Paper 1:A Deep Analysis of Brain Tumour Detection from MR Images Using Deep Learning Networks

1. Methodology

The study proposed a CNN architecture for brain tumour detection from MRI images, consisting of multiple convolutional layers followed by max-pooling layers and a final dense layer with a softmax activation function. The data preprocessing involved normalisation and augmentation (rotation, zoom, shifting). The dataset was divided into 80% training, 10% testing, and 10% validation, with evaluation metrics including accuracy, recall, AUC, and loss.

2. Simulations

Simulations were conducted using Google Colab Pro+ with high-RAM runtime and GPUs (Tesla K80, T4, and P100). The proposed CNN model underwent 80 epochs with a batch size of 18 and a learning rate of 0.01. The Adam optimizer and categorical cross-entropy loss function were used during training.

3. Equations

- **Accuracy**: Accuracy=((TP+TN) / (TP+TN+FP+FN)) × 100%
- **Recall**: Recall=TP / TP+FN
- **AUC (Area Under the Curve)**: A measure of the model's ability to distinguish between classes.

4. Threading to Boost Performance

Although threading was not explicitly implemented in this study, opportunities exist in the preprocessing stage (parallelizing normalisation and augmentation) and CNN operations (parallelizing convolutional and matrix multiplication).

5. Techniques That Use Threading or Parallel Processing

- Parallelized augmentation and preprocessing for image datasets.
- Parallelized convolutional layers in CNN models using frameworks like TensorFlow or PyTorch.
- GPU-based acceleration for deep learning tasks, as used in the study.

6. Existing Bottlenecks in MRI-Based Brain Tumour Detection Workflows

- High computational demand for training large datasets.
- Inefficient preprocessing of low-quality and noisy MRI images.
- Long training times due to hardware limitations.

7. Novelty

- Use of a custom CNN architecture tailored for brain tumor detection.
- Performance comparison with established transfer learning models (ResNet-50, VGG16, Inception V3).
- Utilization of a relatively large dataset (3264 MRI images) compared to many other studies.

8. Result Analysis

- The CNN model achieved the highest validation accuracy (93.3%) and AUC (98.43%) compared to other models.
- It showed lower validation loss (0.25), outperforming transfer learning models like VGG16 and Inception V3.

9. Challenges Faced

- Limited GPU resources, causing prolonged training times.
- Lack of post hoc visualization tools to interpret CNN results and highlight tumor regions.

10. Future Directions

- Incorporating patient-specific data to enhance diagnostic accuracy.
- Exploring fine-tuning of pre-trained models like ResNet-50 and VGG16 for further optimization.
- Implementation of post hoc explanation methods for visualizing key tumor regions.

11. Discussion

- The study highlights the efficacy of CNNs in MRI-based brain tumor detection, showcasing better accuracy and AUC than other models.
- The importance of a robust preprocessing pipeline is emphasized, as MRI images are often low-quality.
- Future work should address challenges like computational efficiency and model interpretability.

12. Research Gap

- Limited focus on the use of patient-specific metadata for improving predictions.
- Lack of implementation of advanced threading techniques for performance optimization.
- Absence of explainable AI methods to visualise model decisions.

Paper 2: A distinctive approach in brain tumour detection and classification using MRI

1. Methodology

- The study presents an automated system for detecting and classifying brain tumours at the lesion and image levels using MRI.
- It involves preprocessing to segment the region of interest, feature extraction based on texture, shape, and intensity, and classification using Support Vector Machine (SVM) with different kernels (Linear, Gaussian, and Cubic).
- Datasets used include Harvard, RIDER, and a local dataset, with cross-validation approaches applied for model evaluation.

2. Simulations

- The experiments were conducted on datasets with preprocessing techniques like skull removal, Gaussian filtering, and clustering methods (e.g., K-means).
- Classification was done using SVM with different kernels, tested across 5, 10, 15, 20, and 30-fold cross-validation setups to evaluate accuracy, AUC, sensitivity, and specificity.

3. Equations

- **Accuracy (ACC):** ACC=TP+TN / TP+TN+FP+FN
- **AUC:** Integrates True Positive Rate (TPR) and False Positive Rate (FPR).
- **Sensitivity (Recall):** Sensitivity=TP / TP+FN
- **Specificity:** Specificity=TN / TN+FP

  Additional equations involve thresholding and morphological operations like dilation and erosion.

4. Threading to Boost Performance

- Parallelism could be introduced in:
  - **Preprocessing:** Parallel execution of skull removal and Gaussian filtering.
  - **Feature Extraction:** Concurrent computation of texture, shape, and intensity features.
  - **Classification:** Batch parallelization in SVM training.

5. Techniques That Use Threading or Parallel Processing

- Using optimized libraries like OpenCV for preprocessing tasks.
- Leveraging multi-threaded SVM implementations for large datasets.
- Employing GPU-based parallel processing for clustering and filtering.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- High computational time for feature extraction and classification.

- Low-quality MRI scans requiring intensive preprocessing.
- Scalability issues with traditional methods for larger datasets.

7. Novelty

- A combination of texture, shape, and intensity features for improved classification.
- Integration of three SVM kernel types for enhanced accuracy and flexibility.
- Superior accuracy (97.1%), sensitivity (91.9%), and specificity (98.0%) compared to existing methods.

8. Result Analysis

- The proposed method outperformed existing techniques in accuracy, sensitivity, and specificity across datasets.
- The Linear kernel SVM achieved the best results for image-level classification, while the Gaussian kernel excelled in lesion-level detection.

9. Challenges Faced

- Variability in MRI quality across datasets.
- Balancing computational efficiency with model complexity.
- Selecting optimal parameters for segmentation and SVM kernels.

10. Future Directions

- Introducing advanced threading for preprocessing and classification tasks.
- Exploring deep learning models like CNNs for automated feature extraction.
- Applying explainable AI for better interpretation of tumor classification.

11. Discussion

- The proposed system demonstrates potential for clinical application due to its high accuracy and efficiency.
- Results suggest improvements in processing speed and classification precision through optimization.

12. Research Gap

- Lack of end-to-end integration of deep learning for tumor detection.
- Minimal focus on real-time processing capabilities.
- Limited explainability of classification results.

Paper 3: A Hybrid CNN-SVM Threshold Segmentation Approach for Tumour Detection and Classification of MRI Brain Images

1. Methodology

- The proposed system utilizes a hybrid model combining CNN for feature extraction and SVM for classification.
- MRI images undergo preprocessing steps such as resizing, skull removal, and filtering to enhance quality.
- Feature extraction is performed using convolution and pooling layers in the CNN. The extracted features are fed into the SVM for classification.
- The system segments MRI images using a threshold-based approach to identify tumor regions.

2. Simulations

- Experiments were conducted using the BRATS 2015 dataset, containing MRI images classified into benign and malignant tumors.
- Implemented on a Dell laptop with a Core i7 CPU, 8GB RAM, and a 4GB NVIDIA GPU.
- The hybrid CNN-SVM model was tested for its ability to classify MRI images, yielding an accuracy of 98.495%.

3. Equations

- **Accuracy**: Accuracy=TP+TN / TP+TN+FP+FN
- **Positive Predictive Value (PPV)**: PPV=TP / TP+FP
- **False Predictive Value (FPV)**: FPV=FP / FP+TP
- Feature selection and segmentation equations were also employed, such as thresholding and entropy calculations.

4. Threading to Boost Performance

- Threading can enhance preprocessing by parallelizing tasks like resizing, skull removal, and noise filtering.
- In CNN operations, parallelizing convolutional layers and pooling operations could improve performance.
- SVM classification may benefit from threading by handling feature subsets concurrently.

5. Techniques That Use Threading or Parallel Processing

- Multi-threaded libraries like OpenCV for image preprocessing.
- GPU-based parallelism for CNN layers to accelerate feature extraction.
- Batch processing with multi-threaded SVM classifiers for large datasets.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- Computationally expensive preprocessing steps.
- High training time for CNN models due to large MRI datasets.
- Inefficient segmentation methods causing delays in tumor identification.

7. Novelty

- Combination of CNN and SVM exploits strengths of both methods: CNN's ability to extract complex features and SVM's accuracy in classification.
- Use of threshold segmentation for better tumor localization.
- High accuracy and precision compared to standalone models.

8. Result Analysis

- The hybrid CNN-SVM model achieved a classification accuracy of 98.495%, outperforming standalone CNN (97.5%) and SVM (72.55%).
- Results show significant improvements in PPV and FPV for the hybrid approach.
- The model effectively distinguished benign and malignant tumors with high precision.

9. Challenges Faced

- Extensive computational requirements during feature extraction and classification.
- Dataset limitations in terms of diversity and quality.
- Difficulty in achieving real-time performance.

10. Future Directions

- Incorporating faster CNN architectures and bio-inspired optimization algorithms to improve efficiency.
- Enhancing segmentation techniques to include tumor size and location.
- Exploring explainable AI to interpret model predictions.

11. Discussion

- The hybrid model demonstrates the potential for automated brain tumor detection and classification with high accuracy.
- Results indicate that combining CNN and SVM provides significant advantages over standalone models.
- Future research should address computational efficiency and dataset diversity for broader applicability.

12. Research Gap

- Limited exploration of real-time processing capabilities in clinical settings.
- Lack of patient-specific contextual data integration for improved accuracy.
- Insufficient focus on model explainability and interpretability for medical professionals.

Paper 4: An Image Processing-based and Deep Learning Model to Classify Brain Cancer

1. Methodology

The research combines image processing and deep learning techniques to classify brain cancer into 14 distinct types. The methodology includes several key steps:

- **Dataset Collection**: A dataset of 4,489 MRI images was utilized, including T1 and T2 contrast-enhanced scans. Images represent 14 cancer types, such as glioblastoma, medulloblastoma, and schwannoma. The dataset includes both common and rare types of brain tumors, with imbalances addressed through data preprocessing.
- **Preprocessing**: Images were resized to 224×224 pixels to standardize input dimensions across models. Noise reduction techniques and data augmentation were applied to enhance generalization. Images were grouped into batches of 32 for efficient model training.
- **Deep Learning Models**: Four state-of-the-art architectures were evaluated:
    - **YOLOv8**: Optimized for real-time applications with fast inference and high accuracy.
    - **ResNet12**: Designed to address the degradation problem in deep networks by using residual connections.
    - **DenseNet169**: Improves information flow and feature reuse with dense layer connections.
    - **MobileNet**: Developed for mobile and embedded applications, using depthwise separable convolutions to reduce computational cost.
- **Model Training**: Hyperparameters such as a learning rate of 0.001, a batch size of 64, and 50 epochs were used. The data split was 80% training, 10% validation, and 10% testing. Training was performed in Google Colab using high-performance GPUs like T4, V100, and A100.

2. Simulations

- The models were trained and evaluated using the Google Colab environment, which supports GPU acceleration.
- The best-performing model, YOLOv8, achieved an inference time of 1.8ms and a preprocessing time of 0.1ms.
- DenseNet169 and ResNet12 required more training time due to their complex architectures, while MobileNet was faster but less accurate.
- Experiments involved tuning hyperparameters such as epochs, learning rate, and batch size to identify the optimal configuration for each model.

