Detection and Classification of Tumors in Medical Images(X-rays and MRIs)

MACHINE VISION

Team Name : **Data Maverick(7805)**

Prakarsh Pathak

Puspkant Kumar

Mentor : Dr Chetan J Shelke

Current Scenario

- The radiology technician will transmit the images or notify the radiologist that the images are available via a secure method.
- Some facilities have a radiologist onsite who reviews the images. Others contract with radiologists who may live in different locations who view and interpret the images.
- A radiologist will then write a report of their findings.
- The doctor who ordered your MRI receives the report and views the images too.
- They will decide if they agree or disagree with the radiologists' findings. They may seek out another opinion from an additional medical specialist.
- The doctor will present the MRI findings to you either over the phone or at a follow-up appointment.
- Generally speaking, the radiologist, doctor, and other medical professionals are all busy. Reading and determining MRI results may not take long to do in terms of time commitment, but may be a longer process.

Problem Statement

The accurate and timely detection of tumors in medical images is critical for diagnosing and treating various medical conditions. This use case focuses on leveraging computer vision technology to detect and classify tumors in medical images, such as X-rays and MRIs, aiding medical professionals in making informed decisions and improving patient outcomes.

Data acquisition

- This dataset contains 7023 images of human brain MRI images .
- This dataset is a combination of the following three datasets :
  - figshare
  - SARTAJ dataset
  - Br35H
- we were not able to get the unified data for all the different type of tumors. we have searched and combined 3 datasets to make our own dataset with 4 classes which are glioma - meningioma - no tumor and pituitary.

- This data was used in the following paper:

1. Cheng, Jun, et al. "Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation and Partition." PloS one 10.10 (2015).
2. Cheng, Jun, et al. "Retrieval of Brain Tumors by Adaptive Spatial Pooling and Fisher Vector Representation." PloS one 11.6 (2016).

About Dataset

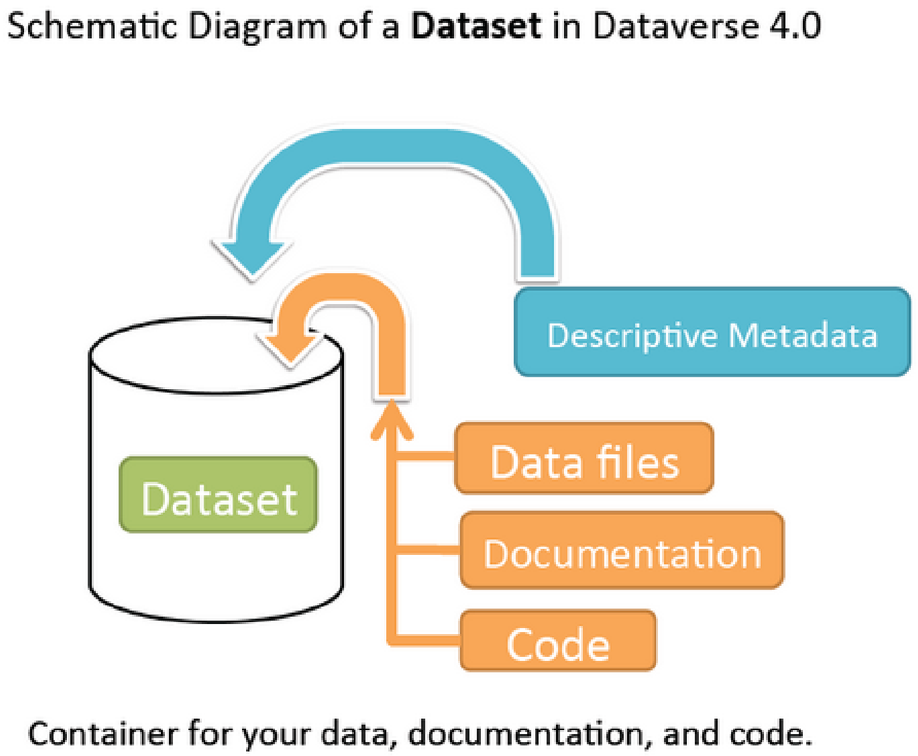
- **This dataset contains 1273 images of human brain MRI images**

