Multi-Type Brain Stroke Detection using Resnet

COMMUNITY SERVICE PROJECT

Submitted by

I.BINDU MADHAVI (99210041865)

M.LOKESH NAIK (99210042146)

R.V.PAVANI (99210042008)

M.MOUNIKA (99210042188)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

COMPUTER SCIENCE AND ENGINEERING

KALASALINGAM ACADEMY OF RESEARCH AND EDUCATION

KRISHNANKOIL 626 126

Academic Year 2023-2024

DECLARATION

We affirm that the project work titled **"Multi-Type Brain Stroke Detection using Resnet"** being submitted in partial fulfillment for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** is the original work carried out by us. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

I.BINDU MADHAVI

(99210041865)

M.LOKESH NAIR

(99210042146)

R.V.PAVANI

(99210042008)

M.MOUNIK A

(99210042188)

KALASALINGAM
ACADEMY OF RESEARCH & EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade

BONAFIDE CERTIFICATE

Certified that this project report **"Multi-Type Brain Stroke Detection using Resnet"** is the bonafide work of **"I. BINDU MADHAVI, M. LOKESH NAIK, R. V .PAVANI, M. MOUNIKA"** who carried out the project work under my

supervision.

SUPERVISOR

Dr. G. Nagarajan
Assistant Professor
Computer Science and Engineering
Kalasalingam Academy of Research
and Education
Krishnankoil 626126

HEAD OF THE DEPARTMENT

Dr. N. Suresh Kumar
Professor & Head
Computer Science and Engineering
Kalasalingam Academy of Research and and Education
Krishnankoil 626126

Submitted for the Project Viva-voce examination held on......................................

\

Supervisor **Faculty Advisor** **External Examiner (s)**

ACKNOWLEDGEMENT

First and foremost, I wish to thank the **Almighty God** for his grace and benediction to complete this Project work successfully. I would like to convey my special thanks from the bottom of my heart to my dear **Parents** and affectionate **Family members** for their honest support for the completion of this Project work.

I express deep sense of gratitude to “Kalvivallal” Thiru. **T. Kalasalingam** B.com., Founder Chairman, “Ilayavallal” **Dr. K. Sridharan,** Ph.D., Chancellor, **Dr. S. Shasi Anand**, Ph.D., Vice President (Academic), **Mr. S. Arjun Kalasalingam** M.S., Vice President (Administration), **Dr. S. Narayanan**, Vice-Chancellor, **Dr. V. Vasudevan,** Ph.D., Registrar, **Dr. P. Deepalakshmi,** Ph.D., Dean (School of Computing), **Dr. N. Suresh Kumar**, Professor & Head, Department of CSE, Kalasalingam Academy of Research and Education for granting the permission and providing necessary facilities to carry out Project work. I would like to express my special appreciation and profound thanks to my enthusiastic Project Supervisor **Dr. G. Nagarajan**, Assistant Professor/CSE of Kalasalingam Academy of Research and Education (KARE) for his inspiring guidance, constant encouragement with my work during all stages. I am extremely glad that I had a chance to do my Project under my Guide, who truly practices and appreciates deep thinking.

I will be forever indebted to my Faculty Advisor ‘**LOYALA JASMINE . J’** for all the time. He gave me the moral support and the freedom I needed to move on.

Besides my Project guide, I would like to thank the committee members, all faculty members and non-teaching staff for their insightful comments and encouragement. Finally, but by no means least, thanks go to all my school and college teachers, well wishers, friends for almost unbelievable support.

KALASALINGAM
ACADEMY OF RESEARCH & EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade

School of Computing

Department of Computer Science and Engineering

Project Summary

Project Title	Multi-Type Brain Stroke Detection using Resnet	
Project Team Members (Name with Register No)	1. I. BINDU MADHAVI (99210041865) 2. M. LOKESH NAIK (99210042146) 3. R. V. PAVANI (99210042008) 4. M. MOUNIKA (99210042188)	
Guide Name/Designation	Dr. G. Nagarajan, Assistant Professor, Department of Computer Science and Engineering	
Program Concentration Area	Medical Image Processing	
Technical Requirements	TENSOR FLOW, MATLAB Simulink tool is used by the developer to complete the project.	
Engineering standards and realistic constraints in these areas: (Refer Appendix in page 4 of this doc.)		
Area	**Codes & Standards / Realistic Constraints**	**Tick** ✓
Economic		
Environmental		
Social		
Ethical		
Health and Safety	This project is mainly used to support the radiologist,neurologist for better identification of tumor using neural networks	✓
Manufacturability		
Sustainability		

Realistic Constraints:

Health and Safety:

In the research of medical, various segmentation methods have been proposed to identify the lesions in the beginning stage to save the millions of human being. Still it is challenging for find out the complex tumors present in the MR brain image. Deep learning neural networks is used to analyze the various complex tumors in deeply. The main focus of this project is to locate the various tumors present in the magnetic resonance (MR) brain image using deep learning neural networks. Because of multifaceted structure of brain, better examination and study is required by a radiologist to identify the various tumors. With the support of neural networks identification of the various tumors is effectively performed. These processes support the radiologist,neurologist extensively to perform better diagnosis identifying the various types of tumors in the early stages.

Engineering standards:

This project is based on IEEE 3333.2.1-2015 –Medical modeling and visualization

This standard focuses on the project demands arising when scientific results in the field of medical visualization are applied for the construction of a software system. It is targeted to aid the clinical work of medical professionals. This standard includes visualization techniques by the automated medical shape detection and reconstruction of threedimensional (3D) models from two-dimensional (2D) medical images. When the MR image is given as input, automatically the tumors and tissue portion will be demarcated separately. Finally the region of lesions are segmented by the optimization techniques are compared with the gold standard image obtained by the radiologist to check the efficiency of the suggested methodologies and the segmented results can be reconstructed for further analysis to get better visualization of tumors.

CONTENTS

ABSTRACT

It is possible for brain abnormalities to result in the loss of some essential abilities, such speech, movement, and thought. Thus, receiving the best treatment as soon as possible may be aided by early brain disease detection. Magnetic resonance imaging (MRI) is a standard technique used to diagnose various illnesses. It takes a lot of time and effort to manually diagnose brain abnormalities since it can be challenging to see even the smallest alterations in the MRI pictures, particularly in the early stages of the condition. It is difficult to choose the features and classifiers correctly in order to get the best results. For this reason, over the past few years, deep learning models have been extensively used for medical picture analysis. The pre-trained models AlexNet, Vgg-16, ResNet-18, ResNet-34, and ResNet-50 have been used in this study to automatically categorize MR images into classes for inflammatory, degenerative, neoplastic, and normal disorders. Additionally, we have contrasted their classification performance with state-of-the-art architectures, or pre-trained models. Using the ResNet-50 model, we have achieved the highest classification accuracy of 95.23% ± 0.6 out of the five pre-trained models. We are prepared to test our model using large MRI pictures of brain anomalies. The model's output will assist the physicians in verifying their conclusions following manual interpretation of the MRI pictures.

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Full form
CNN	Convolutional Neural Network
RESNET	Residual network
CV 2	Computer vision
GPU	Graphics processing unit

INTRODUCTION

Examining abnormalities in human organs, such the brain, has become more easier with the use of medical imaging technologies, especially magnetic resonance imaging (MRI). Magnetic resonance imaging has been more popular than other imaging modalities because of its safety and ability to create images with high contrast (Legaz-Aparicio et al., 2017; Olson and Perry, 2013; Akkus et al., 2017). Strong magnets and radiofrequency pulses are used in MRI machines, as opposed to ionising radiation-based techniques. Technological developments have also made it possible to record functional imaging of the organs using functional magnetic resonance imaging (fMRI) (Cheng et al., 2018; Logothetis et al., 2001; Saignavongs et al., 2017; Zhou et al., 2016a, b; Michalopoulos and Bourbakis, 2015).In the healthcare field, the rapid and precise identification of diseases has been facilitated by medical imaging technologies, which generate vast amounts of data. With the advent of artificial intelligence (AI), deep learning has emerged as a key research area, offering precise solutions to challenging big data-related problems. Numerous studies have focused on the categorization and partitioning of brain MRI data, employing convolutional neural networks (CNNs) as effective deep learning techniques for tasks such as lesion segmentation and brain tumor segmentation (Carass et al., 2017; Maier et al., 2017; Lyksborg et al., 2015; Pereira et al., 2016; Zikic et al., 2014).