3. Equations

The following equations were used to evaluate the performance of the models:

- **Precision**: Precision=True Positives (TP) /True Positives (TP)+False Positives (FP)
- **Recall**: Recall=True Positives (TP) / True Positives (TP)+False Negatives (FN)
- **Accuracy**: Accuracy=True Positives (TP)+True Negatives (TN) / Total Predictions (TP + TN + FP + FN)

4. Threading to Boost Performance

- **Preprocessing**: Threading can be applied to parallelize resizing, noise reduction, and batch creation, significantly reducing preprocessing time.

- **Model Training**: Multi-threaded execution during training allows parallel processing of data batches, improving training efficiency on GPUs.
- **Inference**: Real-time applications like YOLOv8 can utilize threading to process multiple images simultaneously, reducing latency.

5. Techniques That Use Threading or Parallel Processing

- **GPU Acceleration**: Frameworks like TensorFlow and PyTorch inherently support parallel computation for CNN operations, such as convolution and pooling.
- **Batch Parallelization**: Large datasets are split into smaller batches processed in parallel, optimizing GPU usage.
- **YOLOv8**: Employs advanced threading and anchor-free mechanisms to improve speed and accuracy during inference.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- **Data Imbalance**: Some rare cancer types are underrepresented, affecting model generalization.
- **Computational Costs**: Training deep models like DenseNet169 and YOLOv8 demands significant resources, limiting scalability.
- **Preprocessing Overhead**: Steps like resizing, noise reduction, and augmentation can become bottlenecks for large datasets.
- **Inference Challenges**: Real-time applications require fast and accurate models, which can be hindered by hardware limitations.

7. Novelty

- The use of YOLOv8, a model optimized for real-time mobile applications, highlights the novelty of the study.
- The inclusion of 14 diverse cancer types ensures the model's robustness compared to earlier studies with fewer categories.
- The study evaluates model performance across three critical parameters: accuracy, speed, and model size, providing a comprehensive analysis.

8. Result Analysis

- **YOLOv8**: Achieved the best accuracy (97.3%) and the fastest inference time (1.8ms), making it suitable for mobile applications despite its larger model size (57.39MB).
- **DenseNet169**: Performed similarly in accuracy (97%) but required more processing time due to its dense architecture.
- **ResNet12**: Balanced accuracy (92.36%) with moderate model size (38.15MB), making it suitable for scenarios with constrained resources.
- **MobileNet**: Had the smallest size (18.35MB) and faster training but achieved the lowest accuracy (85.17%).

9. Challenges Faced

- **Data Quality**: Variability in MRI quality impacts model performance.
- **Resource Constraints**: Training large models like DenseNet169 and YOLOv8 required access to high-performance GPUs.
- **Real-Time Adaptation**: Balancing model size with accuracy and speed remains a challenge for mobile deployment.

10. Future Directions

- **Dataset Expansion**: Including additional cancer types such as breast and lung cancer to broaden the model's applicability.
- **Model Interpretability**: Integrating explainable AI techniques to help medical professionals understand predictions.
- **Optimization**: Reducing YOLOv8's size without compromising accuracy for better integration into mobile devices.

11. Discussion

- The study demonstrates the effectiveness of deep learning models in brain cancer classification, with YOLOv8 emerging as the best performer.
- The diversity of the dataset contributes to the model's robustness, but imbalanced data for rare cancer types remains a limitation.
- Future work should focus on making models more interpretable and resource-efficient for real-time medical applications.

12. Research Gap

- **Real-Time Processing**: Limited focus on optimizing models for real-time clinical use.
- **Explainability**: Lack of tools to interpret model predictions for medical professionals.
- **Dataset Imbalance**: Need for techniques to address underrepresented cancer types.

Paper 5:Automated Brain Tumour Segmentation and Classification in MRI Using YOLO-Based Deep Learning.

1. Methodology

The study focuses on using YOLOv5 and YOLOv7 models for brain tumor segmentation and classification from MRI scans. The workflow includes:

- **Dataset Collection**: A dataset of 3,064 MRI images from Southern Medical University is used. Tumors are categorized into meningioma, glioma, and pituitary types.
- **Preprocessing**: Conversion of `.mat` files to `.png` images and masks. Morphological operations and annotation alignment standardize the tumor regions for analysis.
- **Model Training**:
  - **YOLOv5**: Employs convolutional layers, C3 blocks, and spatial pyramid pooling to handle multi-scale features.

- ○ **YOLOv7**: Refines the architecture for greater accuracy and efficiency with additional layers and multi-scale feature fusion.
- **Evaluation Metrics**: Precision, recall, F1 score, and mean average precision (mAP) are used for model comparison.

2. Simulations

- Simulations were performed using Google Colab Pro with an NVIDIA A100-SXM4-40GB GPU.
- Hyperparameters included a learning rate of 0.001, batch size of 64, and 100 training epochs.
- Model inference times were 7.0 ms (YOLOv5) and 10.0 ms (YOLOv7), achieving high frame rates (143 FPS for YOLOv5 and 100 FPS for YOLOv7).

3. Equations

- **Precision**: Precision=TP/ TP+FP
- **Recall**: Recall=TP / TP+FN
- **Loss Function**: loss=l_box +l_cls+l_obj

  Where l_box,l_cls and l_obj represent bounding box, classification, and confidence losses, respectively.

4. Threading to Boost Performance

- **Preprocessing**: Threading can speed up image conversion and mask alignment tasks.
- **Model Training**: Parallel processing across GPU cores for batch updates can optimize computation.
- **Inference**: Real-time segmentation and classification benefit from multi-threaded processing to reduce latency.

5. Techniques That Use Threading or Parallel Processing

- YOLO models inherently leverage GPU parallelism for convolution and pooling layers.
- Frameworks like TensorFlow and PyTorch enable efficient multi-threading for training and inference.
- Advanced data loaders handle batched data preprocessing in parallel.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- Manual annotation and alignment of tumor masks are time-intensive.
- Computational cost increases with higher-resolution images and larger datasets.
- Real-time detection requires optimization to achieve low latency.

7. Novelty

- The study is the first to apply YOLOv7 on the BT dataset for brain tumor segmentation and classification.
- High mAP scores for multi-class tumor detection demonstrate the efficacy of YOLO-based methods.
- Standardized tumor mask alignment improves model training and evaluation outcomes.

8. Result Analysis

- **YOLOv5**: Achieved mAP@0.5 scores of 0.947 (box) and 0.947 (mask) with faster inference times.
- **YOLOv7**: Achieved mAP@0.5 scores of 0.94 (box) and 0.941 (mask), outperforming YOLOv5 in recall and segmentation loss.
- Both models struggled with glioma detection compared to meningioma and pituitary tumors.

9. Challenges Faced

- High false positives for glioma segmentation.
- Substantial computational resources were needed for training.
- Limited dataset diversity restricted model generalizability.

10. Future Directions

- Integrating lightweight deep learning models to improve efficiency and deployment on resource-constrained devices.
- Expanding datasets with additional tumor classes and modalities (e.g., CT scans).
- Implementing explainable AI techniques to enhance model interpretability.

11. Discussion

- YOLOv7 demonstrated superior segmentation accuracy across all tumor types, except for glioma.
- The study highlights the potential of YOLO models for medical applications, particularly for real-time diagnostics.
- Future improvements should address computational efficiency and expand tumor type coverage.

12. Research Gap

- Limited focus on real-time processing optimization for clinical use.
- Lack of methods to handle data imbalance, especially for glioma cases.
- Insufficient model interpretability for aiding medical professionals in decision-making.

Paper:6 : Brain Tumour Detection Based on Deep Learning Approaches and Magnetic Resonance Imaging

1. Methodology

The study proposes an enhanced YOLOv7-based model for multiclass brain tumor detection and classification. The methodology includes:

- **Data Collection**: A dataset with 10,288 images was used, containing glioma, pituitary, meningioma, and non-tumor categories, sourced from Kaggle.
- **Preprocessing**: MRI images were preprocessed using techniques such as Gaussian blur for noise reduction, high-pass filters for edge sharpening, and normalization to improve data quality.
- **Data Augmentation**: Techniques like rotation, flipping, and scaling were applied to increase dataset size, resulting in 51,448 images for training.
- **Model Architecture**:
    - Integrated Convolutional Block Attention Module (CBAM) for improved feature selection.
    - Added Spatial Pyramid Pooling Fast+ (SPPF+) for better multi-scale feature extraction.
    - Used BiFPN for efficient bidirectional feature fusion.
    - Employed a decoupled head (DP) for precise classification and regression.
- **Training and Evaluation**: The model was trained using transfer learning with weights pre-trained on the COCO dataset, fine-tuned for three tumor types.

2. Simulations

- **Setup**: The model was trained using an 8-core CPU, 32 GB RAM, and NVIDIA GeForce 1080Ti GPUs.
- **Hyperparameters**: Learning rate of 0.001, batch size of 64, and 100 epochs.
- **Evaluation Metrics**: Accuracy, precision, recall, specificity, sensitivity, and F1-score were calculated using confusion matrix analysis.

3. Equations

- **Precision (PR)**: PR=TP/ TP+FP
- **Recall (RE)**: RE=TP/TP+FN
- **Specificity (SP)**: SP=TN/TN+FP
- **Accuracy (AC)**: AC=TP+TN / TP+TN+FP+FN
- **F1-Score**: F1-Score=2×PR×RE / PR+RE

4. Threading to Boost Performance

Threading could be applied to:

- **Data Preprocessing**: Parallelize Gaussian blur, edge sharpening, and morphological operations.
- **Training**: Optimize batch processing using multi-threaded GPU operations.
- **Inference**: Utilize threading for simultaneous classification of multiple images.

5. Techniques That Use Threading or Parallel Processing

- **YOLOv7 Features**: Inherently supports GPU-based parallelism for real-time object detection.
- **Data Augmentation**: Parallel processing of transformations using libraries like Albumentations.
- **Feature Fusion**: BiFPN employs multi-threading for efficient feature aggregation.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- **Manual Annotation**: Time-consuming and prone to human error.
- **Data Imbalance**: Limited representation of rare tumor types.
- **Small Tumor Detection**: Difficulty in identifying tiny or low-contrast tumors in MRI images.

7. Novelty

- Incorporation of CBAM for better feature representation.
- Enhanced multi-scale feature extraction using SPPF+.
- Efficient handling of feature fusion with BiFPN.
- Achieved 99.5% accuracy, outperforming state-of-the-art models.

8. Result Analysis

- **YOLOv7 with Enhancements**: Delivered precision (99.5%), recall (99.3%), and F1-score (99.4%).
- Outperformed models like EfficientNet (97.8%) and YOLOv4 (97.8%) in accuracy.
- Demonstrated high sensitivity and specificity in multiclass tumor classification.

9. Challenges Faced

- **Computational Requirements**: Training required high-performance hardware.
- **Small Tumor Detection**: Despite enhancements, detecting minute tumors remains challenging.
- **Imbalanced Dataset**: Rare tumor types required additional data augmentation.

10. Future Directions

- **Dataset Expansion**: Include more tumour types and imaging modalities.
- **Lightweight Models**: Develop resource-efficient models for deployment in low-resource settings.
- **Explainable AI**: Enhance model interpretability for medical applications.

