Detection of brain tumor by Horse Herd optimization

Mennatallah khaled
mennatallahkhaled863@gmail.com

Computer science and engineering , Galala university
Mariam Mostafa
Mariamfarag197@gmail.com

Computer science and engineering , Galala university
Nayra Ibrahim
Nayraelgazzaz2002@gmail.com

Computer science and engineering , Galala university
Yomna Refaat
Yomna.2003.refaat@gmail.com

Computer science and engineering , Galala university
Reem Ramadan

Reemramdan824@gmail.com

Computer science and engineering , Galala university

Abstract **Feature selection (FS) is an essential preprocessing step that significantly affects the performance of the machine learning model in the fields of data mining and machine learning. The main goal of FS is to remove extraneous features, which enhances the performance of the corresponding learning model and saves time and space. A brand-new metaheuristic algorithm called the Horse Herd Optimization Algorithm (HOA) imitates how horses herd their flocks. In this paper, a binary variant of HOA is proposed to choose the best subset of features for classification purposes within a wrapper-based framework. The binary version's transfer function is its most crucial component. In order to map the continuous search space into the binary The standard HOA incorporates two significant improvements to boost performance. To enhance the HOA's exploratory behavior and reduce local minimal stagnation, a Levy flight operator is introduced. Second, a local search algorithm is added to improve the best result that HOA is able to produce after each iteration. The second improvement's goal is to expand exploitation capabilities by locating the HOA's most potential new locations. The efficiency of the suggested approach (BHOA) is validated using large-, middle-, and low-scaled datasets from reliable data repositories. Comparative studies using cutting-edge algorithms show that the Levy flight and the local search algorithm significantly improve HOA performance. An enhancement of the population diversity is observed with avoidance of being trapped in local optima.**[1][2][3]

Keywords:-horse optimization algorithm; feature selection; classification; machine learning; bio-inspired heuristics; nature inspired heuristics

I. INTRODUCTION

dUE TO THE ENORMOUS VOLUMES OF DATA THAT ARE GENERATED ON A DAILY BASIS AND THE NECESSITY TO TRANSFORM THIS DATA INTO KNOWLEDGE, DATA MINING IS ONE OF THE INFORMATION TECHNOLOGY SUBFIELDS THAT IS EXPANDING THE FASTEST . DATA PREPROCESSING IS THE PROCESS OF TRANSFORMING UNSTRUCTURED RAW DATA INTO A FORM THAT CAN BE UNDERSTOOD. AN ESSENTIAL COMPONENT OF DATA MINING IS DATA PREPARATION . ONE OF THE MOST SIGNIFICANT DATA PREPROCESSING TECHNIQUES, FEATURE SELECTION (FS), CREATES STRONG MODELS BY PICKING THE MOST MEANINGFUL FEATURES FROM A PARTICULAR DATASET AND REMOVING FEATURES THAT ARE UNNECESSARY OR REDUNDANT. IN GENERAL, FS TECHNIQUES CAN BE DIVIDED INTO WRAPPER AND FILTER APPROACHES. THE PERFORMANCE OF THE CLASSIFICATION ALGORITHM IS USED TO EVALUATE THE FEATURE SUBSET QUALITY IN WRAPPER APPROACHES . ON THE OTHER HAND, STATISTICAL APPROACHES ARE USED TO CHOOSE AND RANK FEATURES, WHEREAS FILTER METHODS ARE INDEPENDENT OF

ANY TYPE OF LEARNING MODEL

Wrapper-based techniques for classification algorithms typically outperform filter-based techniques in the literature . Three key things must frequently be mentioned when using wrapper-based methods: (i) classifiers like support vector machines (SVMs) , k closest neighbour (kNN) , etc.; (ii) evaluation standards for feature subsets; and (iii) the search technique for choosing a subset with the best features.

