



Flu-Net: two-stream deep heterogeneous network to detect flu like symptoms from videos using grey wolf optimization algorithm

Himanshu Gupta¹ · Javed Imran² · Chandani Sharma³

Received: 10 September 2022 / Accepted: 20 February 2023 / Published online: 31 March 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

The outbreak of COVID-19 (also known as Coronavirus) has put the entire world at risk. The disease first appears in Wuhan, China, and later spread to other countries, taking a form of a pandemic. In this paper, we try to build an artificial intelligence (AI) powered framework called Flu-Net to identify flu-like symptoms (which is also an important symptom of Covid-19) in people, and limit the spread of infection. Our approach is based on the application of human action recognition in surveillance systems, where videos captured by closed-circuit television (CCTV) cameras are processed through state-of-the-art deep learning techniques to recognize different activities like coughing, sneezing, etc. The proposed framework has three major steps. First, to suppress irrelevant background details in an input video, a frame difference operation is performed to extract foreground motion information. Second, a two-stream heterogeneous network based on 2D and 3D Convolutional Neural Networks (ConvNets) is trained using the RGB frame differences. And third, the features extracted from both the streams are combined using Grey Wolf Optimization (GWO) based feature selection technique. The experiments conducted on BII Sneeze-Cough (BIISC) video dataset show that our framework can 70% accuracy, outperforming the baseline results by more than 8%.

Keywords COVID-19 · Deep learning · Two-stream ConvNet · Action recognition · Grey Wolf optimizer

1 Introduction

COVID-19 (2019) is a highly infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). Compared to the previous epidemic diseases like Ebola (2014) and SARS (2002), COVID-19 has a much lower fatality rate but spreads much faster. As of now, approximately 225 countries and territories have reported more than 591 million cases of COVID-19. Due to this, the

World Health Organization (WHO) has declared COVID-19 a global pandemic.¹ Further, the initial symptoms of COVID-19 are quite similar to common flu, as both are respiratory diseases and could cause fever, cough, and fatigue. Thus, it is very important to quickly identify the people with such symptoms and send them to quarantine, where formal tests like Reverse Transcription Polymerase Chain Reaction (RT-PCR) could be conducted to verify the presence of coronavirus.

As per WHO, coronavirus can spread through contact, droplets and fomites, and may survive on surfaces for up to 72 h. Further, since no vaccine has 100% efficacy against COVID-19, the most effective method to prevent the spreading of the virus is to implement a total lockdown, where the general public is advised to stay in their homes. However, lockdown is only a short-term solution to this health crisis. Once the lockdown is lifted, people will come out in the open environment, and again become susceptible to infection. The riskiest places are airports,

✉ Javed Imran
javed.imran@thapar.edu

Himanshu Gupta
hguptasb@gmail.com

Chandani Sharma
chandani19nov@gmail.com

¹ Department of Computer Science and Engineering, Quantum University, Roorkee, India

² Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, India

³ Department of Computer Science and Engineering, Quantum University, Roorkee, India

¹ <https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19—11-march-2020>.

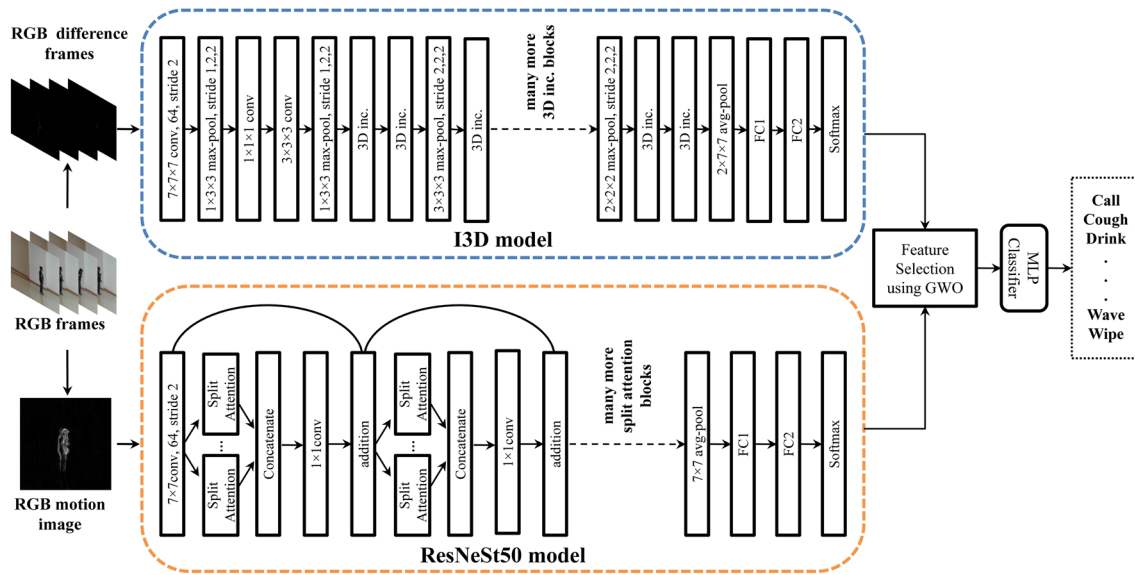


Fig. 1 Architecture of Flu-Net based on ResNeSt50 and I3D models. Our proposed framework has three steps. First, frame difference operation is performed to remove background details and retain only foreground motion information. Second, frame differences are directly

transport stations, shopping malls, conference venues, public administrations, etc., where the chance of people coming in contact with each other is very high. This could result in a new wave of coronavirus, which will put our healthcare systems under unimaginable strain, especially in densely populated countries like India and China. However, if we could easily detect people in public places who show potential symptoms of the infections, we could isolate them fast to prevent spreading the infection further. To this end, we propose a two-stream heterogeneous network, called Flu-Net, to recognize Flu-like symptoms in people present in video streams. As shown in Fig. 1, our framework is built upon ResNeSt (Zhang et al. 2020) and I3D (Carreira and Zisserman 2017) models to learn complementary information from input RGB frames. Specifically, we first perform frame difference operation to suppress the influence of background information and focus on foreground motion areas. These RGB frame differences are directly fed into the I3D model to learn spatio-temporal features from the input video sequences, while for the ResNeSt50 model, we first stack frame differences together to generate RGB motion image (RGBMI) (Imran and Raman 2019), and then use them to train the network. Finally, the features extracted from both the streams are carefully fused together using GWO-based feature selection technique and classified using a Multi-layer Perceptron (MLP). To validate the efficacy of our technique, we conduct experiments on a Sneeze-Cough video dataset (Thi et al. 2014) containing eight common activities like waving, sneezing, coughing, etc. The results show that

fed into I3D stream, while an RGBMI is created for ResNeSt50 stream. Finally, an optimal set of features are selected using GWO technique to train a MLP classifier

such a system could be installed at surveillance stations to quickly identify human subjects with potential Flu-like symptoms.

To summarize, our contributions are two-fold:

1. We are the first to propose such a heterogeneous network based on ResNeSt and I3D networks for video classification task.
2. We also optimized the feature selection process using GWO algorithm and achieve superior results than other state-of-the-art methods.

The rest of the paper is organized as follows. Section 3 presents the explanation of our two-stream heterogeneous network. Section 4 discusses the findings of our experimental results. Finally, Sect. 5 concludes the paper.