- **Glioma: Gliomas are tumors that arise from glial cells in the brain. They can be low-grade (less aggressive) or high-grade (more aggressive).**

- **Meningioma: Meningiomas are typically benign tumors that form in the meninges, the protective layers around the brain.**

- **Pituitary Tumor (Pituitary Adenoma): Pituitary tumors form in or around the pituitary gland, which is located at the base of the brain.**

- **No tumor : Means that there are no signs of cancerous or non-cancerous growths in the imaged area.**



Preprocessing

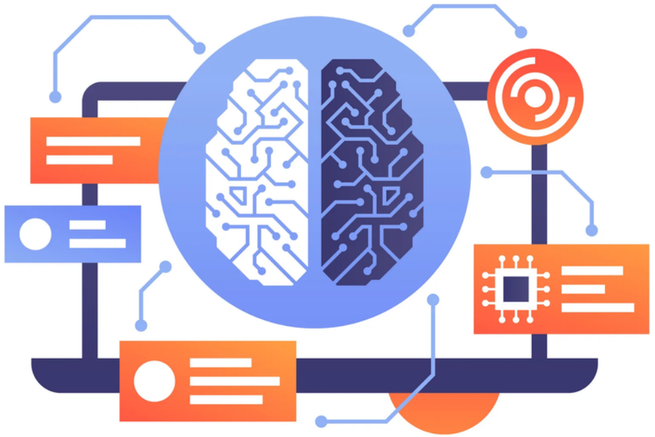
Techniques used:–

1. **Resizing** : - to ensure all images are of same dimensions. In our case our image size = 150 (cv2.resize)
2. **Normalization** :- to a common scale (e.g., [0, 1] or [-1, 1]) helps in improving convergence during training and reducing the impact of varying illumination conditions. (cv2.normalize)
3. **Grayscale conversion**:- cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
4. **Data Augumentation** :- applied during preprocessing to create additional training examples by applying various transformations like rotation, flipping, and scaling

Model Used:

EfficientNet B0 is a convolutional neural network (CNN) architecture designed for efficient and accurate computer vision tasks. It features a compound scaling method that adjusts the network's depth, width, and resolution to optimize performance.

1. **Efficiency:** high accuracy with significantly fewer parameters compared to other large CNN architectures, making it computationally efficient.
2. **Compound Scaling**: EfficientNet balances depth, width, and resolution with a user-defined parameter, allowing you to trade off between efficiency and accuracy.
3. **Transfer Learning:** EfficientNet B0 is suitable for transfer learning, making it valuable for various applications by fine-tuning pre-trained models on specific tasks.
4. **The architecture of the classification model:**
   - Use EfficientNetB0 as a base model with pre-trained weights.
   - Add a Global Average Pooling layer.
   - Apply Dropout regularization.
   - Add a Dense output layer with softmax activation for classification.
5. **Versatility:** Apart from medical applications, EfficientNet B0 can be applied to tasks like image classification, object detection, and semantic segmentation, and is deployable in resource-constrained environments.



Implementation

The implementation consists of a tumor classification system using a pre-trained deep learning model (EfficientNetB0). It allows users to upload medical images, predicts the type of tumor present, and sends real-time alert messages via WhatsApp when a tumor is detected.
Key points:

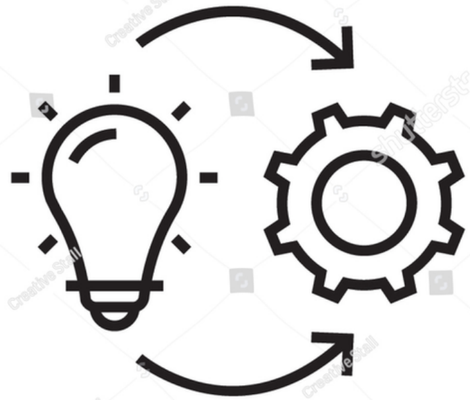
1. **Import Libraries**:
2. **Load Trained Model**:
3. **Define Labels**:
4. **Load Image Data**:
5. **Data Shuffling and Splitting** :Shuffle the training data to randomize the order of samples. Split the data into training and testing sets using **train_test_split**
6. **One-Hot Encoding**: Perform one-hot encoding for the target labels (**y_train** and **y_test**) using **tf.keras.utils.to_categorical()**
7. **Model Compilation**: Compile the model by specifying the loss function ('categorical_crossentropy'), optimizer ('Adam'), and evaluation metric ('accuracy').
8. **Callbacks**: Define callbacks for model training:

   TensorBoard for logging training information.