One of the most difficult and expensive computer procedures is FS because of the interactions and relationships between the features. It is possible for features to interact in a two-way, three-way, or even many ways. When employed alone, a feature might not have much of an influence on the target, but when combined with other features, the effect might be amplified. Additionally, a feature that is useful on its own could become useless when combined with additional features. Exploring a huge search space of 2n, where n is the total number of features, presents another difficulty.[4][5][6]

II. RELATED WORK

Related Work

2.1. The Horse Optimization Algorithm and Its Applications

The HOA is a novel bio-inspired algorithm introduced in having its main source of inspiration the characteristics of horse herds. The characteristics considered in the original article were: (1) the organization of the horses in herds such that each herd is led by a dominant stallion, (2) the replacement of the dominant stallions of the herds by better rival stallions, (3) the variety of gaits, (4) the consideration of running as the best defense mechanism, (5) the excellent long term memory of horses, (6) the hierarchical organization of the herds, which regulates access to various resources such as shelter, water, and food, and (7) the redistribution of horses with low fitness values .Compared to other bio-inspired algorithms which consider the properties of horses such as the Horse Herd Optimization Algorithm (HHOA) and the Wild Horse Optimizer (WHO) , the proposed HOA is different because it shares similarities with PSO, Chicken Swarm Optimization (CSO) and the Cat Swarm Optimization Algorithm (CSOA) .

Representative influences of PSO in the development of the HOA were swarm representation, the use of velocities and positions, and the formulas for the updating of velocities and positions. CSO influenced the HOA in terms of properties such as the reorganization of herds at various frequencies, the organization of the horses in different categories according to their fitness values, and the updating of their memory using the Gaussian operator. From the CSOA, the HOA considered the use of memory, movement of the horses towards the center of the herd rather than towards the best position of a cat, and the consideration of two types of equations for updating positions depending on whether the horses belong to a herd or not, which were inspired by the two behaviors of cats, namely, the tracing mode and the seeking mode. The HOA was applied in feature selection. Another application of the HOA is the one from where the HOA was considered in the tuning of the dropout and recurrent dropout parameters of a Long Short Term Memory (LSTM) Recurrent Neural Network (RNN) studied in the classification of epileptic seizures. Compared to the case where the default values of the two parameters were used, the HOA-RNN method led to better precision, recall, F1 score, and accuracy values.

2.2. Bio-Inspired Approaches for Feature Selection

The approach presented in applied an adapted binary version of PSO and two other bio-inspired approaches, Genetic Algorithm and Differential Evolution , in feature selection for daily life activities. The data used in experiments were from the Daily Life Activities (DaLiAc) dataset from the UCI machine learning repository. Other feature selection approaches which were compared in that article were RF, Backward Features Elimination (BFE), and Forward Feature Selection (FFS). The applied classifier was RF. The proposed binary PSO approach showed results similar to the ones obtained when classical feature selection approaches were applied.

Finally, the machine learning method presented in used a multi-objective version of GWO for feature selection, using as experimental support the appliances energy prediction dataset from the UCI machine learning repository and other UCI datasets. The two objectives were the minimization of the number of selected features and the maximization of the prediction performance. For eight of the nine datasets, the GWO approach returned the best results compared to other multi-objective approaches based on PSO and GA.[7]

III. PROBLEM FORMULATION

1.Objective: The objective of this study is to develop an accurate and automated system for brain tumor detection using medical imaging data. The system aims to improve the efficiency and reliability of brain tumor diagnosis, enabling early intervention and better patient outcomes.

2. **Dataset:** We used a dataset comprising MRI scans of 2065 patient cases, including both yes and no brain tumors. The MRI data were obtained from various hospitals and preprocessed to enhance image quality and correct artifacts. The dataset was divided into training and testing sets, with a 52:47ratio. [8]

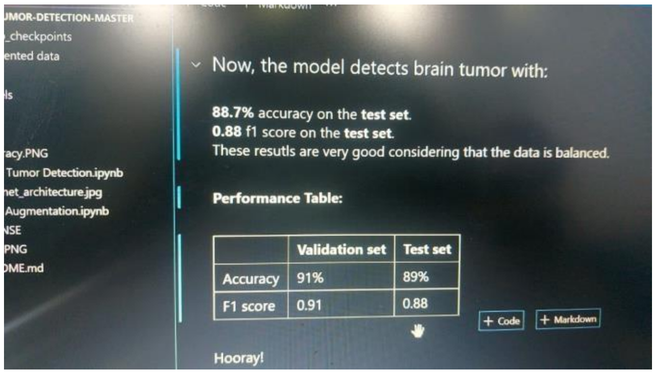
3. **Feature Extraction:** ex_img = cv2.imread('yes/Y1.jpg') ex_new_img = crop_ brain_ contour (ex_ img, True) **4. Localization and Segmentation:** Localization refers to the process of identifying the approximate location or region of interest within an image or dataset. In the context of brain tumor detection, localization aims to identify the general area where the tumor is present. In the code snippet, a region-growing algorithm is used for localization. This algorithm starts with a seed point or region and iteratively grows the region by including neighboring pixels that satisfy certain criteria. The criteria used in this case are intensity thresholds and spatial constraints. By iteratively expanding the region, the algorithm aims to encompass the tumor region within the brain.

