

Classification Performance of Violence Content by Deep Neural Network with Monarch Butterfly Optimization

Ashikin Ali¹, Norhalina Senan²
Faculty of Computer Science and
Information Technology, Universiti
Tun Hussein Onn Malaysia
Parit Raja, Malaysia

Iwan Tri Riyadi Yanto³
Faculty of Mathematics and Natural
Sciences, Ahmad Dahlan University
Yogyakarta, Indonesia

Saima Anwar Lashari⁴
College of Computing and Informatics
Saudi Electronic University
Dammam, Kingdom of Saudi Arabia

Abstract—Violence is self-sufficient, it is perplexing due to visibility of content dissimilarities among the positive instances that been displayed on media. Besides, the ever-increasing demand on internet, with various types of videos and genres, causes difficulty for a proper search of these videos to ensure the contents is humongous. It involves in aiding users to choose movies or web videos suitable for audience, in terms of classifying violence content. Nevertheless, this is a cumbersome job since the definition of violence is broad and subjective. Detecting such nuances from videos becomes technical without a human's supervision that can lead to conceptual problem. Generally, violence classification is performed based on text, audio, and visual features; to be precise, it is more relevant to use of audio and visual base. However, from this perspective, deep neural network is the current build-up in machine learning approach to solve classification problems. In this research, audio and visual features are learned by the deep neural network for more specific violence content classification. This study has explored the implementation of deep neural network with monarch butterfly optimization (DNNMBO) to effectively perform the classification of the violence content in web videos. Hence, the experiments are conducted using YouTube videos from VSD2014 dataset that are publicly available by Technicolor group. The results are compared with similar modified approaches such as DNNPSO and the original DNN. The outcome has shown 94% of violence classification rate by DNNMBO.

Keywords—Deep learning; monarch butterfly; violence video; classification

I. INTRODUCTION

Sensible content filtering of media is necessary, as the multimedia content is affecting the internet users significantly. Eventually, every user can provide and be exposed to sensitive media contents over the internet. One sensitive media content type is violence. Through unlimited hours of video being uploaded every day over the internet, the access is by considerable portion of internet users with limited control [1]. Subjective characterization of violence makes impenetrable the violence content displayed based on audio signals, like screams or gunshots. This situation may result in harmful situation to users either mentally or physically. It is evident that numerous web-filtering systems are highly available on internet. Yet, these approaches did not meet user needs to

accurately classify violence content that is exposed on internet or social media. To resolve this drawback, artificial neural network will be a feasible solution and has been an efficient method to classify violence content [2],[3],[4]. Currently, Deep neural network (DNN) technique is broadly used for video classification tasks [5],[6],[7],[8],[9],[10]. There are few drawbacks of DNN that it still faces issue regarding complex architecture, specifically on multiple hidden layers, high tendency to get trapped in local minima; moreover, it is also computationally exhaustive. In the long run to ensure DNN perform reliably, big or ample amount of data is required to avoid overfitting [11],[12]. To combat this deficiency, optimization techniques are constantly considered as a solution to this problem. In the relevant literature, these are communal, especially using several optimization algorithms, namely, particle swarm optimization (PSO), chicken swarm optimization (CSO), firefly optimization (FO), whale optimization algorithm (WOA) on deep learning algorithms [13]. Meanwhile, monarch butterfly optimization (MBO) was proposed by Gai-Ge Wang [33], is a new nature-based metaheuristic algorithm, it is carried out by studying the migration behavior of monarch butterflies. Researchers have conducted a comparative study of the performance of MBO with other metaheuristic algorithms, which has shown promising performances [14]. The inspiring simple and robust nature of the algorithm with the ability to deal with the stirring between exploration and exploitation has given strength to the choice. This has motivated this study to embed the MBO with DNN in order to improve the performance accuracy based on the accuracy, precision and recall. From the dataset perspective, researchers focus on audio features [15][16], audio-visual features [17],[18],[19], inter feature and inter-classes [20], multimodal features [21] and exploiting feature and class relationships [22]. Thus, in this study, both the audio-visual features and all the attributes of the features have been used to feed into the proposed classifier.

