameters to suit application dependent datasets. Additionally, the proposed approach allows integrating various architectures, including residual and inception architectures. Building ConvLSTM-based architectures from scratch, not restricted by predefined structures, allows exploring a wider range of solutions with the potential of finding an optimal solution.

In this paper, we present an approach to automatically design ConvLSTM-based architectures using GA to solve the HAR problem. Our approach uses the following layers as the basic building blocks of the architecture: ConvLSTM, Residual ConvLSTM, Inception ConvLSTM, and Residual Inception ConvLSTM layers [Khater et al. 2022]. Our approach proposes and uses a new multi-objective fitness function to evaluate each generated architecture. The objective function maximizes recognition accuracy and minimizes the number of trainable parameters and overfitting measure [Pavlitskaya et al. 2022] (difference between training and validation set accuracies).

The novelty of our work is introducing a new multi-objective function and using Residual Inception ConvLSTM layer, along with ConvLSTM, Residual ConvLSTM, and Inception ConvLSTM to automatically design an architecture from scratch to solve the HAR problem.

Our approach is trained and tested against KTH, Weizmann, and UCF Sports datasets. The classification accuracies yielded from the best-generated models are 97.92%, 96.77%, and 94.87% for KTH, Weizmann, and UCF Sports datasets, respectively. Also, our approach is tested against some of the state-of-the-art architectures. The proposed approach

is found to outperform some of these architectures with percentages up to 9.92%, 5.77% and 23.64% for KTH, Weizmann, and UCF Sports, respectively. Our approach outperforms some of the introduced NAS approaches with percentages of approximately 2%, 11%, and 4% for KTH, Weizmann, and UCF Sports, respectively.

2 Literature Review

Our proposed approach addresses solving HAR using NAS. In this section, we present a quick review of different approaches used to solve HAR. We also provide an overview on NAS methods.

2.1 HAR Approaches

In general, there are sensor-based HAR approaches [Kuncan et al. 2022, Tuncer et al. 2020] and vision-based HAR approaches. In our work, we focus more on vision-based approaches. We classify the presented HAR approaches from three different perspectives.

The first perspective is the feature extraction process. Feature extraction is done either manually or automatically using a learning method. Manual feature extraction may use body parts, moving object, or a hybrid between these features [Jalal et al. 2012, Jalal et al. 2015, Kumar and John 2016, Niu and Abdel-Mottaleb 2004, Althloothi et al. 2014]. Examples of manually extracted features are 3D Harris space-time interest point detector, 3D Scale-Invariant Feature Transform (3DSIFT) descriptor [Nazir et al. 2018], and Gray Level Co-Occurrence Matrix (GLCM) [Kuncan et al. 2022]. These features can then be arranged for each sequence of frames as a multidimensional feature to train a model like a support vector machine (SVM) model or an artificial neural network (ANN). On the other hand, automatic feature learning can be done using famous deep learning methods like CNN [Khair et al. 2018], RNN [Qi et al. 2018], CNN-RNN [Singh and Singhal 2023], R-CNN [Liu et al. 2021], ANN [Kuncan et al. 2019], GNN [Jlidi et al. 2024], and ConvLSTM [Khater et al. 2022], or using non-deep learning methods like dictionary learning [De et al. 2017] or Bayesian networks [Wang and Ji 2012].

The second perspective is how the features are represented. Features can be defined to preserve spatial, temporal, dimensional, and color information. This famous representation uses a sampled frame sequence as the input to the model. In this representation, the pixel values in each frame are considered the features. Other approaches flatten the features, losing their spatial information, as in Bag of Features method [Aly and Sayed 2019]. Other feature representations are used, such as Silhouette representation, where only the object of interest is outlined in the frame [Ramya and Rajeswari 2021]. Bit map feature representation is also used to encode the movement of the object of interest in a sequence of frames [Arunnehr et al. 2018]. Binary input representation is also one type of the used representations.

The third perspective is the supervision level. HAR approaches can be supervised, semi-supervised, or unsupervised. Supervised learning approaches require the presence of large labelled datasets [Han et al. 2018, Zhang et al. 2018]. In case the datasets are not big enough, some augmentation techniques are proposed to overcome the overfitting problem arising from the absence of massive datasets [Han et al. 2018]. On the contrary, unsupervised learning is used when labeled datasets are unavailable. Discriminative features are learned from unlabeled data. [Abdelbaky and Aly 2021] and [Haddad et al. 2021] are examples of research proposed to solve the HAR problem based on unsupervised learning. Semi-supervised learning combines the benefits of both supervised and

unsupervised learning. Semi-supervised can learn visual features from labelled data, and learn hidden non-visual features [Singh et al. 2021, Jing et al. 2021].

2.2 NAS Approaches

NAS is one of the most widely explored topics in machine learning. Many interesting research work and approaches are frequently introduced in this topic [Salehin et al. 2024]. In this subsection, we present a quick review of NAS approaches from the following aspects: search space, search strategy, and candidate architecture evaluation. Additionally, we provide a more detailed review of the use of GA in NAS. Also, we highlight the most significant steps towards ConvLSTM-based architectures automatic generation.

The first aspect is the different search spaces. The search space can be defined by some parameters such as the maximum number of layers, the possible types of each layer, and the possible hyperparameters of each layer (e.g., the number of filters and kernel size) [Elsken et al. 2019]. Some search spaces incorporate skip connections [O'Neill et al. 2021], multi-branch [Ahn and Cho 2021, El Assal et al. 2023], or inception blocks, which give some complexity to the architecture. This type of search space is called global search space [Kang et al. 2023]. Global search space describes an architecture as a directed acyclic graph. Building an architecture requires finding a series of operators [Cai et al. 2023]. The NAS space can also be described as a sequence of neural network predefined building blocks or units, also known as cell-based search space [Kang et al. 2023]. Cell-based search space is based on reusing successful neural network structures or blocks to optimize new architectures [Tu et al. 2023, Liu et al. 2021]. Some research studies impose restrictions on different parameters of the explored search space, e.g., in [Xie and Yuille 2017], the authors proposed a fixed length network representation. Also, autoencoders are proposed to extract a more meaningful representation of the neural network under assessment [Tang et al. 2020].

The second aspect is the different search strategies that automatically search for an artificial neural network in the designated search space. The most common strategies used to automatically generate an artificial neural network are:

- Evolutionary algorithms [Miikkulainen et al. 2019, Tirumala et al. 2016, Shang et al. 2022, David and Greental 2014, Lu et al. 2023]: Evolutionary algorithms imitate nature to find an optimal architecture. A lot of research is done on this point. In [Liu et al. 2018], authors proposed a sequential model-based optimization method that combines evolutionary algorithm and reinforcement learning method. In [O'Neill et al. 2021], the authors suggested using evolutionary algorithm to explore the Dense net structure with various skip connections. Also, in [Xie and Yuille 2017], Xie and Yuille proposed a CNN search approach that uses GA. In [Ishwarya and Nithya 2023], Ishwarya and Alice Nithya proposed a method for human pose estimation using squirrel search optimization (SSO) technique applied on CNN.
- Traditional learning: This approach includes supervised learning, semi-supervised learning, unsupervised learning, and reinforcement learning. In [Li et al. 2020], the authors used supervised learning to modularize the NAS search space to ensure the proper training of all candidate architectures to get a correct candidate evaluation. In [Luo et al. 2020], the authors proposed an approach that uses a minimized set of architecture-accuracy pairs to train an initial architecture accuracy predictor and expands the predictor with generated data pairs. Other literature used variations of semi-supervised learning approaches [Li et al. 2021, Kaplan and Giryes 2020, Ducros

2023], and very little work is done using unsupervised learning [Xue et al. 2021]. [Zoph and Le 2016, Jaafra et al. 2019, Hsu et al. 2018] explored reinforcement learning to solve NAS problem. In [Hsu et al. 2018], the authors proposed a multi-objective reinforcement approach to automatically search for high-performance neural network architectures. The approach optimizes both prediction accuracy and energy consumption.

- Transfer learning: Transfer learning improves the learning process by reusing knowledge gained while solving a related problem [Torrey and Shavlik 2010]. In [Lu et al. 2021], transfer learning is combined with multi-objective evolutionary search to automatically generate custom models that compete with state-of-the-art models.
- Meta learning: Meta learning dynamically improves its bias by accumulating meta-knowledge [Vilalta and Drissi 2002]. Much work explored this approach in NAS [Lian et al. 2019, Wang et al. 2020, Elsken et al. 2020].
- Bayesian optimization: Bayesian optimization is one of the most promising methods in NAS after being a real success in hyperparameter optimization [White et al. 2021, Shen et al. 2023, Nandagopal et al. 2023].
- Many other techniques are examined to solve this problem, for e.g., one-shot method, this method is still not very effective on large datasets [Guo et al. 2020], but still, it compresses lengthy training process by parameter sharing across different candidate solutions. Remarkable research is done using this method [Dong and Yang 2019, Zela et al. 2020, Shi et al. 2020]. Also, fusion methods have a share in NAS research work [Tran et al. 2021, Peng et al. 2020].