11. Discussion

The proposed YOLOv7 model demonstrated superior performance in brain tumour detection. Its architecture, combining CBAM, SPPF+, and BiFPN, ensured precise detection even for small tumours. The study highlights the importance of real-time detection for improving diagnostic workflows in clinical settings.

12. Research Gap

- Limited exploration of transfer learning for rare tumour types.
- Insufficient focus on explainability for clinical adoption.
- Lack of scalability for resource-constrained environments.

Paper 7 : Brain tumor detection from images and comparison with transfer learning methods and 3-layer CNN

1. Methodology

The paper outlines a structured approach to detect brain tumors using Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN). The key steps include:

- **Dataset Preparation**: The dataset, comprising 2065 MRI images (1085 tumor images and 980 non-tumor images), was sourced from GitHub. Images were resized to a uniform size of 256x256 pixels and normalized for consistency. The data was split into training (1672 images), validation (186 images), and testing (207 images).
- **ANN Architecture**: A seven-layer ANN was designed with one input layer, five hidden layers using the ReLU activation function, and one output layer using the sigmoid activation function. The hidden layers contained 128, 256, 512, 256, and 128 neurons, respectively.
- **CNN Architecture**: The CNN employed a sequential model with convolutional layers, max-pooling, and dropout layers. Five convolutional layers used 32, 32, 64, 128, and 256 filters. Each convolutional layer was followed by max-pooling (2x2) and dropout (20%). The final layers consisted of flattening, dense layers, and an output layer with a sigmoid activation function for binary classification.
- **Training and Evaluation**: Both models were trained using the Adam optimizer and binary cross-entropy loss. Training involved 200 epochs for both models, with performance monitored on validation and test sets.

2. Simulations

The experiments were conducted using Python on Google Colab. The models were trained on the dataset, and performance was analyzed across training, validation, and test sets:

- **ANN Simulation**: Achieved a training accuracy of 97.13%, validation accuracy of 71.51%, and test accuracy of 80.77%.
- **CNN Simulation**: Exhibited superior performance with a validation accuracy of 94.00% and a test accuracy of 89.00%.

- Graphical results compared accuracy and loss across epochs for both models, confirming CNN's robustness in handling image data.

3. Equations

The following equations are central to the methodology:

- **Convolution Operation (CNN)**:

$$(I * K)(x, y) = \sum_m \sum_n I(x+m, y+n) \cdot K(m, n)$$

  where I is the input image and K is the kernel.
- **Activation Functions**:
  - ReLU: f(x)=max(0,x)
  - Sigmoid: f(x)=1/1+e^ (−x)
- **Loss Function (Binary Cross-Entropy)**:

$$L = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

  where y_i is the true label and y^_i is the predicted probability.

4. Threading to Boost Performance

Threading can enhance performance in several stages of the workflow:

- **Data Preprocessing**: Parallelize image resizing and normalization.
- **Data Loading**: Use multithreaded pipelines to load large datasets efficiently.
- **Model Training**: Implement threading for batch processing, ensuring GPU and CPU resources are utilized effectively.
- **Inference**: Enable simultaneous predictions on multiple MRI scans.

5. Techniques That Use Threading or Parallel Processing

- **Threading in Preprocessing**: Resize and normalize images concurrently using libraries like OpenCV with Python threading.
- **GPU Acceleration**: Leverage CUDA or OpenCL to parallelize convolution operations in CNNs.
- **Data Augmentation**: Use parallelization for generating augmented datasets dynamically during training.
- **Batch Processing**: Parallelize training batches using frameworks like TensorFlow's `tf.data` API.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- **Preprocessing**: Handling large, high-resolution MRI scans is computationally expensive and time-intensive.

- **Model Training**: Training deep models on limited datasets often leads to overfitting and prolonged training times.
- **Data Imbalance**: Uneven distribution of tumor and non-tumor images affects model generalization.
- **Real-Time Inference**: Delays in processing MRI scans hinder the applicability of models in clinical settings.

7. Novelty

The paper introduces a self-defined architecture for both ANN and CNN tailored to the problem of brain tumor detection. It provides a comparative analysis of the two models, highlighting the CNN's superior performance in terms of accuracy and computational efficiency.

8. Result Analysis

- **ANN Performance**:
    - Training Accuracy: 97.13%
    - Validation Accuracy: 71.51%
    - Test Accuracy: 80.77%
- **CNN Performance**:
    - Validation Accuracy: 94.00%
    - Test Accuracy: 89.00%
- CNN outperformed ANN due to its ability to extract spatial hierarchies from image data effectively.

9. Challenges Faced

- **Data Limitations**: A relatively small dataset limited the model's capacity to generalize.
- **Computational Resources**: Training deep models on high-resolution MRI images required significant resources.
- **Hyperparameter Tuning**: Determining the optimal configuration of layers and filters was challenging.

10. Future Directions

- **Enhanced Data Augmentation**: Apply advanced augmentation techniques to expand the dataset and reduce overfitting.
- **Transfer Learning**: Utilize pre-trained CNNs like ResNet or Inception for better feature extraction.
- **Real-Time Systems**: Develop faster preprocessing and inference pipelines for real-time diagnosis.
- **Optimization**: Implement pruning and quantization techniques to reduce model complexity.

11. Discussion

The study demonstrates the effectiveness of CNNs over ANNs in detecting brain tumors from MRI scans. CNN's architecture, with convolutional layers and pooling operations, proved crucial for capturing image features. However, the dependency on dataset size and computational power underscores the need for efficient workflows.

12. Research Gap

- Limited exploration of hybrid models combining ANN and CNN features.
- Inadequate focus on advanced parallel processing techniques for real-time deployment.
- Lack of studies on model robustness across different MRI modalities and equipment.

Paper 8: Brain tumour detection from MRI images using deep learning techniques

1. Methodology

The study employs two deep learning techniques, Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN), to classify MRI images as tumor or non-tumor. Key steps include:

1. **Dataset Preparation**:
   - MRI images are resized to 256x256 pixels and normalized.
   - Data is split into training (1672 images), validation (186 images), and testing (207 images).
   - Labels are assigned (1 for tumor, 0 for non-tumor).
2. **ANN Model**:
   - Composed of seven layers: one input, five hidden layers (128-512 neurons with ReLU activation), and one output layer (sigmoid activation).
   - Optimized using Adam and binary cross-entropy loss.
3. **CNN Model**:
   - Includes convolutional layers with filters (32-256), max pooling, dropout layers (20%), and a dense output layer.
   - Trained using a sequential architecture to extract features hierarchically.
4. **Evaluation**:
   - Models are trained for 200 epochs.
   - Performance is evaluated on validation and test sets with metrics like accuracy, precision, and recall.

2. Simulations

Experiments were conducted using Python on Google Colab. Simulations involved training and testing the models:

- ANN achieved 97.13% training accuracy but only 71.51% validation accuracy.
- CNN demonstrated superior generalization with 94.00% validation accuracy and 89.00% test accuracy.

- Training and validation loss curves were plotted to monitor overfitting and convergence.

3. Equations

- **Convolution Operation (CNN)**:

$$(I * K)(x, y) = \sum_m \sum_n I(x + m, y + n) \cdot K(m, n)$$

  where I is the input image and K is the kernel.

- **Activation Functions**:
  - ReLU: f(x)=max(0,x)
  - Sigmoid: f(x)=1/1+e^ (−x)
- **Loss Function (Binary Cross-Entropy)**:

$$L = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

4. Threading to Boost Performance

Threading can optimize multiple aspects:

1. **Preprocessing**:
   - Resize and normalize images in parallel using Python's `concurrent.futures`.
2. **Data Loading**:
   - Use multi-threaded data pipelines to manage large batches efficiently.
3. **Model Training**:
   - Parallelize batch computations with GPU acceleration.
4. **Inference**:
   - Process multiple MRI scans concurrently for real-time predictions.

5. Techniques That Use Threading or Parallel Processing

- GPU-based computations for CNN layers using frameworks like TensorFlow or PyTorch.
- Parallel image augmentation during training.
- Concurrent data fetching to minimize input-output bottlenecks.
- Multi-threaded execution for batch processing and inference.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- **Preprocessing**: High-resolution MRI images require significant computational time.
- **Training**: Limited datasets lead to overfitting and increased training epochs.

- **Real-Time Inference**: Lack of optimization for concurrent processing slows predictions.
- **Data Imbalance**: Skewed distribution of tumor vs. non-tumor images reduces model robustness.

7. Novelty

This study introduces a custom ANN and CNN architecture, allowing a comparative analysis. CNN's feature extraction capabilities outperform traditional ANNs, demonstrating its effectiveness for MRI-based tumor detection.

8. Result Analysis

- **ANN Results**:
  - Training accuracy: 97.13%
  - Validation accuracy: 71.51%
  - Test accuracy: 80.77%
- **CNN Results**:
  - Validation accuracy: 94.00%
  - Test accuracy: 89.00%
- CNN exhibited better generalization and higher precision, recall, and F1 scores.

9. Challenges Faced

- Dataset limitations affected generalization capabilities.
- Computational resources were required to handle high-dimensional data.
- Optimizing hyperparameters like layer configurations and learning rates was time-consuming.

10. Future Directions

1. **Data Augmentation**: Explore advanced techniques to balance the dataset.
2. **Transfer Learning**: Use pre-trained CNN architectures for improved accuracy.
3. **Hybrid Models**: Combine ANN and CNN strengths.
4. **Real-Time Applications**: Develop lightweight, optimized pipelines for faster deployment.

11. Discussion

CNN significantly outperforms ANN in handling high-dimensional MRI data. The results emphasize the importance of feature extraction and model optimization for medical image classification. However, reliance on dataset quality and computational power remains a challenge.

12. Research Gap

- Limited exploration of ensemble methods or hybrid architectures.
- Lack of robust real-time systems integrating threading and parallel processing.

- Underutilization of 3D MRI data for volumetric analysis.

Paper 9: Brain tumour diagnosis from MRI based on Mobilenetv2 optimised by contracted fox optimization algorithm

1. Methodology

The research employs the MobileNetV2 deep learning model, optimized using the Contracted Fox Optimization Algorithm (CFOA), for brain tumor diagnosis from MRI images. The methodology combines:

1. **Dataset**: A publicly available Figshare dataset of 3064 MRI images, comprising three tumor types: glioma, meningioma, and pituitary.
2. **Preprocessing**:
   - CLAHE (Contrast Limited Adaptive Histogram Equalization) for enhancing contrast.
   - Min-Max Normalization for consistent pixel scaling.
   - Data augmentation techniques, including rotation, translation, and scaling.
3. **MobileNetV2 Architecture**:
   - Optimized with CFOA to select optimal hyperparameters (e.g., depth multiplier, input resolution, dropout rate).
   - Reduced computational complexity with Separable Depth-Wise Convolutions and Inverted Residual blocks.
4. **Optimization**:
   - CFOA improves hyperparameter tuning, balancing exploration and exploitation during training.

2. Simulations

Simulations were performed on a system with:

- Intel Core i7-8700K processor.
- 16 GB DDR4 RAM.
- NVIDIA GeForce GTX 1080 Ti GPU. The dataset was split into training (80%) and testing (20%), with MobileNetV2 trained using CFOA-optimized parameters. Results demonstrated superior performance compared to other methods like CNN and VGG19.