2 Related works

Deep neural networks like ConvNets and Long short-term memory (LSTM) have shown tremendous success in a variety of computer vision tasks like image classification (Krizhevsky et al. 2012; Li et al. 2022), video classification (Wang et al. 2016; Kujani and Kumar 2021), object detection (Wang et al. 2022a), gesture recognition (Zhang et al. 2018; Mohammed et al. 2022), video captioning (Natarajan et al. 2022; Wang et al. 2022b), medical image analysis (Deepak and Ameer 2021; Zhou et al. 2022), etc. ConvNets are loosely based on how all humans perceive the world

around them. This basically involves a hierarchical series of feature recognition, starting off with simpler features like edges, blobs, etc., and progresses towards complex or abstract recognitions, like combinations of shapes, and finally, the classification of entire objects. A ConvNet architecture is built by stacking together a series of convolutional layers (Conv), pooling layers (Pool), batch normalization (BN) layers, fully connected (FC) layers, and softmax layers. A Conv layer is made up of a number of learnable filters, which convolve with the given input to extract features and pass them to the next layer. Pool layer is used to combine the output of the previous layer into a single value in the next layer. This is achieved by either taking the average of all the neurons (called AvgPool) or by selecting the maximum value among all the neurons (called MaxPool). BN helps to speed up the training process by standardizing and normalizing the output of the previous layer. FC layer connects every neuron in the previous layer to all the neurons in the next layer. Softmax layer is typically present at the end and performs classification by generating a probability distribution of all the classes.

In this section, we briefly discuss the application of deep neural networks in the area of video action recognition which has applications in several domains like human-computer interaction (HCI), healthcare services, video surveillance, video summarization, and similar other tasks.

2.1 Video action recognition using 2D ConvNets

The first application of ConvNets in video classification tasks is by Karpathy et al. (2014). They propose to input successive RGB frames into a multi-resolution AlexNet-based model for performing action recognition in videos. They highlight that training the network at two different resolutions could help maintaining the same level of accuracy without downgrading the network size. However, this method only fuses spatial features, while giving no consideration to motion features which are most critical for any video classification task. Simonyan and Zisserman (2014a) propose a two-stream network consisting of a spatial stream and a temporal stream. The spatial stream is trained using RGB video frames (spatial information) while the temporal stream utilizes optical flow (motion information). The softmax scores obtained from both the streams are normalized to train an SVM. Wang et al. (2015) propose trajectory-pooled deep convolutional descriptor (TDD) to integrate ConvNet features with handcrafted features to build a more robust feature descriptor for the video classification task. Donahue et al. (2015) propose an encode-decoder architecture called Long-term recurrent convolutional network (LRCN) by stacking LSTM layer with ConvNets, and training the entire network in an end-to-end fashion. Yue-Hei Ng et al. (2015) study two methods for handling full length videos.

In the first method, they examine various convolutional temporal feature pooling architectures to analyze various ConvNet architectures. In the second method, they employ LSTM units on top of ConvNet features to realize video classification as a sequence classification task. Feichtenhofer et al. (2016) explore different fusion strategies of ConvNet streams, and concluded that fusion at the convolutional layer can reduce the number of parameters without any loss in accuracy. They also show that pooling of abstract convolutional features along with the use of high-quality optical flow methods also improves the results. Wang et al. (2016) propose Temporal Segment Network (TSN) to model long-range temporal dependencies in a video sequence, along with several other best practices like sparse sampling, network pre-training, and weighted average fusion of different modalities. Diba et al. (2017) propose a temporal linear encoding (TLE) layer which could be trained in an end-to-end fashion with any ConvNet model. TLE is capable of aggregating sparse feature maps over the entire video, and then projecting it to lower dimensional feature space, making it compact and computationally efficient to process. Girdhar et al. 2017 discuss learnable video-level feature aggregation using a vector of locally aggregated descriptors (VLAD) by splitting the descriptor space into k cells and pooling inside each of the cells. Wu et al. 2018 suggest that videos could be compressed to remove redundant information. They directly feed the motion vectors obtained from video compression into ConvNets, and extract more meaningful information for improved classification. Temporal Relation Network (Zhou et al. 2018) focus on modeling multi-scale temporal relation in videos under a sparse sampling strategy. Lin et al. (2019) propose a Temporal Shift Module (TSM) that can be plugged into 2D ConvNets and could perform temporal modeling by shifting the channels along the temporal dimensions in both forward and backward directions without increasing any computational cost. Feichtenhofer et al. (2019) develop SlowFast network which has two pathways: Slow pathway to capture the semantics of an object, and Fast pathway to capture motion. There are lateral connections from fast pathway to slow pathway so that the motion information can be combined with the semantic information, which ultimately improves the model's performance. Recently, Ryoo et al. (2019) have propose a method called AssembleNet to automatically find neural network architectures at multiple temporal resolutions using different modalities like RGB and optical flow.

2.2 Video action recognition using 3D ConvNets

In addition to previous techniques, where spatial and temporal information is modeled separately, some recent works are based on 3D convolutions, which can directly extract spatio-temporal features over multiple frames. Ji et al. (2012) first

apply 3D convolutions to perform human action recognition. Tran et al. (2015) propose C3D model capable of performing 3D convolutions and 3D pooling operations. Varol et al. 2017 propose long-term temporal convolutions (LTC) to prove that the action recognition accuracy of 3D ConvNets could be improved by learning long-term video representations. Carreira and Zisserman (2017) propose Inception 3D (I3D) model by inflating kernels and filters of 2D ConvNets in the temporal dimension, thereby reusing already established 2D ConvNet architectures pre-trained on ImageNet dataset. Qiu et al. (2017) propose Pseudo-3D Residual Net (P3D ResNet) by combining one $1 \times 3 \times 3$ convolutional layer and one $3 \times 1 \times 1$ convolutional layer so that both spatial and temporal information could be learned simultaneously using non-linear residual connections. Tran et al. (2018) present R(2+1)D model based on a single type of spatiotemporal residual block consisting of 2D spatial convolution followed by 1D temporal convolution. Hara et al. (2018) demonstrate that Kinetics dataset has sufficient data to train deep 3D ConvNets, and such models can easily outperform complex 2D architectures on smaller datasets. Xie et al. (2018) investigate I3D architecture to find out how to reduce its space and time complexity. They show that using separable 3D convolutions (S3D), network parameters could be reduced significantly while achieving higher accuracy than I3D.

Compared to all these techniques, we propose to combine 2D and 3D ConvNets to build our heterogeneous network. This leverages to extract complementary features from videos that could accurately detect flu-like symptoms in an efficient manner. In order to further boost the recognition accuracy, we propose to apply GWO to perform feature pruning. To summarize, this paper has three main contributions: First, we propose Flu-Net which is a heterogeneous network based on ResNeSt50 and I3D architectures. Second, we investigate the application of GWO and other evolutionary algorithms to select the best features that enhance the performance of the proposed system. And finally, our Flu-Net obtains state-of-the-art results on BIISC dataset along with flu-specific activities extracted from NTU-RGBD dataset (Shahroudy et al. 2016).