The last aspect is the objective function on which an architecture is elected as a better solution. Different approaches are used to evaluate the generated architectures. Many research studies use recognition accuracy to assess the network [Wang et al. 2021, Zhao et al. 2021]. Some work uses multi-objective functions that optimize different objectives, e.g., energy, number of floating-point operations per second, number of parameters, latency, and recognition accuracy [Lu et al. 2023, Hsu et al. 2018, Liu et al. 2023, El Assal et al. 2023]. More complex assessment approaches have been suggested. In [Tang et al. 2020], Tang et al. proposed a semi-supervised predictor to assess the performance of each neural network. [Shang et al. 2022] used an evaluation correction technique that helps isolate poorly performing architectures from mating with other promising architectures. In [Zheng et al. 2020], authors proposed a minimum importance pruning technique so that less promising solutions are pruned to reserve more computational resources for more important ones.

Since our proposed approach leverages GA to address NAS, we provide a comprehensive review of existing work in this area. The following are the most remarkable studies done to solve NAS problem using GA. Some approaches relied on proposing novel fitness functions. In [Thanh et al. 2024], the authors proposed a hardware-aware NAS approach. The approach considers latency of the candidate architecture in the filtering process. In [Liang et al. 2024], the authors proposed a multi-objective GA-based approach that optimizes both accuracy and size of the generated CNN-based architecture. In [Lu et al. 2019], the authors proposed a fitness function that minimizes both error metric and number of floating-point operations, representing computational complexity. The proposed approach was applied on image classification. Other approaches tackled the computational costs in evaluating a neural architecture. In [Lin and Tsai 2024], Lin

and Tsai proposed a training-free fitness function. The training-free fitness function is used to evaluate a model or its outputs without requiring the model to undergo training. The function evaluates the generated architectures based on the number of architecture parameters and the number of CNN layers. Some of the proposed GA approaches rely on the novelty of the chromosome representation. The chromosome representation plays a crucial role in shaping the efficiency and effectiveness of the proposed approach. In [Wen et al. 2022], the authors proposed a fixed length encoding approach to be utilized by GA to generate variable depth VGG-based architectures. The approach was applied on image classification. In [Ghosh and Jana 2020], the authors used GA to automatically design a feed forward network to solve image classification. Each chromosome is represented as four parameters defining the generated architecture. These parameters are the number of hidden layers, the number of neurons per hidden layer, the activation function, and the network error optimization technique.

Given that our proposed approach automatically generates ConvLSTM-based architectures, we provide an in-depth review of relevant work in this domain. The following part of the review explores key steps towards ConvLSTM automatic generation, offering context for the relevance and effectiveness of our method. Some approaches target exploring different layers configurations. In [Jahlan and Elrefaei 2021], the authors proposed an approach to recognize human violence in spatiotemporal data. The authors used a CNN-based architecture search approach to find the best layers configuration to extract spatial features. The authors suggested aggregating the generated CNN-based architecture with a ConvLSTM layer to capture both spatial and temporal features. Some approaches rely on tuning critical hyperparameters, including number of layers or learning rate. In [Vrskova et al. 2021], the authors proposed a hyperparameter tuning approach to tune ConvLSTM-based architectures. The approach tunes parameters like filter size, number of filters, batch size, number of epochs, and training optimization algorithm. Some approaches utilize existing architectures to automatically generate suitable architecture for a particular problem. In [Houreh et al. 2021], the authors proposed an approach that uses GA to generate a U-Net architecture by combining the most relevant U-Net architectures. The proposed approach was used in segmenting retinal blood vessels.

In general, the proposed ConvLSTM-based architecture generation approaches rely mainly on tuning architecture parameters, combining existing remarkable architectures, or generating architectures composed of a single type of operation and stacking multiple instances of this operation together [Jahlan and Elrefaei 2021, Vrskova et al. 2021, Houreh et al. 2021]. Our approach tackles this gap in research. Our proposed approach builds ConvLSTM-based architectures from scratch, not restricted by predefined structures. Also, the proposed approach incorporates different architectures, residual and inception architectures. The proposed approach allows exploring a wider range of solutions that increases the likelihood of discovering an optimal solution.

3 Generated Model Building Blocks

This section briefly illustrates the basic building blocks used in the generated architecture. The basic building blocks are ConvLSTM [Shi et al. 2015], Residual ConvLSTM [Wei et al. 2018], Inception ConvLSTM [Song et al. 2018], and Residual Inception ConvLSTM [Khater et al. 2022].

ConvLSTM architecture was first introduced by X. Shi et al. in [Shi et al. 2015]. It uses the recurrent nature of the LSTM unit to manipulate spatiotemporal data by applying convolution operations to perform state-to-state and state-to-output transformations.

Fig. 1 shows the unfolded structure of ConvLSTM layer labelled with its main tensors: input tensor X_1, \dots, X_t , cell current state tensor C_1, \dots, C_t , and hidden state tensor, also referred to as cell output tensor, H_1, \dots, H_t . State-to-state and state-to-output transformations are illustrated by ConvLSTM fundamental equations, shown in 1-5, given that i_t, f_t and o_t are 3D tensors and the basic operations are: (i) nonlinear activation function ' σ ', (ii) convolution operator ' $*$ ', and (iii) Hadamard product ' \circ '. Each grid cell value is calculated using the current input and the current neighboring cells values.

Residual ConvLSTM Architecture was introduced to incorporate residual concept [He et al. 2016] with ConvLSTM. Residual blocks are motivated by the vanishing gradient problem. In the backpropagation method, theoretically, as the architecture goes deeper, the model ability to learn a more complex function increases. But, in practice, as the model goes deeper, the gradients become vanishingly small and no more learning takes place. A residual block uses shortcuts to skip two or three layers. These shortcuts help with the vanishing gradient problem by reusing activations from previous layers until adjacent layers adjust their values. Fig. 2 shows Residual ConvLSTM layer.

Inception ConvLSTM Architecture combines both inception [Szegedy et al. 2015] and ConvLSTM concepts. Inception uses repeated components to extract features at different scales, then these features are combined to construct a more general block for object recognition. Fig. 3 shows Inception ConvLSTM layer.

Residual Inception ConvLSTM Architecture aggregates the three concepts of residual, inception, and ConvLSTM. Fig. 4 shows Residual Inception ConvLSTM layer.

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \quad (2)$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \quad (4)$$

$$H_t = o_t \circ \tanh(C_t) \quad (5)$$

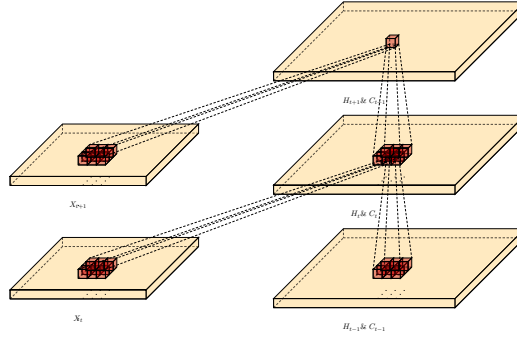


Figure 1: ConvLSTM unfolded Structure.

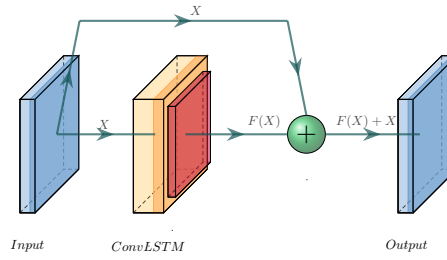


Figure 2: Residual ConvLSTM Architecture.

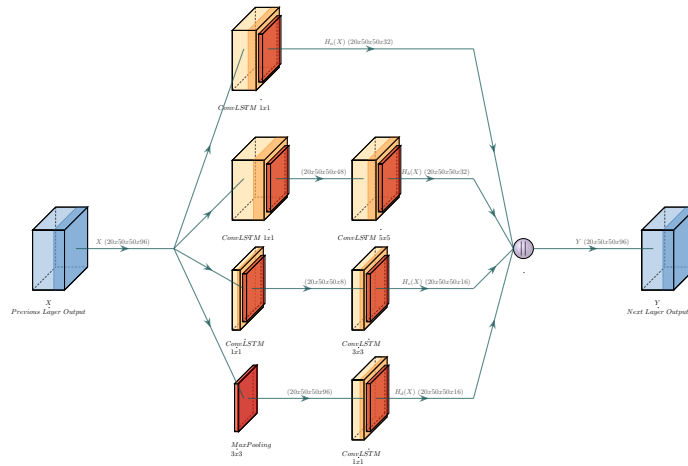


Figure 3: Inception ConvLSTM architecture.