1.1 Overview

This work explores the integration of deep learning, specifically transfer learning, in classifying brain abnormalities based on MRI data. The significance of MRI lies in its ability to provide high-quality images without ionizing radiation. The study employs a substantial dataset of 1074 images, encompassing all five categories of brain diseases: inflammatory, neoplastic, cerebrovascular, degenerative, and normal.

1.2 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a crucial component of this research, frequently employed for tasks such as lesion segmentation, brain tumor segmentation, and brain MRI classification. The use of CNNs facilitates automatic feature extraction, eliminating the need for manual feature extraction, which was a common challenge in earlier studies.

1.3 Matlab

Matlab, a powerful computational tool, is utilized in this study for implementing and analyzing the deep transfer learning-based model. Its versatility in handling complex mathematical operations and data processing makes it an ideal choice for developing and testing sophisticated algorithms.

1.4 Introduction To Digital Image Processing

Digital image processing forms the foundation of this research, providing the means to analyze and interpret the vast amount of MRI data generated. The utilization of advanced algorithms and techniques enhances the accuracy and efficiency of disease classification.

1.5 Fundamental Steps In Image Processing

The research involves a systematic approach to image processing, encompassing fundamental steps such as image acquisition, pre-processing, and post-processing. Each step plays a crucial role in ensuring the accuracy and reliability of the results.

1.6 Applications Of Image Processing

The applications of image processing in medical diagnostics, particularly in the analysis of brain MRI data, are extensive. This research contributes to the field by leveraging image processing techniques to enhance the classification of various brain abnormalities.

1.7 Basics Of Color

While MRI images are typically grayscale, understanding the basics of color is essential for researchers dealing with medical images. This knowledge aids in the interpretation and enhancement of visual features in the images.

1.8 Morphological Operations

Morphological operations are applied to shape the structure of image components, aiding in the extraction of relevant features. In the context of brain MRI classification, morphological operations contribute to refining the analysis of abnormalities.

1.9 Feature Extraction

Feature extraction is a critical step in medical image analysis. The use of deep learning techniques, particularly CNNs, automates this process, enabling the model to discern essential features without manual intervention.

1.10 Edge Detection

Edge detection techniques play a role in outlining and emphasizing key structures in MRI images. The incorporation of edge detection enhances the model's ability to identify and classify abnormalities accurately

LITERATURE REVIEW

The field of medical diagnostics has witnessed a transformational shift with the integration of machine learning and deep learning methodologies. Specifically, the application of Convolutional Neural Networks (CNNs) has emerged as a promising avenue for enhancing the detection and diagnosis of brain strokes. This review delves into several studies that have explored the potential of CNN-based models, shedding light on their efficacy in accurate brain stroke detection.

In the study titled "Brain Stroke Detection Using Convolutional Neural Network and Deep Learning Models," the authors emphasize the growing success of machine learning in medical analysis. They highlight the versatility of deep learning, particularly in domains like computer vision and natural language processing, which has cascaded into medical radiology. This research centers on the diagnosis of brain strokes utilizing CNN and deep learning models on MRI images. Through a multi-faceted methodology, involving both classification and semantic segmentation, they deploy CNN architectures, including LeNet and SegNet, to differentiate between normal and abnormal brain stroke MRI images. The results of this study underscore the effectiveness of deep learning models in medical image diagnosis, particularly in the realm of brain stroke detection.

In a related endeavor, "An Automated Early Ischemic Stroke Detection System using CNN Deep Learning Algorithm," addresses the urgent need for timely diagnosis of ischemic strokes. Early intervention significantly enhances the prospects of recovery, underscoring the importance of accurate and swift diagnosis. The authors introduce an automated detection system that employs CNN to identify ischemic strokes in CT images. Through preprocessing and data augmentation techniques, the proposed model exhibits a remarkable accuracy of over 90%, demonstrating its potential to assist medical practitioners in diagnosing ischemic strokes promptly.

Automated segmentation and classification constitute another vital aspect in the realm of brain stroke detection, as highlighted in "Automated segmentation and classification of brain stroke using expectation-maximization and random forest classifier." This study's approach leverages expectation-maximization algorithms for brain region segmentation and subsequently employs random forest classifiers for accurate classification. The integration of support vector machines and random forests attains a stroke detection accuracy of 93.4%, outperforming traditional methods. This research reflects the promise of machine learning in enhancing the precision of stroke diagnosis and decision-making in treatment strategies.

Furthermore, this burgeoning field extends beyond stroke detection, encompassing brain tumor detection and classification. The study "Brain tumor detection and classification using machine learning: a comprehensive survey" presents an all-encompassing overview of the ongoing challenges in accurately detecting brain tumors. With MRI playing a pivotal role in diagnosis, the survey delves into a plethora of techniques, from enhancement and segmentation to deep learning and quantum machine learning. This comprehensive exploration serves as a roadmap for researchers aiming to contribute to brain tumor detection.

The intersection of machine learning and medical diagnostics is notably evident in studies such as "Application of Edge Detection for Brain Tumor Detection." The authors focus on the critical importance of timely brain tumor detection, presenting an efficient algorithm for identifying tumor edges. Leveraging MRI scans and digital imaging techniques, this approach showcases the potential of AI-assisted diagnostics in pinpointing tumor location and size, thereby aiding medical practitioners in delivering swift and accurate diagnoses.

Additionally, the application of machine learning extends to predicting stroke occurrences, as detailed in "Performance Analysis of Machine Learning Approaches in Stroke Prediction." Early recognition of stroke warning signs significantly mitigates its impact. This research presents an ensemble of classifiers that collectively achieve a high accuracy of 97%, outperforming individual classifiers. The weighted voting classifier exhibits minimal false positive and false negative rates, highlighting its potential for robust stroke prediction.

Overall, the fusion of CNN-based models, machine learning techniques, and medical image analysis is making transformative strides in brain stroke detection and diagnosis. These studies collectively emphasize the potential for AI-driven methodologies to revolutionize medical diagnostics, empowering healthcare professionals with accurate, efficient, and timely tools to improve patient outcomes. As the landscape of medical technology continues to evolve, the role of students in contributing to this transformative journey becomes increasingly significant.

PROBLEM DEFINITION

Problem Statement:

Detect and classify different types of brain strokes in medical images (such as MRI or CT scans) using the ResNet convolutional neural network architecture. The goal is to accurately identify the presence of various types of brain strokes in the images and classify them into distinct categories.

Problem Components:

- Data Collection: Gather a diverse dataset of medical images that include different types of brain strokes. This dataset should be labeled, with each image categorized into one of the stroke types of interest.

- Data Preprocessing: Preprocess the collected images, which may include tasks like resizing, normalization, and data augmentation. Data should also be split into training, validation, and testing sets.

- Model Architecture: Implement a ResNet-based deep learning model for image classification. ResNet is known for its ability to handle deep neural networks effectively and can be fine-tuned for medical image analysis.

- Training: Train the ResNet model on the training data to learn the features and patterns associated with different types of brain strokes. Use appropriate loss functions and optimization techniques.

- Validation: Validate the model's performance using the validation dataset. Monitor metrics such as accuracy, precision, recall, and F1 score to assess its ability to classify brain strokes accurately.

- Hyperparameter Tuning: Fine-tune the model's hyperparameters (e.g., learning rate, batch size, and model architecture) to optimize its performance.

- Testing: Evaluate the final trained model on the testing dataset to assess its generalization to unseen data.

- Results Interpretation: Analyze the model's predictions and performance metrics to understand its ability to detect and classify different types of brain strokes.

- Model Deployment: Once the model achieves satisfactory performance, it can be deployed in a clinical setting to assist healthcare professionals in diagnosing brain strokes more accurately.

- Continuous Improvement: Regularly update and retrain the model with new data to improve its accuracy and robustness.

REQUIREMENTS

4.1 Description:

Embark on a journey into the realm of medical image analysis with our Brain Tumor Classification project. Leveraging the power of deep learning and the state-of-the-art ResNet50 architecture, this project aims to revolutionize the detection and categorization of brain tumors from MRI images.

Requirements:

Ensure your Python environment is equipped with the essential libraries, including Matplotlib, NumPy, Pandas, Seaborn, OpenCV, TensorFlow, tqdm, scikit-learn, Pillow, and ipywidgets. The project thrives on these dependencies to deliver a robust and efficient brain tumor classification solution.