3 Proposed method

Although ConvNets are a very powerful tool for any AI-based task, however, deciding the number and types of layers in a ConvNet architecture can be quite tricky and time-consuming. To overcome this issue, the most common solution prescribed in various computer vision literature is to pick a pre-trained ConvNet (like AlexNet (Krizhevsky et al. 2012), ResNet (He et al. 2016), MobileNet (Howard et al. 2017), C3D (Tran et al. 2015), etc.), and fine-tune it according to

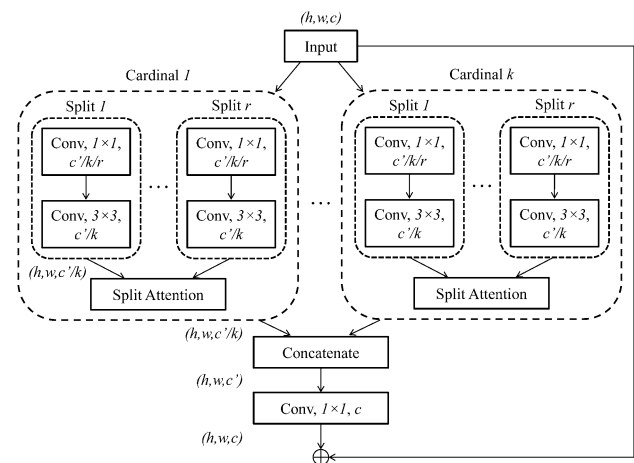


Fig. 2 A ResNeSt block has two hyperparameters: Cardinality (k) and radix (r). Each input feature is divided into k groups with r splits within each group. Attention weights are first calculated across all groups which are then multiplied by each feature map to generate the output representation

the task at hand. Following this guideline, we now explain the design of our two-stream heterogeneous network.

3.1 2D ConvNet stream based on ResNeSt50 model

In 2012, (Krizhevsky et al. 2012) first proposed a deep ConvNet model called AlexNet, and won the ImageNet challenge (Karpathy et al. 2014). Later, deeper models like VGG-16 (Simonyan and Zisserman 2014b) and Inception (Szegedy et al. 2015) are proposed to further improve the recognition accuracy. However, (He et al. 2016) show that with increasing network depth, accuracy gets saturated, and then degrades rapidly. This is because error gradients are not able to propagate back, and the network starts overfitting. To overcome this, the authors propose to add skip connection between the layers to provide an alternate pathway for data and gradients to flow, thus simplifying the process of training deeper networks. As a result, the layers in ResNet architecture are explicitly reformulated to learn residual functions as:

$$H(x) = F(x) + x \quad (1)$$

where $F(x)$ denotes the output (also called as feature maps) of one or more convolution and pooling layers.

However, the main drawback of ResNet architecture is its inability to exploit cross-channel information. To overcome this limitation, (Zhang et al. 2020) propose to incorporate the attention mechanism into the ResNet model to come up with an improved architecture called *Split-Attention Network* (ResNeSt). As shown in Fig. 2, a ResNeSt block is based on the idea of cardinality from ResNeXt model (Xie et al.

Fig. 3 Samples of video frames (top row) and corresponding RGBMI (bottom row) for six action classes in BIISC dataset (Thi et al. 2014)

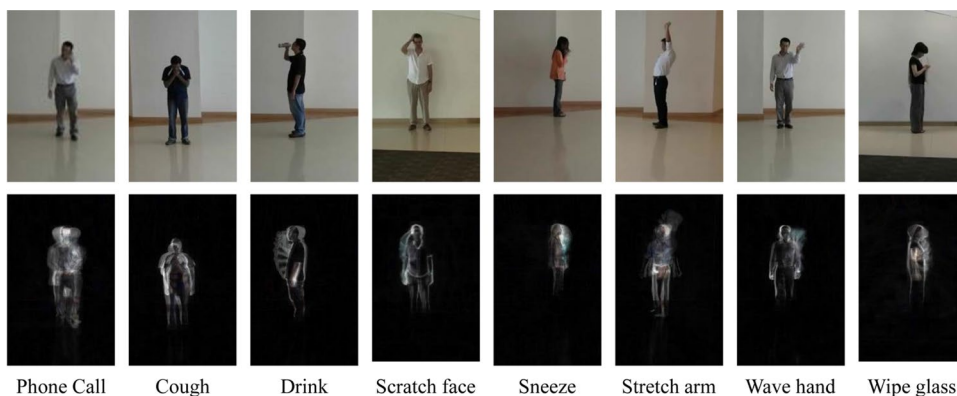
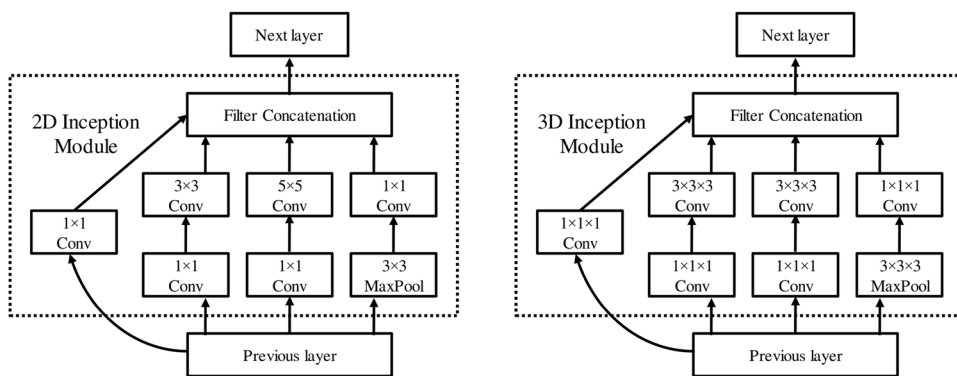


Fig. 4 The 2D Inception module from Inception-V1 architecture (Szegedy et al. 2015) (left), and inflated 3D Inception module from I3D architecture (Carreira and Zisserman 2017) (right). 3D convolutions have the advantage of directly extraction features from spatial and temporal dimensions simultaneously



2017) and attention mechanism from Squeeze and Excitation Network (SENet) (Hu et al. 2018). The term *cardinality* (k) defines the number of bottleneck blocks which breaks channel information into smaller groups, thereby increasing the recognition accuracy of the model. The SENet then *squeezes* each channel into a single numeric value by applying Global Average Pooling. Finally, *excitation* is performed by adding non-linearity using ReLU and smooth gating using sigmoid functions.

In our proposed framework, we use ResNeSt50 model and add two FC layers of 256 neurons with 60% dropout before the Softmax layer. Further, since ResNeSt50 accepts an RGB image (typically of size 224×224) as input, we first convert each video into a single motion template called RGB motion image (RGBMI). As described in (Imran and Raman 2019), an RGBMI is computed by stacking absolute frame difference over the entire video sequence as:

$$RGBMI = \sum_{k=2}^N k * |frame^k - frame^{k-1}|, \tag{2}$$

where N denotes the number of frames.

Some samples of RGBMI are shown in Fig. 3. We can observe that frame difference operation removes most of the irrelevant background information while stacking all the frame differences together helps to summarize the motion

pattern into a single image. Each RGBMI is then resized to 224×224 , and used to train ResNeSt50 stream. Once the network is sufficiently fine-tuned, $256 - D$ feature vectors for all the RGBMIs are extracted from the FC layer.

3.2 3D ConvNet stream based on I3D model

(Tran et al. 2015) first proposed a 3D ConvNet called C3D, which could perform convolution and pooling operations in temporal domain. However, C3D contains a huge number of parameters, making it extremely difficult to fine-tune on small datasets. Recently, (Carreira and Zisserman 2017) proposed an Inflated 3D ConvNet (I3D) based on Inception-V1 (Szegedy et al. 2015) as the backbone network. Specifically, all the $N \times N$ 2D filters (pre-trained on ImageNet dataset) present in Inception-V1 are repeated N times in time dimension to create $N \times N \times N$ 3D filters in I3D model (Fig. 4). Then the resulting network is trained on approximately 16K clips belonging to 400 classes of Kinetics dataset. This provides a strong initialization point to I3D model, making it possible to fine-tune even on small datasets.

