International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

CPW Optimization a Novel Optimization Algorithm to Improve Classification of Objects in Video Content Detection

Sita Yadav[1], Sandeep Chaware[2]

Submitted: 30/09/2023 **Revised:** 15/11/2023 **Accepted:** 28/11/2023

Abstract: Optimizers are essential for video object detection because they facilitate training and enhance model performance. During training, optimizers are in charge of the loss function minimization. The gradients of the loss parameters are used to iteratively update the model parameters. Optimizers steer the model towards convergence by iteratively changing the parameters in the direction of the steepest descent, which lowers the loss and enhances object detection performance. The hybrid optimizer known as the chaser priori wolf optimizer is proposed in the paper. The hybridization of coyote and cat swarm optimization forms the foundation of the chaser priori wolf optimization. The CPW optimizer was introduced in the proposed work to enhance feature selection and convergence in classification. The comparative outcome demonstrated that the CNN-based YOLO model performed better. In terms of sensitivity, specificity, and accuracy, the results are contrasted. The findings unequivocally demonstrated improvements in every performance metric, with an average improvement of 10.3% when compared to state-of-the-art architecture.

Keywords: *Coyote , Cat Swarm , chaser priori wolf optimization, deep learning, object detection*

1. Introduction

The conventional optimizer cannot handle sparse gradients and requires manual learning rate adjustment. In classification and detection problems, common optimizers include SGD, RMSProp, Adagrad, Adadelta, and ADAM. When compared to other methods, ADAM has demonstrated superior performance overall. Nonetheless, researchers are drawn to find more effective optimization strategies by the growing amount of video data and the need for detection and classification jobs. SGD converges slowly while dealing with complex data. When choosing an earning rate incorrectly, SGD produces local minima. Root Mean Square Propagation, or RMSProp The core principle of RMSprop is to maintain the moving average of the squared gradients for every weight. The gradient should then be multiplied by the mean square root. With various datasets, it performs inconsistently. Adagrad focuses on utilizing historical observational data [1]. As a result of the parameters' squared gradients building up over time, learning rates gradually decrease. Stochastic optimization technique Adadelta makes the per-dimension learning rate method for SGD possible. The Adam optimizer, an adaptive gradient-based optimization technique that has become quite popular in deep learning, is proposed in paper [2]. Adam provides effective and adaptive learning rates for various parameters by combining root mean square propagation (RMSprop) with momentum-based optimisation techniques. But it's necessary to compare with a fresh optimizer. Many evolutionary optimization techniques have been applied to detection and classification applications in recent years. This study presents a novel optimizer called Chaser Prairie Wolf (CPW) optimization, which is based on the coyote optimization technique and cat swarm optimization. In this case, the cooperative nature of prairie wolves aids in the chasers' decision-making. Via the effective modification of the classifier settings, the enabled optimization contributed to better results.

[1] Pimpri Chinchwad College of Eng., Pune, India
Army Institute of Technology, Pune, India
ORCID ID : 0000-0001-9510-4336
[2] JSPM RSCOE, Pune, India
** Corresponding Author Email: yadav.sita1@gmail.com*

2. Literature Survey

Photovoltaic power point tracking systems use quantum physics in conjunction with cat swarm optimization (CSO) to prevent local optima [1]. To improve search capability in solutions, the CSO algorithm is employed in numerous hybrid algorithm generations. The application of CSO lowers the computing cost [1][2]. By using various objective function terrains, the CSO is utilized to update processes and regulate individual solutions. In order to improve the search technique, the modified CSO is suggested.

A compact algorithm for optimizing cat swarms that utilizes a small sample probability model reduces computation costs while enhancing the search capacity for a possible global best solution [2]. The scalability of the performance is not addressed; instead, it is measured in particular scenarios. In [3], a novel class discovery network with a two-stage detector is proposed. In both labelled and unlabeled datasets, the approach greatly enhances class detection and is applied to unlabeled datasets to verify detection accuracy. The two-stage detector model does, however, come at a higher computational cost [2][3]. A thorough analysis of the bio-inspired algorithm is presented in a paper, demonstrating how the CSO algorithm on modification with exact features can produce better outcomes on a range of issues. The proposed model enhanced the CSO algorithm by adding focus boost search strategy, introducing variable mode ratio control strategy, and improving the tracing model process.