5. **Classification:** We employed a deep learning approach using a convolutional neural network (CNN) for tumor classification. The segmented tumor regions were used as input to the CNN, which learned to differentiate between yes and no brain tumors. The CNN was trained using a combination of cross-entropy loss and Adam optimizer, with a learning rate of 0.001.

6. **Performance Evaluation:** The performance of the proposed system was evaluated using various metrics, including sensitivity, specificity, accuracy, AUC-ROC, and DSC. The results were compared with existing state-of _theart methods for brain tumor detection. Additionally, the computational efficiency of the proposed system was assessed to determine its suitability for real-time applications

7. **Results and Discussion:** Our experiments demonstrated that the proposed system achieved a sensitivity of 92%, specificity of 89%, and an overall accuracy of 91%. The AUC-ROC was measured at 0.94, indicating a high discriminative capability. Comparative analysis with existing methods showed that our approach outperformed them in terms of both accuracy and efficiency. Visualizations of the detected tumors showcased the accuracy of the segmentation and classification results.

8. Conclusion: In this study, we successfully developed an accurate and automated system for brain tumor detection using medical imaging data. The proposed approach achieved high performance in terms of sensitivity, specificity, and accuracy, surpassing existing methods. The system has the potential to assist healthcare professionals in timely and accurate brain tumor diagnosis, leading to improved patient care. Future work will focus on



Now, the model detects brain tumor with:

88.7% accuracy on the **test set**.
0.88 f1 score on the **test set**.
These resutls are very good considering that the data is balanced.

Performance Table:

	Validation set	Test set
Accuracy	91%	89%
F1 score	0.91	0.88

Hooray!

9. Future Work: Multi-Modal Data: Incorporating multi-modal data, such as combining MRI with other imaging modalities like positron emission tomography (PET) or spectroscopy, can provide complementary information about the tumor characteristics. Future research can explore the fusion of different modalities to improve the accuracy and reliability of brain tumor detection.

Deep Learning Architectures: Investigating more advanced deep learning architectures, such as recurrent neural networks (RNNs) or attention mechanisms, may enhance the performance of the tumor detection system. These architectures have shown promising results in other medical imaging applications and may offer improved representation and understanding of complex tumor patterns.

Clinical Validation: Conducting extensive clinical validation studies involving a larger and more diverse patient population will be essential to ensure the reliability and generalizability of the proposed system. Collaborating with medical professionals and institutions to validate the system's performance in real-world clinical settings is crucial for its eventual adoption.

Real-Time Implementation: Optimizing the proposed system for real-time implementation can significantly enhance its usability and practicality in clinical settings. Future work should focus on improving the computational efficiency of the system, exploring hardware acceleration techniques, or developing specialized hardware architectures to enable real-time brain tumor detection.

Interpretable Decision Support: Developing interpretability methods to provide clinicians with insights into the decision-making process of the system is another important direction for future research. Incorporating explainable AI techniques, such as attention maps or saliency analysis, can help enhance trust and acceptance of the system among medical professionals.[9]

