BRAIN TUMOR DETECTION AND CLASSIFICATION USING MRI SCANS

OVERVIEW:

This project utilizes deep learning techniques to detect and classify brain tumors from MRI scans into categories such as glioma, meningioma, pituitary tumors, or normal. The aim is to assist radiologists by enhancing diagnostic accuracy and efficiency.

DATA APPROCH:

1. Data Collection

- **Dataset Source**:
  - Kaggle: Brain Tumor MRI Dataset
- **Dataset Description**:
  - MRI images categorized into four classes:
    - Glioma
    - Meningioma
    - Pituitary Tumors
    - Normal
  - Labeled dataset with clear classification for supervised learning.

2. Data Preprocessing

- **Image Resizing**:
  - All MRI images resized to a uniform size of (299, 299, 3) to match the input shape of the Xception model.
- **Normalization**:

  - Normalize pixel intensity values to a range of [0, 1] by dividing by 255. This ensures faster convergence during training.
- **Data Augmentation**:
  - To increase dataset diversity and prevent overfitting:
    - **Rotation**: Random rotations between 0°–30°.
    - **Flipping**: Horizontal and vertical flips.
    - **Zooming**: Random zoom with a factor range of 0.8–1.2.
    - **Shifts**: Random width and height shifts within 10% of the image dimensions.
- **Label Encoding**:
  - Convert categorical labels into one-hot encoding for multi-class classification.
  - Example: Normal → [1, 0, 0, 0].

3. Data Splitting

- **Training Set**:
  - ~70% of the dataset.
  - Used for training the deep learning model.
- **Validation Set**:
  - ~15% of the dataset.
  - Used for hyperparameter tuning and performance monitoring.
- **Test Set**:
  - ~15% of the dataset.
  - Kept aside for final evaluation to ensure the model generalizes to unseen data.
- Ensure random shuffling while splitting to maintain class balance.

- **4. Exploratory Data Analysis (EDA)**
- Analyze the dataset for:
  - Class distribution to detect and address any imbalance.
  - Visual inspection of MRI images to understand variations in features.
  - Histogram of pixel intensity values.
- **Addressing Imbalance**:
  - Use techniques such as oversampling, undersampling, or weighted loss functions if one class dominates.

5. Data Pipeline

- Create an efficient **data generator pipeline** using frameworks like TensorFlow to:
  - Load batches of data into memory.
  - Apply real-time augmentations.
  - Ensure optimized GPU usage during training.

6. Data Quality Checks

- Check for:
  - Corrupted images and remove them.
  - Duplicate entries and ensure they don't skew results.
- Validate the consistency of image labels.

7. Evaluation Data

- **Grad-CAM Visualization**:

- Use Grad-CAM to validate that the model focuses on relevant tumor regions during prediction.

- **Metrics**:
    - Compute metrics like accuracy, precision, recall, and F1-score for each class.
    - Analyze confusion matrices to identify misclassified images.

MODEL ARCHITECTURE:

The deep learning model utilizes the **Xception** pre-trained network as the base for feature extraction, with modifications for brain tumor classification.

Key Components:

1. **Base Model**:
    - Pre-trained on ImageNet.
    - Xception model with frozen weights.
    - Input shape: (299, 299, 3).
2. **Custom Layers**:
    - Input layer with shape (299, 299, 3).
    - Flattening layer.
    - Dropout layers (0.3 and 0.25) for regularization.
    - Dense layers (128 units with ReLU activation).
    - Final Dense layer with 4 units and softmax activation for multi-class classification.
3. **Compilation**:
    - Optimizer: Adamax with a learning rate of 0.001.

- Loss: categorical_crossentropy.
- Metrics: Accuracy, Precision, Recall.

Training and Evaluation

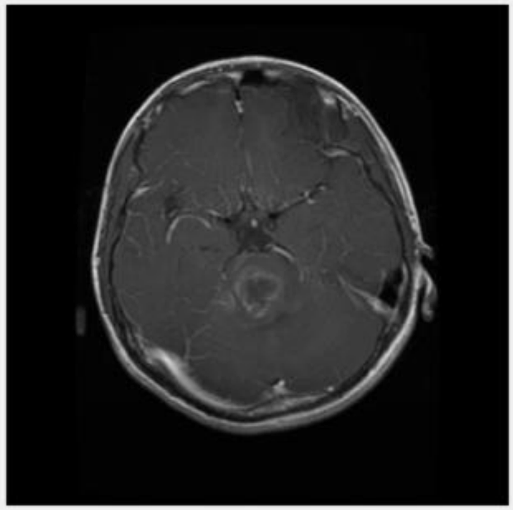
- **Training**:
  - Dataset split into training, validation, and testing sets.
  - Hyperparameter tuning for optimizers, learning rates, and dropout rates.
- **Evaluation Metrics**:
  - Accuracy
  - Precision
  - Recall
  - F1-Score
  - AUC-ROC