The first stage of the multi-stage object tracking method is referred to as preprocessing, and the second stage addresses multiple object identification in frames. To compute the cost function, the CSO algorithm's seeking and tracing mode is modified [4]. Improved search performance and local exploitation capability are demonstrated in the paper [5], which proposes an updated version of the FDB and Levy flight in the COA. However, the paper does not provide a thorough analysis using a real-time database. The research demonstrated that COYOTE produces better results than other optimization algorithms by offering a thorough comparison with them [5]. Coyote hunting behavior serves as the foundation for the Coyote optimization algorithm. It's a meta-heuristic method for solving a range of issues. In search spaces, it maintains equilibrium between both investigation and utilization. In computer vision problems, coyote optimization is used to locate and classify objects. By lowering losses, it maximizes detection efficiency [5][6]. In [6], a local optimum, a stochastic global optimization technique and optimizations algorithm are presented. The paper proposes a new stochastic global optimization technique for continuous search-space problems. The recommended method is a swarm-based strategy that uses spherical bounds to search a vector search space for the best solution [6]. The work proposes a novel combination of differential evolution and multi-objective particle swarm optimizations, which combines the latter with new procedures to solve particular optimization issues [7-8]. The literature review reveals that known optimizers are not as effective as nature-inspired optimization algorithms. In order to improve object classification and detection algorithm optimization, the hybridization of CSO and COYOTE optimization is the focus of the proposed work[9].

3. Algorithm

3.1. Cat Swarm Optimization (CSO)[7][8]

The hunting techniques of animals in the cat family serve as the basis for the nature-inspired optimization algorithm known as CSO. The best solution was found by following the behaviors of cats. Based on particular mixture ratios, the population was initially split into seeking and tracing modes. The algorithm find best answer by iteratively updating values stored to distributed cats. Seeking mode and tracing mode are the two main steps of the CSO algorithm [1–5]. The counting of dimensions and seeking memory pool are created by the seeking mode. To verify the suitability of the optimal solution, the dimensions are altered in every duplicate. The feature selection phase is the tracing mode in CSO, and it is shown in the following formula,

$$Vk+1 = wt * Vk + a*b*(Xj - Xi) \tag{1}$$

where k is the feature that was previously chosen and Vk is the previous velocity at that iteration. For the I iteration, Xi stands for the previous iteration position. In j dimensions, new features are represented by Xj. A and B represent the constant value and, respectively, a randomly generated number between 0 and 1. The cats are moved to the most recent optimal position, Xi+1, to provide the new feature that was chosen by taking into account their position as of late.

$$Xi+1= i + Vk+1 \tag{2}$$

3.2. Coyote optimization algorithm(COA)[9]

The coyote optimization algorithm is based on the idea of coyote behavior and social structure sharing [10]. The best feature is chosen based on the coyote's position. Using data from the coyotes, the COA determines the pack's cultural inclination. The random coyotes are selected with the help of uniform distribution of probability represented through values of δ_1 and δ_2.

$$\delta_1 = \alpha^{p,t} - soc_{cr1}^{p,t} \tag{3}$$

$$\delta_2 = c^{p,t} - soc_{cr2}^{p,t} \tag{4}$$

In the equation 3 & 4 cr1 and cr2 used to give random coyote packs. The values of δ_1 and δ_2 are taken with ‘p’ pack and ‘t’ instance.
The new social condition for coyote is given below ,

$$nsoc^{p,t} = soc_c^{p,t} + r_1.\delta_1 + r_2.\delta_2 \tag{5}$$

4. Proposed Algorithm

4.1. Chaser Priorie Wolf (CPW)

A hybrid optimization algorithm for classification was proposed in the paper. The work is based on the CSO and COA algorithms, which are inspired by nature. Features were selected based on coyotes' accuracy and cats' alertness. The two modes of the work are the tracing mode and the seeking mode. The cost function is decreased and the weights are adjusted using the CPW algorithm. The feature selection formula is represented by the equation below.

$$\alpha = Ynp + soc \tag{6}$$

where Ynp denotes the new object-containing box and Y is a random number between 0 and 1. Soc stands for shared characteristics. Turning on CPW optimization reduces the error occurrence, which is necessary for effective classification. Rather than maintaining a constant learning rate throughout the training process, the optimization approach adjusts learning rate for network weights. To get the best outcome, the weight is carefully adjusted to show how the input will affect the output. By using this CPW technique, the classifiers' bias and noise are also eliminated. To effectively pursue and seize the prey, the chaser must modify its position by applying the proper angle and speed.