4 Proposed Approach

In this section, we discuss the details of the proposed approach. Our approach is a ConvLSTM-based architecture search using multi-objective GA. The proposed approach automatically designs an architecture using a combination of four basic building blocks: ConvLSTM, Residual ConvLSTM, Inception ConvLSTM, and Residual Inception ConvLSTM.

The algorithm searches for an architecture with the following parameters: i) number of layers, ii) layer types, iii) filters number, and iv) kernel sizes for each layer. Our approach uses a multi-objective fitness function to evaluate each architecture. The multi-objective function maximizes recognition accuracy and minimizes the number of trainable parameters and overfitting measure [Pavlitskaya et al. 2022]. The following subsections describe the used GA chromosome representation and initialization, GA operators, and the proposed fitness function.

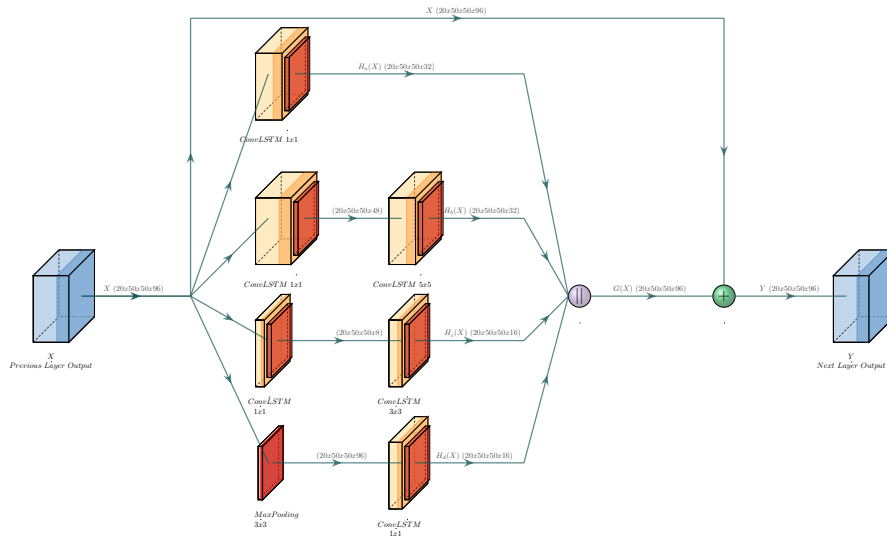


Figure 4: Residual Inception ConvLSTM architecture.

4.1 Chromosome representation and initialization

Each architecture is encoded in the chromosome as a string of genes, with each gene representing a single basic building block. A gene structure encapsulates the basic building block type. In case of ConvLSTM or Residual ConvLSTM building blocks, the gene structure also includes the number of filters and the kernel size. The chromosome, representing the generated architecture to be trained, is initialized using the following characteristics:

1. Number of layers, the number of genes in a chromosome, is between 2 and 15 layers. As the number of layers increases, the trainable parameters increase drastically. Therefore, we decided on limiting the number of generated layers to 15 layers due to computational restrictions and based on state-of-the-art architectures depth [Yang et al. 2017, Yin et al. 2021].
2. The type of each layer is ConvLSTM, Residual ConvLSTM, Inception ConvLSTM, or Residual Inception ConvLSTM.
3. For ConvLSTM and Residual ConvLSTM layers, batch normalization and activation layers are subsequently applied.
4. For Inception ConvLSTM and Residual Inception ConvLSTM layers, layer structures are shown in Fig. 3 and Fig. 4.
5. In the initialized architecture, layers are configured as follows:
 - The first layer is always a ConvLSTM layer. This layer is added to adjust the number of input channels for the following layers.

- The following layers are initialized with probabilities according to their depth. Early layers are designed to be less complex [Szegedy et al. 2015] with probabilities 0.5, 0.2, 0.2, and 0.1; for ConvLSTM, Residual ConvLSTM, Inception ConvLSTM, and Residual Inception ConvLSTM, respectively.
 - Higher layers are designed to be more complex [Szegedy et al. 2015] with probabilities 0.1, 0.3, 0.3, and 0.3; for ConvLSTM, Residual ConvLSTM, Inception ConvLSTM, and Residual Inception ConvLSTM, respectively.
6. ConvLSTM and Residual ConvLSTM layers are initialized with one of the following number of filters: 16, 32, 64, 96, or 128, and kernel sizes (2, 2), (3, 3), (4, 4), or (5, 5).
 7. Maxpooling layers are added randomly in-between layers.
 8. Each architecture must have at least one non-linear activation layer.
 9. Each architecture ends with a fully connected layer.

4.2 Algorithm Operators

4.2.1 Crossover Operator

In our approach, we use single point crossover, as illustrated in Fig. 5, where a single point is picked in each chromosome and two new chromosomes are created by exchanging the genetic material with crossover probability P_{cr} . If any of the generated chromosomes violates the chromosome structure, the chromosome is adjusted according to the chromosome description mentioned in Section 4.1.

4.2.2 Mutation operator

In our approach, we use single point mutation, as illustrated in Fig. 6, where each gene, layer, is subject to mutation with probability P_m . When a gene is selected to be mutated, a new layer is randomly generated to replace the old one.

4.2.3 Selection operator

In our approach, we use roulette selection. The probability of selecting a chromosome is proportional to its calculated multi-objective fitness. Although local minima problem may arise from using roulette selection method, but the results analysis invalidates this problem in our case.

4.2.4 Proposed Fitness Function

Each chromosome is evaluated by constructing the architecture, that maps to the chromosome, then training the constructed architecture. The model is then assessed using a test set, different from the training and validation sets. Each model is evaluated according to a multi-objective fitness function. The multi-objective fitness function depends on three factors: recognition accuracy, the number of trainable parameters, and the standard overfitting measure [Pavlitskaya et al. 2022]. The function maximizes the recognition accuracy on the test set. The function minimizes the number of trainable parameters and the

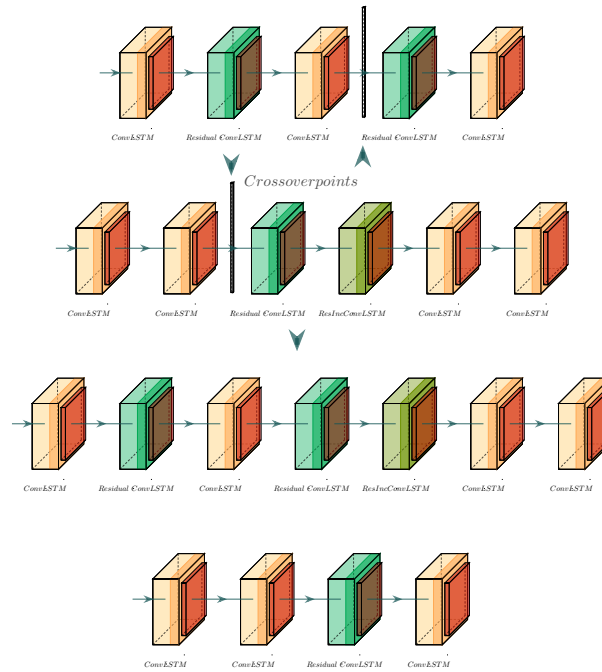


Figure 5: Crossover Operator.

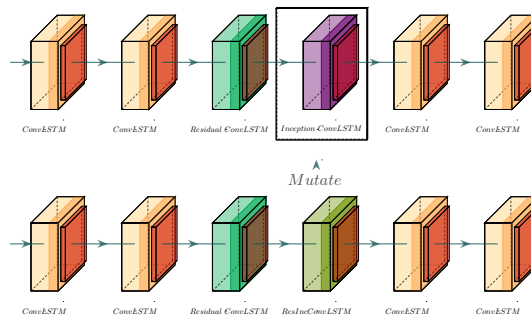


Figure 6: Mutation Operator.

standard overfitting measure. The standard overfitting measure is the difference between training and validation set accuracies, as illustrated in Eq. 6, where, $P_{train_accuracy}$ and $P_{val_accuracy}$ are the training and validation recognition accuracies, respectively. Although overfitting measure has its limitations, which are:

1. There is no guarantee that the validation set is representative of unseen data. It may not describe the underlying data distribution.
2. This measure gives false estimation when comparing different architectures trained using various datasets.

To overcome the limitation of overfitting measure:

1. The validation set is randomly picked from the dataset.
2. Also, we use the same predefined dataset splits to compare different architectures.