In our proposed I3D stream, we add two FC layers with 512 neurons and 70% dropout before the Softmax layer as shown in Fig. 1. For input, we first perform frame difference operation and resize the resulting images to 112×112 to reduce computational complexity. 64 RGB frame

differences are then randomly selected from each video sequence to fine-tune I3D stream. Finally, 512 – D feature vector corresponding to each sample is extracted from the last FC layer.

3.3 Fusion of two-streams

Let $F^{2D} = \{f_1^{2D}, f_2^{2D}, \dots, f_n^{2D}\}$ and $F^{3D} = \{f_1^{3D}, f_2^{3D}, \dots, f_n^{3D}\}$ denotes the set of feature vectors extracted from our 2D and 3D ConvNet streams corresponding to n video samples, respectively. As stated previously, $f_i^{2D} \in \mathbb{R}^{256}$ and $f_i^{3D} \in \mathbb{R}^{512}$. Now there could be different fusion strategies (Feichtenhofer et al. 2016) to predict the actual class label:

- 1. Softmax Score Fusion** The Softmax scores of both streams can be combined by using Sum rule (Equ. 3) or Product rule (Equ. 4).

$$c = \operatorname{argmax}_j \sum_{i=1}^n P(\hat{y}_j | \mathbf{f}_i) \tag{3}$$

$$c = \operatorname{argmax}_j \prod_{i=1}^n P(\hat{y}_j | \mathbf{f}_i) \tag{4}$$

where k is the number of streams (in our case $k = 2$), and c is the number of classes. $P(\hat{y}_c | \mathbf{f}_i)$ denotes the probability of predicting class \hat{y}_c for a given feature vector \mathbf{f}_i .

- 2. Feature Selection using GWO** The feature vectors extracted from both the streams can be stacked together and then passed through a classifier like linear SVM or MLP. However, simply concatenating the feature vectors not only increases the computational complexity but also results in lower accuracy due to the addition of irrelevant features. To overcome these limitations, we propose to perform feature selection using GWO algorithm developed by Mirjalili et al. (2014). GWO is a population-based meta-heuristics algorithm that simulates the leadership hierarchy and hunting mechanism of grey wolves. The social hierarchy of grey wolves is divided into four levels: alpha, beta, delta, and omega. The alpha wolf is considered the dominant wolf in a pack and all his/her orders should be followed by pack members. The second level is beta which are discipliner for the pack and also subordinate and helps the alpha in decision making. The beta wolf can be either male or female and is considered the best candidate to be the alpha when the alpha passes away or become very old. The third level is called delta. The delta wolves have to submit to the alpha and beta but they dominate the omega. The lowest level is omega who have to obey all their superiors in the hierarchy.

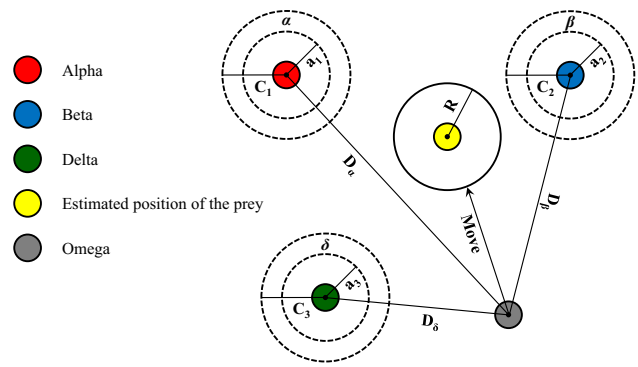


Fig. 5 Position updating of alpha, beta, delta and omega wolves during hunting operation in GWO algorithm

During hunting, all the wolves try to surround the prey as shown in Fig. 5. Mathematically, this behavior can be represented as

$$\vec{D} = \left| \vec{C} \vec{V}_p(t) - \vec{V}(t) \right|, \tag{5}$$

$$\vec{V}(t + 1) = \vec{V}_p(t) - \vec{A} \cdot \vec{D}, \tag{6}$$

where t is the current iteration, V_p is the position vector of the prey, V indicates the position vector of a grey wolf, and A and C are coefficient vectors which are calculated as

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}, \tag{7}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{8}$$

where components of a are linearly decreased from 2 to 0 over the course of iterations and r_1, r_2 are random vectors in $[0,1]$. During each iteration, the position of omega wolves is updated based on the positions of the best three wolves (alpha, beta, delta) in the current generation of the population. The following equations are used:

$$\vec{D}_\alpha = \left| \vec{C}_1 \vec{V}_\alpha - \vec{V} \right|, \quad \vec{D}_\beta = \left| \vec{C}_1 \vec{V}_\beta - \vec{V} \right|, \quad \vec{D}_\delta = \left| \vec{C}_1 \vec{V}_\delta - \vec{V} \right| \tag{9}$$

$$\vec{V}_1 = \vec{V}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha, \quad \vec{V}_2 = \vec{V}_\beta - \vec{A}_2 \cdot \vec{D}_\beta, \quad \vec{V}_3 = \vec{V}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \tag{10}$$

$$\vec{V}(t + 1) = \frac{\vec{V}_1 + \vec{V}_2 + \vec{V}_3}{3} \tag{11}$$

The entire evolutionary process is shown in Fig. 6, which is optimized using a two-stage feature selection (2SFS)

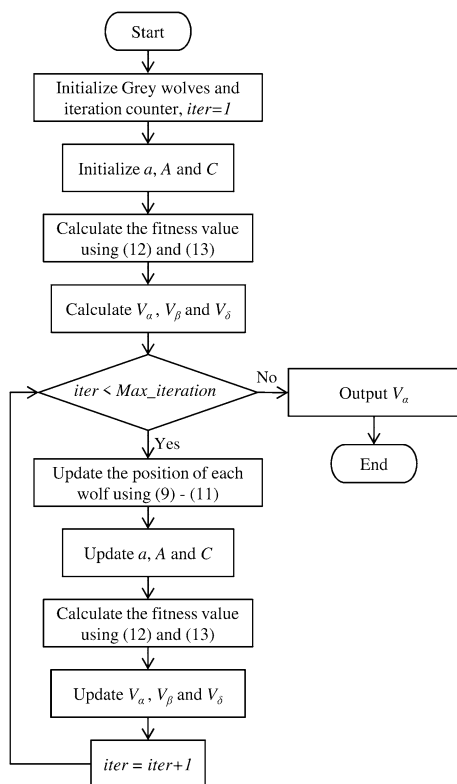


Fig. 6 Flowchart of GWO algorithm

approach proposed by Xue et al. (2012). In stage 1, the fitness function (Equ. 12) is used to minimize the classification error rate obtained by the selected feature subset during the evolutionary training process.

$$Fitness1 = Error\ Rate = \frac{FP + FN}{TP + TN + FP + FN} \tag{12}$$

where TP, TN, FP and FN denote the true positives, true negatives, false positives, and false negatives, respectively. Stage 2 begins with the solution achieved in stage 1 and tries to minimize the number of features while maximizing the classification performance using the fitness function shown in Eq. 13.

$$Fitness2 = \alpha * \frac{\#Selected\ Features}{\#All\ Features} + (1 - \alpha) * \frac{Error\ Rate}{ER} \tag{13}$$

where α is a constant and $\alpha \in [0, 1]$. $\#SelectedFeatures$ denotes the number of features selected and $\#AllFeatures$ represents the total number of available features. $ErrorRate$ is the classification error obtained on $SelectedFeatures$, while ER denotes the classification error corresponding to the $AllFeatures$ on the training set. α decides the tradeoff between classifier performance (classification accuracy in our case) and the number of selected features with respect to the number of all features.