$$\delta = P_{C.j.new} + new\_soc_{dw}^{r,T} \tag{7}$$

$$\delta = \left(P_{C.j.old} + C_{vy.j}\right) + new\_soc_{dw}^{r,T} \tag{8}$$

$P_{C.j.new}$ shows updated position of the chaser with dimension ‘ j ’ at a specific time instance ‘ T ’, and prairie wolves represented by the symbol ‘ r ’. Detailed comparison of various optimizers is shown in Table 1.

Table 1: Comparison of various Optimizers

Optimization	Learning Rate	Convergence Speed	Strengths
Adam [15]	Momentum used with adaptive learning rate	Fast	Efficient memory utilization
Adagrad [16]	Adaptive learning rates parameter wise	Slow	Useful in sparse parameter situations
Adadelta [18][19]	Per-dimension learning rate method	Moderate	Good for online learning, no explicit learning rate
Coyote Optimization [10]	Confidence Intervals used	Fast	Strong convergence, suitable for non-convex issues
Cat Swarm Optimization [1][2]	Combining exploitation and exploration with a large number of agents	Moderate	Sufficient for multimodal problems and efficient for global optimization
Chaser Priori Wolf (CPW)	Swarm intelligence combined with	Fast	Robust and multimodal support for ensemble models

Optimization (Proposed)	adaptive learning rates		

5. Methodology

One of the main tasks in the video content detection task is object localization and classification. One method that has been shown to work well for improving classifier performance is hyper parameter tweaking of CNN-based models. However, it depends on carefully choosing the parameters. The CPW algorithm simplifies and optimizes. The standard hybridization of the traits of the prairie wolf and the chaser leads to the development of the suggested Chaser prairie wolf optimization. Results are better with the hybrid classifier that combines CPW and Deep CNN. Figure 1 illustrates the steps in the process.

MSCOCO and PASCAL VOC are the datasets that were used. With about 328,000 images, the COCO image dataset was carefully chosen to improve image recognition capabilities. The inclusion of difficult and excellent visual data for a range of computer vision tasks makes this dataset noteworthy. Annotations in the COCO dataset cover a variety of tasks, including segmentation, object detection, captioning, and more. In contrast, PASCAL VOC is a widely used benchmark dataset that addresses issues with object detection, semantic segmentation, and classification. This dataset is divided into three subsets: 1,449 images assigned to validation, 1,464 images allocated for training, and a private testing set for evaluation.

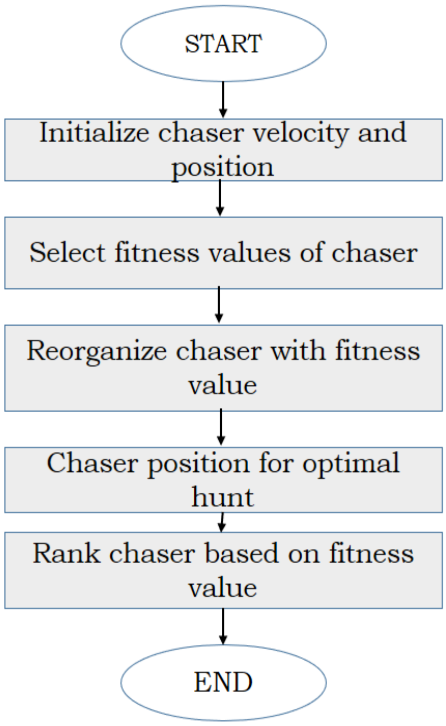

Fig 1. Methodology

Fig. 2: Results (a) Occluded objects, (b) Multiple Objects in occlusion

6. Experimental setup

Python and the Keras library were used to conduct the experiment on a Windows 10 PC with 16 GB of RAM. The implementation made use of both the PASCAL and COCO datasets. As test inputs, arbitrary video clips were used to evaluate the model's performance.