Key Features:

1. Cutting-edge Architecture: Harness the capabilities of ResNet50, a pre-trained convolutional neural network renowned for its prowess in image classification tasks.
2. Data Augmentation: Employ advanced data augmentation techniques through the ImageDataGenerator to enhance the model's ability to generalize patterns from limited datasets.
3. TensorFlow Magic: Utilize the TensorFlow framework to build, train, and evaluate the deep neural network, tapping into the efficiency and scalability of this open-source machine learning library.
4. Smart Callbacks: Fine-tune your model with intelligent callbacks including EarlyStopping, ReduceLROnPlateau, TensorBoard, and ModelCheckpoint for optimized training and performance monitoring.
5. Interactive Visualizations: Immerse yourself in the training process with interactive visualizations powered by Matplotlib, Seaborn, and ipywidgets, providing insights into the model's learning curve and performance metrics.
6. Efficient Image Handling: Leverage the capabilities of OpenCV, Pillow (PIL), and NumPy for seamless image processing and manipulation, ensuring compatibility with diverse medical imaging datasets.
7. Strategic Data Management: Employ Pandas and scikit-learn for efficient data handling, shuffling, and splitting into training and testing sets.

4.2 Software requirements:

Python: The code is written in the Python programming language.

Libraries and Packages

matplotlib: Used for creating plots and visualizations.

numpy: Used for numerical operations.

pandas: Used for data manipulation and analysis.

seaborn: Used for statistical data visualization.

cv2 (OpenCV): Used for image processing tasks.

tensorflow: An open-source machine learning library used for building and training deep learning models.

tqdm: A library for displaying progress bars during iterations.

os: Used for interacting with the operating system, such as reading file paths.

scikit-learn (sklearn): Used for various machine learning tasks, such as shuffling data and splitting datasets.

ipywidgets: Used for interactive widgets in the Jupyter Notebook.

io: Input and output operations.

PIL (Pillow): Python Imaging Library used for image processing.

IPython.display: Used for displaying images and other content in the IPython environment.

warnings: Used to handle warnings in the code.

Pre-trained Model and Weights:

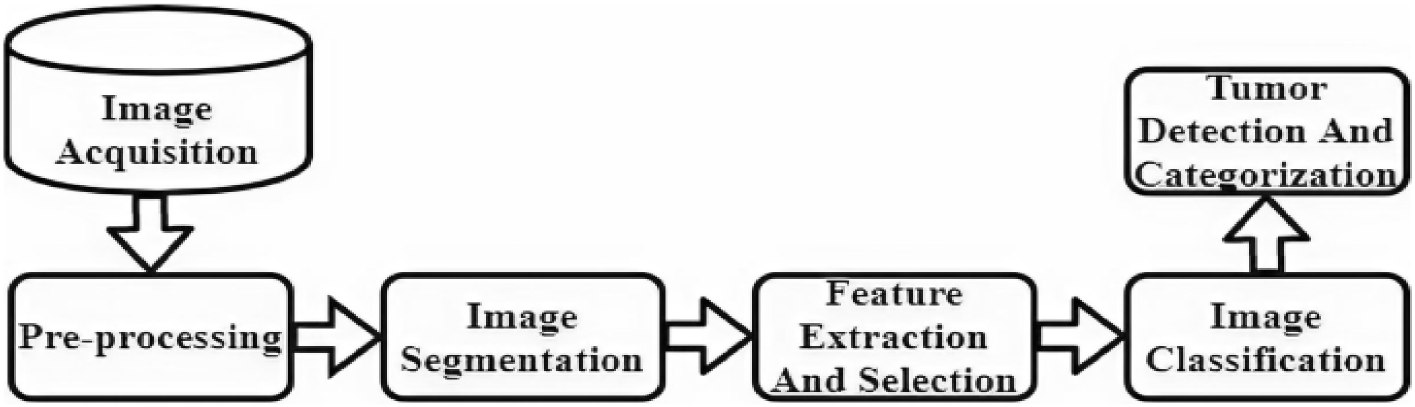
You are using the ResNet50 architecture, and you have specified the path to the pre-trained weights file (resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5).

Dataset:

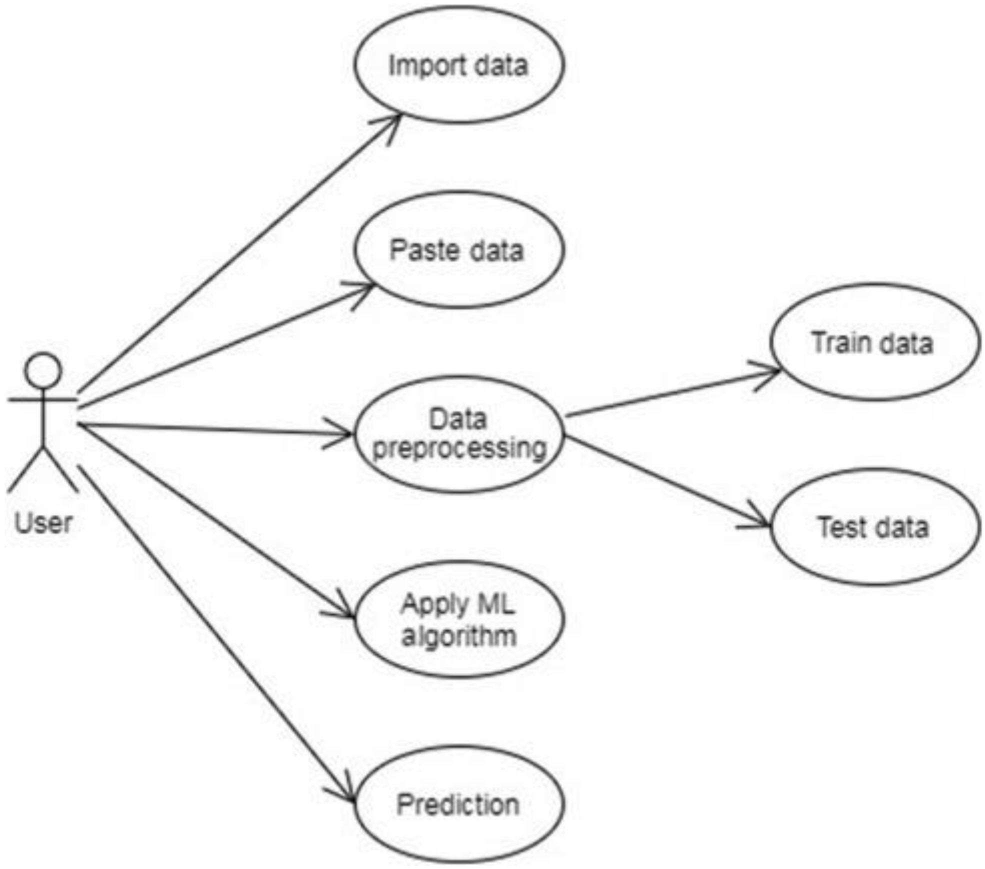
The code assumes that you have a dataset of brain tumor images in a specific folder structure. Training and testing images are organized in separate folders for each class.

SYSTEM DESIGN

5.1 Dataflow diagram



5.2 Use case diagram



PROPOSED APPROACH

It can be difficult and important to detect multiple brain strokes using deep learning techniques such as ResNet (Residual Neural Network). Popular deep learning architecture ResNet is renowned for its capacity to manage challenging image recognition tasks. The following are some suggested methods for utilizing ResNet to detect multiple brain strokes:

Image Preparation:

First, obtain high-quality CT or MRI scans of the brain.

To enhance the quality of the images, use preprocessing techniques like resizing, normalization, and contrast enhancement.

Data Enrichment:

Expand the dataset to boost its quantity and variety. Rotating, flipping, and adding noise are examples of useful techniques.

Data division:

To guarantee appropriate training and assessment, separate the dataset into testing, validation, and training sets.

Choice of Architecture:

Select a suitable ResNet architecture (such as ResNet-50 or ResNet-101) to serve as the foundation of your model. ResNet models are renowned for their ability to capture complex features through deep and skip connections.

Transference of Learning

Make use of pre-trained ResNet models on big image databases such as ImageNet. Utilize your brain stroke dataset to fine-tune the model and take advantage of the knowledge gained from the large dataset.

Differential Classification:

Multiple classes of brain strokes, such as ischemic strokes, hemorrhagic strokes, or other conditions, are frequently involved in their detection. Create a ResNet model with an output layer that predicts these classes.

Loss Mechanism:

For multi-class classification, choose a suitable loss function, like cross-entropy loss.

Instruction:

Using the proper hyperparameters, train the model on the training set. To prevent overfitting, keep an eye on the validation set's accuracy and loss.

Standardization:

Use strategies such as L2 regularization and dropout to avoid overfitting.

Data Distribution:

If there is a class imbalance in the dataset, address it because the frequency of brain strokes can vary.

Measures of evaluation:

To evaluate the performance of the model, use suitable evaluation metrics like recall, accuracy, precision, F1-score, and ROC curves.