3. Equations

1. **CLAHE Redistribution**: $C'(k)=(L-1)\times C(k)/MN$

   where $C(k)$ is the cumulative distribution function, L is intensity levels, and M,N are image dimensions.

2. **Normalization**: $x\_norm=x-min / max-min$
3. **Accuracy**: $ACC=TP+TN / TP+TN +FP+FN$ where TP,TN,FP,FN are true positives, true negatives, false positives, and false negatives, respectively.

4. Threading to Boost Performance

- Parallelize preprocessing (e.g., CLAHE) and normalization across multiple CPU threads.
- Use multi-threaded pipelines to load and augment images concurrently.
- Accelerate MobileNetV2 training with GPU threading (e.g., TensorFlow or PyTorch).

5. Techniques That Use Threading or Parallel Processing

- **Preprocessing**: Distributed histogram equalization and normalization.
- **Optimization**: CFOA leverages exploration-exploitation phases with parallel calculations for evaluating fitness functions.
- **Model Inference**: Utilize multi-threaded batch predictions for real-time analysis.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- High computational cost for preprocessing and augmentation.
- Imbalanced datasets reduce model robustness.
- Overfitting due to limited sample size.
- Difficulty in tuning hyperparameters for optimal performance.

7. Novelty

- Integration of MobileNetV2 with CFOA for hyperparameter optimization.
- Efficient preprocessing pipeline using CLAHE and normalization.
- Demonstrated applicability on a diverse dataset (glioma, meningioma, pituitary tumors).

8. Result Analysis

- **Accuracy**: 97.32%.
- **Precision**: 97.68%.
- **F1-Score**: 86.22%.
- **Sensitivity**: 80.12%. The proposed MN-V2/CFO model outperformed other benchmarks (e.g., CNN, VGG19), validating its potential for clinical applications.

9. Challenges Faced

- Computational overhead from CFOA's iterative optimization process.
- Balancing exploration and exploitation phases.
- Addressing dataset imbalance without sacrificing diagnostic accuracy.

10. Future Directions

1. Incorporate transfer learning with pre-trained models to enhance accuracy.
2. Develop lightweight MobileNetV2 variants for deployment on edge devices.
3. Explore 3D MRI datasets for volumetric analysis.

11. Discussion

The study underscores the effectiveness of combining deep learning with metaheuristic optimization. While MobileNetV2 provides a computationally efficient backbone, CFOA ensures robust hyperparameter selection. The approach addresses common bottlenecks in medical imaging and sets the foundation for real-world deployment.

12. Research Gap

- Limited exploration of real-time implementation on low-power devices.
- Lack of validation on large, diverse datasets from multiple institutions.
- Underutilization of ensemble methods to combine strengths of different models.

Paper 10: Deep-Learning Detection of Cancer Metastases to the Brain on MRI

1. Methodology

The study introduces a Computer-Aided Detection (CAD) pipeline leveraging deep learning for detecting brain metastases in T1-weighted MRI scans. The primary components include:

- **Preprocessing**: Zero-padding, mean subtraction for normalization, and data augmentation (left-right flips).
- **Detection**: Faster R-CNN was used for region proposal and initial detection of metastases.
- **Skull Stripping**: Removed non-brain elements using FreeSurfer to eliminate irrelevant detections.
- **False-Positive Reduction**: Applied RUSBoost, a hybrid undersampling and boosting algorithm, to reduce false-positive predictions.

2. Simulations

- A dataset of 361 scans from 121 patients was used. The training set included 270 scans with 1565 lesions, and the testing set had 91 scans with 488 lesions.
- A holdout validation approach was employed for both the Faster R-CNN and RUSBoost steps.
- Simulations were performed on a system with Intel i7 CPU and NVIDIA GTX 1080 Max-Q GPU.

3. Equations

The Faster R-CNN training involved optimizing the following loss function:

$$L(p_i, t_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*),$$

Where:

- pi: Predicted probability of an anchor being a metastasis.
- p_i^*: Ground truth label (1 for metastasis, 0 otherwise).
- t_i: Predicted bounding box coordinates.
- t_i^*: Ground truth bounding box coordinates.
- L_{cls}: Classification loss (log loss).
- L_{reg}: Regression loss (smooth L_1 loss).
- λ: Balancing parameter, set to 10.

4. Threading to Boost Performance

While the paper does not explicitly discuss threading, the following components could benefit:

- **Preprocessing**: Parallelizing data augmentation.
- **Detection**: Faster R-CNN's region proposal step could use threading for analyzing multiple slices simultaneously.
- **False-Positive Reduction**: RUSBoost can be parallelized to handle subsets of data concurrently.

5. Techniques Using Threading or Parallel Processing

Potential threading opportunities include:

- **Region Proposal**: Multi-threading Faster R-CNN for parallel analysis.
- **Data Augmentation**: Parallel execution for creating augmented datasets.
- **Training Optimization**: Using frameworks like TensorFlow or PyTorch with multi-threading or GPU acceleration for Faster R-CNN.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- **Small Metastases Detection**: Challenges due to size and low contrast-to-background ratios.
- **False Positives**: Misclassification of small blood vessels as metastases.
- **Time-Consuming Manual Interpretation**: Radiologists face increasing workloads due to the complexity of scans.

7. Novelty

The study's novelty lies in combining Faster R-CNN with RUSBoost for high sensitivity and reasonable specificity. It integrates deep learning with classic sampling methods to reduce false positives effectively.

8. Result Analysis

- Achieved **96% sensitivity** with a false-positive rate of **20 per scan**.

- Detected even small metastases, with a noted decrease in detection sensitivity for lesions with low contrast or small size.
- Outperformed many traditional and machine learning approaches in sensitivity and specificity metrics.

9. Challenges Faced

- **False Positives**: Small vessels were often misclassified as metastases.
- **Limited 3D Context**: The method processes slices individually, losing 3D spatial information.
- **Data Limitation**: The dataset, while large, required extensive manual annotations.

10. Future Directions

- Extend the approach to 3D CNNs for better context preservation.
- Explore advanced imaging techniques (e.g., "black-blood imaging") to reduce false positives from blood vessels.
- Incorporate larger datasets to improve model generalization.

11. Discussion

The approach demonstrated the potential of CAD systems in aiding radiologists, significantly improving detection accuracy while reducing manual workloads. However, the method's practical deployment requires further refinement, particularly in minimizing false positives and integrating real-time 3D analysis.

12. Research Gap

- Lack of clinically deployable CAD systems for brain metastases detection.
- Need for enhanced sensitivity to differentiate metastases from similar structures like blood vessels.
- Absence of robust, large-scale datasets for training advanced deep-learning models.

Paper 11: Detection of Brain Tumour based on Features Fusion and Machine Learning

1. Methodology

The paper proposes a four-step automated system for brain tumor detection:

1. **Lesion Enhancement**: Applied histogram matching to enhance MR images, improving contrast for lesion visibility.
2. **Lesion Segmentation**: Used an unsupervised clustering algorithm with five clusters for precise tumor segmentation.
3. **Feature Extraction**: Extracted fused features combining Gabor wavelet features (GWF), histograms of oriented gradient (HOG), local binary patterns (LBP), and segmentation-based fractal texture analysis (SFTA).

4. **Classification**: Classified tumor sub-regions (complete, enhancing, and non-enhancing) using a Random Forest (RF) classifier.

2. Simulations

- Experiments were conducted on benchmark datasets, including BRATS 2012–2015 and ISLES 2015.
- Performance evaluation utilized fivefold cross-validation and 0.5 holdout methods to ensure robustness and reduce overfitting.
- Pixel-based comparisons were made between predicted segmentations and ground truth annotations.

3. Equations

Key equations used in the methodology:

- **Lesion Normalization**:
- $$\text{Normalized Intensity} = \frac{\text{Pixel Intensity - Min}}{\text{Max - Min}}$$
- **Feature Fusion**: Fused feature vector (fv) is created as:

  $$fv = GWF + HOG + LBP + SFTA$$

- **Segmentation Metrics**: Dice Similarity Coefficient (DSC):

  $$DSC = \frac{2TP}{2TP + FP + FN}$$

4. Threading to Boost Performance

Threading could be used in:

- **Lesion Segmentation**: Parallelize cluster formation and centroid updates during the segmentation phase.
- **Feature Extraction**: Simultaneously compute GWF, HOG, LBP, and SFTA features across multiple MRI slices.
- **Classification**: Use multi-threading for the ensemble decision-making process in the Random Forest classifier.

5. Techniques that Use Threading or Parallel Processing

- Parallel processing in histogram computation and clustering can significantly speed up lesion enhancement and segmentation.
- GPU-based implementations for feature extraction and classification using libraries like PyTorch or TensorFlow.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- **Low Contrast in Tumors**: Makes segmentation challenging, especially for gliomas with poorly defined boundaries.
- **Time-Consuming Computations**: High computational cost for multi-feature extraction and fusion.
- **False Positives**: High sensitivity methods often misclassify non-tumorous regions as tumors.

7. Novelty

- Integration of diverse texture features into a fused vector for better tumor characterization.
- Use of an unsupervised clustering algorithm tailored for multi-modal MRI data.
- Achieved high Dice Similarity Coefficients (DSCs) on challenging datasets, outperforming existing methods.

8. Result Analysis

- Achieved **94% DSC** on BRATS 2014 for complete tumor segmentation.
- Outperformed traditional methods like Conditional Random Fields and CNNs in accuracy and sensitivity.
- Pixel-based results showed strong alignment with ground truth annotations.

9. Challenges Faced

- **High Computational Demand**: Feature extraction and segmentation algorithms require optimization for large datasets.
- **Segmentation Errors**: Difficulty in segmenting necrotic tumor cores and separating edema from healthy tissue.

10. Future Directions

- Extend to real-time processing using parallel algorithms for clinical applications.
- Incorporate advanced deep learning models like 3D CNNs for improved segmentation of volumetric data.
- Explore hybrid approaches combining traditional feature-based and deep learning methods.

11. Discussion

The paper demonstrates that feature fusion and RF classification enhance detection accuracy. However, its reliance on handcrafted features may limit adaptability to diverse datasets. The results underline the need for robust, real-time, and automated systems.

12. Research Gap

Lack of Deep Learning Integration

- The proposed method relies on handcrafted feature extraction techniques such as Gabor wavelets, HOG, LBP, and SFTA. While these methods are effective, they may not capture complex patterns as comprehensively as deep learning models, particularly Convolutional Neural Networks (CNNs) or 3D CNNs, which can learn features directly from the data.
- State-of-the-art techniques often use deep learning for end-to-end systems, bypassing the need for manual feature engineering, which is absent in this study.

Limited Generalization to Diverse Datasets

- The algorithm was primarily evaluated on benchmark datasets like BRATS (2012–2015) and ISLES (2015). These datasets, while extensive, do not fully represent the diversity of real-world clinical MRI data, including variations in imaging protocols, tumor types, and patient demographics.
- A lack of external validation on unseen, real-world datasets restricts confidence in the method's robustness and generalizability.