The multi-objective fitness, F , can be seen in Eq. 7, where, $P_{test_accuracy}$ is test recognition accuracy, R_{tp} is the ratio between the number of trainable parameters of the current solution and the maximum allowed number of trainable parameters of the generated solutions, and $R_{overfitting}$ is overfitting measure. In case that overfitting measure is zero, the chromosome is assigned the highest fitness value; this usually happens when validation accuracy approaches 100%.

$$R_{overfitting} = |P_{train_accuracy} - P_{val_accuracy}| \quad (6)$$

$$F = \frac{P_{test_accuracy}}{R_{tp} * R_{overfitting}} \quad (7)$$

5 Experimental Results

This section describes and discusses the conducted experiments to evaluate our approach. The section is organized as follows. Section 5.1 gives a brief description of the datasets used in the experiments. Section 5.2 describes the experiments conducted and presents the results, along with a discussion of these results. Section 5.3 compares our approach with some state-of-the-art architectures.

5.1 Datasets

We trained and tested our approach against the following datasets: KTH, Weizmann, and UCF Sports datasets.

KTH is a public dataset. It consists of six actions. The actions are performed by 25 actors, each actor performs the six actions in four separate videos. The actions are running, jumping, clapping, walking, two-hand waving, and skipping. The dataset consists of 600 videos with resolution (160 x 120) pixels. The average video length is 4 s. Fig. 7 shows sample frames of KTH dataset, as an illustration. Weizmann is a public dataset that consists of 10 actions. The actions are performed by nine actors. The actions are walking, running, jumping, gallop sideways, bending, one-hand waving, two-hand waving, jumping in place, jumping jacks, and skipping. The dataset consists of 90 videos with resolution (80 x 144) pixels. The average video length is 1.5 s. Fig. 8 shows sample frames of Weizmann dataset, as an illustration.

UCF Sports is a public dataset of 13 actions, collected from different sports featured on television. These actions are diving, golf swing (back, front, and side), kicking (side and front), lifting, horseback riding, running, skateboarding, swing (bench and side), and walking. The dataset consists of 150 videos with resolution (720 x 480) pixels. The minimum and maximum number of videos per class are six and 22, respectively. The average, maximum, and minimum video lengths are 6.39 s, 2.2 s, and 14.4 s, respectively. Fig. 9 shows sample frames of UCF Sports dataset, as an illustration.

Table 1 shows a brief description of KTH, Weizmann, and UCF Sports datasets.

In our experiments, the evaluation of each model uses the train-test split method, 90% - 10%. Train split encapsulates both training and validation sets. Each video is sampled

into 20 frames and resized into (50 x 50) pixels. For KTH, we convert the frames into gray scale. KTH dataset is grouped by actions, and each split has videos performed by the same actors. For Weizmann, we use multiple samples per video for better training. During GA epochs, each architecture is evaluated using the train-test split method.

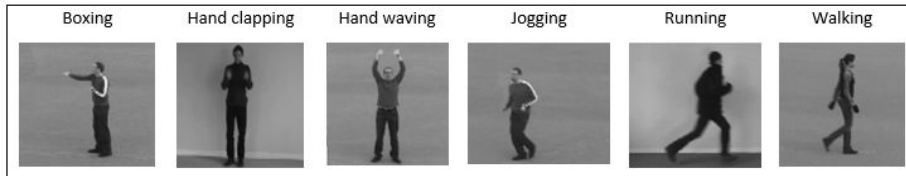


Figure 7: Examples from KTH dataset

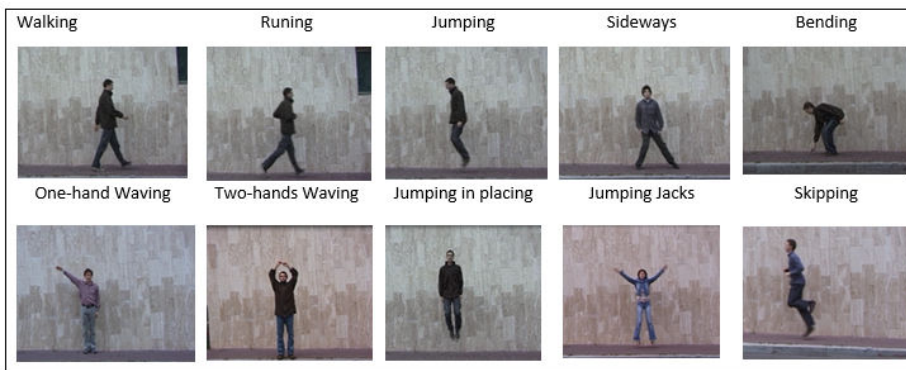


Figure 8: Examples from Weizmann dataset

Dataset	Number of actions	Number of videos	Average video length	Resolution
KTH	6	600	4s	160 x 120
Weizmann	10	90	1.5s	180 x 144
UCF Sports	13	150	6.39s	720 x 480

Table 1: KTH, Weizmann, and UCF Sports Brief Description

5.2 Experiments

The experiments are held on a system with the following specs: NVIDIA A100 Tensor Core GPU with 80 Gigabyte High-bandwidth memory 2nd generation with GPU memory

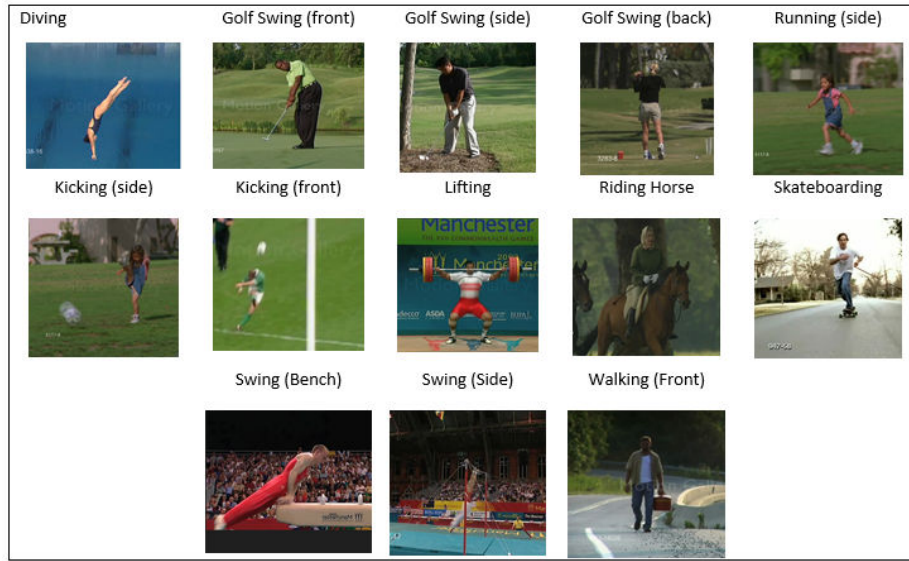


Figure 9: Examples from UCF Sports dataset

Parameters	Datasets	Proposed Approach	GA Standard Approach
Recognition Accuracy	KTH	97.92%	95.83%
	Weizmann	96.77%	100%
	UCF Sports	94.87%	92.31%
Number of Trainable Parameters	KTH	2,545,830	3,685,510
	Weizmann	4,594,794	20,739,594
	UCF Sports	3,798,445	6,404,429
Overfitting Measure	KTH	0.0135	0.0417
	UCF Sports	0.0336	0.0718

Table 2: Comparison between best performing architectures of the proposed approach and GA standard approach for KTH, Weizmann, and UCF Sports datasets.

bandwidth of over 2TB/s. Our work is based on the following experiments: Examining our approach, GA using the proposed multi-objective fitness function, and comparing the results with GA standard approach, GA with recognition accuracy as the fitness function [Ijjina and Chalavadi 2016], refined with insights from [Houreh et al. 2021]. The experiments are held on the datasets mentioned in Section 5.1. The experiments yielded the following statistics.

In our approach, 20% of the generated architectures achieved more than 92%, 91.5%, and 90% recognition accuracy for KTH, Weizmann, and UCF Sports datasets, respectively. Also, the generated architectures in the final population achieved average recognition accuracy of 90.885%, 90.32%, and 84.616%, and worst recognition accuracy of 79.17%, 64.52%, and 61.54%, for KTH, Weizmann, and UCF Sports datasets, respectively.

Table 2 compares the results of the best performing architectures among both our

proposed approach and GA standard approach for KTH, Weizmann, and UCF Sports datasets. The table compares the best-performing architectures in terms of recognition accuracy, the number of trainable parameters, and overfitting measure.

For KTH dataset, the conducted experiments yielded recognition accuracy of 97.92%, for our proposed approach, and 95.83%, for the standard approach, which show an increase in recognition accuracy with approximately 2% for our proposed approach. Also, the results show that the proposed approach yielded an architecture with fewer trainable parameters. The numbers of trainable parameters for the generated architectures are 2,545,830, for our approach, vs. 3,685,510, for the standard approach, with a reduction percentage of 30%. The results show an overfitting measure of 0.0135, for the proposed approach, and 0.0417, for the standard approach.