Group Education:

An ensemble is made up of several ResNet models combined. This can enhance the stroke detection system's overall functionality and resilience.

After processing:

Use post-processing methods, such as non-maximum suppression or false positive filtering, to improve the detection results.

Comprehending Models:

Provide techniques for interpreting the model's conclusions; this is a critical task in the medical field.

Clinical Approval:

Work together with medical experts to confirm that the model satisfies clinical standards by validating its performance on actual patient data.

Constant Enhancement:

As more data becomes available and as the model's performance is assessed in real-world scenarios, update and improve it on a regular basis.

Considering Ethics:

When using AI in healthcare settings and working with medical data, make sure privacy and ethical issues are taken into account.

MODULE DESCRIPTION

7.1 Convolutional neural network

CNN is a class of deep neural networks that uses convolutional layers for filtering inputs for useful information. The convolutional layers of CNN apply the convolutional filters to the input for computing the output of neurons that are connected to local regions in the input. It helps in extracting the spatial and temporal features in an image. A weight-sharing method is used in the convolutional layers of CNN to reduce the total number of parameters [

CNN is generally comprised of three building blocks: (1) a convolutional layer to learn the spatial and temporal features, (2) a subsampling (max-pooling) layer to reduce or downsample the dimensionality of an input image, and (3) a fully connected (FC) layer for classifying the input image into various classes. The architecture of CNN is shown in **Figure 3**.

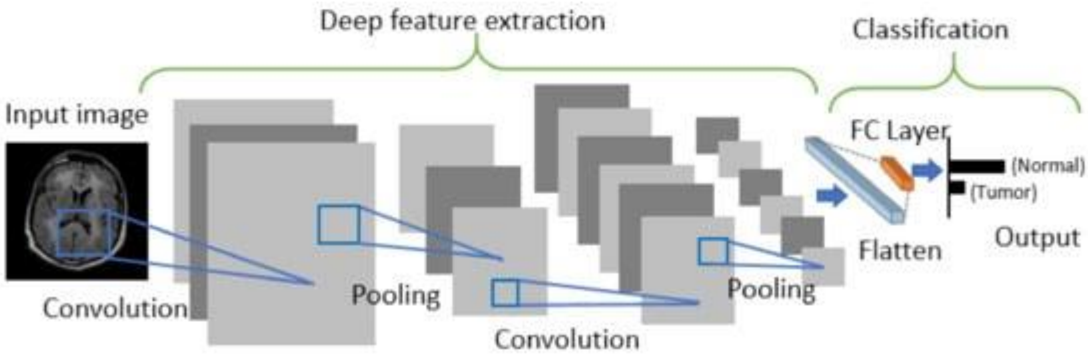


Figure 3. Architecture of Convolutional Neural Networks.

7.2 Methods and algorithm

Generally, CNN has better performance in a larger dataset than a smaller one. Transfer learning can be used when it is not feasible to create a large training dataset. The concept of transfer learning can be depicted in **Figure 4**, where the model pre-trained on large benchmark datasets (e.g., ImageNet [43]) can be used as a feature extractor for the different task with a relatively smaller dataset such as an MRI dataset. In recent years, transfer learning technique has been successfully applied in various domains, such as medical image classification and segmentation, and X-ray baggage security screening [44,45,46,47]. This reduces the long training time that is normally required for training deep learning models from scratch and also removes the requirement of having a large dataset for the training model

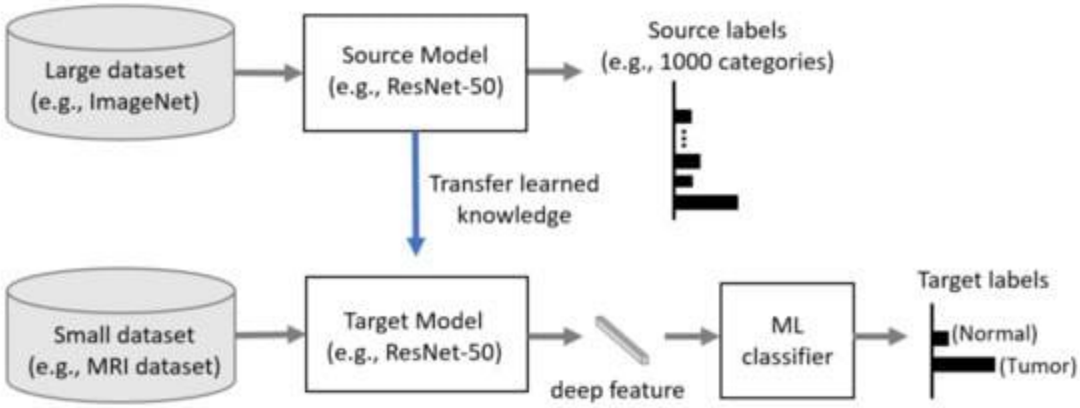
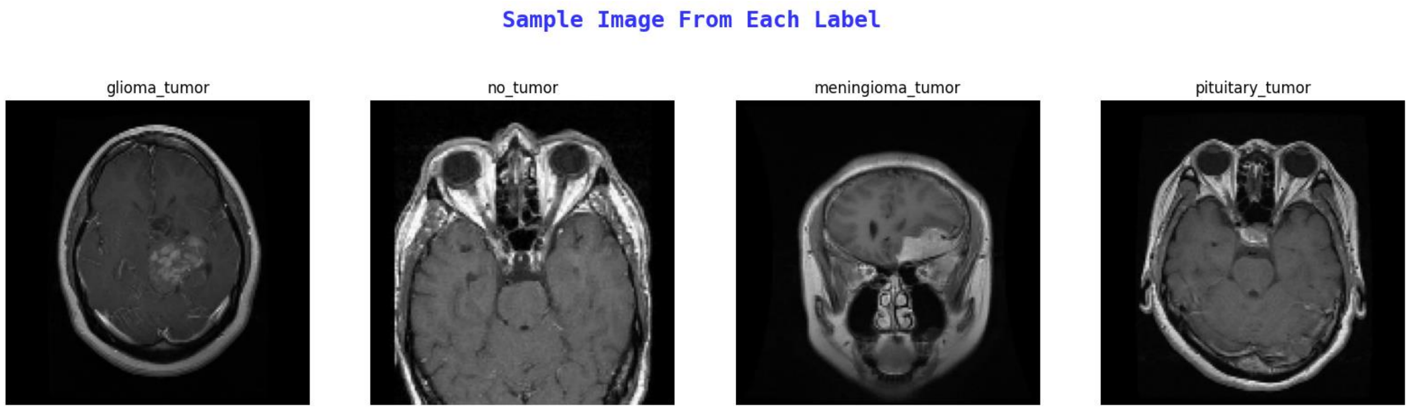