   ModelCheckpoint to save the best model during training.

   ReduceLROnPlateau to adjust the learning rate if validation accuracy plateaus.

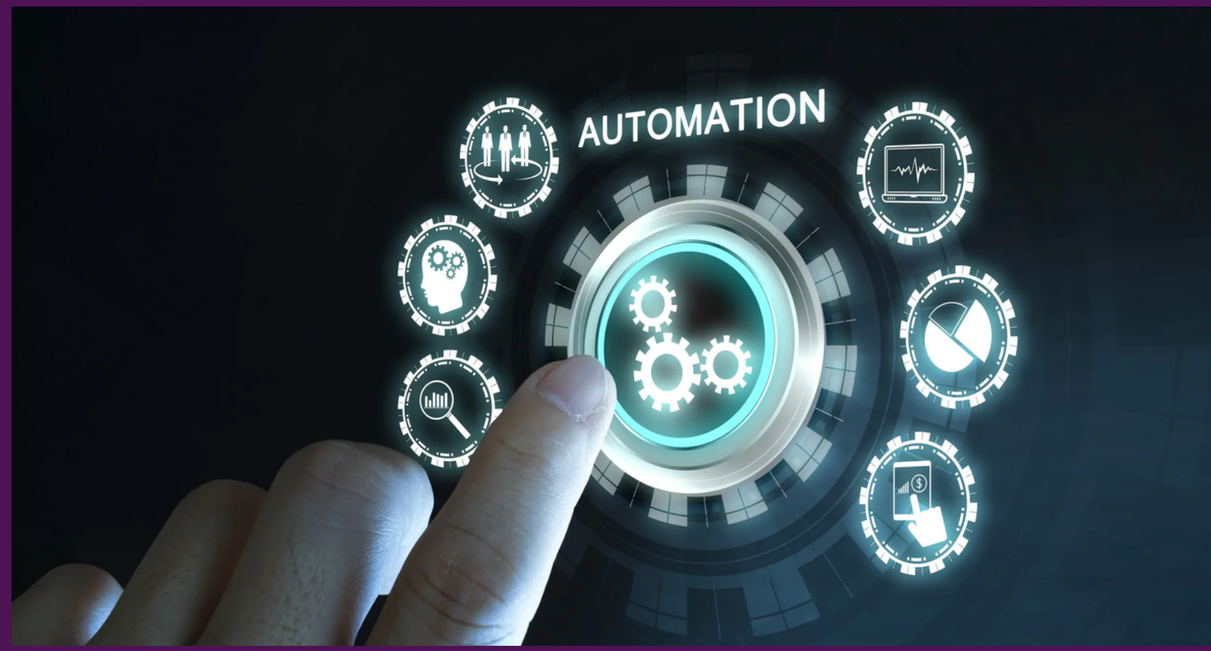
- **Model Training:**
- **Plot Training History:**
- **Make Predictions:**
- **Classification Report and Confusion Matrix:**
- **Alert Generation:** Define a function img_pred to predict tumor type for an uploaded image.
  Load an image using file upload functionality.
  Preprocess the image and make predictions.
  Send a WhatsApp alert message with the prediction using pywhatkit.
- **File Upload Widget:**
- **Prediction Button:**
- **Display Results:**
- **Alert via WhatsApp:** When a tumor is detected, send an alert message via WhatsApp using **pywhatkit**. The alert includes the prediction (tumor type) along with additional information.



Implementation

Model automation

We have used widgets from the **ipywidgets** library to create a basic user interface for image upload and prediction triggering within a Jupyter Notebook.