For UCF Sports dataset, the experiments resulted in recognition accuracy of 94.87%, for our proposed approach, and 92.31%, for the standard approach. The results show an increase in recognition accuracy by 2.56% for our proposed approach. Also, the results show that the proposed approach yielded an architecture with fewer trainable parameters. The numbers of trainable parameters for the generated architectures are 3,798,44, for our approach, vs. 6,404,429, for the standard approach, with reduction percentage of 40%. Also, the results show an overfitting measure of 0.0336, for the proposed approach, and 0.0718, for the standard approach.

For Weizmann dataset, the experiments show accuracy of 96.77%, for our proposed approach, and 100%, for the standard approach. In the proposed approach, the misclassifications are only in the skip action test cases. Some of the skip action test cases are classified as run and jump actions. The resemblance between these three actions can be seen in Fig. 8, the actors are moving quickly showing their sideview with one or both legs not touching the ground. But still, the results show that the proposed approach yielded an architecture with fewer trainable parameters. The numbers of trainable parameters for the generated architectures are 4,594,794, for our approach, vs. 20,739,594, for the standard approach.

Fig. 10, Fig. 11, and Fig. 12 show the best performing generated architectures elected from the last epoch. The figures are labelled with the layer types, kernel sizes, number of filters, input, and output.

The results show that the proposed approach generated architectures that yielded higher recognition accuracy, along with fewer trainable parameters, and less overfitting measure for both KTH and UCF Sports datasets. For Weizmann dataset, the generated architecture yielded less recognition accuracy, yet comparable. Also, our approach generated an architecture with substantially fewer trainable parameters for Weizmann dataset.

At early GA epochs, models generated from population chromosomes are trained for 20 epochs. The number of training epochs increases with the number of GA epochs till each model is trained for 80 epochs. Models generated from chromosomes in the same GA epoch are evaluated based on training for the same number of epochs.

5.3 Comparison with existing approaches and state-of-the-art architectures

In this section, we compare our GA-based generated architectures with some of the existing approaches and state-of-the-art architectures. We use recognition accuracy as the comparison metric because it serves as a reliable measure for evaluating the performance of various approaches. The comparison is illustrated in Table 3. We compare our approach with SVM method [Schuldt et al. 2004], conventional CNN architecture, RNN architecture, two stream CNN architecture, squirrel search optimization technique,

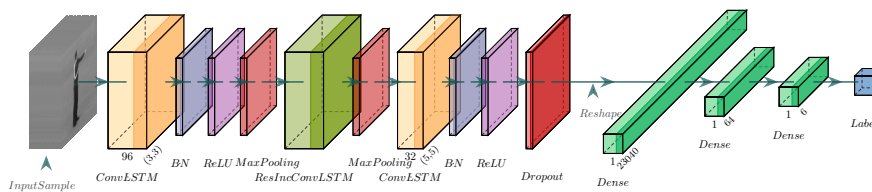


Figure 10: KTH Generated Architecture.

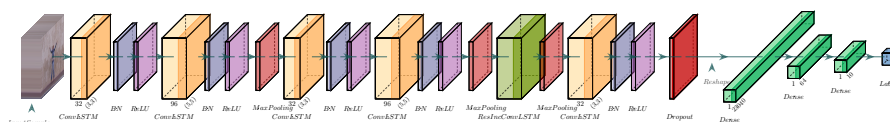


Figure 11: Weizmann Generated Architecture.

and hyperparameters optimization technique. When compared using KTH dataset, our approach is found to outperform [Rao et al. 2023, El Assal et al. 2023, Singh and Singhal 2023, Jaouedi et al. 2020, Schuldt et al. 2004]. Also, our approach shows comparable performance when compared with [Mahmoud et al. 2022, Liu et al. 2021]. When compared using Weizmann dataset, our approach is found to outperform [Singh and Singhal 2023, Schuldt et al. 2004]. Also, our approach shows comparable performance when compared with [Rao et al. 2023, Mahmoud et al. 2022, Liu et al. 2021]. When compared using UCF Sports dataset, our approach is found to outperform [El Assal et al. 2023, Nandagopal et al. 2023, Ishwarya and Nithya 2023, Jaouedi et al. 2020, Schuldt et al. 2004].

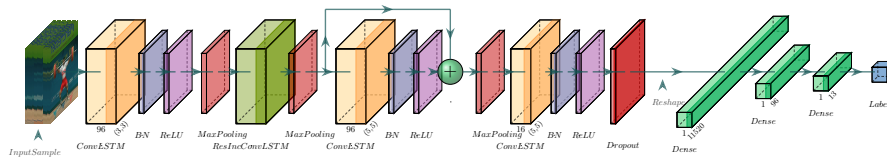


Figure 12: UCF Sports Generated Architecture.

Reference	Method	Year	KTH Acc. (%)	Weiz. Acc. (%)	UCF Sports Acc. (%)
Rao et al. [Rao et al. 2023]	CNN with Silhouette representation	2023	95.6%	98%	-
El-Assal et al. [El Assal et al. 2023]	Two Stream CNN	2023	64.97%	-	71.23%
Singh and Singhal [Singh and Singhal 2023]	CNN+RNN	2023	88%	91%	-
Nandagopal et al. [Nandagopal et al. 2023]	Hyperparameter Optimization	2023	-	-	85.44%
Ishwarya and Alice Nithya [Ishwarya and Nithya 2023]	SSO	2023	-	-	91.2%
Mahmoud et al. [Mahmoud et al. 2022]	DNN	2022	97.5 %	98.7%	-
Liu et al. [Liu et al. 2021]	Two Stream Network with Faster R-CNN	2021	98.83%	99.10%	-
Jaouedi et al. [Jaouedi et al. 2020]	Hybrid Deep Learning	2020	96.30%	-	89.01%
Schuldt et al. [Schuldt et al. 2004]	SVM	2004	95.17%	91.11%	78%
Ijjina and Chalavadi 2016 [Ijjina and Chalavadi 2016] Hourch et al. 2021 [Hourch et al. 2021]	Standard GA with ConvLSTM layer	2021	95.83%	100%	92.31%
Proposed approach	Multi-objective GA	2023	97.92%	96.77%	94.87%

Table 3: Comparison with Existing Approaches and State-of-the-art Architectures.

6 Conclusions and Future Work

In this paper, we addressed the HAR problem by proposing a NAS approach. Our method automatically designs ConvLSTM-based architectures from scratch using a multi-objective GA that maximizes recognition accuracy, and minimizes the number of trainable parameters and overfitting measure, using ConvLSTM, Residual ConvLSTM, Inception ConvLSTM, and Residual Inception ConvLSTM layers as basic building blocks. We demonstrated the effectiveness of our method by achieving recognition accuracies of 97.92%, 96.77%, and 94.87% on KTH, Weizmann, and UCF Sports datasets, respectively, outperforming state-of-the-art HAR solutions by up to 9.92%, 5.77%, and 23.64%. Additionally, our method surpassed some existing NAS approaches with improvements of approximately 2%, 11%, and 4% for KTH, Weizmann, and UCF Sports, respectively. Our approach addresses key research gaps by generating ConvLSTM-based architectures from scratch, rather than refining existing models, allowing for greater flexibility in identifying optimal designs. Moreover, it tailors the optimization process to specific datasets and offers flexibility through the integration of various layer architecture types, enhancing its adaptability to diverse tasks.

Although our work shows robust and promising results, certain limitations provide opportunities for improvement. While we proposed exploring different layer types in ConvLSTM-based architecture generation, our study may still be constrained by the specific types of layers used. Adding more layer types could have a significant impact without requiring much additional effort. Another key limitation is the time-consuming nature of NAS, especially during the early stages of training and evaluating each candidate architecture. The resource-intensive process points to the potential for optimizing resource allocation through a training-free objective function. In the future, several directions can be explored to enhance the scope and impact of our approach. Expanding the architectural search space by experimenting with a wider variety of layer types and testing the approach on additional HAR datasets would improve its robustness and adaptability. Another promising avenue involves the use of a training-free objective function during the initial stages of training, followed by our proposed fitness function, to optimize resource usage. Refining the multi-objective function further could also enhance the precision and effectiveness of the search process. Additionally, for domain-specific applications, integrating state-of-the-art architectures as initial chromosomes alongside randomly initialized ones would leverage existing high-performing models while maintaining flexibility, broadening the applicability without restricting it to known designs. Lastly, while validated on HAR, the method could be extended to other NAS tasks, such as speech recognition and semantic segmentation, increasing its potential across different domains.

References

- [Abdelbaky and Aly 2021] A. Abdelbaky, S. Aly, "Human action recognition using three orthogonal planes with unsupervised deep convolutional neural network," *Multimedia Tools and Applications*, vol. 80, no. 13, pp. 20019–20043, 2021, <https://doi.org/10.1007/s11042-021-10636-2>.
- [Ahn and Cho 2021] J. Y. Ahn, N. I. Cho, "Multi-branch neural architecture search for lightweight image super-resolution," *IEEE Access*, vol. 9, pp. 153633–153646, 2021, <https://doi.org/10.1109/ACCESS.2021.3127437>.
- [Althloothi et al. 2014] S. Althloothi, M. H. Mahoor, X. Zhang, R. M. Voyles, "Human activity recognition using multi-features and multiple kernel learning," *Pattern recognition*, vol. 47, no. 5, pp. 1800–1812, 2014, <https://doi.org/10.1016/j.patcog.2013.11.032>.