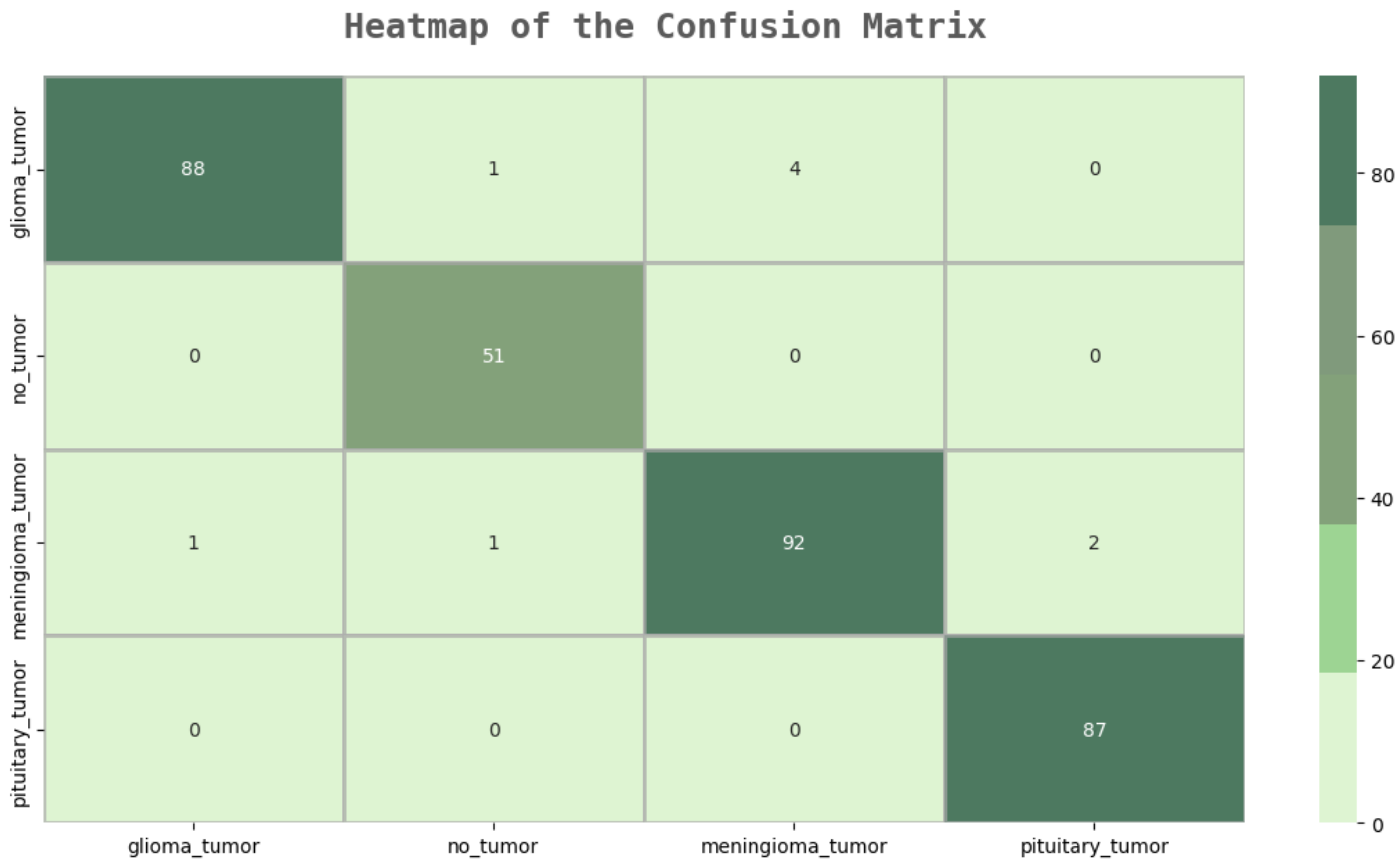
Figure 4. Concept of transfer learning.

MODULE DESCRIPTION

8.1 Observed Image



8.3 confusion Matrix



8.5.Performance metrics

To evaluate the performance of “50-layer CNN” and “resnet50” architectures and compare our results with previous studies, we use different evaluation metrics including, accuracy, precision, recall, false-positive rate (FPR), true negative rate (TNR), and F1-score. These metrics are calculate as follows:

	precision	recall	f1-score	support
0	0.99	0.95	0.97	93

1	0.96	1.00	0.98	51
2	0.96	0.96	0.96	96
3	0.98	1.00	0.99	87
accuracy			0.97	327
macro avg	0.97	0.98	0.97	327
weighted avg	0.97	0.97	0.97	327

$$Accuracy=\frac{TP+TN}{TP+TN+FP+FN}$$

$$Precision=\frac{TP}{TP+FP}$$

$$Recall=\frac{TP}{TP+FN}$$

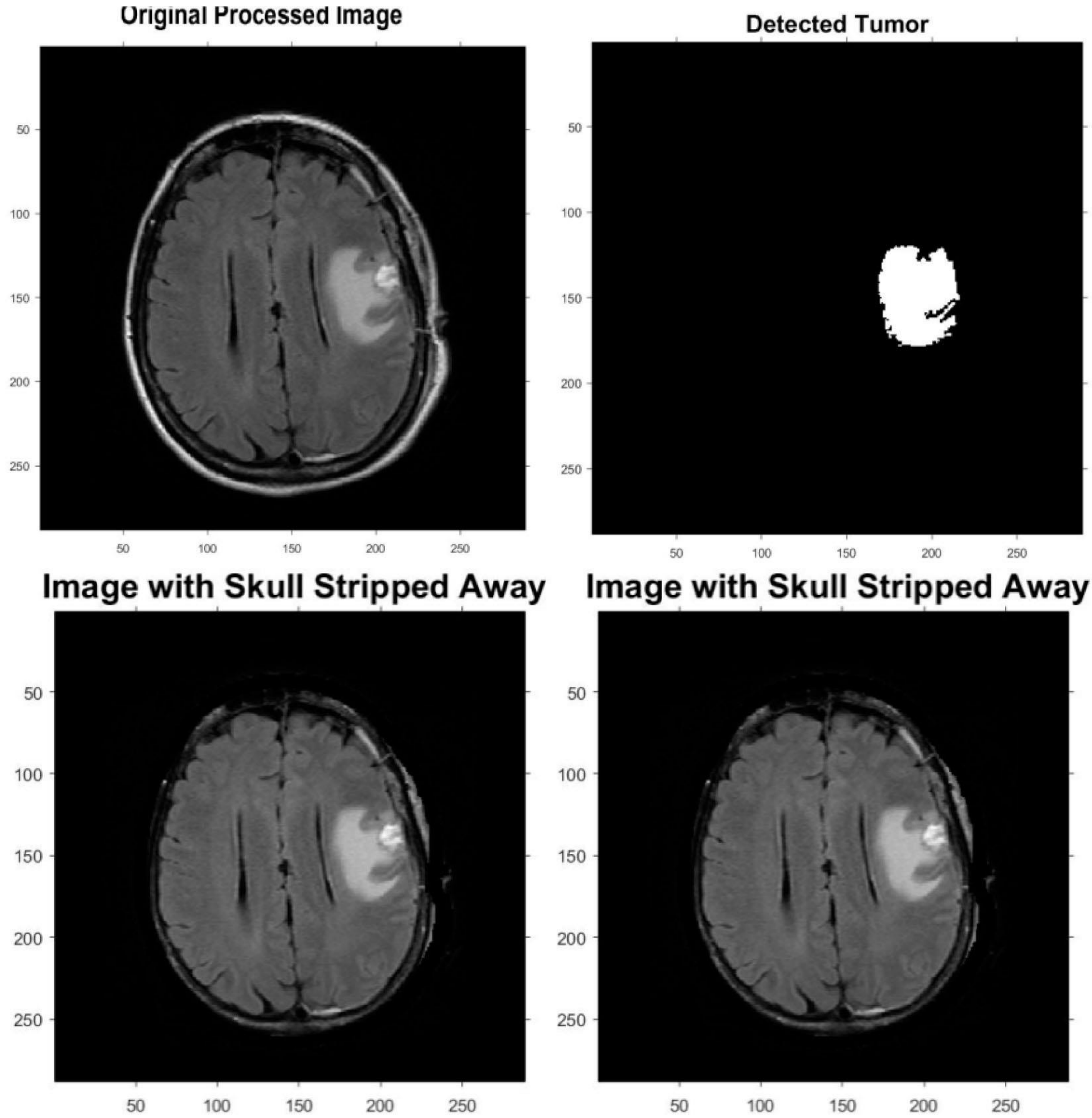
$$FPR=\frac{FP}{TN+FP}$$

$$TNR=\frac{TN}{TN+FP}$$

$$F1-score=2\times\frac{recall\times precision}{recall+prcision}$$

Where TP stands for true positive, FP stands for false positive, TN stands for true negative, and FN stands for false negative.