7. Results

Figure 2 displays the classification and object detection under different object conditions using the suggested CPW optimizer in YOLO V4. Different object positions in frames and images are displayed in Figures 2(a) through 2(d). The suggested model can successfully identify objects that are obscured, in low-light frames, and against difficult backgrounds. Three metrics are used to measure performance: sensitivity, specificity, and accuracy. The MSCOCO dataset yields content detection accuracy of 94% on average, with a few classes reaching up to 94% accuracy. With an average sensitivity of 89%, an average specificity of 89% is attained. 50 epochs at a learning rate of 0.001 were employed. The training dataset was 80 per cent and 20 percent of dataset classes and the average accuracy is 91 percent. With an average sensitivity of 89%, an average specificity of 89% is attained. 50 epochs at a learning rate of 0.001 were employed. Eighty percent of the dataset was used for training, and the remaining twenty percent was used for testing. With the Pascal VOC dataset, the average accuracy obtained was 84%, the specificity obtained was 92%, and the sensitivity obtained was 81%. Figures 3 and 4 illustrate the accuracy, sensitivity, and specificity attained using the MSCOCO and PASCAL VOC dataset.

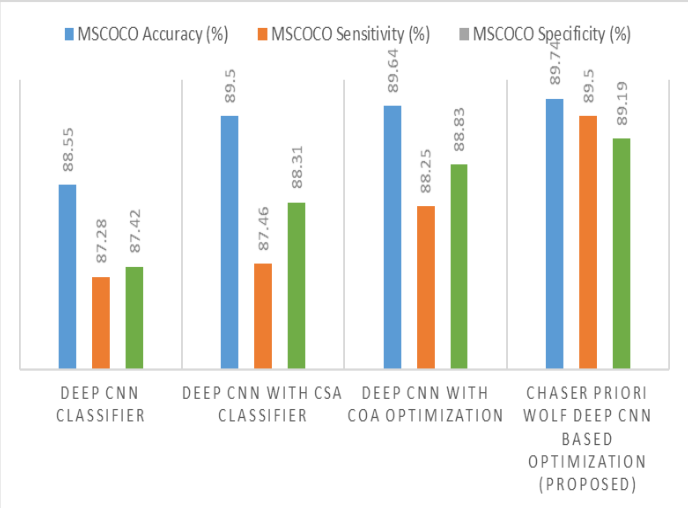

Fig 3: CPW optimizer implemented with CNN algorithms.

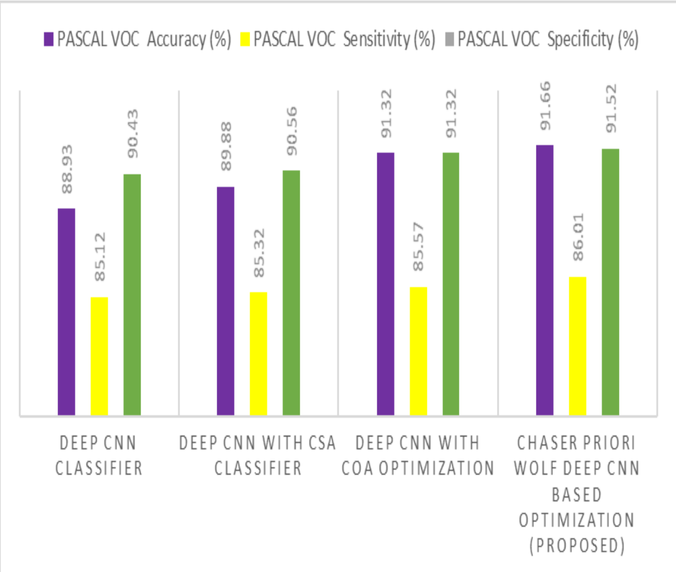

Fig 4: Using the PASCAL dataset, the CPW optimizer with Deep CNN is implemented, and the outcomes are compared with other classification algorithms.

8. Conclusion

The deep CNN classifier is used in conjunction with this suggested CPW optimizer to address the challenge of object detection in video content. Video analytics provide an abundance of important information and data regarding how viewers react to this kind of content. In this work, the classifier's parameters are optimized with the aid of the CPW optimization technique, leading to improved outcomes overall. The performance metrics for MSCOCO are 89.50% sensitivity, 89.19% specificity, and 89.74% accuracy. More astonishingly, PASCAL produces even greater efficiency, highlighting the strength and superiority of our method with values of 91.66% accuracy, 86.01% sensitivity, and 91.52% specificity. In practice, this model can be used for real-time scenarios such as video indexing for surveillance, summarization, and watermarking, among others. These adaptable applications demonstrate how our research can improve different facets of video content management and analysis.