4 Experiments

In this section, we describe the dataset details, network training, and experimental results. We also analyze the results obtained to get the further insight into our proposed framework.

4.1 Dataset

We evaluate our method on BIISC video action dataset (Thi et al. 2014). As shown in Fig. 3, this dataset consists of eight action types: ‘answer phone call’, ‘cough’, ‘drink’, ‘scratch face’, ‘sneeze’, ‘stretch arm’, ‘wave hand’ and ‘wipe glasses’. For each class, there are 120 videos, performed by 20 human subjects (12 males and 8 females) in the age group between 20 to 50 years. Each subject perform each action six times under 3 view-points (front, left and right) in 2 different positions (standing and walking). All the videos are also horizontally flipped, thus producing a total of $20 \times 8 \times 3 \times 2 \times 2 = 1920$ videos. For evaluation, we follow the protocol set by (Thi et al. 2014), and use subjects# 2, 3, 4, 5 and 6 for testing, and the remaining subjects for training.

Since BIISC is the only dataset available for flu detection, we explored other publicly available datasets and found that NTU RGB+D dataset (Shahroudy et al. 2016) contains few actions (out of total 60 actions) that could be useful in evaluating our proposed technique. As shown in Fig. 7, we select 10 actions classes: ‘phone call’, ‘sneeze/cough’, ‘drink’, ‘wipe face’, ‘brush teeth’, ‘hand wave’, ‘rub hands’, ‘clapping’, ‘nausea/vomiting’, and ‘nod head’. We extract 204 videos per class and called it NTU-Flu dataset. Out of the total $204 \times 10 = 2040$ videos, 75% (1530 videos) are randomly selected for training, and the remaining 25% (510 videos) are used for testing. The rest of the details of both datasets are shown in Table 1.

4.2 Network training

We perform experiments on a PC with Intel Corei5(TM) CPU @ 3GHz, 16GB RAM and NVIDIA GTX 1060 GPU. Frame difference operation is performed using Matlab R2018a, while Keras library is used to implement deep neural networks. We use Adam optimizer to fine-tune both the streams with initial learning rate set to 0.0001 and 0.0005 for ResNeSt50 and I3D stream, respectively. The batch size is kept as 16. The learning rate is halved when the training error plateaus.

Fig. 7 Sample frames of 10 classes from NTU-Flu dataset



Table 1 Details of BIISC and NTU-Flu dataset

	BIISC	NTU-Flu
Resolution	290×480	1920×1080
Min frames	30	37
Max frames	180	220
Avg frames	67.7	83.8

Table 2 Comparison of results on 2D ConvNet stream using different state-of-the-art 2D ConvNets as backbone architectures

Model	BIISC	NTU-Flu
MobileNetV3 (Howard et al. 2019)	50.8	68.2
EfficientNetV2-S (Tan and Le 2021)	53.3	71.6
ResNet50 (He et al. 2016)	55.0	72.8
ResNeSt50 (Zhang et al. 2020)	57.5	75.5

Bold represents the highest value

4.3 Results

Results on 2D ConvNet stream:

Table 2 presents the results on 2D ConvNet stream. We utilize four different backbone networks during our

Table 3 Comparison of results on 3D ConvNet stream using different clip size

Clip size	BIISC	NTU-Flu
32-frames	55.8	79.8
48-frames	60.4	81.2
64-frames	64.6	82.6

experiments: MobileNetV3 (Howard et al. 2019), EfficientNetV2-S (Tan and Le 2021), ResNet50 (He et al. 2016) and ResNeSt50 (Zhang et al. 2020). It is clear that ResNeSt50 achieves the best results on both datasets due to its ability to focus on relevant areas using the split-attention mechanism. This can also be visualized from the attention score maps shown in Fig. 8, Therefore, we utilize ResNeSt50 to implement our 2D ConvNet stream.

Results on 3D ConvNet stream:

We test our 3D ConvNet streams at three different settings: 32-frames, 48-frames and 64-frames clips. In each clip, the frames are extracted randomly, and the results obtained are presented in Table 3. We can observe that the best results are obtained for 64-frames clip size. This is because the average number of frames in both datasets is greater than 64, thus, extracting any fewer frames results in a loss of accuracy. We do not increase the clip size beyond 64 due to the increase in computational complexity.

Results using Softmax score fusion:

Fig. 8 Attention score maps obtained on NTU-Flu dataset using ResNeSt50 model. Our trained model accurately focuses on the important area while giving less attention to the surroundings. For instance, the heat-maps generated for ‘Sneeze/Cough’ and ‘Hand wave’ actions clearly shows that our model gives more emphasis (red) to the face and hands region, respectively

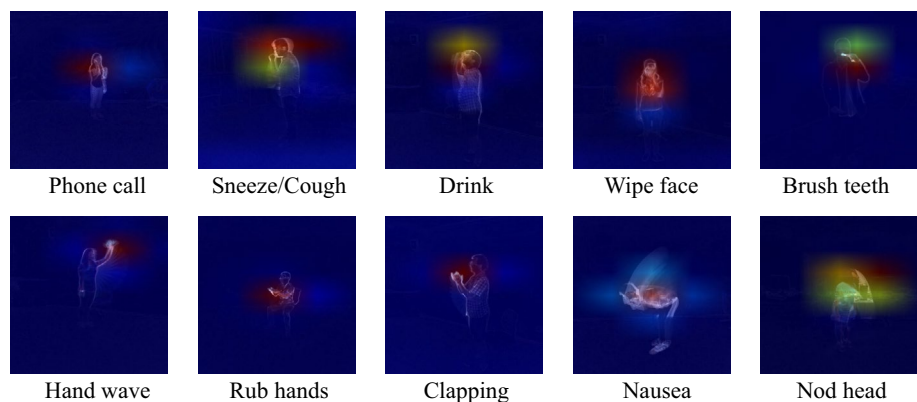


Table 4 Comparison of results obtained using score fusion

Stream	BIISC	NTU-Flu
ResNest50	57.5	75.5
I3D	64.6	82.6
Sum fusion	64.6	82.2
Product fusion	68.8	84.5

Table 4 presents the results obtained by fusing softmax scores of 2D and 3D ConvNet streams. We find that product fusion achieves higher results than score fusion method on both datasets. In fact, the sum fusion degrades the performance of NTU-Flu dataset. Thus, product fusion proves to be more effective for fusing Softmax scores in applications based on human activity recognition.

Results using GWO-based feature selection:

Next, we present the results of the GWO-based feature selection technique. We vary the population size between 5 to 50 (in steps of 5) while keeping the max number of iterations to 10. We could not find any improvement in the results if maximum iterations are increased beyond 10. The features selected using 2SFS defined in equations 12 and 13 are finally classified using MLP. The results obtained for BIISC and NTU-Flu datasets are presented in Table 5 and Table 6, respectively. For BIISC dataset, the best accuracy obtained is 70% when 161 features are selected using a population size of 20. This result is 1.2% higher than product fusion. Similarly, an improvement of about 1.6% is noticed for NTU-Flu dataset when 157 features are selected at a population size of 25, resulting in an overall accuracy of 86.07%.