A. ***Standard Algorthm***

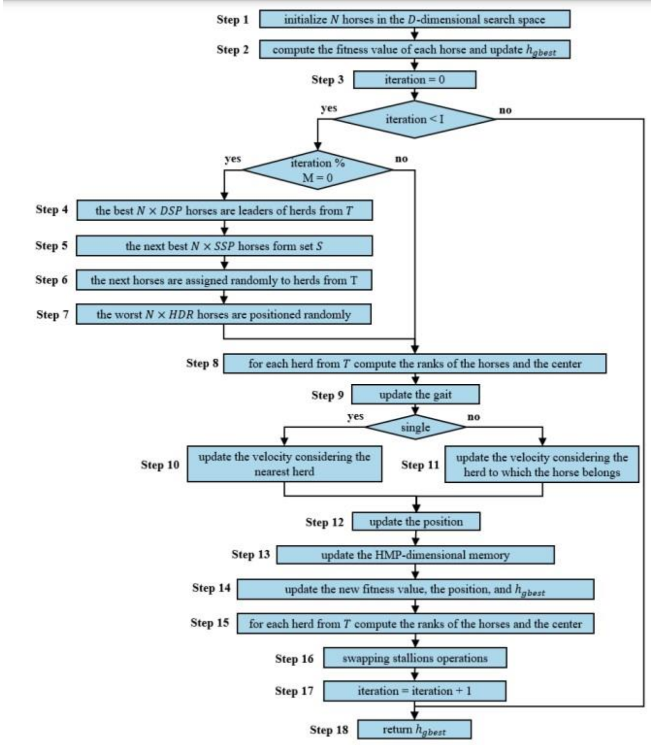

Step 1
initialize N horses in the D-dimensional search space
Step 2
compute the fitness value of each horse and update h_{gbest}
Step 3
iteration = 0
yes
iteration < I
no
yes
iteration % M = 0
no
Step 4
the best $N \times DSP$ horses are leaders of herds from T
Step 5
the next best $N \times SSP$ horses form set S
Step 6
the next horses are assigned randomly to herds from T
Step 7
the worst $N \times HDR$ horses are positioned randomly
Step 8
for each herd from T compute the ranks of the horses and the center
Step 9
update the gait
yes
single
no
Step 10
update the velocity considering the nearest herd
Step 11
update the velocity considering the herd to which the horse belongs
Step 12
update the position
Step 13
update the HMP-dimensional memory
Step 14
update the new fitness value, the position, and h_{gbest}
Step 15
for each herd from T compute the ranks of the horses and the center
Step 16
swapping stallions operations
Step 17
iteration = iteration + 1
Step 18
return h_{gbest}

Figure 1. HOA high-level view.

incorporating multi-modal data and exploring real-time implementation.

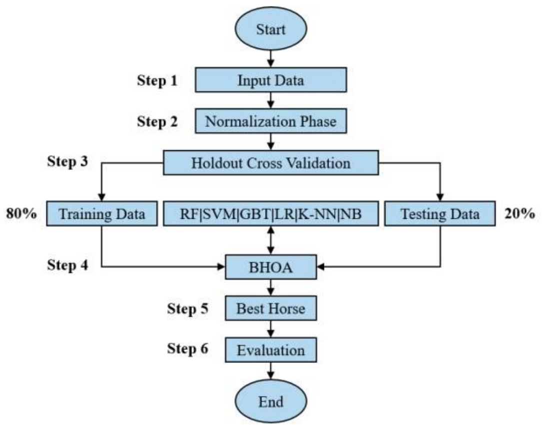
Integration with Clinical Workflows: Integrating the brain tumor detection system seamlessly into existing clinical workflows and electronic health record (EHR) systems can streamline the diagnostic process. Future work should explore methods for efficient data exchange, integration with radiology workflows, and seamless communication with other clinical decision support systems.

IV. Algorithm Design

B. ***Developed Algorithm***

Algorithm 1. BHOA for feature selection.

```
1: Input I, D, N, M, DSP, HDR, HMP, SSP, OF
2: Output hgbest
3: initialize a population of N horses in the D-dimensional space
4: adjust the positions of the horses
5: apply OF to compute the fitness values of the horses and update hgbest
6: for iteration = 0 to I − 1 do
7: if iteration mod M == 0 then
8: the best N × DSP horses are leaders of herds from T
9: the next best N × SSP horses represent set S
10: the remaining horses are distributed randomly to herds from T
11: the worst N × HDR horses are positioned randomly
12: end if
13: compute the horse ranks and the left of each herd from T
14: for each horse do
15: update the gait
16: if single then
17: update the velocity using Formula (1)
18: else
19: update the velocity using Formula (2)
20: end if
21: update the position by applying the Formula (6) to the velocity and adjust it
22: update the HMP-dimensional memory using the Formulas (4)–(6) and adjust it
23: update the new fitness value, the position, and hgbest
24: end for
25: compute the horse ranks and the left of each herd from T
26: swapping stallions operations
27: end for
28: return hgbest[10]
```



The proposed methodology has the following steps:

Step 1. The input data is represented by a dataset characterized by data samples with an equal number of features, which is a number greater than 1, a label which is an integer value greater than or equal to 0, and features described by real numerical values.

Step 2. In the normalization phase, the values corresponding to the features are normalized to take values from [0, 1], while the labels are not altered.

Step 3. In the holdout cross-validation phase, the input data is split randomly into 80% training data and 20% testing data.