The paper organized as follows: Section II highlights research methodology adopted to perform this study. Section III proposes the DNN in integration with MBO. The results are reported and discussed in last section of the paper; duly concludes this study, as well as, provides potential research direction.

II. RESEARCH MOTIVATION

In line with the literature review, violence could define any circumstances that can cause harm physically or mentally to single or many users. For specific problem on audio signals, the related literature is very limited, and, mostly, it examines audio features and visual features (Krizhevsky, 2012; Dai *et al.*, 2015) separately. Complex models trained with insufficient data can be prone to overfitting (Caruana, 2001; Shin, 2016; Cortes, 2017). This study intended to examine both the audio and visual features that calculates 68, 830 data instances ensuring it to perform reliably to achieve the violence classification task.

From a computational perspective, as an advantage with Deep Neural Network (DNN) to its supremacy in terms of accuracy is trained with huge amount of data. However, primary drawbacks of using DNN have been identified from its substantial complexity, which require extensively long periods of training time. These are due to nature of its architecture which consist of trillions hidden layers and hidden nodes depends on the loaded input (Bou-Rabee *et al.*, 2017). Hence, it may cause the algorithm to be stuck at local minima or global minimum. Initially, this research decided to focus to overcome the local minima drawback. Optimization has always been a solution to this problem. However, the implementations are widely used for non-convex (Cortes, 2017), hyper-parameters (Albelwi, 2017), weights (Kietzmann, 2018). Thus, to mitigate the issue this study decided to improve DNN algorithm by optimizing the weights using Monarch Butterfly Optimization (MBO) Algorithm to further reduce the error to an acceptable minimum fitness to improve the classification accuracy. Whereby the initial weighting factor of MBO algorithm proposed by Gai Ge Wang (2015) is $\alpha = S_{max}/t^2$. This study has come to consent that modifying the weighting factor to $\alpha = S_{min}/t^2$ will enable to achieve the goal of minimize the errors and obtain satisfying accuracy.

III. RESEARCH METHODOLOGY

A. Research Framework

The five main phases of the proposed methodology are, namely, data acquisition, feature extraction, data pre-processing, classification and classified output, illustrated according to the experimental need. Fig. 1 is the research design of the study.

As a domain in this study, benchmark violence scene dataset (VSD2014) is acquired from Technicolor Group [23][24]. In the whole, the generalization subset of 86 videos have been used but reduced to a meaningful instance to ensure suitability with the available devices. The videos are downloaded from YouTube and are normalized to a frame rate of 25fps (Fig. 2). This dataset contains two classes: 1 represents violence and 0 represents non-violence, respectively.

In pre-processing stage, features were extracted consisting of 29 audio and 405 visual features by total of 434 instances as the input to the network. The data were intertwined with missing values, those missing values were imputed using the ReplaceMissingValues filters from WEKA. By steering this

process, it provides eloquent dataset for the learning algorithm, normalized and segregated to provide meaningful dataset for the network.

Classification phase evolves the modified DNNMBO. The optimization process during the training part of deep neural network is performed using MBO algorithm to optimize the weights in DNN to classify the violence content. Performance evaluation is based on the classification metrics like precision, recall and accuracy.