Comparison with other optimization algorithms:

In order to verify the efficacy of the GWO algorithm, we compare our results with five other optimization algorithms (OAs) that could be used for feature selection:

1. Bat Optimization Algorithm (BAT) (Yang 2010)
2. Bees Optimization Algorithm (BEE) (Pham et al. 2006)
3. Cuckoo Search Algorithm (CSA) (Yang and Deb 2009)

4. Moth-Flame Optimization (MFO) (Mirjalili 2015)
5. Particle Swarm Optimization (PSO) (Kennedy and Eberhart 1995)

Again, all the experiments are performed by varying the population size between 5 to 50 while fixing the number of iterations to 10. Classification is performed using MLP and the best results thus obtained are reported in Table 7. We can observe that without any feature selection (FS) algorithm (row 1 of Table 7) yields the worst performance. The GWO algorithm (last row of Table 7) achieves the highest accuracies on both datasets while selecting the minimum number of features. Other OAs are not as effective in reducing the dimensionality of features extracted from both streams. Only MFA is able to achieve the same accuracy as GWO on NTU-Flu dataset, but the number of features selected by MFA (314) is twice as more than selected by GWO (157).

Comparison with previous results:

Table 8 shows the comparison of our results against the four baseline results reported in (Thi et al. 2014): Cuboid + BoW + χ^2 , HOGHOF + BoW + χ^2 , Cuboid + AMK II, and HOGHOF + AMK II. AMK I and II are two new types of Action Machine Kernels proposed by Thi et al. (Thi et al. 2014) to integrate space-time layout and Bag-of-Words-based local features. Both of them fall under the category of handcrafted features. Compared to them, our ResNeSt50

Table 7 Comparison of accuracy using different optimization algorithms (OAs) for feature selection (FS)

OA	BIISC		NTU-Flu	
	Acc (%)	FS	Acc (%)	FS
Without FS	66.66	768	85.88	768
PSO (Kennedy and Eberhart 1995)	67.50	282	85.68	300
MFA(Mirjalili 2015)	67.91	312	86.07	314
BAT (Yang 2010)	68.33	272	85.88	353
BEE (Pham et al. 2006)	68.75	312	85.68	314
CSA(Yang and Deb 2009)	68.75	301	85.88	268
GWO (Mirjalili et al. 2014)	70.00	161	86.07	157

Bold represents the highest value

Table 5 Comparison of results obtained on BIISC dataset using GWO algorithm

Population size	5	10	15	20	25	30	35	40	45	50
#Selected Features	155	172	152	161	165	149	157	136	138	132
Accuracy (%)	65.41	68.33	65.00	70.00	65.83	66.66	67.91	64.58	67.50	67.91

Bold represents the highest value

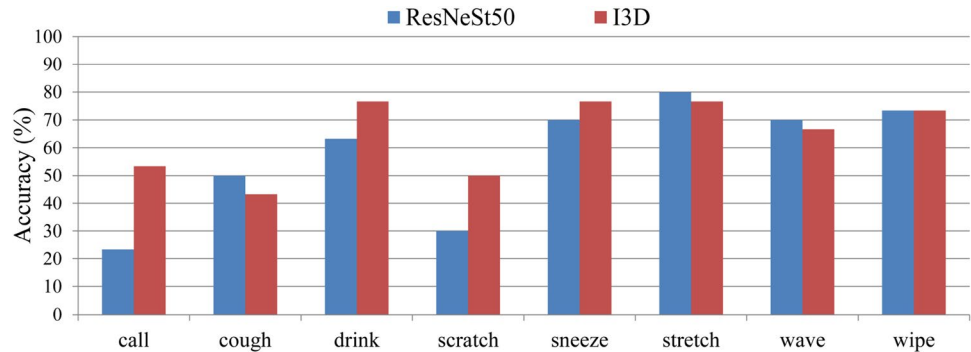
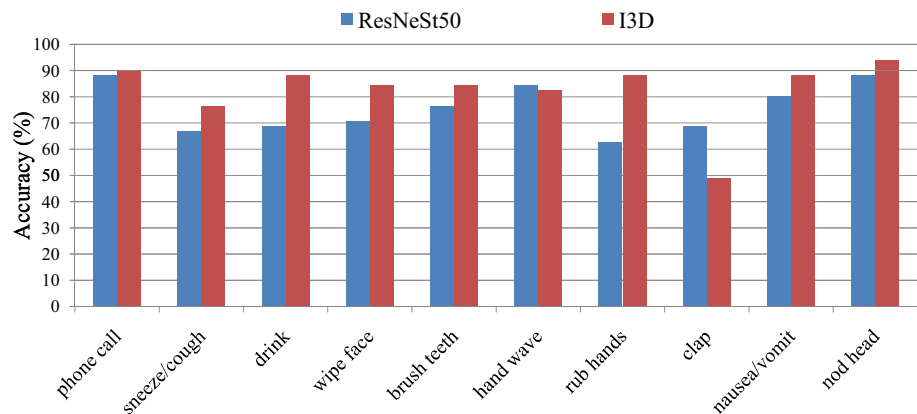
Table 6 Comparison of results obtained on NTU-Flu dataset using GWO algorithm

Population size	5	10	15	20	25	30	35	40	45	50
#Selected features	185	185	143	161	157	137	143	131	134	138
Accuracy (%)	85.49	85.49	85.29	84.90	86.07	82.94	85.29	84.70	85.49	84.70

Bold represents the highest value

Table 8 Accuracy comparison with existing methods on BIISC dataset

Method	Accuracy (%)
Cuboid + BoW + χ^2 (Thi et al. 2014)	53.5
HOGHOF + BoW + χ^2 (Thi et al. 2014)	47.0
Cuboid + AMK II (Thi et al. 2014)	59.0
HOGHOF + AMK II (Thi et al. 2014)	62.1
Ours	
ResNeSt50 stream	57.5
I3D stream	64.6
Both streams (Product fusion)	68.8
Both streams (GWO fusion)	70.0

Fig. 9 Class-wise comparison of accuracy between ResNeSt50 vs I3D streams on BIISC dataset. I3D stream performs better than ResNeSt50 streams on most of the actions like ‘call’, ‘drink’ and ‘sneeze’**Fig. 10** Class-wise comparison of accuracy between ResNeSt50 vs I3D streams on NTU-Flu dataset. I3D stream performs better than ResNeSt50 streams on most of the actions like ‘sneeze/cough’, ‘drink’, ‘wipe face’ and ‘rub hands’

stream gives only 57.5% accuracy, while I3D performs much better to recognize 64.6% of actions correctly. On combining both the streams, we get 68.8% and 70% accuracy using Product and GWO-based fusion, respectively.

4.4 Analysis

To get further insight into the performance of individual streams, we plot the comparison between ResNeSt50 and I3D models in Figs. 9 and 10 on BIISC and NTU-Flu datasets, respectively. For BIISC dataset, I3D performs significantly better on ‘call’, ‘drink’, and ‘scratch’ classes, while ResNeSt50 is more suitable for ‘cough’ class. The remaining

classes are almost equally recognized by both the streams. A similar observation is made for NTU-Flu dataset, where I3D achieves superior accuracy than ResNeSt50 on all classes except ‘hand wave’ and ‘clapping’ actions. This is because I3D is pre-trained on both ImageNet and Kinetics datasets, and could perform convolutions in both spatial and temporal dimensions.