References

[1] Pei-Wei Tsai, Xingsi Xue, Jing Zhang, Vaci Istanda, "Adjustable mode ratio and focus boost search strategy for cat swarm optimization", Applied Computing and Intelligence, 2021, 1(1): 75-94. doi: 10.3934/aci.2021005.

[2] Zeyu, Ming Zhao, Tie Luo, and Yimin Yang, "A Compact Cat Swarm Optimization Algorithm Based on Small Sample Probability Model" Applied Sciences 12, no. 16: 8209. https://doi.org/10.3390/app12168209.

[3] Sabah, Mustafa, "An Enhanced Video Tracking Technique Based on Improved Cat Swarm Optimization",2015, 10.13140/RG.2.2.31093.37604.

[4] Duman, S., Kahraman, H.T., Guvenc, U. et al., "Development of a Lévy flight and FDB-based coyote optimization algorithm for global optimization and real-world ACOPF problems", Soft Computing 25, 6577–6617 (2021). https://doi.org/10.1007/s00500-021-05654-z.

[5] Rasoul Rahmani, Rubiyah Yusof, "A new simple, fast and efficient algorithm for global optimization over continuous search-space problems: Radial Movement Optimization", Applied Mathematics and Computation, Volume 248, 2014, Pages 287-300, ISSN 0096-3003, https://doi.org/10.1016/j.amc.2014.09.102.

[6] Dhafar Al-Ani, Saeid Habibi, "A New Particle Swarm Optimization and Differential Evolution Technique for Constrained Optimization Problems", https://doi.org/10.1115/IMECE2013-63877, 2014.

[7] Arora, S., Singh, S., " Butterfly optimization algorithm: a novel approach for global optimization", Soft Computing 23, 715–734 (2019). https://doi.org/10.1007/s00500-018-3102-4.

[8] Ferrante Neri, Ernesto Mininno, Giovanni Iacca, "Compact Particle Swarm Optimization", Information Sciences, Volume 239, 2013, Pages 96-121, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2013.03.026.

[9] Ming Zhao (2020) A Novel Compact Cat Swarm Optimization Based on Differential Method, Enterprise Information Systems, 14:2, 196-220, DOI: 10.1080/17517575.2018.1462405.

[10] J. Pierezan and L. Dos Santos Coelho, "Coyote Optimization Algorithm: A New Metaheuristic for Global Optimization Problems," *2018 IEEE Congress on Evolutionary Computation (CEC)*, Rio de Janeiro, Brazil, 2018, pp. 1-8, doi: 10.1109/CEC.2018.8477769.

[11] Choi, D., Shallue, C. J., Nado, Z., Lee, J., Maddison, C. J., Dahl, G. E., "On Empirical Comparisons of Optimizers for Deep Learning", ICLR 2020.

[12] H. Sikkandar, R. Thiyagarajan, Deep learning based facial expression recognition using improved Cat Swarm Optimization, J. Amb. Intel. Hum. Comp., 12 (2021), 3037–3053.

[13] S. Das, S. Maity, B.-Y. Qu and P.N. Suganthan, "Real-parameter evolutionary multimodal optimization — a survey of the state-of-the-art", Swarm and Evolutionary Computation, vol. 1, no. 2, pp. 71-88, June 2011

[14] Gupta et. al., "Adam vs. SGD: Closing the generalization gap on image classification", OPT2021: 13th Annual Workshop on Optimization for Machine Learning, 2021.

[15] P. Kingma and J. Ba, "Adam: A method for stochastic optimization", 3rd International Conference on Learning Representations, (ICLR), 2015.

[16] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, Deva Ramanan, "Object detection with discriminatively trained part-based models", Pascal Challenge , 2010.

[17] Choi, D., Shallue, C. J., Nado, Z., Lee, J., Maddison, C. J., Dahl, G. E., "On Empirical Comparisons of Optimizers for Deep Learning", ICLR 2020.

[18] S. Bera and V. K. Shrivastava, "Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification", Int. J. Remote Sens., vol. 41, no. 7, pp. 2664–2683, 2020 .

[19] Everingham, M. and VanGool, L. and Williams, C. K. I. and Winn, J. and Zisserman, A., "Pascal 2007 Dataset", http://www.pascalnetwork.org/challenges/VOC/voc2007/workshop/index.html, 2007.