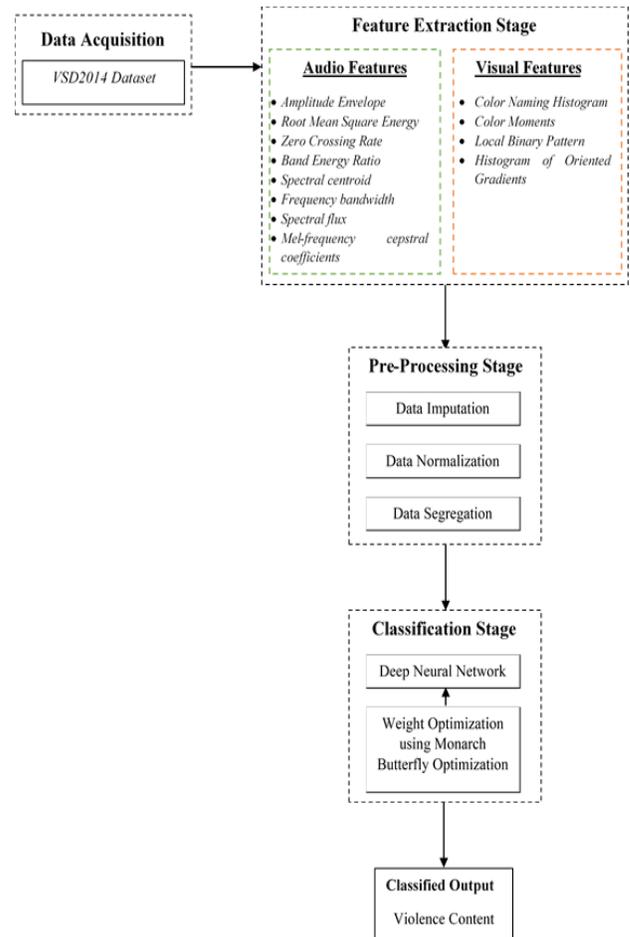


Fig. 1. Methodology of the Study.



Fig. 2. Example Images from the VSD2014 Dataset.

B. Deep Neural Network

The backpropagation is a supervised learning algorithm, acute part of layered feed-forward deep neural network, provided with the inputs and outputs that the network must compute, afterwards calculate the error [25]. Backpropagation algorithm is aware to reduce this error, until the DNN (Fig. 3) learns the training data. The training will begin with random weights till optimized, and the goal is to adjust them so that the error is minimal. The weighted sum of a neuron is written as Equation (1):

$$A_j(x, w) = \sum_{i=0}^n X_i W_{ji}, \quad (1)$$

where the sum of input X_i is multiplied by their respective weights W_{ji} . If the output function is the identity, then the neuron is called linear. The most used output function is sigmoid function [26], as formulated by Equation (2):

$$O_j(x, w) = \frac{1}{1+e^{-A(x,w)}} \quad (2)$$

The sigmoid function allows a smooth transition between the low and high output of the neuron. The output depends on the activation, which in turn depends on the values of the inputs and their respective weights. The target is to obtain a desired output based on the input given. Since the error is the difference between the actual and desired output, it depends on the weights and preferred to be adjusted to minimize. The error function for the output of each neuron defined as in Equation (3):

$$E_j(x, w, d) = (O_j(x, w) - d_j)^2 \quad (3)$$

The desired target will be greater when the difference is big and lesser, the output considered positive. The error of the network will simply be the sum of the errors of all the neurons in the output layer as in Equation (4):

$$E(x, w, d) = \sum_j (O_j(x, w) - d_j)^2 \quad (4)$$

where O_j is the output and d_j is the desired output of the experiment. Afterwards, the weights will be adjusted using the method of gradient descent, Equation (5):

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \quad (5)$$

The Equation (5) infer the adjustments of each weight (Δw_{ji}) will be the negative of a constant eta (η), while η is the learning rate. The size of the adjustment rely on η , and derivative of E in respect to w_{ji} discovered. Initially, the error is dependent on the output, which is the derivative of E in respect to O_j from Equation (3).

$$\frac{\partial E}{\partial O_j} = 2(O_j - d_j) \quad (6)$$

As of Equation (1) and Equation (2), will alter the output of the activation. That can be seen from Equation (6) and Equation (7):

$$\frac{\partial O_j}{\partial w_{ji}} = \frac{\partial O_j}{\partial A_j} \frac{\partial A_j}{\partial w_{ji}} = O_j(1 - O_j)x_i \quad (7)$$

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial O_j} \frac{\partial O_j}{\partial w_{ji}} = 2(O_j - d_j)O_j(1 - O_j)x_i \quad (8)$$

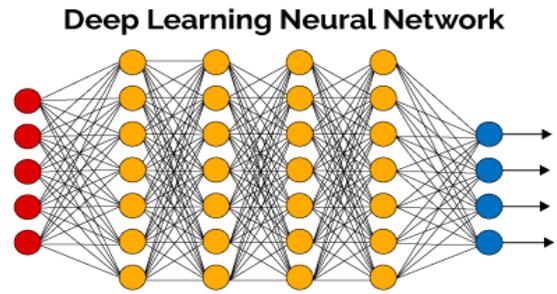


Fig. 3. Architecture of Deep Neural Network.