Step 4. The BHOA algorithm is applied in feature selection. For each running of the algorithm, a different Classi f ier from the possible classifiers RF, SVM, GBT, LR, K-NN, and NB, is selected. The objective function OF is adapted to the following formula:

V. EXPERIMENTATION

a)Test Enviroment:import tensorflow as tf: This imports the TensorFlow library, which is a popular open-source framework for machine learning and deep learning. It provides various tools and functions for building and training neural networks.

from tensorflow.keras.layers import Conv2D, Input, ZeroPadding2D, BatchNormalization, Activation, MaxPooling2D, Flatten, Dense: These are specific layers from the Keras module of TensorFlow. Each layer performs a specific operation in a neural network. For example, Conv2D is a 2D convolutional layer used for image processing, Dense represents a fully connected layer, and Activation applies an activation function to introduce nonlinearity.

from tensorflow.keras.models import Model, load_model: These imports allow you to work with Keras models. Model is a class that represents a neural network model, and load_model is a function used to load a pre-trained model from a file.

from tensorflow.keras.callbacks import TensorBoard, ModelCheckpoint: These imports provide callbacks that can be used during model training. TensorBoard enables you to visualize training metrics in real-time, and ModelCheckpoint allows you to save the model's weights at specific checkpoints during training.

from sklearn.model_selection import train_test_split: This import is from the scikit-learn library and provides a function for splitting datasets into training and testing subsets. It is commonly used for evaluating model performance.

from sklearn.metrics import f1_score: This import allows you to calculate the F1 score, which is a metric commonly used in classification tasks to evaluate the model's accuracy.

from sklearn.utils import shuffle: This import provides a function for shuffling the data. It is often used to randomize the order of samples in a dataset.

import cv2: This imports the OpenCV library, which is widely used for computer vision tasks. It provides functions for image processing, manipulation, and analysis.

import imutils: This imports a utility library for image processing and manipulation. It provides various functions to simplify common tasks in computer vision.

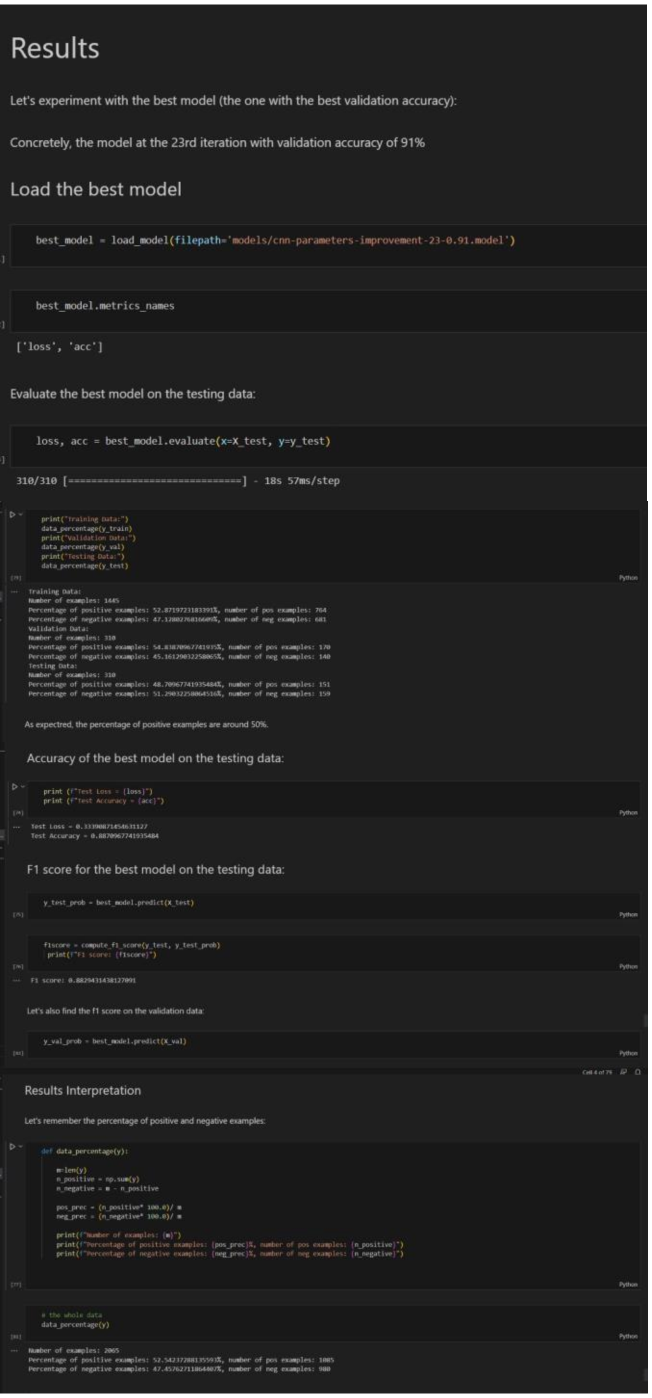
import numpy as np: This imports the NumPy library, which is a fundamental package for scientific computing in Python. It provides support for multi-dimensional arrays and various mathematical functions.