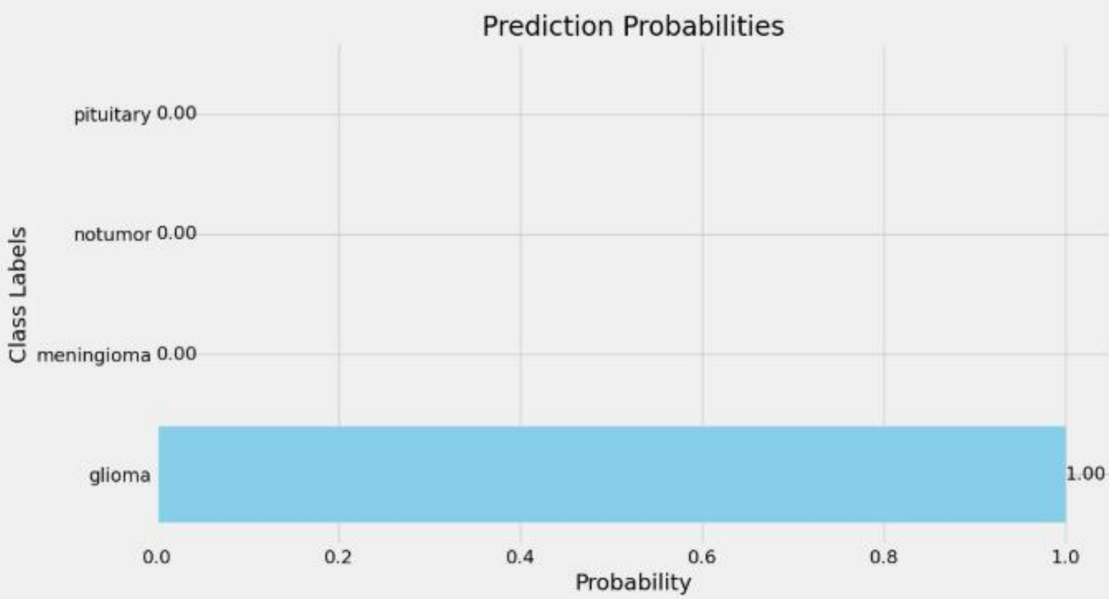
OUTPUT:

```
    plt.tight_layout()
    plt.show()
```

```
[78]: predict(r"C:\Users\HP\.cache\kagglehub\datasets\masoudnickparvar\brain-tumor-mri-dataset\versions\1\Testing\glioma\Te-glTr_0007.jpg")
```

```
Processed image shape: (1, 299, 299, 3)
Model input shape: (None, 299, 299, 3)
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 208ms/step
```

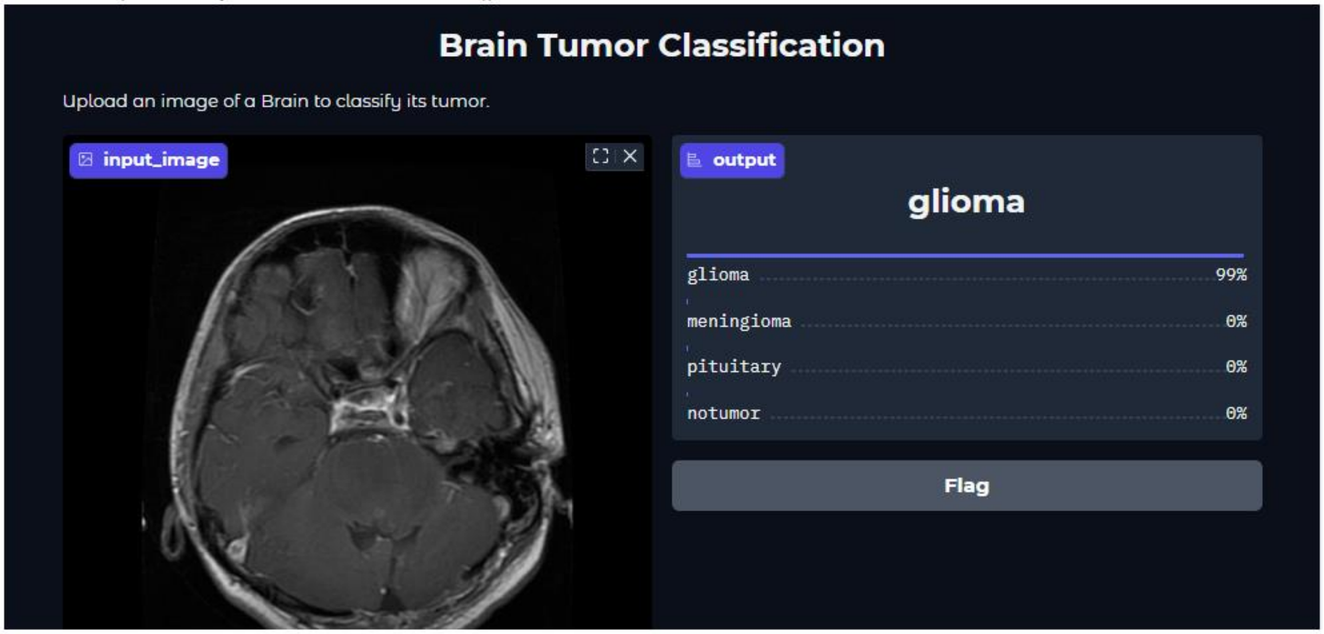





```
[79]: predict(r"C:\Users\HP\.cache\kagglehub\datasets\masoudnickparvar\brain-tumor-mri-dataset\versions\1\Testing\meningioma\Te-me_0210.jpg")
```

```
* Running on local URL:  http://127.0.0.1:7860

To create a public link, set `share=True` in `launch()`.
```




```
]:
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 200ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 223ms/step
```

CONCLUSION:

The project Brain Tumor Detection and Classification Using MRI Scans demonstrates the potential of deep learning in revolutionizing medical imaging diagnostics. By leveraging pre-trained models like Xception and applying transfer learning techniques, the solution efficiently classifies brain tumors into categories such as glioma, meningioma, pituitary tumors, and normal cases. The integration of robust preprocessing methods, including normalization and data augmentation, ensures high-quality input for model training, improving its generalization to unseen data.