The adjustment to each weight will begin from Equation (5) and Equation (8):

$$\Delta w_{ji} = -2\eta(O_j - d_j)O_j(1 - O_j)x_i \quad (9)$$

Equation (9) can be used as it is for training DNN. These processes clearly visualized in the following architecture of DNN:

The fact that different deep learning networks in general from the common feed-forward multilayer networks that DNN have many hidden layers that are hard to train. More neurons than previous networks and complex way of connecting layers [27],[28]. With backpropagation inside, the errors are sent back through the network again and weights are adjusted to improve the model. This process is constant by adjusting the weights until the error rate cannot be reduced any more. During this process the layers learn the optimal features for the model which derives to learn the optimal weights.

C. Monarch Butterfly Optimization (MBO)

In MBO [29], the spots of the monarch butterflies are updated in two ways. Initially, the offspring are generated to update position by migration operator, which can be adjusted by the migration ratio. Tailed through tuning the positions for other butterflies by means of butterfly adjusting operator. To minimize fitness evaluations and keep the population consistent, the original population and the amount of newly generated butterflies endures equal to the original population which later will be divided into two depending on the fitness. The number of monarch butterflies in land₁ and land₂ are calculated as subpopulation1 (NP_1) = ceil ($p*NP$)/(NP_1) and subpopulation2 (NP_2) = $NP - NP_1$, respectively, where 'p' is the migration ratio. The positions of monarch butterfly individuals are updated using migration operator (MO) and butterfly adjusting operator (BAO). Next section explains the process of DNNMBO.

IV. DEEP NEURAL NETWORK WITH MONARCH BUTTERFLY OPTIMIZATION (DNNMBO)

A. Butterfly Adjusting and Migration Operator

Meanwhile, MBO is a latest metaheuristic algorithm, and undoubtedly, it has shown better performance on benchmark evaluation [30],[31],[35]. Hence, this has become a motivation for this study to improve the DNN with MBO. As for migration operator (MO), the offspring are generated to update the position in NP_1 by MO and adjust based on ratio of monarch butterflies in land₁. The position of each butterfly

from NP1 is based on individuals from NP1 and NP2. This initialization aids on finding the optimum weights to provide best error rate and performance accuracy of DNNMBO.

B. Proposed DNNMBO

The initialization of DNNMBO, selected monarch butterflies are considered as mb_1 and mb_2 from NP1 and NP2 respectively. Meanwhile, $r = rand * peri$, where $rand$ is random number between [0,1] and $peri$ is the migration period. Fig. 4 visualizes the implementation process of DNNMBO.

Each one mb_i from the population of NP is represented as given as Equation (10):

$$mb_i = \{ w_i, b_i \} \tag{10}$$

where w_i denotes the weights of input and hidden layer, b_i denotes bias for hidden layer and output neuron. These weights and biases fulfil the criteria of butterfly positions in the network. The performance is based on the entire training data of size n . The MSE is calculated as in Equation (11). Hence the fitness function, FES , of MBO is the average MSE of the entire training set, which is formulated as:

$$\text{Minimise } FES = \sum_{j=1}^n \frac{E_j}{j} \tag{11}$$

To visualize the proposed DNNMBO, the Algorithm shows the process of DNN training by MBO.

Based on the above clarifications, it is believed that the proposed DNNMBO promises better outcome with best global optimum of 50 generations.