Next, we plot the class-wise accuracy obtained using Product and GWO-based fusion techniques on BIISC and NTU-Flu dataset in Figs. 11 and 12, respectively. Except for ‘wave’ class, the GWO fusion achieves better results on all classes of BIISC dataset. Similarly, GWO fusion is better for all except ‘rub hands’ and ‘nausea/

Fig. 11 Class-wise comparison of accuracy between Product and GWO fusion on BIISC dataset

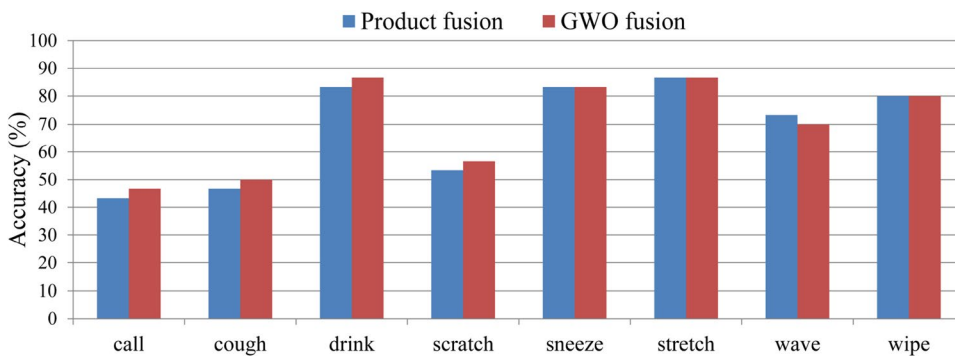
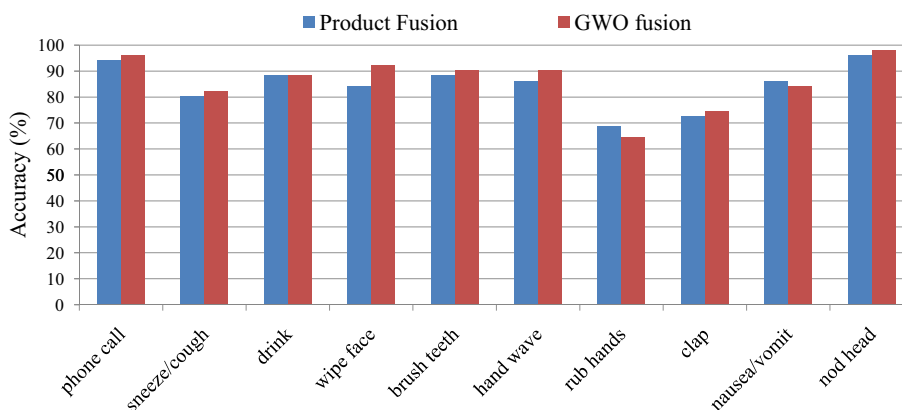


Fig. 12 Class-wise comparison of accuracy between Product and GWO fusion on NTU-Flu dataset



vomiting’ classes on NTU-Flu dataset. These results verify the utilization of GWO-based optimization technique of feature selection when combining features extracted from different streams. This method not only reduces the dimension of the input feature vector but also increases the recognition accuracies significantly.

Finally, we analyze the normalized confusion matrix obtained by GWO fusion in Figs. 13 and 14 corresponding to two datasets. We see observe that ‘call’ action has the least accuracy of 46.67% in BIISC dataset. This is because, while answering a phone call, a person moves his/her hand towards the head, which has a very close resemblance to other actions like ‘cough’ and ‘scratch’ classes. Similarly, ‘cough’ and ‘sneeze’ classes also show high confusion between them due to their identical motion patterns. However, most of the classes show high accuracies in NTU-Flu dataset. This is because NTU-Flu contains high resolution (1920×1080) videos, resulting in much cleaner features extracted by both the streams. Only ‘rub hands’ and ‘clapping’ actions show some confusion due to the involvement of hands in both classes.

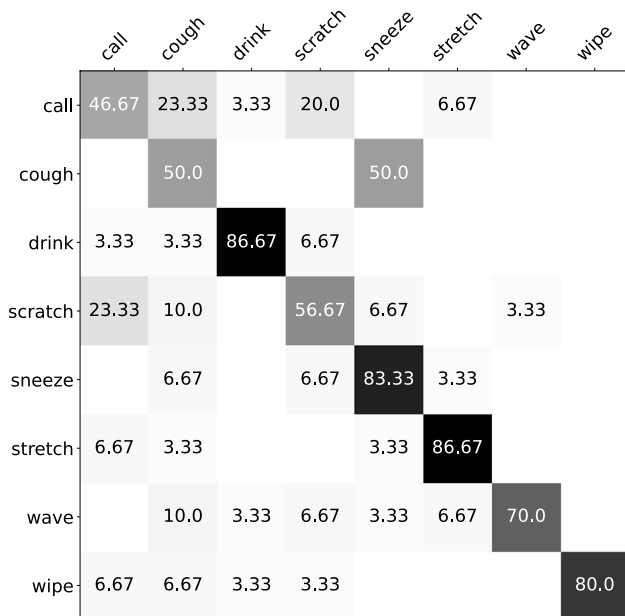


Fig. 13 Normalized confusion matrix of GWO fusion on BIISC dataset. Highest confusion between ‘cough’ and ‘sneeze’ actions occurs due to their similar spatial and temporal pattern

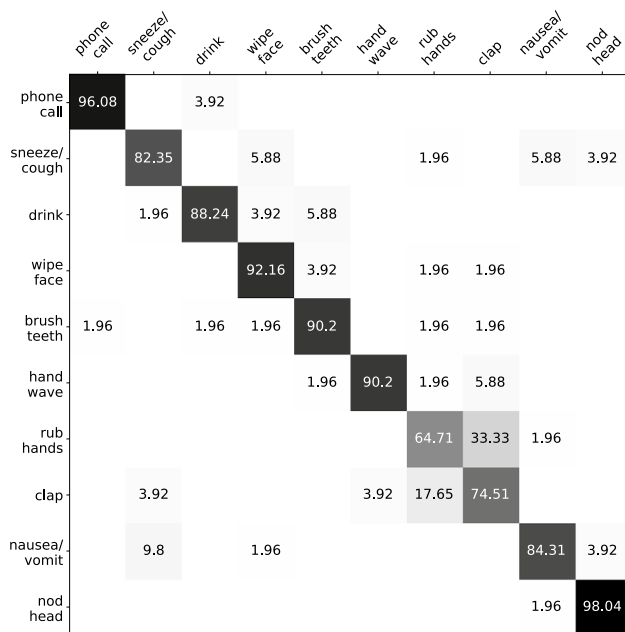


Fig. 14 Normalized confusion matrix of GWO fusion on NTU-Flu dataset. Highest confusion between ‘rub hands’ and ‘clap’ actions occurs due to their similar spatial and temporal pattern

5 Conclusion

In this paper, a deep learning-based recognition method has been used to identify Flu-like symptoms in videos. We built a two-stream heterogeneous network to extract complementary features from RGB frame differences using 2D and 3D ConvNets. We performed multiple experiments to decide the best models for our framework, i.e., ResNeSt50 as 2D ConvNet and I3D as 3D ConvNet. The split-attention block in ResNeSt50 helps to focus on important regions within the input RGBMIs, while a larger (64-frame) clip size helps to capture the entire spatio-temporal information in the I3D stream. To further boost the final performance, we use GWO-based feature selection algorithm to select the most relevant feature set, thereby reducing computational complexity and improving classification accuracy. Finally, we perform an in-depth analysis to know the strengths and weaknesses of our proposed framework. In the future, we would consider collecting larger datasets and adding other modalities (like depth or infrared videos) to improve the robustness of our technique. Inertial sensors like Accelerometers and Gyroscopes data could also be incorporated in future experiments so as to overcome the limitations of occlusions and further improving the results by adding an extra modality into our existing framework.