Absence of Volumetric and Spatial Context

- The segmentation process operates in 2D, focusing on individual MRI slices. This ignores the volumetric nature of brain tumors and the spatial continuity across slices, which could provide additional contextual information for better accuracy.
- Modern deep learning methods like 3D CNNs or hybrid models incorporating both 2D and 3D features have shown promise in addressing this limitation.

Computational Complexity

- The feature extraction and fusion process, while detailed, is computationally intensive. This makes the approach less suitable for real-time applications, particularly in resource-constrained environments such as smaller clinics.
- Optimizing the system using parallel processing, threading, or GPU acceleration could address these challenges but has not been explored in the study.

Over-Sensitivity to Tumor Contrast

- The method struggles with low-contrast lesions, such as those found in gliomas or necrotic regions. This is a common limitation in handcrafted feature-based approaches, which are less adaptive to such variations compared to deep learning systems.

Lack of Clinical Validation

- Although the study shows promising results in terms of segmentation accuracy and Dice Similarity Coefficients, it lacks real-world clinical trials to validate its effectiveness in assisting radiologists or improving patient outcomes.
- Integrating the method into clinical workflows and assessing its impact on diagnosis, treatment planning, and monitoring remains unexplored.

Limited Focus on Automation

- The system requires manual preprocessing steps, such as histogram matching and unsupervised clustering for segmentation. Fully automated systems would be more practical in clinical settings, reducing the need for expert intervention and manual adjustments.

Incomplete Handling of Multimodal Data

- While the study considers multiple MRI modalities (e.g., T1, T2, Flair), it does not explore advanced fusion techniques to fully leverage the complementary information from these modalities.
- Techniques such as attention mechanisms or modality-specific feature extraction networks could enhance the fusion process.

Need for Explainability

- The study does not address the interpretability of the results, which is critical for clinical adoption. Radiologists and clinicians often require transparent models that provide insights into why a particular region was classified as tumorous.

Absence of Comparative Analysis with Advanced Techniques

- Although the study compares its results with some traditional methods, it does not benchmark against the latest deep learning frameworks or hybrid approaches, leaving uncertainty about its relative performance in the current state of the field.

Addressing these research gaps would involve incorporating deep learning for feature extraction, expanding evaluations to more diverse datasets, integrating spatial and volumetric context, optimizing computational efficiency, and validating the method in clinical settings. Bridging these gaps will enhance the system's adaptability, efficiency, and real-world applicability, moving closer to an automated and robust solution for brain tumor detection.

Paper 12: Employing Deep Learning and Transfer Learning for Accurate Brain Tumour Detection

1. Methodology

The study explores the application of transfer learning for brain tumor detection using MRI images. Four pre-trained models—ResNet152, VGG19, DenseNet169, and MobileNetv3—were fine-tuned and applied to classify MRI images into four categories: meningioma, glioma, pituitary tumor, and normal brain. Key steps include:

- **Data Augmentation**: Techniques like rotations, flips, and zooming were applied using Keras' `ImageDataGenerator` to expand the dataset and improve model robustness.
- **Model Training**: The models were trained on an 80-20 split of the Kaggle dataset using five-fold cross-validation. A batch size of 128 and Adam optimizer with a learning rate of 0.001 were used.

- **Evaluation Metrics**: Accuracy, precision, recall, and F1 scores were computed to assess the models.

2. Simulations

- Experiments were conducted on Google Colab with access to high-end GPUs.
- All models were trained for 50 epochs. Cross-entropy loss was used for both training and testing datasets.
- MobileNetv3 exhibited the most stable training, achieving the highest validation accuracy of **98.52%**.

3. Equations

- **Cross-Entropy Loss**:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(\hat{y}_{ij}),$$

where yij is the ground truth label, y^ij is the predicted probability, N is the number of samples, and C is the number of classes.

4. Threading to Boost Performance

Although the paper does not explicitly mention threading, performance can be improved by:

- **Data Augmentation**: Parallelizing the augmentation process.
- **Model Training**: Distributing computations across multiple GPUs or using multi-threaded data pipelines to reduce I/O bottlenecks.
- **Prediction**: Employing batch-level parallelism for inference.

5. Techniques That Use Threading or Parallel Processing

- **Distributed Training**: Leveraging TensorFlow's `tf.data` API for multi-threaded data loading.
- **Inference Optimization**: Batch processing multiple MRI scans simultaneously on GPUs.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- **Imbalanced Datasets**: The dataset had an unequal distribution of tumor classes, which may affect model performance.
- **Computational Overhead**: Fine-tuning deep networks like ResNet152 requires significant computational resources.
- **Overfitting**: DenseNet169 exhibited overfitting due to its complex architecture.

7. Novelty

- Fine-tuning multiple pre-trained transfer learning models for classifying brain tumors.
- MobileNetv3 achieved **99.75% training accuracy**, showcasing its potential for deployment on resource-constrained devices.

8. Result Analysis

- ResNet152 achieved the best overall accuracy (**98.86%**) but required more training time.
- MobileNetv3 demonstrated the highest training efficiency with minimal fluctuations in accuracy and loss.
- VGG19 and DenseNet169 performed well but were less efficient compared to MobileNetv3.

9. Challenges Faced

- **Data Limitation**: Although augmented, the dataset size was still a limiting factor for generalizability.
- **Overfitting**: Some models, like DenseNet169, showed overfitting tendencies despite data augmentation.
- **Resource Intensive**: Training deep networks on large datasets required significant computational power.

10. Future Directions

- **Model Generalization**: Testing the models on external datasets and modalities (e.g., CT scans) to improve robustness.
- **Lightweight Models**: Exploring smaller architectures for real-time deployment in clinical settings.
- **Advanced Preprocessing**: Incorporating noise reduction and multi-modal fusion techniques for better segmentation.

11. Discussion

The study highlights the potential of transfer learning for accurate brain tumor detection. MobileNetv3 emerges as a lightweight and efficient model, making it suitable for real-world deployment. However, the dependence on pre-existing datasets and limited generalization across modalities remain key challenges.

12. Research Gap

- **Limited Dataset Diversity**: The Kaggle dataset may not represent real-world variability in MRI data.
- **Overlooked Spatial Context**: The models analyze images slice-wise, ignoring the 3D context of MRI scans.
- **Lack of Explainability**: The study does not provide insights into the interpretability of predictions, which is crucial for clinical adoption.
- **Resource Constraints**: Computational requirements for training large models are prohibitive for many institutions.

Paper 13: Improved Multiclass Brain Tumour Detection Using Convolutional Neural Networks and Magnetic Resonance Imaging

1. Methodology

The proposed study utilises a CNN-based model to classify brain tumours into four distinct categories: Normal, Glioma, Meningioma, and Pituitary. The methodology emphasises efficient and accurate medical image analysis using deep learning.

- **Data Collection**: MRI data was obtained from publicly available datasets such as Kaggle, Figshare, and the Sartaj dataset. The combined dataset contained 7022 images classified into four tumor types.
- **Preprocessing**: Images were resized to 224×224 dimensions and converted to RGB format for consistency. Batch size was set to 32.
- **Model Architecture**: The CNN model included:
  - Five convolutional layers, each with increasing filters (32, 64, 128, 256, and 512).
  - Max-pooling layers after each convolutional layer to downsample feature maps.
  - A flattening layer to transform multi-dimensional feature maps into a 1D vector.
  - Two dense layers: one with 128 neurons and another with 4 neurons (for multi-class classification) using the SoftMax activation function.
- **Optimization**: The model was trained using the Adam optimizer, with a categorical cross-entropy loss function. Training was conducted for 50 epochs with a learning rate of 1e-4.
- **Evaluation Metrics**: Precision, recall, F1-score, and confusion matrix were used to evaluate the model's performance.

2. Simulations

Simulations were performed to evaluate the proposed CNN model's efficiency:

- **Training Setup**:
  - The dataset was split into 80% training and 20% testing sets.
  - Images were processed using data augmentation to enhance generalization and prevent overfitting.
- **Hardware**: Simulations were run on a high-performance computing environment, potentially using GPUs for faster computation.
- **Training and Validation**:
  - Training accuracy and loss were monitored over 50 epochs, demonstrating convergence to optimal performance.
  - Validation datasets were used to ensure model generalisation.
- **Visualisation**: Accuracy and loss curves were plotted to showcase the model's performance during training and validation.

3. Equations

Several performance metrics were calculated to evaluate the CNN model:

1. **Accuracy**:

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

2. **Precision**:

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

3. **Recall**:

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

4. **F1-Score**:

$$F1 = 2 \times \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

4. Threading to Boost Performance

Threading can be utilised to enhance the performance of the proposed workflow:

- **Preprocessing**: Parallel threads can load and preprocess MRI images, reducing data preparation time.
- **Training**: Utilise multi-threaded data loaders for simultaneous image processing and model training.
- **Inference**: Perform parallel predictions on batches of MRI scans to speed up tumor classification in real-time.
- **Libraries**: Tools like Python's `threading` module or parallel processing frameworks in TensorFlow and PyTorch can be leveraged for implementation.

5. Techniques That Use Threading or Parallel Processing

- **Data Parallelism**: Splitting batches of MRI data across multiple GPUs to accelerate training.
- **Model Parallelism**: Dividing CNN layers across multiple computational units to enhance throughput.
- **Threaded Data Loaders**: Using frameworks like TensorFlow’s `tf.data` or PyTorch’s `DataLoader` to enable concurrent data augmentation and loading.

- **Asynchronous Processing**: Overlapping preprocessing and training steps using multi-threading.

6. Existing Bottlenecks in MRI-Based Brain Tumour Detection Workflows

1. **High Data Volume**: MRI scans are high-resolution, requiring significant storage and computational resources.
2. **Feature Variability**: Tumours exhibit variations in size, shape, and intensity, complicating accurate detection.
3. **Manual Annotation**: Reliance on radiologists for tumour labelling introduces subjectivity and delays.
4. **Computational Costs**: Deep learning models demand substantial hardware resources, which can be a bottleneck in resource-limited settings.
5. **Real-Time Processing**: Existing workflows struggle with providing real-time results due to computational overheads.

7. Novelty

- The proposed model achieved **95.44% accuracy**, surpassing existing techniques.
- It effectively classifies brain tumours into four types, addressing a significant challenge in multi-class medical diagnosis.
- Utilisation of an optimised CNN architecture tailored for medical imaging improves diagnostic accuracy and efficiency.

8. Result Analysis

- **Performance Metrics**:
  - Accuracy: 95.44%.
  - Recall: 95%.
  - Precision: High precision across all four classes.
- **Confusion Matrix**: Demonstrated strong performance in correctly classifying the four tumor categories, with minimal misclassifications.
- **ROC Curves**: Highlighted the model's ability to distinguish between classes with high sensitivity and specificity.

9. Challenges Faced

- **Dataset Diversity**: Limited variability in training data could hinder model generalisation.
- **Misclassification**: Tumour classes with similar features occasionally led to incorrect predictions.
- **Infrastructure Requirements**: Training a deep CNN model requires substantial computational power.

10. Future Directions

- **Dataset Expansion**: Incorporate more diverse MRI datasets to improve model robustness.

- **Transfer Learning**: Adapt pre-trained models to further enhance performance on medical imaging tasks.
- **Real-Time Applications**: Optimise the model for deployment in real-time diagnostic systems.
- **Hybrid Models**: Explore combining CNNs with attention mechanisms or capsule networks for refined detection.
- **Integration**: Develop a pipeline integrating MRI classification with other diagnostic tools for comprehensive analysis.