- [Aly and Sayed 2019] S. Aly, A. Sayed, "Human action recognition using bag of global and local zernike moment features," *Multimedia Tools and Applications*, vol. 78, no. 17, pp. 24923–24953, 2019, <https://doi.org/10.1007/s11042-019-7674-5>.
- [Anguita et al. 2013] Anguita, D., Ghio, A., Oneto, L., Llanas Parra, F. & Reyes Ortiz, J. Energy efficient smartphone-based activity recognition using fixed-point arithmetic. *Journal Of Universal Computer Science*. 19, 1295-1314 (2013)
- [Arshad et al. 2022] M. H. Arshad, M. Bilal, A. Gani, "Human activity recognition: Review, taxonomy and open challenges," *Sensors*, vol. 22, no. 17, p. 6463, 2022, <https://doi.org/10.3390/s22176463>.
- [Arunnehru et al. 2018] J. Arunnehru, G. Chamundeeswari, S. P. Bharathi, "Human action recognition using 3d convolutional neural networks with 3d motion cuboids in surveillance videos," *Procedia computer science*, vol. 133, pp. 471–477, 2018, <https://doi.org/10.1016/j.procs.2018.07.059>.
- [Cai et al. 2023] Z. Cai, L. Chen, S. Zeng, Y. Lai, H.-I. Liu, "Est-nas: An evolutionary strategy with gradient descent for neural architecture search," *Applied Soft Computing*, vol. 146, p. 110624, 2023, <https://doi.org/10.1016/j.asoc.2023.110624>.
- [David and Greental 2014] O. E. David, I. Greental, "Genetic algorithms for evolving deep neural networks," in *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 2014, pp. 1451–1452.
- [De et al. 2017] P. De, A. Chatterjee, A. Rakshit, "Recognition of human behavior for assisted living using dictionary learning approach," *IEEE Sensors Journal*, vol. 18, no. 6, pp. 2434–2441, 2017, <https://doi.org/10.1109/JSEN.2017.2787616>.
- [Ding et al. 2023] W. Ding, M. Abdel-Basset, R. Mohamed, "Har-deepconvlg: Hybrid deep learning-based model for human activity recognition in iot applications," *Information Sciences*, vol. 646, p. 119394, 2023.
- [Dong and Yang 2019] X. Dong, Y. Yang, "One-shot neural architecture search via self-evaluated template network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3681–3690.
- [Ducros 2023] S. Ducros, "Self-supervised learning for neural architecture search (nas)," *arXiv preprint arXiv:2304.01023*, 2023, <https://doi.org/10.48550/arXiv.2304.01023>.
- [El Assal et al. 2023] M. El-Assal, P. Tirilly, I. M. Bilasco, "Spiking two-stream methods with unsupervised stdp-based learning for action recognition," *arXiv preprint arXiv:2306.13783*, 2023, <https://doi.org/10.48550/arXiv.2306.13783>.
- [Elsken et al. 2019] T. Elsken, J. H. Metzen, F. Hutter, "Neural architecture search: A survey," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1997–2017, 2019, <https://doi.org/10.48550/arXiv.1808.05377>.
- [Elsken et al. 2020] T. Elsken, B. Staffler, J. H. Metzen, F. Hutter, "Meta-learning of neural architectures for few-shot learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12365–12375.
- [Ghosh and Jana 2020] Ghosh, A. & Jana, N. Neural architecture search with improved genetic algorithm for image classification. *2020 International Conference On Computational Performance Evaluation (ComPE)*. pp. 344-349 (2020)
- [Guo et al. 2020] Z. Guo, X. Zhang, H. Mu, W. Heng, Z. Liu, Y. Wei, J. Sun, "Single path one-shot neural architecture search with uniform sampling," in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, 2020, pp. 544–560, Springer.
- [Haddad et al. 2021] M. Haddad, V. K. Ghassab, F. Najjar, N. Bouguila, "A statistical framework for few-shot action recognition," *Multimedia Tools and Applications*, vol. 80, no. 16, pp. 24303–24318, 2021, <https://doi.org/10.1007/s11042-021-10721-6>.

- [Han et al. 2018] Y. Han, P. Zhang, T. Zhuo, W. Huang, Y. Zhang, “Going deeper with two-stream convnets for action recognition in video surveillance,” *Pattern Recognition Letters*, vol. 107, pp. 83–90, 2018, <https://doi.org/10.1016/j.patrec.2017.08.015>.
- [He et al. 2016] K. He, X. Zhang, S. Ren, J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [Heravi et al 2024] Heravi, M., Jang, Y., Jeong, I. & Sarkar, S. Deep learning-based activity-aware 3D human motion trajectory prediction in construction. *Expert Systems With Applications*. 239 pp. 122423 (2024)
- [Houreh et al. 2021] Y. Houreh, M. Mahdinejad, E. Naredo, D. M. Dias, C. Ryan, “Hnas: Hyper neural architecture search for image segmentation,” in *ICAART (2)*, 2021, pp. 246–256.
- [Houreh et al. 2021] Houreh, Y., Mahdinejad, M., Naredo, E., Mota Dias, D. & Ryan, C. HNAS: hyper neural architecture search for image segmentation. (University of Limerick, 2021)
- [Hsu et al. 2018] C.-H. Hsu, S.-H. Chang, J.-H. Liang, H.-P. Chou, C.-H. Liu, S.-C. Chang, J.-Y. Pan, Y.-T. Chen, W. Wei, D.-C. Juan, “Monas: Multi-objective neural architecture search using reinforcement learning,” *arXiv preprint arXiv:1806.10332*, 2018, <https://doi.org/10.48550/arXiv.1806.10332>.
- [Ijjina and Chalavadi 2016] E. P. Ijjina, K. M. Chalavadi, “Human action recognition using genetic algorithms and convolutional neural networks,” *Pattern recognition*, vol. 59, pp. 199–212, 2016, <https://doi.org/10.1016/j.patcog.2016.01.012>.
- [Rao et al. 2023] N. S. Rao, G. Shanmugapriya, S. Vinod, S. Raju, S. P. Mallick, et al., “Detecting human behavior from a silhouette using convolutional neural networks,” in *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, 2023, pp. 943–948, IEEE.
- [Ishwarya and Nithya 2023] K. Ishwarya, A. A. Nithya, “Squirrel search optimization with deep convolutional neural network for human pose estimation,” *Computers, Materials & Continua*, vol. 74, no. 3, 2023, <https://doi.org/10.32604/cmc.2023.034654>.
- [Jaafra et al. 2019] Y. Jaafra, J. L. Laurent, A. Deruyver, M. S. Naceur, “Reinforcement learning for neural architecture search: A review,” *Image and Vision Computing*, vol. 89, pp. 57–66, 2019, <https://doi.org/10.1016/j.imavis.2019.06.005>.
- [Jahlan and Elrefaei 2021] Jahlan, H. & Elrefaei, L. Mobile neural architecture search network and convolutional long short-term memory-based deep features toward detecting violence from video. *Arabian Journal For Science And Engineering*. 46, 8549-8563 (2021)
- [Jalal et al. 2012] A. Jalal, S. Lee, J. T. Kim, T.-S. Kim, “Human activity recognition via the features of labeled depth body parts,” in *International Conference on Smart Homes and Health Telematics*, 2012, pp. 246–249, Springer.
- [Jalal et al. 2015] A. Jalal, Y. Kim, S. Kamal, A. Farooq, D. Kim, “Human daily activity recognition with joints plus body features representation using kinect sensor,” in *2015 International Conference on Informatics, Electronics & Vision (ICIEV)*, 2015, pp. 1–6, IEEE.
- [Jing et al. 2021] L. Jing, T. Parag, Z. Wu, Y. Tian, H. Wang, “Videoss semi-supervised learning for video classification,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1110–1119.
- [Jlidi et al. 2024] Jlidi, N., Kouni, S., Jemai, O. & Bouchrika, T. MediaPipe with GNN for Human Activity Recognition.. *Journal Of Universal Computer Science (JUICS)*. 30 (2024)
- [Kang et al. 2023] J.-S. Kang, J. Kang, J.-J. Kim, K.-W. Jeon, H.-J. Chung, B.-H. Park, “Neural architecture search survey: A computer vision perspective,” *Sensors*, vol. 23, no. 3, p. 1713, 2023, <https://doi.org/10.3390/s23031713>.
- [Kaplan and Giryes 2020] S. Kaplan, R. Giryes, “Self-supervised neural architecture search,” *arXiv preprint arXiv:2007.01500*, 2020, <https://doi.org/10.48550/arXiv.2007.01500>.