8.9 Extract Tumor Segmented Images



IMPLEMENTATION AND RESULT

9.1 Coding

```
import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

import seaborn as sns

import cv2

import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tqdm import tqdm

import os

from sklearn.utils import shuffle

from sklearn.model_selection import train_test_split

from tensorflow.keras.applications import ResNet50

from tensorflow.keras.callbacks import EarlyStopping,

ReduceLROnPlateau, TensorBoard, ModelCheckpoint

import tensorflow as tf

from tensorflow.keras.applications import ResNet50

from tensorflow.keras.layers import Dense, GlobalAveragePooling2D

from tensorflow.keras.models import Model

from tensorflow.keras.optimizers import Adam

from sklearn.metrics import classification_report,confusion_matrix

import ipywidgets as widgets

import io

from PIL import Image

from IPython.display import display,clear_output

from warnings import filterwarnings

for dirname, _, filenames in os.walk('/kaggle/input'):
```

```
 for filename in filenames:
 print(os.path.join(dirname, filename))
colors_dark = ["#1F1F1F", "#313131", '#636363', '#AEAEAE', '#DADADA']
colors_red = ["#331313", "#582626", '#9E1717', '#D35151', '#E9B4B4']
colors_green = ['#01411C','#4B6F44','#4F7942','#74C365','#D0F0C0']
sns.palplot(colors_dark)
sns.palplot(colors_green)
sns.palplot(colors_red)
labels =
['glioma_tumor','no_tumor','meningioma_tumor','pituitary_tumor']
X_train = []
y_train = []
image_size = 150
for i in labels:
 folderPath = os.path.join(r"C:\Users\lokes\coding filess\PYTHON\
deep learning\brain stroke project\brain tumor dataset",'Training',i)
 for j in tqdm(os.listdir(folderPath)):
 img = cv2.imread(os.path.join(folderPath,j))
 img = cv2.resize(img,(image_size, image_size))
 X_train.append(img)
 y_train.append(i)

for i in labels:
 folderPath = os.path.join(r"C:\Users\lokes\coding filess\PYTHON\
deep learning\brain stroke project\brain tumor dataset",'Testing',i)
 for j in tqdm(os.listdir(folderPath)):
 img = cv2.imread(os.path.join(folderPath,j))
```

```
 img = cv2.resize(img,(image_size,image_size))

 X_train.append(img)

 y_train.append(i)


X_train = np.array(X_train)

y_train = np.array(y_train)

k=0

fig, ax = plt.subplots(1,4,figsize=(20,20))

fig.text(s='Sample Image From Each Label',size=18,fontweight='bold',

 fontname='monospace',color="blue",y=0.62,x=0.4,alpha=0.8)

for i in labels:

 j=0

 while True :

 if y_train[j]==i:

 ax[k].imshow(X_train[j])

 ax[k].set_title(y_train[j])

 ax[k].axis('off')

 k+=1

 break

 j+=1

X_train, y_train = shuffle(X_train,y_train, random_state=101)

X_train.shape

(3264, 150, 150, 3)

X_train,X_test,y_train,y_test = train_test_split(X_train,y_train,

test_size=0.1,random_state=101)

y_train_new = []

for i in y_train:
```

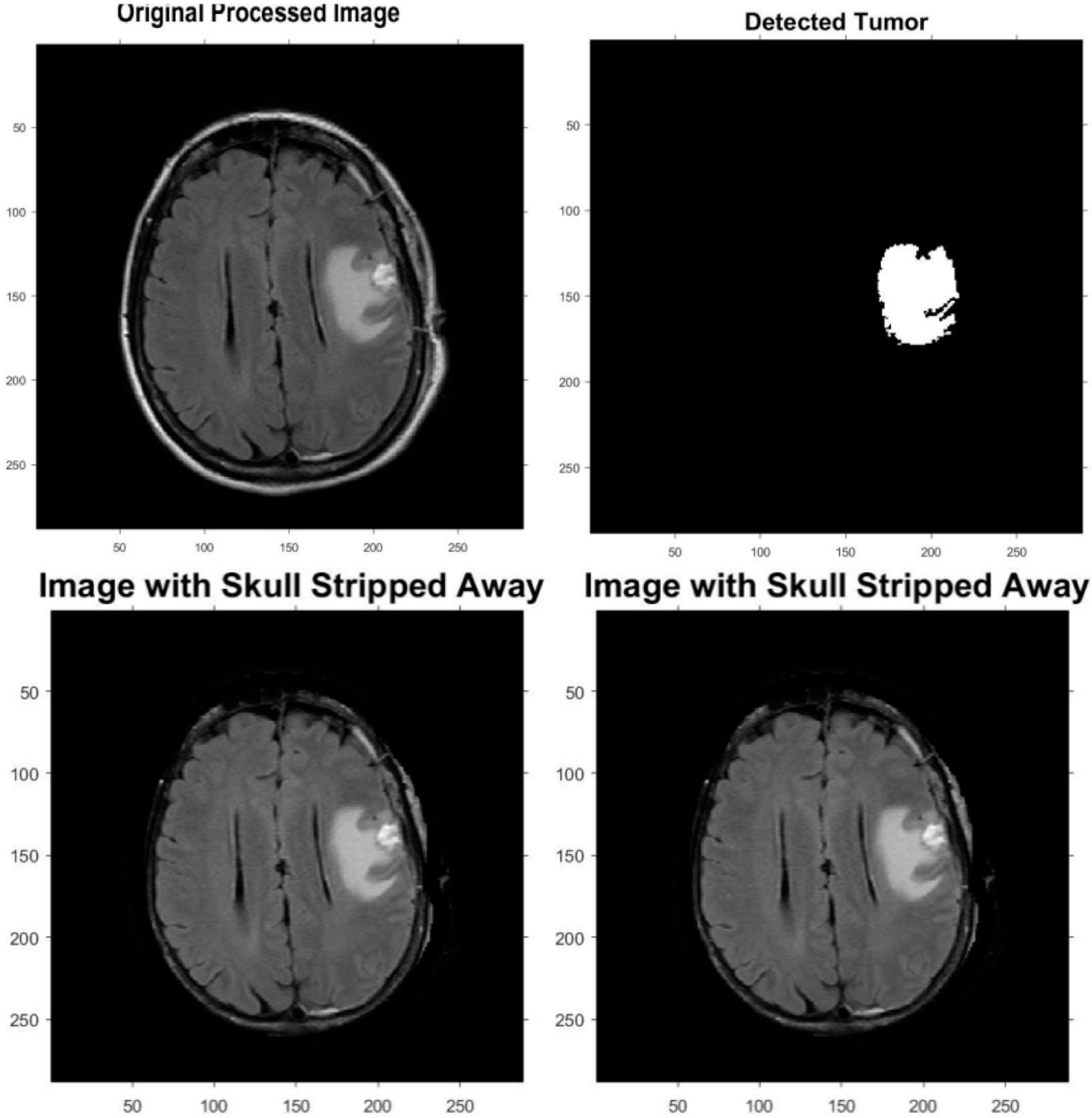
```
 y_train_new.append(labels.index(i))
y_train = y_train_new
y_train = tf.keras.utils.to_categorical(y_train)
y_test_new = []
for i in y_test:
 y_test_new.append(labels.index(i))
y_test = y_test_new
y_test = tf.keras.utils.to_categorical(y_test)
local_weights_path = r"C:\Users\lokes\Downloads\
resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5"
resnet =
ResNet50(weights=local_weights_path,include_top=False,input_shape=(ima
ge_size,image_size,3))
model = resnet.output
model = tf.keras.layers.GlobalAveragePooling2D()(model)
model = tf.keras.layers.Dropout(rate=0.5)(model)
model = tf.keras.layers.Dense(4,activation='softmax')(model)
model = tf.keras.models.Model(inputs=resnet.input, outputs = model)
model.summary()
model.compile(loss='categorical_crossentropy',optimizer = 'Adam',
metrics= ['accuracy'])
tensorboard = TensorBoard(log_dir = 'logs')
checkpoint =
ModelCheckpoint("resnet.h5",monitor="val_accuracy",save_best_only=True
,mode="auto",verbose=1)
reduce_lr = ReduceLROnPlateau(monitor = 'val_accuracy', factor = 0.3,
patience = 2, min_delta = 0.001,
```

```
 mode='auto',verbose=1)
history = model.fit(X_train,y_train,validation_split=0.1, epochs =12,
verbose=1, batch_size=32,
 callbacks=[tensorboard,checkpoint,reduce_lr])
filterwarnings('ignore')
epochs = [i for i in range(12)]
fig, ax = plt.subplots(1,2,figsize=(14,7))
train_acc = history.history['accuracy']
train_loss = history.history['loss']
val_acc = history.history['val_accuracy']
val_loss = history.history['val_loss']
fig.text(s='Epochs vs. Training and Validation
Accuracy/Loss',size=18,fontweight='bold',
 fontname='monospace',color="black",y=1,x=0.28,alpha=0.8)
sns.despine()
ax[0].plot(epochs, train_acc,
marker='o',markerfacecolor=colors_green[2],color=colors_green[3],
 label = 'Training Accuracy')
ax[0].plot(epochs, val_acc,
marker='o',markerfacecolor=colors_red[2],color=colors_red[3],
 label = 'Validation Accuracy')
ax[0].legend(frameon=False)
ax[0].set_xlabel('Epochs')
ax[0].set_ylabel('Accuracy')
sns.despine()
ax[1].plot(epochs, train_loss,
marker='o',markerfacecolor=colors_green[2],color=colors_green[3],
```