Funding No funding source available

Data availability No associated data

Declarations

Conflict of interest The authors declare that they have no conflict of interest

References

- Carreira J, Zisserman A (2017) Quo vadis, action recognition? a new model and the kinetics dataset. In: IEEE Conference on Computer Vision and Pattern Recognition, pages 6299–6308
- Deepak S, Ameer PM (2021) Automated categorization of brain tumor from mri using cnn features and svm. *J Ambient Intell Human Comput* 12(8):8357–8369
- Diba A, Sharma V, Van Gool L (2017) Deep temporal linear encoding networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pages 2329–2338
- Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2015) Long-term recurrent convolutional networks for visual recognition and description. In: IEEE Conference on Computer Vision and Pattern Recognition, pages 2625–2634
- Feichtenhofer C, Fan H, Malik J, He K (2019) Slowfast networks for video recognition. In: IEEE International Conference on Computer Vision, pages 6202–6211
- Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pages 1933–1941
- Girdhar R, Ramanan D, Gupta A, Sivic J, Russell B (2017) Actionvlad: Learning spatio-temporal aggregation for action classification. In: IEEE Conference on Computer Vision and Pattern Recognition, pages 971–980
- Hara K, Kataoka H, Satoh Y (2018) Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In: IEEE Conference on Computer Vision and Pattern Recognition, pages 6546–6555
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778
- Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- Howard A, Sandler M, Chu G, Chen L-C, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V et al (2019) Searching for mobilenetv3. In: IEEE International Conference on Computer Vision, pages 1314–1324
- Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pages 7132–7141
- Imran J, Raman B (2019) Deep motion templates and extreme learning machine for sign language recognition. *Vis Comput*, pages 1–14
- Ji S, Wei X, Yang M, Kai Yu (2012) 3d convolutional neural networks for human action recognition. *IEEE Trans Pattern Anal Mach Intell* 35(1):221–231
- Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-FL (2014) Large-scale video classification with convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, pages 1725–1732
- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: International Conference on Neural Networks 4:1942–1948

- Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pages 1097–1105
- Kujani T, Dhilip Kumar V (2021) Head movements for behavior recognition from real time video based on deep learning convnet transfer learning. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–15
- Lin J, Gan S, Han S (2019) Tsm: Temporal shift module for efficient video understanding. In: *IEEE International Conference on Computer Vision*, pages 7083–7093
- Li K, Wang Y, Zhang J, Gao P, Song G, Liu Y, Li H, Qiao Y (2022) Uniformer: Unifying convolution and self-attention for visual recognition. arXiv preprint [arXiv:2201.09450](https://arxiv.org/abs/2201.09450)
- Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. *Knowl-Based Syst* 89:228–249
- Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
- Mohammed AAQ, Lv J, Islam M, Sang Y et al (2022) Multi-model ensemble gesture recognition network for high-accuracy dynamic hand gesture recognition. *J Ambient Intell Human Comput*, pages 1–14
- Natarajan B, Elakkiya R, Prasad M (2022) Sentence2signgesture: a hybrid neural machine translation network for sign language video generation. *J Ambient Intell Human Comput*, pages 1–15
- Pham DT, Ghanbarzadeh A, Koç E, Otri S, Rahim S, Zaidi M (2006) The bees algorithm—a novel tool for complex optimisation problems. In: *Intelligent Production Machines and Systems*, pages 454–459. Elsevier
- Qiu Z, Yao T, Mei T (2017) Learning spatio-temporal representation with pseudo-3d residual networks. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5533–5541
- Ryoo MS, Piergiovanni AJ, Tan M, Angelova A (2019) Assemblenet: Searching for multi-stream neural connectivity in video architectures. In: *International Conference on Learning Representations*
- Shahroudy A, Liu J, Ng T-T, Wang G (2016) Ntu rgb+d: A large scale dataset for 3d human activity analysis. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1010–1019
- Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. *Adv Neural Inform Process Syst* 2:568–576
- Simonyan K, Zisserman A (2014b) Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)
- Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9
- Tan M, Le Q (2021) Efficientnetv2: Smaller models and faster training. In: *International Conference on Machine Learning*, pages 10096–10106. PMLR
- Thi TH, Wang L, Ye N, Zhang J, Maurer-Stroh S, Cheng L (2014) Recognizing flu-like symptoms from videos. *BMC Bioinform* 15(1):300
- Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. In: *IEEE International Conference on Computer Vision*, pages 4489–4497
- Tran D, Wang H, Torresani L, Ray J, LeCun Y, Paluri M (2018) A closer look at spatiotemporal convolutions for action recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6450–6459
- Varol G, Laptev I, Schmid C (2017) Long-term temporal convolutions for action recognition. *IEEE Trans Pattern Anal Mach Intell* 40(6):1510–1517
- Wang C-Y, Bochkovskiy A, Mark LH-Y (2022a) Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint [arXiv:2207.02696](https://arxiv.org/abs/2207.02696),
- Wang L, Qiao Y, Tang X (2015) Action recognition with trajectory-pooled deep-convolutional descriptors. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4305–4314
- Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L (2016) Temporal segment networks: Towards good practices for deep action recognition. In: *European Conference on Computer Vision*, pages 20–36
- Wang H, Xu D, Dongliang H, Li F, Ji Z, Han J, Ding E (2022b) Boosting video-text retrieval with explicit high-level semantics. In: *ACM International Conference on Multimedia*, pages 4887–4898
- Wu C-Y, Zaheer M, Hu H, Manmatha R, Smola AJ, Krähenbühl P (2018) Philipp Compressed video action recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6026–6035
- Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500
- Xie S, Sun C, Huang J, Tu Z, Murphy K (2018) Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: *European Conference on Computer Vision*, pages 305–321
- Xue B, Zhang M, Browne WN (2012) New fitness functions in binary particle swarm optimisation for feature selection. In: *IEEE Congress on Evolutionary Computation*, pages 1–8
- Yang X-S, Deb S (2009) Cuckoo search via lévy flights. In: *IEEE World Congress on Nature & Biologically Inspired Computing*, pp 210–214
- Yang X-S (2010) A new metaheuristic bat-inspired algorithm. *Nature inspired cooperative strategies for optimization*. Springer, Berlin, pp 65–74
- Yue-Hei Ng J, Hausknecht M, Vijayanarasimhan S, Vinyals O, Monga R, Toderici G (2015) Beyond short snippets: Deep networks for video classification. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp 4694–4702
- Zhang Z, Tian Z, Zhou M (2018) Handsense: smart multimodal hand gesture recognition based on deep neural networks. *J Ambient Intell Human Comput*, pp 1–16
- Zhang H, Wu C, Zhang Z, Zhu Y, Lin H, Zhang Z, Sun Y, He T, Mueller J, Manmatha R, et al (2020) Resnest: Split-attention networks. arXiv preprint [arXiv:2004.08955](https://arxiv.org/abs/2004.08955)
- Zhou B, Andonian A, Oliva A, Torralba A (2018) Temporal relational reasoning in videos. In: *European Conference on Computer Vision*, pp 803–818
- Zhou Z, Yu L, Tian S, Xiao G (2022) Diagnosis of alzheimer’s disease using 2d dynamic magnetic resonance imaging. *J Ambient Intell Human Comput*, pp 1–11

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.