- [Kaya and Kuncan 2022] Kaya, Y. & Kuncan, F. A hybrid model for classification of medical data set based on factor analysis and extreme learning machine: FA+ ELM. *Biomedical Signal Processing And Control*. 78 pp. 104023 (2022)
- [Khaire et al. 2018] P. Khaire, P. Kumar, J. Imran, "Combining cnn streams of rgb-d and skeletal data for human activity recognition," *Pattern Recognition Letters*, vol. 115, pp. 107–116, 2018, <https://doi.org/10.1016/j.patrec.2018.04.035>.
- [Khater et al. 2022] S. Khater, M. Hadhoud, M. B. Fayek, "A novel human activity recognition architecture: using residual inception convlstm layer," *Journal of Engineering and Applied Science*, vol. 69, no. 1, p. 45, 2022, <https://doi.org/10.1186/s44147-022-00098-0>.
- [Kumar and John 2016] S. S. Kumar, M. John, "Human activity recognition using optical flow based feature set," in 2016 IEEE international Carnahan conference on security technology (ICCST), 2016, pp. 1–5, IEEE.
- [Kumar et al. 2024] Kumar, M., Patel, A. & Biswas, M. Real-time detection of abnormal human activity using deep learning and temporal attention mechanism in video surveillance. *Multimedia Tools And Applications*. 83, 55981-55997 (2024)
- [Kuncan et al. 2019] Kuncan, F., Kaya, Y. & Kuncan, M. A novel approach for activity recognition with down-sampling 1D local binary pattern. *Advances In Electrical And Computer Engineering*. 19, 35-44 (2019)
- [Kuncan et al. 2022] Kuncan, F., Kaya, Y., Yiner, Z. & Kaya, M. A new approach for physical human activity recognition from sensor signals based on motif patterns and long-short term memory. *Biomedical Signal Processing And Control*. 78, pp. 103963 (2022)
- [Kuncan et al. 2022] Kuncan, F., Kaya, Y., Tekin, R. & Kuncan, M. A new approach for physical human activity recognition based on co-occurrence matrices. *The Journal Of Supercomputing*. 78, 1048-1070 (2022)
- [Li et al. 2020] C. Li, J. Peng, L. Yuan, G. Wang, X. Liang, L. Lin, X. Chang, "Block-wisely supervised neural architecture search with knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1989–1998.
- [Li et al. 2021] C. Li, T. Tang, G. Wang, J. Peng, B. Wang, X. Liang, X. Chang, "Bossnas: Exploring hybrid cnn-transformers with block-wisely self-supervised neural architecture search," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12281–12291.
- [Lian et al. 2019] D. Lian, Y. Zheng, Y. Xu, Y. Lu, L. Lin, P. Zhao, J. Huang, S. Gao, "Towards fast adaptation of neural architectures with meta learning," in *International Conference on Learning Representations*, 2019.
- [Lin and Tsai 2024] Lin, J. & Tsai, C. A Lightweight Training-Free Method for Neural Architecture Search. 2024 IEEE Congress On Evolutionary Computation (CEC). pp. 1-8 (2024)
- [Lin et al. 2020] Z. Lin, M. Li, Z. Zheng, Y. Cheng, C. Yuan, "Self-attention convlstm for spatiotemporal prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, 2020, pp. 11531–11538.
- [Liu et al. 2018] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, K. Murphy, "Progressive neural architecture search," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 19–34.
- [Liu et al. 2023] J. Liu, R. Cheng, Y. Jin, "Bi-fidelity evolutionary multiobjective search for adversarially robust deep neural architectures," *Neurocomputing*, p. 126465, 2023, <https://doi.org/10.1016/j.neucom.2023.126465>.
- [Zheng et al. 2020] X. Zheng, R. Ji, Q. Wang, Q. Ye, Z. Li, Y. Tian, Q. Tian, "Rethinking performance estimation in neural architecture search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11356–11365.

- [Lu et al. 2019] Lu, Z., Whalen, I., Boddeti, V., Dhebar, Y., Deb, K., Goodman, E. & Banzhaf, W. Nsga-net: neural architecture search using multi-objective genetic algorithm. *Proceedings Of The Genetic And Evolutionary Computation Conference*. pp. 419-427 (2019)
- [Lu et al. 2021] Z. Lu, G. Sreekumar, E. Goodman, W. Banzhaf, K. Deb, V. N. Boddeti, "Neural architecture transfer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 9, pp. 2971–2989, 2021, <https://doi.org/10.1109/TPAMI.2021.3052758>.
- [Lu et al. 2023] Z. Lu, R. Cheng, Y. Jin, K. C. Tan, K. Deb, "Neural architecture search as multi-objective optimization benchmarks: Problem formulation and performance assessment," *IEEE transactions on evolutionary computation*, 2023, <https://doi.org/10.1109/TEVC.2022.3233364>.
- [Luo et al. 2020] R. Luo, X. Tan, R. Wang, T. Qin, E. Chen, T.-Y. Liu, "Semi-supervised neural architecture search," *Advances in Neural Information Processing Systems*, vol. 33, pp. 10547–10557, 2020.
- [Mahmoud et al. 2022] R. Mahmoud, S. Belgacem, M. N. Omri, "Towards an end-to-end isolated and continuous deep gesture recognition process," *Neural Computing and Applications*, vol. 34, no. 16, pp. 13713–13732, 2022, <https://doi.org/10.1007/s00521-022-07165-w>.
- [Jaouedi et al. 2020] N. Jaouedi, N. Boujnah, M. S. Bouhlel, "A new hybrid deep learning model for human action recognition," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 4, pp. 447–453, 2020, <https://doi.org/10.1016/j.jksuci.2019.09.004>.
- [Miikkulainen et al. 2019] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, et al., "Evolving deep neural networks," in *Artificial intelligence in the age of neural networks and brain computing*, R. Kozma, C. Alippi, Y. Choe, F. C. Morabito Eds., Elsevier, 2019, pp. 293–312, <https://doi.org/10.1016/B978-0-323-96104-2.00002-6>.
- [Nandagopal et al. 2023] S. Nandagopal, G. Karthy, A. S. Oliver, M. Subha, "Optimal deep convolutional neural network with pose estimation for human activity recognition.," *Computer Systems Science & Engineering*, vol. 44, no. 2, 2023, <https://doi.org/10.32604/csse.2023.028003>.
- [Nazir et al. 2018] S. Nazir, M. H. Yousaf, S. A. Velastin, "Evaluating a bag-of-visual features approach using spatio-temporal features for action recognition," *Computers & Electrical Engineering*, vol. 72, pp. 660–669, 2018, <https://doi.org/10.1016/j.compeleceng.2018.01.037>.
- [Niu and Abdel-Mottaleb 2004] F. Niu, M. Abdel-Mottaleb, "View-invariant human activity recognition based on shape and motion features," in *IEEE Sixth International Symposium on Multimedia Software Engineering*, 2004, pp. 546–556, IEEE.
- [O'Neill et al. 2021] D. O'Neill, B. Xue, M. Zhang, "Evolutionary neural architecture search for high-dimensional skip-connection structures on densenet style networks," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 6, pp. 1118–1132, 2021, <https://doi.org/10.1109/TEVC.2021.3083315>.
- [Pavlitskaya et al. 2022] S. Pavlitskaya, J. Oswald, J. M. Zöllner, "Measuring overfitting in convolutional neural networks using adversarial perturbations and label noise," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2022, pp. 1551–1559, IEEE.
- [Peng et al. 2020] Y. Peng, L. Bi, M. Fulham, D. Feng, J. Kim, "Multi-modality information fusion for radiomics-based neural architecture search," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VII 23*, 2020, pp. 763–771, Springer.
- [Qi et al. 2018] M. Qi, J. Qin, A. Li, Y. Wang, J. Luo, L. Van Gool, "stagnet: An attentive semantic rnn for group activity recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 101–117.
- [Ramya and Rajeswari 2021] P. Ramya, R. Rajeswari, "Human action recognition using distance transform and entropy based features," *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 8147–8173, 2021, <https://doi.org/10.1007/s11042-020-10140-z>.