11. Discussion

The proposed CNN model is a promising step toward automating brain tumour diagnosis, achieving high accuracy and reducing manual workload. However, the study acknowledges limitations in dataset diversity and computational demand. The model demonstrates the potential for real-world applications, but further validation on diverse datasets is necessary. Optimising for real-time performance and expanding its scope to include other medical imaging modalities could enhance its clinical utility.

12. Research Gap

Data Limitations:

- Datasets lack diversity in demographics, MRI scanner types, and tumor stages, limiting model generalizability.
- Imbalanced classes and inconsistent annotations hinder effective training and evaluation.

Model Limitations:

- Current models struggle with small or irregular tumours and lack interpretability for clinical use.
- Variability in tumour characteristics, such as shape and size, is not well addressed.

Workflow Challenges:

- Real-time processing and full automation of MRI workflows remain unachieved.
- High computational demands limit deployment in low-resource settings.

Standardisation Issues:

- Inconsistent evaluation metrics and limited use of multi-modal MRI data make cross-comparison difficult.

Clinical Adoption Barriers:

- Few models are validated in real-world settings or integrated into hospital systems.
- Ethical concerns around data privacy and accountability persist.

Underutilized Technologies:

- Techniques like hybrid architectures, federated learning, and threading are underexplored but hold promise for efficiency and accuracy.

Validation Gaps:

- Most studies lack cross-institutional validation or comparisons with radiologists to establish clinical relevance.

Paper 14 :MRI Brain Tumor Detection Using Deep Learning and Machine Learning Approaches

1. Methodology

The paper proposes a hybrid approach combining deep learning and machine learning techniques for brain tumour detection.

- **Dataset**: A publicly available Kaggle dataset with 255 MRI images, categorised into 98 healthy and 155 tumorous brain slices.
- **Preprocessing**: Adaptive Contrast Enhancement Algorithm (ACEA) was used to enhance image quality, and a median filter was applied to remove noise while preserving edges.
- **Segmentation**: The fuzzy c-means (FCM) algorithm was employed to segment the tumour regions.
- **Feature Extraction**: Gray-level co-occurrence matrix (GLCM) was used to extract features such as energy, entropy, contrast, and mean from the segmented images.
- **Classification**: The proposed Ensemble Deep Neural Support Vector Machine (EDN-SVM) model classified the images as normal or tumorous. The EDN-SVM combines neural networks with SVM for improved generalisation and accuracy.

2. Simulations

- Implemented in Python 3.7, the model was evaluated on MRI scans with various preprocessing, segmentation, and classification techniques.
- The simulations aimed to reduce computation time and improve accuracy, sensitivity, and specificity.
- Comparisons were made with existing methods like CNN, Random Forest Classifier, and ANN.

3. Equations

The study employed the following performance metrics:

1. **Accuracy**: Accuracy=TP+TN / TP+TN+FP+FN
2. **Sensitivity**: Sensitivity= TP / TP+FN
3. **Specificity**: **Specificity = TN/TN+FP**
4. **Jaccard Coefficient**: $J=A \cap B / A \cup B$
5. **Peak Signal-to-Noise Ratio (PSNR)**:

$$PSNR = 10 \cdot \log_{10}\left(\frac{255^2}{MSE}\right)$$

4. Threading to Boost Performance

While the paper doesn't explicitly focus on threading, threading can be applied to enhance performance in:

- **Preprocessing**: Parallelize ACEA and median filtering for faster image enhancement.
- **Segmentation**: Implement multi-threaded FCM for large datasets.
- **Feature Extraction**: Utilize parallel computation for GLCM-based feature extraction across images.
- **Classification**: Train EDN-SVM on multiple GPUs using data parallelism.

5. Techniques That Use Threading or Parallel Processing

- **Data Parallelism**: Distribute data batches across multiple processors during training.
- **Model Parallelism**: Split EDN-SVM layers across processors.
- **Parallel Feature Extraction**: Compute GLCM features for multiple images simultaneously.
- **Threaded Data Loading**: Use Python's `concurrent.futures` or deep learning libraries' built-in threading support to preprocess data in parallel.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- **High Variability**: Tumors differ in size, shape, and intensity, complicating detection.
- **Noise and Artifacts**: MRI images often contain noise that can impact segmentation.
- **Manual Processing**: Preprocessing and annotation are labor-intensive and prone to human error.
- **Computational Costs**: Large-scale processing of MRI scans requires significant resources.

7. Novelty

- The study introduced the EDN-SVM model, integrating neural networks with SVM to enhance classification accuracy.
- The hybrid model achieved superior performance (97.93% accuracy, 92% sensitivity, and 98% specificity) compared to traditional classifiers.

8. Result Analysis

- **Accuracy**: Achieved 97.93%, outperforming CNNs and other classifiers.
- **Sensitivity and Specificity**: High sensitivity (92%) ensured minimal false negatives, and specificity (98%) indicated accurate exclusion of non-tumorous cases.
- **Efficiency**: Lower computational time compared to existing methods like R-CNN.

9. Challenges Faced

- **Dataset Imbalance**: A limited dataset (255 MRI images) may hinder the model's generalization.
- **Computational Demand**: High complexity of EDN-SVM increases resource requirements.
- **Segmentation Errors**: Noise and overlapping features in MRI scans sometimes lead to inaccurate segmentation.

10. Future Directions

- **Enhanced Data Diversity**: Incorporate larger and more varied datasets.
- **3D Analysis**: Extend the methodology to 3D MRI scans for more accurate tumor segmentation.
- **Color Images**: Adapt the model for multi-channel color image analysis.
- **Real-Time Systems**: Develop integrated systems for real-time brain tumor detection in clinical settings.

11. Discussion

The proposed EDN-SVM outperformed conventional methods by combining the strengths of deep learning and SVMs. The automated approach reduced feature extraction time while improving classification accuracy. However, further validation on larger datasets and real-world scenarios is necessary to enhance clinical applicability.

12. Research Gap

- **Dataset Diversity**: Lack of diverse and large-scale MRI datasets restricts model robustness.
- **Automation**: Many steps, such as annotation and segmentation, still require manual intervention.
- **Real-Time Processing**: Current methods are not optimized for real-time detection.
- **Explainability**: Models like EDN-SVM lack interpretability, making clinical adoption challenging.
- **Scalability**: High computational demand limits usage in resource-constrained environments.

Paper 15 : Optimized Brain Tumor Detection: A Dual-Module Approach for MRI Image Enhancement and Tumor Classification

1. Methodology

The study proposes a two-module approach for brain tumor detection and classification:

1. **Image Enhancement Module**:
    - **Techniques Used**: Adaptive Wiener filtering for noise reduction, Independent Component Analysis (ICA) for contrast normalization, and Radial Basis Function (RBF) Neural Networks for pre-classification enhancement.

    - **Steps**: Noise suppression, resolution enhancement, and image coherence improvement.
2. **Tumour Classification Module**:
    - **Techniques Used**: Support Vector Machines (SVM) for tumor segmentation and classification.
    - **Workflow**: Preprocessed images are segmented and classified into tumor types (gliomas, meningiomas, pituitary tumors).

The method was validated using the CE-MRI dataset, encompassing 3064 MRI images across three tumour types.

2. Simulations

- **Dataset**: CE-MRI dataset with 3064 images, manually annotated by radiologists.
- **Performance Metrics**: Evaluated using parameters such as sensitivity, specificity, Dice Similarity Coefficient (DSC), and accuracy.
- **Comparison**: The proposed method was compared with traditional enhancement and classification techniques, demonstrating superior performance.

3. Equations

1. **Adaptive Wiener Filtering**:

   $$g(x,y) = w(x,y) \cdot [f(x,y) - m(x,y)] + m(x,y)$$

   Where w(x,y) is the filter, m(x,y) is the local mean, and f(x,y) is the input image.
2. **PSNR for Image Quality**:

   $$PSNR = 20 \cdot \log_{10}\left(\frac{R}{\sigma}\right)$$

3. **Sensitivity**:

   $$\text{Sensitivity} = \frac{TP}{TP + FN}$$

4. **Specificity**:

   $$\text{Specificity} = \frac{TN}{TN + FP}$$

5. **Dice Similarity Coefficient**:

   $$DSC = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}$$

4. Threading to Boost Performance

Threading can be incorporated at various stages:

- **Preprocessing**: Parallelize noise reduction and contrast enhancement for multiple images.

- **Segmentation**: Implement multi-threaded ICA computations across slices.
- **Classification**: Parallelize SVM computations for large datasets.

5. Techniques that Use Threading or Parallel Processing

- **Data Parallelism**: Distribute image batches across processors.
- **Feature Extraction Parallelism**: Compute RBF and ICA features for multiple images simultaneously.
- **Model Parallelism**: Spread SVM training across GPUs/cores.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

1. **Noise and Artifacts**: MRI images often suffer from noise, impacting segmentation accuracy.
2. **Manual Annotation**: Reliance on expert radiologists for annotations increases time and subjectivity.
3. **Computational Costs**: High-resolution images demand significant resources for processing.
4. **Contrast Variability**: Inconsistent contrast across MRI images complicates tumor classification.

7. Novelty

- **Dual-Module Design**: Combining image enhancement and classification improves tumor detection.
- **Techniques Used**: Innovative application of ICA and adaptive Wiener filtering ensures superior image quality.
- **Metrics Achieved**: Sensitivity (0.99), Specificity (0.99), Accuracy (0.989), and Dice Similarity Coefficient (0.981).

8. Result Analysis

- **Image Quality**: Improved contrast by 40% and PSNR by 3dB across all tumor types.
- **Classification Accuracy**: Outperformed existing methods with an accuracy of 98.9%.
- **Processing Time**: Averaged 0.43 seconds per image, significantly faster than alternatives.

9. Challenges Faced

1. **Dataset Limitations**: CE-MRI dataset lacks diversity in demographics and imaging conditions.
2. **Noise Suppression**: Achieving optimal denoising without losing critical tumor features.
3. **Computational Overhead**: High processing demands due to complex enhancement and classification steps.

10. Future Directions

1. **Deep Learning Integration**: Combine ICA with CNNs for end-to-end classification.
2. **Dataset Expansion**: Incorporate larger, multi-center datasets for broader validation.
3. **Real-Time Processing**: Optimize methods for clinical deployment in real-time systems.
4. **Multimodal Analysis**: Use additional imaging modalities like PET or CT for enhanced accuracy.

11. Discussion

The dual-module approach successfully addressed challenges in MRI-based brain tumor detection by improving contrast, reducing noise, and delivering highly accurate classification. While promising, scalability and generalisation remain areas for improvement. Enhanced datasets and integration of advanced deep learning models can further refine the methodology.

12. Research Gap

1. **Limited Dataset Diversity**: Current datasets do not represent varied demographics or MRI conditions.
2. **Real-Time Deployment**: Lack of focus on real-time performance for clinical usage.
3. **Automation**: Segmentation and classification steps still require significant manual intervention.
4. **Explainability**: The model lacks transparency, hindering clinician trust and adoption.