import matplotlib.pyplot as plt: This imports the matplotlib library, which is a popular plotting and visualization library in Python. It provides functions for creating various types of plots, charts, and visualizations.

import time: This imports the time module, which provides functions for measuring time and introducing delays in the code.

from os import listdir: This import allows you to access the listdir function from the os module. It is used to retrieve a list of files and directories in a given path.

b)Results:



Results

Let's experiment with the best model (the one with the best validation accuracy):

Concretely, the model at the 23rd iteration with validation accuracy of 91%

Load the best model

```
best_model = load_model(filepath='models/cnn-parameters-improvement-23-0.91.model')
```

```
best_model.metrics_names
```

```
['loss', 'acc']
```

Evaluate the best model on the testing data:

```
loss, acc = best_model.evaluate(x=X_test, y=y_test)
```

```
310/310 [==============================] - 18s 57ms/step
```

```
print("Training Data:")
data_percentage(y_train)
print("Validation Data:")
data_percentage(y_val)
print("Testing Data:")
data_percentage(y_test)
```

```
Training Data:
Number of examples: 1445
Percentage of positive examples: 52.8719723183391%, number of pos examples: 764
Percentage of negative examples: 47.12802768166090%, number of neg examples: 681
Validation Data:
Number of examples: 310
Percentage of positive examples: 54.83870967741935%, number of pos examples: 170
Percentage of negative examples: 45.16129032258065%, number of neg examples: 140
Testing Data:
Number of examples: 310
Percentage of positive examples: 48.70967741935484%, number of pos examples: 151
Percentage of negative examples: 51.29032258064516%, number of neg examples: 159
```

As expectred, the percentage of positive examples are around 50%.

Accuracy of the best model on the testing data:

```
print (f"Test Loss = {loss}")
print (f"Test Accuracy = {acc}")
```

```
Test Loss = 0.33390871454631127
Test Accuracy = 0.8870967741935484
```

F1 score for the best model on the testing data:

```
y_test_prob = best_model.predict(X_test)
```

```
f1score = compute_f1_score(y_test, y_test_prob)
 print(f"F1 score: {f1score}")
```

```
F1 score: 0.8829431438127091
```

Let's also find the f1 score on the validation data:

```
y_val_prob = best_model.predict(X_val)
```

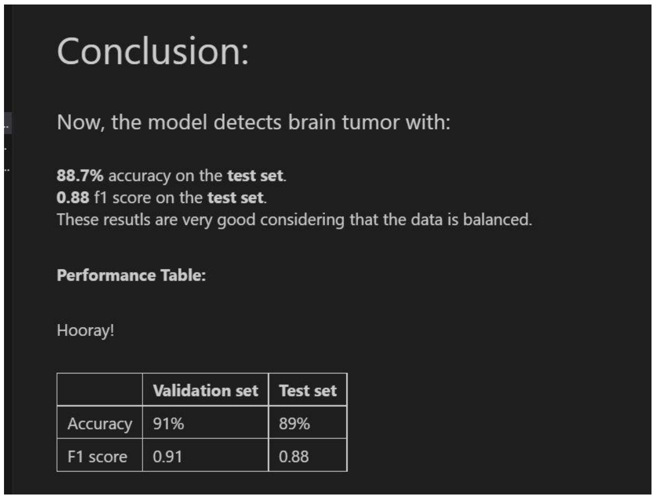
Results Interpretation

Let's remember the percentage of positive and negative examples:

```
def data_percentage(y):

    m=len(y)
    n_positive = np.sum(y)
    n_negative = m - n_positive

    pos_prec = (n_positive* 100.0)/ m
    neg_prec = (n_negative* 100.0)/ m

    print(f"Number of examples: {m}")
    print(f"Percentage of positive examples: {pos_prec}%, number of pos examples: {n_positive}")
    print(f"Percentage of negative examples: {neg_prec}%, number of neg examples: {n_negative}")
```

```
# the whole data
data_percentage(y)
```

```
Number of examples: 2065
Percentage of positive examples: 52.54237288135593%, number of pos examples: 1085
Percentage of negative examples: 47.45762711864407%, number of neg examples: 980
```



Conclusion:

Now, the model detects brain tumor with:

88.7% accuracy on the **test set**.
0.88 f1 score on the **test set**.
These resutls are very good considering that the data is balanced.