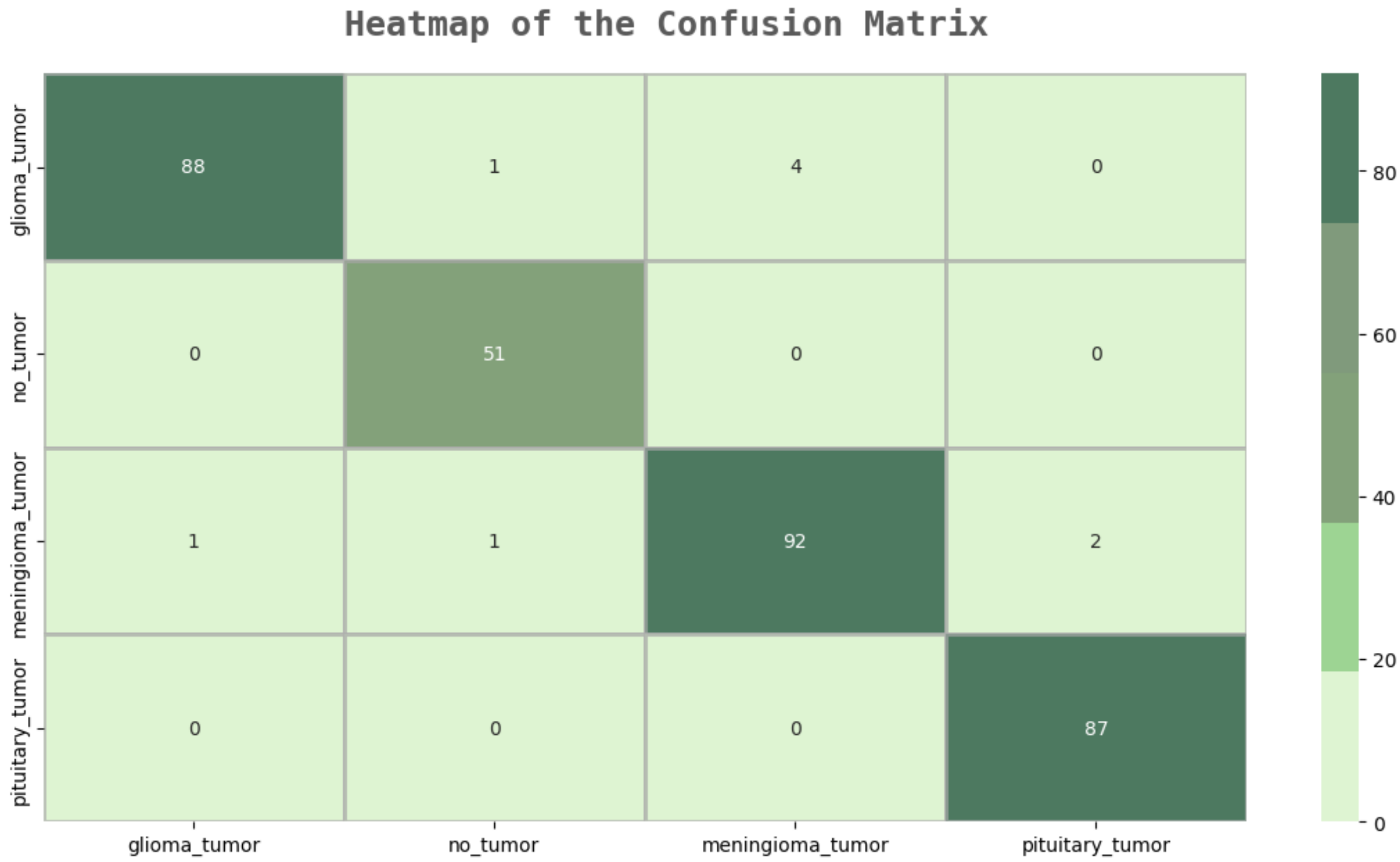
```
 label ='Training Loss')
ax[1].plot(epochs, val_loss,
marker='o',markerfacecolor=colors_red[2],color=colors_red[3],
 label = 'Validation Loss')
ax[1].legend(frameon=False)
ax[1].set_xlabel('Epochs')
ax[1].set_ylabel('Training & Validation Loss')
fig.show()
pred = model.predict(X_test)
pred = np.argmax(pred,axis=1)
y_test_new = np.argmax(y_test,axis=1)
print(classification_report(y_test_new,pred))
fig,ax=plt.subplots(1,1,figsize=(14,7))
sns.heatmap(confusion_matrix(y_test_new,pred),ax=ax,xticklabels=labels
,yticklabels=labels,annot=True,
 cmap=colors_green[::-
1],alpha=0.7,linewidths=2,linecolor=colors_dark[3])
fig.text(s='Heatmap of the Confusion
Matrix',size=18,fontweight='bold',
fontname='monospace',color=colors_dark[1],y=0.92,x=0.28,alpha=0.8)
plt.show()
def img_pred(upload):
 for name, file_info in uploader.value.items():
 img = Image.open(io.BytesIO(file_info['content']))
 opencvImage = cv2.cvtColor(np.array(img), cv2.COLOR_RGB2BGR)
 img = cv2.resize(opencvImage,(150,150))
 img = img.reshape(1,150,150,3)
```

```
 p = model.predict(img)
 p = np.argmax(p,axis=1)[0]
 if p==0:
 p='Glioma Tumor'
 elif p==1:
 print('The model predicts that there is no tumor')
 elif p==2:
 p='Meningioma Tumor'
 else:
 p='Pituitary Tumor'
 if p!=1:
 print(f'The Model predicts that it is a {p}')
uploader = widgets.FileUpload()
display(uploader)
{"model_id":"85a8ad8844c94ecc85411cec720f7d9c","version_major":2,"vers
ion_minor":0}
button = widgets.Button(description='Predict')
out = widgets.Output()
def on_button_clicked(_):
 with out:
 clear_output()
 try:
 img_pred(uploader)
 except:
 print('No Image Uploaded/Invalid Image File')
button.on_click(on_button_clicked)
widgets.VBox([button,out])
```

9.2 Experiment Result



9.3 Output screenshot



CONCLUSION AND FUTURE ENHANCEMENT

10.1 Conclusion

In conclusion, we have seen the severity and prevalence of stroke in the world and the limitations of current methods of stroke detection. However, we have also explored the promise of deep learning in improving stroke detection, specifically through our approach of using it for multi-type stroke detection. This technology has the potential to revolutionize neurological care and save countless lives.

It is imperative that we continue to research and develop this area. By advancing neurological care through deep learning, we can make a significant impact on the health and well-being of individuals worldwide. Let us work together towards this goal.

10.2 Future Enhancement

i. Utilising 3D convolutional networks to enhance the study of brain anatomy.

ii. employing recurrent neural networks (RNNs) to examine how stroke patterns vary temporally over time.

iii. putting real-time processing into practise to provide quick clinical decision support.

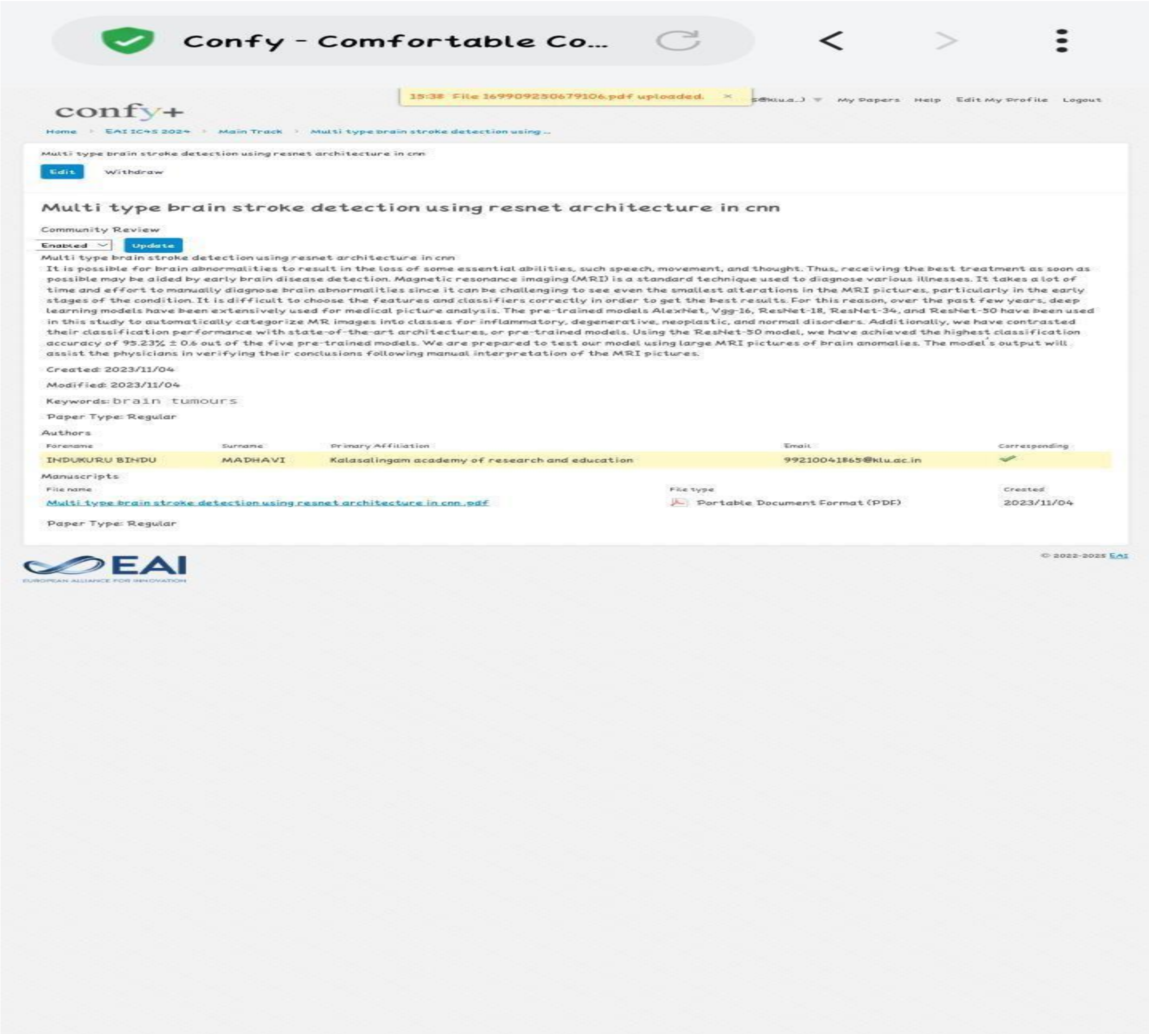
iv. Collaborative learning is being introduced to boost model generalisation by securely aggregating data from several healthcare facilities.