Paper 16: MRI-based Brain Tumor Image Detection Using CNN-Based Deep Learning Method

1. Methodology

The paper proposes a convolutional neural network (CNN)-based method for brain tumor detection using MRI images. Key steps include:

- **Data Collection**: Images from the BraTS 2020 dataset were used, with T1, T2, and FLAIR modalities. Tumor and non-tumor images were classified into two categories.
- **Preprocessing**:
  - All images were resized to 128×128×3 for uniformity.
  - Gaussian blur and high-pass filters were applied to reduce noise and enhance image features.
- **Model Architecture**:
  - A 9-layer CNN with 14 stages was designed.
  - Convolutional layers with 32 and 64 filters were followed by ReLU activation and batch normalization.
  - Max-pooling was employed for dimensionality reduction.
  - Fully connected layers (512 hidden units in the first layer and 2 output units in the final layer) were added for classification.
- **Training Configuration**:

- The model used the RMSProp optimizer, softmax activation for the output layer, and the categorical cross-entropy loss function.
- A split ratio of 9:1 was employed for training and testing, with 11 epochs used for training.

2. Simulations

- **Dataset**: The BraTS dataset comprised 2892 MRI images, split into 2473 training images and 273 testing images.
- **Environment**: Training was conducted using TensorFlow and Keras libraries in Python.
- **Evaluation**: Metrics such as accuracy, precision, and loss were computed for both training and validation datasets.

3. Equations

- **Accuracy (AC)**: AC=TP+TN / TP+TN+FP+FN
- **Dice Score**: Dice(P,T)=2 ∣P∩T∣ / ∣P∣ + ∣T∣

  Where PPP is the predicted segmentation and T is the ground truth.

4. Threading to Boost Performance

- Threading can be implemented in:
  - **Preprocessing**: Parallelization of resizing and filtering processes across multiple cores.
  - **Model Training**: Batch processing can be parallelized using GPU cores for improved computation speed.
  - **Inference**: Multi-threaded pipelines can enable faster predictions for real-time clinical applications.

5. Techniques That Use Threading or Parallel Processing

- The model leverages GPU parallelism for CNN layers during convolution and pooling.
- TensorFlow and Keras libraries optimize data loading and batch processing.
- Parallel computations during max-pooling and fully connected layers enhance speed.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- **Data Imbalance**: Limited representation of non-tumor images can affect model generalization.
- **Small Tumor Detection**: Challenges in accurately identifying small or low-contrast tumors.
- **Computational Cost**: High resource requirements for training deep CNN models.

7. Novelty

- Introduction of a 9-layer CNN optimized for MRI-based brain tumor classification.
- Achieved an accuracy of 99.74%, surpassing previous methods.
- Utilized techniques like batch normalization and ReLU activation to enhance model stability and performance.

8. Result Analysis

- **Performance**: The CNN model achieved 99.74% accuracy, outperforming prior approaches by Seetha et al. (97.50%) and Tonmoy Hossain et al. (97.87%).
- **Validation**: Results indicated high sensitivity and specificity across tumor classes, ensuring reliable classification.

9. Challenges Faced

- **Overfitting**: Addressed by dropping approximately 5% of the dataset during training.
- **Data Noise**: Overcame noise using preprocessing techniques like Gaussian blur.
- **Computational Efficiency**: Achieving high accuracy with limited epochs (11) required careful hyperparameter tuning.

10. Future Directions

- Expanding datasets to include more diverse tumor types and imaging modalities.
- Developing lightweight CNN architectures for real-time deployment on mobile devices.
- Exploring explainable AI techniques to interpret model predictions and improve clinical adoption.

11. Discussion

The study demonstrated the potential of CNNs for automated brain tumor detection from MRI images. The proposed model achieved high accuracy and efficiency, highlighting its applicability in clinical diagnostics. However, further improvements in scalability, interpretability, and dataset diversity are needed for widespread adoption.

12. Research Gap

- Limited focus on rare tumor types and non-tumor classifications.
- Insufficient exploration of lightweight models for resource-constrained settings.
- Lack of interpretability tools to assist medical professionals in decision-making.

Paper 17: Detection of Tumors on Brain MRI Images Using the Hybrid Convolutional Neural Network Architecture

1. Methodology

- **Hybrid Model Development**: The study uses the ResNet50 architecture as a base. The last 5 layers are removed and replaced with 10 new layers, including ReLU activation, batch normalization, dropout, fully connected layers, and softmax classification.
- **Dataset**: MRI images from Kaggle are categorized into tumor (155 images) and non-tumor (98 images). Images are resized to 224×224 pixels for standardization.
- **Training Configuration**:
    - Mini-batch size: 16
    - Maximum epochs: 5
    - Initial learning rate: 1×10−41 \times 10^{-4}1×10−4
    - Optimizer: Stochastic Gradient Descent with Momentum (SGDM)

2. Simulations

- **Hardware**: Training was conducted using a system with an i7 processor, GPU, and 8GB RAM.
- **Evaluation Metrics**: Confusion matrix, accuracy, sensitivity, specificity, and F1 score were used to evaluate model performance.

3. Equations

- **Accuracy**: $\text{Accuracy}=TP+TN \ / \ TP+TN+FP+FN$
- **Sensitivity**: $\text{Sensitivity}=TP \ / \ TP+FN$
- **Specificity**: $\text{Specificity}=TN \ / \ TN+FP$
- **F1 Score**: $F1=2\times (\text{Precision}\times\text{Recall} \ / \ \text{Precision}+\text{Recall})$
- **Pooling Layer**: $S=(w1-f+2p) \ / \ s+1$

    Where w1 is the input width, f is the filter size, ppp is the padding, and s is the stride.

4. Threading to Boost Performance

- **Preprocessing**: Parallelize resizing and normalization tasks to improve speed.
- **Training**: Utilize GPU threading for batch processing and convolutional operations.
- **Inference**: Multi-threaded pipelines allow simultaneous classification of multiple MRI images.

5. Techniques That Use Threading or Parallel Processing

- **Data Loading**: Parallel data preprocessing using frameworks like TensorFlow/Keras.
- **Convolutional Layers**: GPU-based acceleration for convolutional and pooling operations.
- **Dropout Regularization**: Parallel implementation to manage random neuron deactivation during training.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- **Data Limitations**: Small dataset sizes increase overfitting risk.
- **Manual Annotations**: Labor-intensive and error-prone for MRI image labeling.
- **Hardware Constraints**: High computational demands for training deep models.

7. Novelty

- The hybrid model integrates 10 additional layers into the ResNet50 architecture, improving classification accuracy.
- The study achieves a remarkable accuracy of 97.01%, outperforming traditional ResNet50 and other CNN architectures.

8. Result Analysis

- **Improved Model**:
  - Accuracy: 97.01%
  - Sensitivity: 94.7%
  - Specificity: 100%
  - F1 Score: 96.9%
- Performance comparison with baseline models:
  - ResNet50: 92.54%
  - DenseNet201: 91.04%
  - AlexNet: 89.55%
  - GoogleNet: 71.64%
- The hybrid model achieved the highest accuracy with the lowest false negative rate.

9. Challenges Faced

- **Overfitting**: Mitigated using dropout and data augmentation.
- **Computational Requirements**: Limited hardware capabilities posed challenges for large-scale data processing.
- **Small Tumor Detection**: Difficulty in accurately classifying small or indistinct tumor regions.

10. Future Directions

- **Dataset Expansion**: Increase dataset size and diversity to improve model robustness.
- **Lightweight Architectures**: Develop efficient models suitable for mobile or edge devices.
- **Explainable AI**: Incorporate methods to interpret and visualize CNN decisions for clinical applications.

11. Discussion

- The proposed hybrid ResNet50 model demonstrates superior performance in brain tumor classification, achieving high accuracy and reliability. The study highlights the

potential of deep learning in automating diagnostic workflows, reducing manual effort and errors.
- Limitations include the dataset size and lack of multi-modal imaging data. The study also notes the need for real-time deployment solutions in clinical settings.

12. Research Gap

- **Limited Dataset**: The dataset contains only two classes (tumor and non-tumor) and lacks multi-class classification.
- **Real-Time Processing**: Insufficient focus on optimizing models for real-time clinical use.
- **Interpretability**: Lack of visualization techniques to explain model predictions.

Paper 18:Research on Feature Extraction of Tumor Image Based on Convolutional Neural Network

1. Methodology

- **Feature Extraction Framework**:
  - Introduced a multi-channel input CNN to extract texture features from tumor CT images.
  - Used local binary pattern (LBP) algorithms with rotation invariance for initial texture feature extraction.
  - Developed two CNN models, Xception and DenseNet, to enhance feature extraction and classification accuracy.
- **Dataset**:
  - The study utilized 500 ultrasound tumor images (417 malignant and 83 benign cases) from the CANCER-CAPTAC-GBM database.
- **Preprocessing**:
  - Images were scaled uniformly using autocorrelation functions to ensure consistency.
  - Four data block sizes (13×13, 15×15, 17×17, and 19×19) were experimented with, identifying 17×17 as optimal.
- **Model Training**:
  - Input image blocks were divided into positive (tumor) and negative (non-tumor) samples, ensuring balanced training.

2. Simulations

- **Hardware Setup**:
  - Models were trained on systems with sufficient computational resources (details not explicitly mentioned).
- **Experimentation**:
  - Compared CNNs (Xception and DenseNet) with LBP algorithms across three sample datasets.

- Repeated 150 training iterations to minimize variability and validate the robustness of results.

3. Equations

- **Local Binary Pattern (LBP)**:

$$LBP_{N,R} = \sum_{i=0}^{N-1} 2^i s(g_i - g_c); \; s(x) = \begin{cases} 1 & |x| \geq T_{LBP} \\ 0 & |x| < T_{LBP} \end{cases}$$

  Where gig_igi is the gray value of neighborhood pixels, gc is the central pixel, and T_LBP is the threshold.

- **Accuracy**: ACC=TP+TN / TP+TN+FP+FN
- **Specificity**: SPE=TN / TN+FP
- **Sensitivity**: SEN=TP / TP+FN

4. Threading to Boost Performance

- **Preprocessing**:
  - Parallelize LBP feature extraction and scaling processes.
- **Model Training**:
  - Utilize GPU-based threading for faster training and gradient calculations.
- **Inference**:
  - Optimize feature map processing through multi-threaded operations.

5. Techniques That Use Threading or Parallel Processing

- **Xception Model**:
  - Separates spatial and cross-channel correlations to simplify feature extraction.
  - Implements depth-wise separable convolutions for reduced computational cost.
- **DenseNet Model**:
  - Employs dense concatenation of feature maps to improve gradient flow and feature reuse.
  - Reduces the number of parameters through efficient layer-to-layer connections.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

- **Data Imbalance**: Disproportionate representation of malignant vs. benign cases impacts model generalization.
- **Preprocessing Overheads**: Scaling and rotation invariance in LBP require extensive computation.

- **Limited Sample Size**: Small datasets hinder the training of complex CNN architectures.

7. Novelty

- Integrated rotation-invariant LBP with CNNs to enhance shallow texture feature extraction.
- Developed hybrid Xception and DenseNet models to improve feature representation and classification accuracy.
- Achieved an average classification accuracy of 99.7%, surpassing traditional LBP methods.