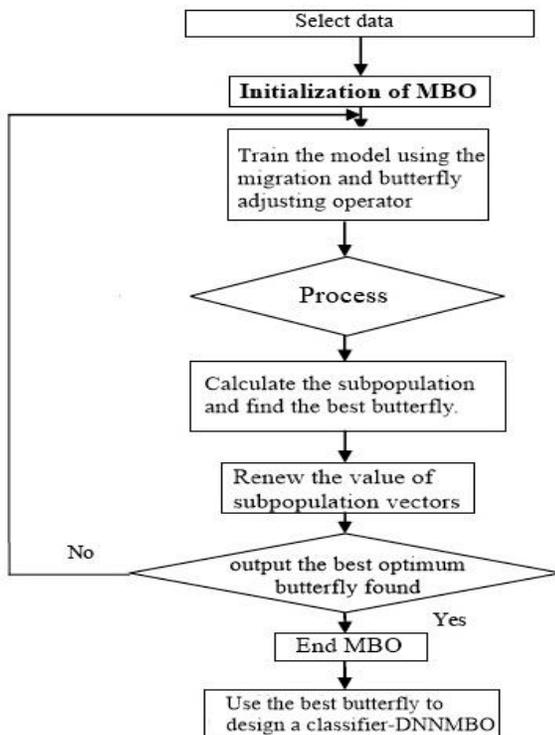


Fig. 4. Flowdiagram of Proposed DNNMBO.

Algorithm: Monarch Butterfly Optimization based training of Deep Neural Network

Step 1: Input X denotes n samples, O as output the maximum selected attributes.

Step 2: While weights of hidden and input layers updated.

Step 3: Set the population of monarch butterfly individuals NP which is the **weight factors of deep neural networks**, W_{ji} , **where the adjustmen is as in Equation 5 to Equation 8**, randomly in [-1,1], set MaxGen, NP1 in Land 1 and NP2 in Land 2, max step is the BAR, $peri$ and ratio p .

Step 4: for randomly generate numbers by uniform distribution, $mb < p$ from subpopulation 1 monarch butterfly randomly selected **else** select monarch butterfly in subpopulation 2 **end if**. Evolves in Migration Operator.

Step 5: for each monarch butterflies in subpopulation 2 **do** calculate the walk step, calculate the weighting factor $S_{max} = 1.0$ **for** randomly generate by uniform distribution, **if** $mb < p$ **then** best monarch butterfly is chosen, **else** randomly select monarch butterfly in subpopulation 2 **end if**.

Step 6: Combine two newly generated subpopulation into one whole population. **Evaluate** monarch butterfly according newly updated positions. The process that evolves weight adjustments in the network.

Step 7: **If** best solution is not found, butterfly individuals sorted according the fitness. **end If**

Step 8: Finally, gives the best **Output** of weight vector for deep neural network, for best solution.

V. RESULTS AND DISCUSSION

A. Experimental Design

The proposed model and similar comparison models are simulated using MATLAB R2016a on MacPro 6, macOS High Sierra operating system, the processor is 8 Core Intel Xeon E5 with the speed of 3 GHz and 32GB RAM.

B. Simulation Results

The experiment is conducted with 5-fold cross validation method [32],[33],[34]. With important parameters of initial momentum of 0.5 and final momentum of 0.9, weight cost of 0.0002. Meanwhile, to obtain optimum weights and biases to present best accuracy, based on [35], the MBO parameters were set as $S_{max} = 1.0$, $peri = 1.2$, and maximum generation = 50. In the result tables, Tables I to V summarizes the results that have been obtained from different hidden layers. These experiments are simulated using the VSD2014 dataset as the focus domain is on violence classification.

The efficiency of the proposed technique has been thoroughly tested with different hidden layers. This has shown the efficiency of the proposed technique, furthermore able to classify the data correctly at satisfying classification rate of 94% with hidden layer 5 with precision of 0.4471 and recall of 0.8892. It is believed that these parameters and arrangements

are appropriate for the classification of violence content in VSD2014 dataset.

TABLE. I. RESULT OF COMPARISONS WITH 3 HIDDEN LAYERS

Approach	Precision	Recall	Acc.	Inc.
DNNMBO	0.5004	0.469	49.98%	50.02%
DNNPSO	0.4947	0.5472	48.60%	51.40%
DNN	0.4638	0.4638	46.29%	53.70%

TABLE. II. RESULT OF COMPARISONS WITH 4 HIDDEN LAYERS

Approach	Precision	Recall	Acc.	Inc.
DNNMBO	0.696	0.702	70.35%	29.65%
DNNPSO	0.5023	0.447	58.67%	41.33%
DNN	0.4856	0.4856	48.57%	51.43%

TABLE. III. RESULT OF COMPARISONS WITH 5 HIDDEN LAYERS

Approach	Precision	Recall	Acc.	Inc.
DNNMBO	0.4471	0.8892	94%	6%
DNNPSO	0.5667	0.6415	75.22%	24.78%
DNN	0.5242	0.5242	52.57%	47.43%