- [Salehin et al. 2024] Salehin, I., Islam, M., Saha, P., Noman, S., Tunj, A., Hasan, M. & Baten, M. AutoML: A systematic review on automated machine learning with neural architecture search. *Journal Of Information And Intelligence*. 2, 52-81 (2024)
- [Schuldt et al. 2004] C. Schuldt, I. Laptev, B. Caputo: "Recognizing human actions: a local svm approach," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 3, 2004, pp. 32–36, IEEE.
- [Shang et al. 2022] R. Shang, S. Zhu, J. Ren, H. Liu, L. Jiao, "Evolutionary neural architecture search based on evaluation correction and functional units," *Knowledge-Based Systems*, vol. 251, p. 109206, 2022, <https://doi.org/10.1016/j.knsys.2022.109206>.
- [Shen et al. 2023] Y. Shen, Y. Li, J. Zheng, W. Zhang, P. Yao, J. Li, S. Yang, J. Liu, B. Cui, "Proxybo: Accelerating neural architecture search via bayesian optimization with zero-cost proxies," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 9792–9801.
- [Shi et al. 2015] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, 2015.
- [Shi et al. 2020] H. Shi, R. Pi, H. Xu, Z. Li, J. Kwok, T. Zhang, "Bridging the gap between sample-based and one-shot neural architecture search with bonas," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1808–1819, 2020.
- [Singh and Singhal 2023] U. Singh, N. Singhal, "Exploiting video classification using deep learning models for human activity recognition," in *Computer Vision and Robotics: Proceedings of CVR 2022*, Springer, 2023, pp. 169–179.
- [Liu et al. 2021] C. Liu, J. Ying, H. Yang, X. Hu, J. Liu, "Improved human action recognition approach based on two-stream convolutional neural network model," *The visual computer*, vol. 37, pp. 1327–1341, 2021, <https://doi.org/10.1007/s00371-020-01868-8>.
- [Singh et al. 2021] A. Singh, O. Chakraborty, A. Varshney, R. Panda, R. Feris, K. Saenko, A. Das, "Semi-supervised action recognition with temporal contrastive learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021*, pp. 10389–10399.
- [Song et al. 2018] H. Song, W. Wang, S. Zhao, J. Shen, K.-M. Lam, "Pyramid dilated deeper convlstm for video salient object detection," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 715–731.
- [Szegedy et al. 2015] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition, 2015*, pp. 1–9.
- [Tang et al. 2020] Y. Tang, Y. Wang, Y. Xu, H. Chen, B. Shi, C. Xu, C. Xu, Q. Tian, C. Xu, "A semi-supervised assessor of neural architectures," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020*, pp. 1810–1819.
- [Thanh et al. 2024] Thanh, T., Doan, L., Luong, N. & Huynh Thi Thanh, B. THNAS-GA: A Genetic Algorithm for Training-free Hardware-aware Neural Architecture Search. *Proceedings Of The Genetic And Evolutionary Computation Conference*. pp. 1128-1136 (2024)
- [Liang et al. 2024] Liang, J., Cao, H., Lu, Y. & Su, M. Architecture search of accurate and lightweight CNNs using genetic algorithm. *Genetic Programming And Evolvable Machines*. 25, 13 (2024)
- [Tirumala et al. 2016] S. S. Tirumala, S. Ali, C. P. Ramesh, "Evolving deep neural networks: A new prospect," in *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2016, pp. 69–74, IEEE.
- [Torrey and Shavlik 2010] L. Torrey, J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, IGI global, 2010, pp. 242–264.

- [Tran et al. 2021] L.-T. Tran, M. S. Ali, S.-H. Bac, "A feature fusion based indicator for training-free neural architecture search," *IEEE Access*, vol. 9, pp. 133914–133923, 2021, <https://doi.org/10.1109/ACCESS.2021.3115911>.
- [Tu et al. 2023] R. Tu, M. Khodak, N. C. Roberts, N. Balcan, A. Talwalkar, "Nas-bench-360: Benchmarking diverse tasks for neural architecture search," 2023.
- [Tuncer et al. 2020] Tuncer, T., Ertam, F., Dogan, S. & Subasi, A. An automated daily sports activities and gender recognition method based on novel multikernel local diamond pattern using sensor signals. *IEEE Transactions On Instrumentation And Measurement*. 69, 9441-9448 (2020)
- [Vilalta and Drissi 2002] R. Vilalta, Y. Drissi, "A perspective view and survey of meta-learning," *Artificial intelligence review*, vol. 18, pp. 77–95, 2002, <https://doi.org/10.1023/A:1019956318069>.
- [Vrskova et al. 2021] Vrskova, R., Sykora, P., Kamencay, P., Hudec, R. & Radil, R. Hyperparameter tuning of ConvLSTM network models. 2021 44th International Conference On Telecommunications And Signal Processing (TSP). pp. 15-18 (2021)
- [Wang and Ji 2012] X. Wang, Q. Ji, "Learning dynamic bayesian network discriminatively for human activity recognition," in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, 2012, pp. 3553–3556, IEEE.
- [Wang and Schmid 2013] Wang, H. & Schmid, C. Action recognition with improved trajectories. *Proceedings Of The IEEE International Conference On Computer Vision*. pp. 3551-3558 (2013)
- [Wang et al. 2018] D. Wang, Y. Yang, S. Ning, "Deepstcl: A deep spatio-temporal convlstm for travel demand prediction," in *2018 international joint conference on neural networks (IJCNN)*, 2018, pp. 1–8, IEEE.
- [Wang et al. 2020] J. Wang, J. Wu, H. Bai, J. Cheng, "M-nas: Meta neural architecture search," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 6186–6193.
- [Wang et al. 2021] D. Wang, M. Li, C. Gong, V. Chandra, "Attentivenas: Improving neural architecture search via attentive sampling," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6418–6427.
- [Wei et al. 2018] H. Wei, H. Zhou, J. Sankaranarayanan, S. Sengupta, H. Samet, "Residual convolutional lstm for tweet count prediction," in *Companion Proceedings of the The Web Conference 2018*, 2018, pp. 1309–1316.
- [Wen et al. 2022] Wen, L., Gao, L., Li, X. & Li, H. A new genetic algorithm based evolutionary neural architecture search for image classification. *Swarm And Evolutionary Computation*. 75 pp. 101191 (2022)
- [White et al. 2021] C. White, W. Neiswanger, Y. Savani, "Bananas: Bayesian optimization with neural architectures for neural architecture search," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 10293–10301.
- [White et al. 2023] White, C., Safari, M., Sukthankar, R., Ru, B., Elsken, T., Zela, A., Dey, D. & Hutter, F. Neural architecture search: Insights from 1000 papers. *ArXiv Preprint ArXiv:2301.08727*. (2023)
- [Wistuba et al. 2019] M. Wistuba, A. Rawat, T. Pedapati, "A survey on neural architecture search," *arXiv preprint arXiv:1905.01392*, 2019, <https://doi.org/10.48550/arXiv.1905.01392>.
- [Xie and Yuille 2017] L. Xie, A. Yuille, "Genetic cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1379–1388.
- [Xue et al. 2021] S. Xue, H. Chen, C. Xie, B. Zhang, X. Gong, D. Doermann, "Fast and unsupervised neural architecture evolution for visual representation learning," *IEEE Computational Intelligence Magazine*, vol. 16, no. 3, pp. 22–32, 2021, <https://doi.org/10.1109/MCI.2021.3084394>.
- [Yang et al. 2017] D. Yang, T. Xiong, D. Xu, S. K. Zhou, Z. Xu, M. Chen, J. Park, S. Grbic, T. D. Tran, S. P. Chin, et al., "Deep image-to-image recurrent network with shape basis learning for automatic vertebra labeling in large-scale 3d ct volumes," in *Medical Image Computing and*

Computer Assisted Intervention- MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part III 20, 2017, pp. 498–506, Springer.

[Yin et al. 2021] Z. Yin, K. Xia, Z. He, J. Zhang, S. Wang, B. Zu, “Unpaired image denoising via wasserstein gan in low-dose ct image with multi-perceptual loss and fidelity loss,” *Symmetry*, vol. 13, no. 1, p. 126, 2021, <https://doi.org/10.3390/sym13010126>.

[Yuan et al. 2018] Z. Yuan, X. Zhou, T. Yang, “Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data,” in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 984–992.

[Zela et al. 2020] A. Zela, J. Siems, F. Hutter, “Nas-bench-1shot1: Benchmarking and dissecting one-shot neural architecture search,” *arXiv preprint arXiv:2001.10422*, 2020, <https://doi.org/10.48550/arXiv.2001.10422>.

[Zhang et al. 2018] K. Zhang, L. Zhang, “Extracting hierarchical spatial and temporal features for human action recognition,” *Multimedia Tools and Applications*, vol. 77, no. 13, pp. 16053–16068, 2018, <https://doi.org/10.1007/s11042-017-5179-7>.

[Zhao et al. 2021] Y. Zhao, L. Wang, Y. Tian, R. Fonseca, T. Guo, “Few-shot neural architecture search,” in *International Conference on Machine Learning*, 2021, pp. 12707–12718, PMLR.

[Zoph and Le 2016] B. Zoph, Q. V. Le, “Neural architecture search with reinforcement learning,” *arXiv preprint arXiv:1611.01578*, 2016, <https://doi.org/10.48550/arXiv.1611.01578>.