**SIR SYED UNIVERSITY OF ENGINEERING & TECHNOLOGY**

**COMPUTER SCIENCE & INFORMATION TECHNOLOGY DEPARTMENT**

**Fall 2023**

**Parallel & Distributed Computing (CS-429)**

**Assignment # 2**

**Dataset of G12:** <https://www.kaggle.com/datasets/alik05/forest-fire-dataset>

**Group Members:**  
**Muhammad Shayan Ashraf**

**BSCS -2020 -097  
Section: C**

**Muhammad Nouman Imran**

**BSCS -2020 -097**

**Section: C**

| **CLO #** | **Course Learning Outcomes (CLOs)** | **PLO Mapping** | **Bloom’s Taxonomy** |
| --- | --- | --- | --- |
| CLO 2 | **Predict** the following program using GPU utilization in python. | PLO\_1  (Academic Education) | C2  (**Predict**) |

Q1. **Predict** the following program using GPU utilization in python by using your existing Model with any PSO variant (Hybrid Model). Assignment must contain/cover the following points.

**Note:** You have already given a dataset in **Assignment 1** for simulation. In **ASSIGNMENT 1** you have already designed a **GPU based model** in Deep learning by using Built in optimizer. In this assignment you have to combine your existing simulation model with **PSO optimization** to make your existing model as **Hybrid optimized model**. You may use any **PSO variant as mentioned below** but **NO** group should practice same PSO variant in same section.

1-Abstract (450 words) one paragraph

2-Detail of dataset (450 words) one paragraph

3-Short detail (450 words) of **type of PSO variant** you have used in program.

4-Program must use auto split function to split dataset into 70, 15, and 15 (Training, Testing, and validation)

5-Base model e.g. VGG, Inception + Optimizer e.g. Adam, RMS prop) **(as per your choice)**

6-Precision, Recall, F1-Score, True Positives, False positives, True Negatives, False Negatives

7- Plot the training and validation accuracy graph & Plot the training and validation loss graph.

8- Plot the confusion matrix for the training and Validation set.

9- Create a line plot graph for the number of images per class

10- Calculate ROC curves, AUC, and error rates for each class

11- Calculate image counts graph/Number of images for each process e.g. testing, train, and validation.

You may use any PSO variant from the following list.

1. Global Best PSO (GB-PSO):

2. Local Best PSO (LB-PSO).

3. Fully Informed PSO (FIPS

4. Ring PSO

5. Clan-Based PSO.

6. Adaptive PSO

7. Constriction Coefficient PSO

8. Chaotic PSO

9. Dynamic PSO

10. Multi-objective PSO (MOPSO).

11. **Hybrid PSO. (Used)**

12. Self-Adaptive PSO.

**Abstract**

We represent a comprehensive solution for fire detection using a deep learning approach, specifically employing the InceptionV3 model. The dataset, stored in Google Drive, is loaded and preprocessed using an Image-Data-Generator for data augmentation and normalization. The InceptionV3 model, pre-trained on ImageNet, is utilized as a base model, with a custom head added for fine-tuning on the fire detection task. The model is compiled with categorical cross entropy loss and the Adam optimizer. To optimize hyperparameters, a hybrid Particle Swarm Optimization (PSO) algorithm is implemented. This optimization process dynamically adjusts learning rates and the number of training epochs to enhance the model's performance. The PSO algorithm is fine-tuned to achieve the best solution in terms of validation accuracy. The trained model is then evaluated on the validation set, and performance metrics such as precision, recall, F1-score, and confusion matrices are calculated and presented. Additionally, the training and validation accuracy and loss are visualized through graphs. The code concludes with an exploration of the dataset's class distribution and a demonstration of Receiver Operating Characteristic (ROC) analysis for fire detection.

2-**Detail of dataset (450 words) one paragraph**

I made a dataset for spotting forest fires. It has 760 pictures of places with fires and 760 pictures of places without fires. I got these pics from Kaggle. To make things quicker, I only used 11 pics for each category during training because going through all of them takes a long time. This smaller set still helps the model learn without being too slow. The goal is to balance efficiency with having good examples for the model to learn from, making it better at spotting forest fires.

**Q2: Type of PSO Variant Used in the Program (450 words)**

The Particle Swarm Optimization (PSO) algorithm implemented in the program is a variant known as Hybrid PSO. Hybrid PSO combines the traditional PSO algorithm with additional components to enhance its effectiveness in optimizing hyperparameters for machine learning models. The primary motivation behind using Hybrid PSO is to strike a balance between exploration and exploitation, improving the algorithm's ability to converge to a globally optimal solution.

In Hybrid PSO, each particle in the swarm represents a candidate solution in the hyperparameter space. The particles traverse the solution space seeking the optimal set of hyperparameters that maximizes the validation accuracy of the deep learning model. The hybrid nature of the algorithm is manifested in the incorporation of three key components: cognitive, social, and memory components.

The algorithm also features self-adaptive inertia weight, which dynamically adjusts the impact of previous velocities on the particle's movement. This self-adaptation aids in balancing exploration and exploitation throughout the optimization process.

The hybridization of these components results in a more robust and adaptive optimization algorithm, making Hybrid PSO suitable for tuning hyperparameters in complex machine learning models, such as deep neural networks. The implementation in the code leverages these principles to fine-tune the learning rate and number of epochs for the InceptionV3 model, ultimately enhancing its performance in fire detection.



















