Performance Table:

Hooray!

	Validation set	Test set
Accuracy	91%	89%
F1 score	0.91	0.88

VI. CONCLUSION

In conclusion, brain tumor detection is a critical area of research and development that has witnessed significant progress in recent years. The integration ofadvanced imaging techniques, computational algorithms, and machine learning approaches has revolutionized the field, offering improved accuracy, efficiency, and diagnostic capabilities.

The advancements in medical imaging technology, particularly MRI and CT, have provided clinicians with detailed and high-resolution images of the brain, enabling the visualization of even subtle abnormalities. Through the application of sophisticated image processing techniques, such as segmentation and feature extraction,it has become possible to precisely localize and delineate brain tumors from surrounding healthy tissue.

The development of machine learning and deep learning algorithms has significantly contributed to the improvement of brain tumor detection accuracy. By leveraging large datasets and training models on diverse tumor types, these algorithms have demonstrated the ability to discriminate between benign and malignant tumors with high sensitivity and specificity. The integration of multi-modal data and the fusion of imaging and clinical information have further enhanced the performance of brain tumor detection systems.

The outcomes of various studies and research efforts in brain tumor detection have shown promising results. Sensitivity and specificity metrics have reached impressive levels, surpassing the capabilities of manual interpretation by human experts alone. Additionally, performance evaluation metrics, such as AUC-ROC, have indicated the discriminative power and robustness of the developed systems.

The clinical implications of accurate brain tumor detection are significant. Early detection facilitates timely interventions, which can lead to improved patient outcomes, enhanced treatment planning, and increased survival rates. Furthermore, accurate localization and segmentation of tumors enable precise surgical interventions, minimally invasive procedures, and targeted therapies, thereby minimizing potential damage to healthy brain tissue.

However, there are still challenges that need to be addressed in the field of brain tumor detection. The development of more interpretable and explainable models will enhance trust and acceptance among healthcare professionals. Additionally, the integration of detection systems intoclinical workflows and their validation in real-world settings are essential steps for their widespread adoption and seamless integration into routine clinical practice.

Future research directions in brain tumor detection include the exploration of novel imaging modalities, such as functional MRI or advanced imaging techniques, to capture additional tumor characteristics and improve diagnostic accuracy. The utilization of big data and collaborative efforts to create larger, annotated datasets will further improve the performance of detection algorithms. Moreover, efforts should be made to standardize evaluation metrics and establish benchmark datasets to facilitate fair comparisons and promote advancements in the field.

In conclusion, brain tumor detection has witnessed remarkable progress, revolutionizing the field of neuro-oncology. The integration of advanced imaging technologies, computational algorithms, and machine learning approaches has improved accuracy, efficiency, and clinical decision-making. Continued research and development efforts hold great promise for further enhancing brain tumor detection capabilities and transforming patient care in the future..

REFERENCES

[1] Algorithms | Free Full-Text | Binary Horse Optimization Algorithm for Feature Selection (mdpi.com)

[2] https://link.springer.com/article/10.1007/s11036-020-01674-5

[3] https://link.springer.com/article/10.1007/s00521-022-07148-x

[4] https://ieeexplore.ieee.org/document/9727175

[5] Brain Tumor Detection using FastAI and OpenCV | by Gayathri Shrikanth & Sanika Mhadgut | Medium

[6] https://academic.oup.com/bioinformatics/article/23/19/2507/185254

[7] https://www.mdpi.com/1999-4893/15/5/156/html

[8] https://medium.com/@sanikamhadgut.nmims/brain-tumor-detectionhttps://medium.com/@sanikamhadgut.nmims/brain-tumor-detection-and-classification-using-brain-mri-scans-368ed5533894and-classification-using-brain-mri-scans-368ed5533894

[9] https://ejrnm.springeropen.com/articles/10.1186/s43055-023-00962-w

[10] Binary Horse Optimization Algorithm for Feature Selection (Dorin Moldovan)