v. creating a mobile application that is easy to use in order to quickly and easily assess stroke risk and provide early intervention

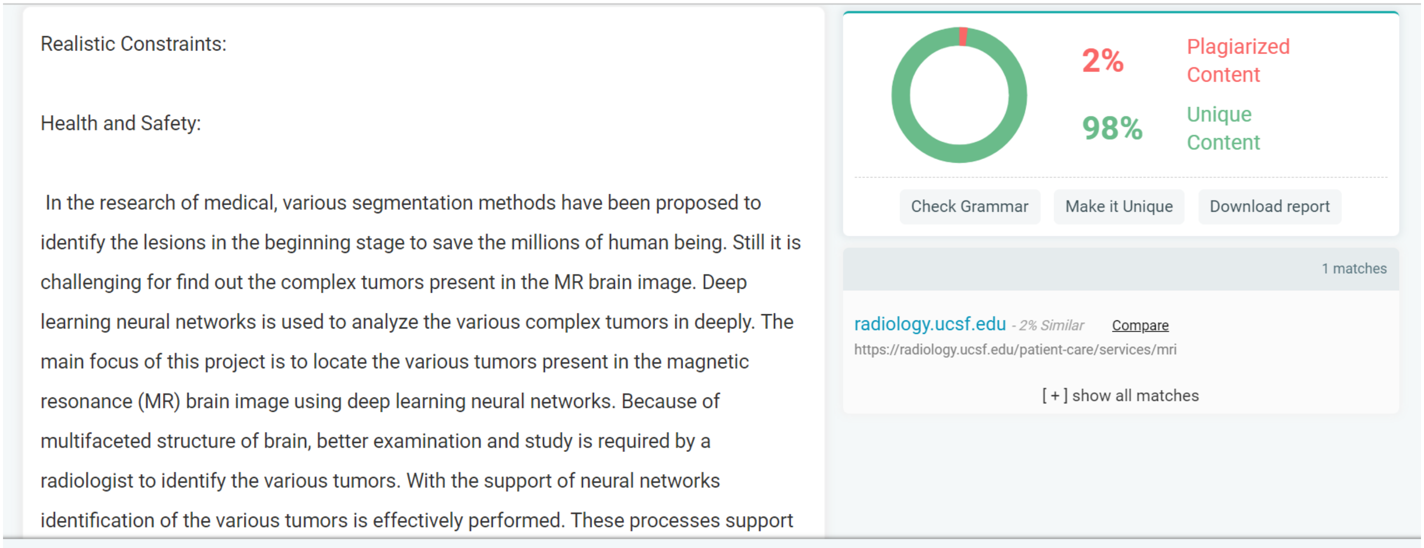
REFERENCES

- Chiun-Li Chin, Bing-Jhang Lin, Guei-Ru Wu, Tzu-Chieh Weng. An.(2017) automated early ischemic stroke detection system using CNN deep learning algorithm. DOI: 10.1109/ICAwST.2017.8256481
- Muhammad Sharif, Anandakumar Haldorai, Mussarat Yasmin & Ramesh Sundar Nayak.(2021).Brain tumor detection and classification using machine learning: a comprehensive survey.3161–3183
- Pratibha Sharma, Manoj Diwakar, Sangam Choudhary.(2012)Application of Edge Detection for Brain Tumor Detection. International Journal of Computer Applications (0975 – 8887) Volume 58– No.16
- Minhaz Uddin Emon, Maria Sultana Keya, Tamara Islam Meghla, Md. Mahfujur Rahman.(2020)Performance Analysis of Machine Learning Approaches in Stroke Prediction.DOI: 10.1109/ICECA49313.2020.9297525
- Protima Khan; Md. Fazlul Kader; S. M. Riazul Islam; Aisha B. Rahman.Machine Learning and Deep Learning Approaches for Brain Disease Diagnosis: Principles and Recent Advances.DOI: 10.1109/ACCESS.2021.3062484

APPENDICES:



PLAGARISM:



Plagiarism Result

Share It

It can be difficult and important to detect multiple brain strokes using deep learning techniques such as ResNet (Residual Neural Network). Popular deep learning architecture ResNet is renowned for its capacity to manage challenging image recognition tasks. The following are some suggested methods for utilizing ResNet to detect multiple brain strokes:

Image Preparation:

First, obtain high-quality CT or MRI scans of the brain.

To enhance the quality of the images, use preprocessing techniques like resizing, normalization, and contrast enhancement.

Data Enrichment:

Expand the dataset to boost its quantity and variety. Rotating, flipping, and adding

0% Plagiarized Content

100% Unique Content

Check Grammar | Make it Unique | Download report

0 matches

[+] show all matches

Scan Properties

Number of Words : 970

Results Found : 3

To or From | To or From

Binary Translator | PDF Converter

6% Plagiarism | 94% Unique

Make it Unique | Start New Search

Check Grammar | Detect AI Content

To check plagiarism in photos click here

Reverse Image Search

Feature Extraction

Feature extraction is a critical step in medical image analysis. The use of deep learning techniques, particularly CNNs, automates this process, enabling the model to discern essential features without manual intervention.

Edge Detection

Edge detection techniques play a role in outlining and emphasizing key structures in MRI images. The incorporation of edge detection enhances the model's ability to identify and classify abnormalities accurately

Literature Survey

In the ever-evolving landscape of medical diagnostics, the fusion of advanced technology and medical expertise holds the promise of transforming patient outcomes. This review delves into the domain of brain stroke detection, where the integration of Convolutional Neural

Similarity 25%

Medical Imaging Interview Questions Answers - PYCAD

https://pycad.co/medical-imaging-interview-questions-answers/

Similarity 7%

Performance Analysis of Machine Learning Approaches in ...

https://ieeexplore.ieee.org/document/9297525

Similarity 5%

Machine Learning and Deep Learning Approaches for ...

https://ieeexplore.ieee.org/document/9363896

Scan Properties

Number of Words : 791

Results Found : 0

To or From | To or From

Binary Translator | PDF Converter

0% Plagiarism | 100% Unique

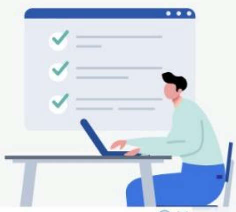
Make it Unique | Start New Search

Check Grammar | Detect AI Content

To check plagiarism in photos click here

Reverse Image Search

Abstract— It is possible for brain abnormalities to result in the loss of some essential abilities, such speech, movement, and thought. Thus, receiving the best treatment as soon as possible may be aided by early brain disease detection. Magnetic resonance imaging (MRI) is a standard technique used to diagnose various illnesses. It takes a lot of time and effort to manually diagnose brain abnormalities since it can be challenging to see even the smallest alterations in the MRI pictures, particularly in the early stages of the condition. It is difficult to choose the features and classifiers correctly in order to get the best results. For this reason, over the past few years, deep learning models have been extensively used for medical picture analysis. The pre-trained models



close