TABLE. IV. RESULT OF COMPARISONS WITH 6 HIDDEN LAYERS

Approach	Precision	Recall	Acc.	Inc.
DNNMBO	0.788	0.631	80.30%	19.70%
DNNPSO	0.5318	0.5222	73.50%	26.50%
DNN	0.5378	0.5374	53.71%	46.29%

TABLE. V. RESULT OF COMPARISONS COMPARISON WITH 7 HIDDEN LAYERS

Approach	Precision	Recall	Acc.	Inc.
DNNMBO	0.5972	0.7963	60.00%	40.00%
DNNPSO	0.5004	0.469	53.58%	46.42%
DNN	0.4571	0.5	45.71%	54.29%

VI. CONCLUSION

As a conclusion, this study implemented MBO optimization algorithm in integration with DNN in favor to enhance the accuracy rate of the conventional DNN. From the conventional DNN, researcher learned that untreated data can be a reason of inefficacy of performance. Thus, in this approach, the data is treated especially on missing values problem and modification of DNN with optimization algorithm has seen the desired outcome. Hence, implementation of monarch butterfly optimization (MBO) algorithm with DNN has contributed in the improvement of the accuracy as it is compatible to each other. As a summary, this employment of DNNMBO considered suitable for violence classification domain.

As a potential research direction, the proposed technique can also be tested with other violence datasets or other focus

domain; multiple classifications with more meaningful features, such as using features ranks or feature selection. It is advisable to improve on the weighting factor of monarch butterfly optimization $S_{max} = 1.0$ to MinStep.

ACKNOWLEDGMENT

This research is fully funded by Ministry of Higher Education (MOHE) under the Fundamental Research Grant Scheme (FRGS) – Vot. No. 1608; in part supported by Research Management Center (RMC), UTHM.

REFERENCES

- [1] Cantor, J. (2000). Media violence. *Journal of adolescent health*, 27(2), 30-34.
- [2] Peixoto, B. M., Avila, S., Dias, Z., & Rocha, A. (2018, August). Breaking down violence: A deep-learning strategy to model and classify violence in videos. In *Proceedings of the 13th International Conference on Availability, Reliability and Security* (p. 50). ACM.
- [3] Fenil, E., Manogaran, G., Vivekananda, G. N., Thanjaivadivel, M., Jeeva, S., & Ahilan, A. (2019). Real time Violence Detection Framework for Football Stadium comprising of Big Data Analysis and Deep Learning through Bidirectional LSTM. *Computer Networks*.
- [4] Sudhakaran, S., & Lanz, O. (2017). Convolutional long short-term memory networks for recognizing first person interactions. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2339-2346).
- [5] Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4694-4702).
- [6] Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems* (pp. 568-576).
- [7] Wu, Z., Jiang, Y. G., Wang, J., Pu, J., & Xue, X. (2014, November). Exploring inter-feature and inter-class relationships with deep neural networks for video classification. In *Proceedings of the 22nd ACM international conference on Multimedia* (pp. 167-176). ACM.
- [8] Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 689-696).
- [9] Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015, September). Deep face recognition. In *bmvc* (Vol. 1, No. 3, p. 6).
- [10] Levi, G., & Hassner, T. (2015). Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 34-42).
- [11] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436.
- [12] EnterpricTech, <https://www.enterprisetech.com/2017/07/10/deep-neural-networks-not-use/>.
- [13] Fathollahi-Fard, A. M., Hajiaghahi-Keshтели, M., & Tavakkoli-Moghaddam, R. (2018). The social engineering optimizer (SEO). *Engineering Applications of Artificial Intelligence*, 72, 267-293.
- [14] Wang, GG., Deb, S. & Cui, Z. *Neural Comput & Applic* (2015). Monarch Butterfly Optimization. *Neural Computing & Applications*, <https://doi.org/10.1007/s00521-015-1923-y>.
- [15] Giannakopoulos, T., Kosmopoulos, D., Aristidou, A., & Theodoridis, S. (2006, May). Violence content classification using audio features. In *Hellenic Conference on Artificial Intelligence* (pp. 502-507). Springer Berlin Heidelberg.
- [16] Mu, G., Cao, H., & Jin, Q. (2016, November). Violent Scene Detection Using Convolutional Neural Networks and Deep Audio Features. In *Chinese Conference on Pattern Recognition* (pp. 451-463). Springer Singapore.
- [17] Mironică, I., Duță, I. C., Ionescu, B., & Sebe, N. (2016). A modified vector of locally aggregated descriptors approach for fast video classification. *Multimedia Tools and Applications*, 75(15), 9045-9072.