- **File Upload Widget** :- This widget allows users to upload image files from their local machine to the Jupyter Notebook environment.
- **Event Handling (on_button_clicked):** An event handler function, on_button_clicked, is defined to handle the button click event. When the "Predict" button is clicked, this function is executed.
- **User Interaction:** Users can interact with the interface by uploading an image using the file upload widget and then clicking the "Predict" button.
- **Displaying Results:** The prediction results, including the type of tumor (if detected), are displayed within the out widget.

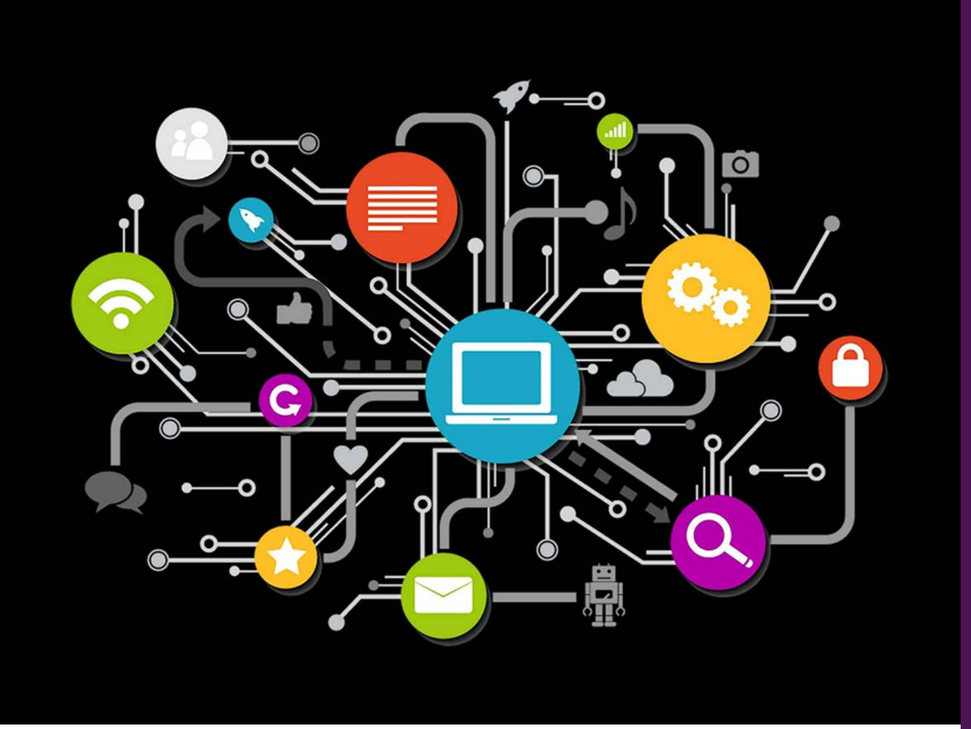
Auto Message Generation

PyWhatKit is a Python library that allows you to automate various tasks on WhatsApp, including sending messages and scheduling messages.

- uses the customers **phone number**, message and time of scheduling as input.
- it requires the whatsapp web login of the sender.
- we have used the **DATETIME** library to get the current hour and minute. we have added extra 1 minute to smoothen the process.
- the **whatsapp** will be open automatically and the message will be send directly to the specified user without any manual intervention.

```
# Send the prediction via WhatsApp
phone_no = "+916280408473"  # Replace with the recipient's ph
myobj = datetime.now()
time_hour = myobj.hour
time_min = myobj.minute + 1
info = "This is to inform you about the tumor prediction"
message = info + ": " + p

# Send the message
pywhatkit.sendwhatmsg(phone_no, message, time_hour, time_min)
```



Application and Benefits:

01

- Rapid Diagnosis:
- Timely Treatment:
- Improved Patient Outcomes:
- Reduced Uncertainty: facilitating informed decision-making.
- Streamlined Communication:
- Minimized Delays:

- easy to use for both the patients and doctors.
- gives the results instantly to the patients using whatsapp automation.
-

Future work

- can create a web application for efficient communication between patient and doctors.
- better designing of the application using flask and django to give patients a better experience.
- can similarly be used in CT scans, X- Ray scans.
- sending messages can be more fast and effective by using images and hospitals logos.
- can be used for business like sending fast blood reports,RT-PCR reports directly on whatsapp.


thank
you