- [18] Ali, A., & Senan, N. (2016, August). A Review on Violence Video Classification Using Convolutional Neural Networks. In International Conference on Soft Computing and Data Mining (pp. 130-140). Springer, Cham.
- [19] Dai, Q., Wu, Z., Jiang, Y. G., Xue, X., & Tang, J. (2014, October). Fudan-NJUST at MediaEval 2014: Violent Scenes Detection Using Deep Neural Networks. In MediaEval.
- [20] Wu, Z., Jiang, Y. G., Wang, J., Pu, J., & Xue, X. (2014, November). Exploring inter-feature and inter-class relationships with deep neural networks for video classification. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 167-176). ACM.
- [21] Wu, Z., Jiang, Y. G., Wang, J., Pu, J., & Xue, X. (2014, November). Exploring inter-feature and inter-class relationships with deep neural networks for video classification. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 167-176). ACM.
- [22] Jiang, Y. G., Wu, Z., Wang, J., Xue, X., & Chang, S. F. (2015). Exploiting feature and class relationships in video categorization with regularized deep neural networks. arXiv preprint arXiv:1502.07209.
- [23] Zhang, B., Yi, Y., Wang, H., & Yu, J. (2014, October). MIC-TJU at MediaEval Violent Scenes Detection (VSD) 2014. In MediaEval.
- [24] Schedi, M., Sjöberg, M., Mironică, I., Ionescu, B., Quang, V. L., Jiang, Y. G., & Demarty, C. H. (2015, June). Vsd2014: a dataset for violent scenes detection in hollywood movies and web videos. In Content-Based Multimedia Indexing (CBMI), 2015 13th International Workshop on (pp. 1-6). IEEE.
- [25] Hecht-Nielsen, R. (1992). Theory of the backpropagation neural network. In Neural networks for perception (pp. 65-93).
- [26] Fenil, E., Manogaran, G., Vivekananda, G. N., Thanjaiavadivel, M., Jeeva, S., & Ahilan, A. (2019). Real time Violence Detection Framework for Football Stadium comprising of Big Data Analysis and Deep Learning through Bidirectional LSTM. Computer Networks.
- [27] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine, 29(6), 82-97.
- [28] Towards Data Science. (2017). The 10 Deep Learning Methods AI Practitioners Need to Apply. Available: <https://towardsdatascience.com/the-10-deep-learning-methods-ai-practitioners-need-to-apply-885259f402c1>
- [29] Wang, Gai-Ge & Deb, Suash & Cui, Zhihua. (2015). Monarch Butterfly Optimization. Neural Computing and Applications. 10.1007/s00521-015-1923-y.
- [30] Wang, Gai-Ge & Deb, Suash & Cui, Zhihua. (2015). Monarch Butterfly Optimization. Neural Computing and Applications. 10.1007/s00521-015-1923-y.
- [31] Wang G-G, Zhao X, Deb S (2015c) A novel monarch butterfly optimization with greedy strategy and selfadaptive crossover operator. Paper presented at the 2015 2nd international conference on soft computing & machine intelligence (ISCMi 2015), Hong Kong, Nov 23–24.
- [32] Piczak, K. J. (2015, September). Environmental sound classification with convolutional neural networks. In 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP) (pp. 1-6). IEEE.
- [33] Shin, H. C., Roth, H. R., Gao, M., Lu, L., Xu, Z., Nogues, I., ... & Summers, R. M. (2016). Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE transactions on medical imaging, 35(5), 1285-1298.
- [34] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.
- [35] G.G. Wang, S. Deb, Z.H. Cui, Monarch butterfly optimization, Neural Comput. Appl. (2015) <http://dx.doi.org/10.1007/s00521-015-1923-y>.