8. Result Analysis

- **Performance Metrics**:
  - Xception and DenseNet achieved superior recognition rates compared to LBP algorithms.
  - Average AUC: 99.7%
- **Comparative Results**:
  - CNNs consistently outperformed LBP in feature extraction and classification across different databases.

9. Challenges Faced

- **Feature Overlap**: Difficulty distinguishing between benign and malignant features due to texture similarities.
- **Computational Intensity**: High resource requirements for training deep CNN models.
- **Generalization**: Limited dataset diversity posed challenges for broader applicability.

10. Future Directions

- **Dataset Expansion**:
  - Incorporate more diverse imaging modalities, such as MRI and PET.
- **Algorithm Optimization**:
  - Enhance CNN architectures for resource-efficient training and real-time application.
- **Explainability**:
  - Develop methods to interpret CNN decision-making for clinical use.

11. Discussion

- The combination of LBP and CNN provides a robust framework for tumor classification. While LBP extracts shallow texture features effectively, CNNs enhance deeper feature representation.

- Models demonstrated high accuracy and sensitivity but faced challenges in computational efficiency and dataset limitations.

12. Research Gap

- **Real-Time Deployment**: Limited focus on optimizing CNNs for real-time applications.
- **Scalability**: Lack of exploration in scaling models for larger datasets.
- **Explainability**: Insufficient tools to interpret CNN predictions for medical professionals.

Paper 19: Detection of brain tumors from MRI images base on deep learning using hybrid model CNN and NADE.

1. Methodology

The study proposes a hybrid architecture combining Convolutional Neural Networks (CNN) and Neural Autoregressive Distribution Estimation (NADE) to improve the accuracy and efficiency of brain tumor classification from MRI images. The methodology is organized into three stages:

1. **Density Estimation:**
    - The NADE model estimates the joint probability distribution of MRI pixel intensities. This step reduces the effects of imbalanced datasets and noise, which are common challenges in medical imaging.
    - NADE processes input MRI slices to produce denoised and enhanced outputs that are then used for feature extraction.
    - It also handles variations in tumor shapes and sizes, improving robustness.
2. **Feature Extraction:**
    - Two parallel CNNs extract features from:
        - The raw MRI image.
        - The output of the NADE model.
    - This dual-pathway design enables complementary feature extraction, where one CNN focuses on raw patterns while the other processes enhanced, denoised inputs.
3. **Classification:**
    - Features extracted by the CNNs are merged and passed through a fully connected network.
    - Activation functions used include ReLU for the hidden layers and softmax for the output layer.
    - Cross-entropy loss is employed for optimization during training, and six-fold cross-validation ensures robust performance assessment.

2. Simulations

Simulations were conducted using a dataset of 3064 T1-weighted contrast-enhanced MRI images from 233 patients. These images were resized to 64×64 pixels to reduce computational costs. Key simulation details:

- **Software Environment:** The hybrid model was developed using Python (version 3.6.9) with TensorFlow and Keras libraries.
- **Hardware:** Simulations were executed on Google Colab, which provides access to GPUs like Nvidia K80, T4, and P100. This cloud-based setup ensured scalability and efficiency.
- **Validation:** Six-fold cross-validation was used to train and test the model, ensuring reliability in performance metrics.

3. Equations

Key equations in the study include:

1. **Autoregressive Property of NADE:**

$$p(x) = \prod_{d=1}^{D} p(x_d | x_{<d})$$

Where $x_{<d}$ refers to all variables preceding x_d in the sequence. This ensures that the joint distribution is factorized and computationally tractable.

2. **Log Probability Loss Function:**

$$l(x) = -\sum_{d=1}^{D} \log p(x_d | x_{<d})$$

This loss function measures the discrepancy between the predicted and actual distributions.

3. **Gaussian Mixture Model for Real-Valued Data:**

$$p(x_d | x_{<d}) = \sum_{c=1}^{C} p_{d,c} \mathcal{N}(x_d; \mu_{d,c}, \sigma_{d,c}^2)$$

1. Here, $p_{d,c}$ is the mixture weight, and $\mathcal{N}$ represents a Gaussian distribution with mean $\mu_{d,c}$ and variance $\sigma_{d,c}^2$.

These equations underpin the NADE model's ability to estimate density distributions accurately.

4. Threading to Boost Performance

Threading is essential for optimizing computational performance, particularly for tasks like:

1. **Preprocessing:** Parallel resizing, denoising, and segmentation of MRI slices.

2. **Dual CNN Training:** Separate CNNs can process raw and NADE-enhanced images concurrently, leveraging GPU threading.
3. **Data Loading:** Threaded data pipelines can accelerate the flow of images into the model during training.

Threading also ensures that GPU resources are fully utilized, minimizing idle times during both training and inference

5. Techniques that Use Threading or Parallel Processing

- **Data Augmentation:** Real-time augmentation such as rotation, flipping, and noise addition can be parallelized.
- **Concurrent Model Training:** Multi-threaded processing allows simultaneous optimization of the two CNN paths.
- **Batch Processing:** Efficient handling of batches of MRI images ensures the GPU operates at peak throughput.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

1. **Data Imbalance:** Limited samples for specific tumor types affect classification accuracy.
2. **Noise and Artifacts:** Variability in MRI quality, including motion artifacts, complicates image analysis.
3. **Computational Complexity:** Traditional CNNs with many layers are resource-intensive, leading to longer training times.

7. Novelty

The hybrid CNN-NADE model introduces several innovations:

1. **Denoising and Density Estimation:** NADE preprocesses MRI images to remove noise and smooth tumor boundaries.
2. **Complementary Feature Extraction:** Dual CNNs leverage different input sources for enhanced accuracy.
3. **Efficiency:** The model achieves a high classification accuracy (95%) while reducing computational costs compared to traditional CNN architectures like AlexNet.

8. Result Analysis

Key findings from the simulations include:

- **Accuracy:** The hybrid CNN-NADE achieved an overall accuracy of 95%, outperforming simpler CNN models.
- **Confusion Matrix:** High sensitivity and specificity were observed across all tumor types, with minimal misclassification.
- **Comparison:** The proposed method rivaled AlexNet in accuracy but required significantly less computational time, emphasizing its efficiency

9. Challenges Faced

1. **Handling Small Datasets:** Limited MRI data necessitated robust density estimation techniques like NADE.
2. **Computational Constraints:** Reducing computational overhead without compromising accuracy was a key focus.
3. **Noise Sensitivity:** Ensuring reliable performance despite artifacts in medical images posed a significant challenge.

10. Future Directions

1. **Enhanced Preprocessing:** Incorporate segmentation or advanced filtering techniques to improve feature extraction.
2. **Lightweight Models:** Explore architectures with fewer parameters for deployment in resource-limited environments.
3. **Real-Time Applications:** Adapt the model for real-time diagnostic use in clinical settings.

11. Discussion

The hybrid CNN-NADE model provides an effective solution for brain tumor classification, combining the strengths of NADE's density estimation and CNN's feature extraction capabilities. It addresses challenges like imbalanced datasets and computational constraints, achieving state-of-the-art performance while being computationally efficient.

12. Research Gap

1. **Preprocessing Limitations:** Many studies depend heavily on segmentation or filtering, whereas this work bypasses such steps, relying solely on the hybrid architecture.
2. **Data Imbalance:** Existing methods struggle with class imbalance; this model leverages NADE to mitigate the issue.
3. **Efficiency vs. Accuracy:** The hybrid approach achieves competitive accuracy with lower computational demands, addressing gaps in traditional CNN-based methods

Paper 20: Brain tumor MRI images identification and classification based on the recurrent convolutional neural network

1. Methodology

The proposed methodology combines adaptive preprocessing, segmentation, feature extraction, and classification. Key components include:

1. **Preprocessing:**
    - Adaptive filtering is used to remove noise and improve the clarity of MRI images.
    - The adaptive filter adjusts locally based on noise variance to enhance edge retention.

2. **Segmentation:**
    - An improved K-means clustering (IKMC) algorithm segments the MRI images into meaningful regions.
    - This method clusters pixels into non-overlapping areas and identifies tumor boundaries with high precision.
3. **Feature Extraction:**
    - The Gray-Level Co-Occurrence Matrix (GLCM) extracts texture-based features, such as contrast, homogeneity, and dissimilarity.
4. **Classification:**
    - A Recurrent Convolutional Neural Network (RCNN) is employed to classify brain tumors (glioma, meningioma, pituitary tumors, and no-tumor) using contextual information from image features.

2. Simulations

- **Dataset:** The Kaggle dataset with 2870 training images and 394 testing images.
- **Implementation:**
    - Developed using Python and the Keras library with TensorFlow backend.
    - Hyperparameters were tuned through grid search, with the learning rate starting at 0.003 and gradually reduced.
- **Evaluation Metrics:** Performance was evaluated based on accuracy, sensitivity, and specificity.

3. Equations

1. **Adaptive Filtering:**

$$\hat{I}(x,y) = \mu_L + \frac{\sigma_y^2}{\hat{\sigma}_y^2}\left(I(x,y) - \mu_L\right)$$

   Where:

   - μ_L: Local mean
   - σ_y^2: Global noise variance
   - $\hat{\sigma}_y^2$: Local variance

2. **Sensitivity:**

1.
$$S_t = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

3. **Specificity:**

$$S_p = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}$$

4. **Accuracy:**

$$A = \frac{\text{True Positive} + \text{True Negative}}{\text{Total Population}}$$

5. **RCNN Behavior:**

- Net input for RCL:

$$z_{lmn}(t) = w_n^f u_{(l,m)}(t) + w_n^r x_{(l,m)}(t-1) + b_n$$

- Output state:

$$x_{lmn}(t) = g(f(z_{lmn}(t)))$$

2. Where w_n^f and w_n^r are feedforward and recurrent weights, respectively

4. Threading to Boost Performance

Threading could optimize:

1. **Data Preprocessing:** Parallel processing of adaptive filtering and segmentation tasks across multiple slices.
2. **RCNN Training:** Utilize GPU-based parallelism for training the recurrent and convolutional layers concurrently.
3. **Feature Extraction:** Multithreaded computation for GLCM matrices across different image patches.

5. Techniques that Use Threading or Parallel Processing

- **Parallel Segmentation:** Applying IKMC to multiple regions concurrently.
- **Batch Processing:** Efficient feeding of training data to the RCNN via parallel threads.
- **GPU Acceleration:** Utilizing TensorFlow's support for multi-GPU setups for faster model training.

6. Existing Bottlenecks in MRI-Based Brain Tumor Detection Workflows

1. **Noise Sensitivity:** MRI images often contain noise that complicates segmentation.
2. **Manual Segmentation Dependency:** Traditional methods require manual intervention, increasing time and error rates.
3. **Model Complexity:** Existing deep learning methods like CNNs and U-Nets are computationally expensive, requiring high memory and processing power.

7. Novelty

The RCNN architecture introduces:

1. **Contextual Learning:** Captures temporal and spatial relationships using recurrent layers.
2. **Improved Segmentation:** Enhanced clustering accuracy with IKMC.
3. **Efficiency:** Achieves 95.17% classification accuracy while reducing computational complexity compared to U-Net and BP methods.

8. Result Analysis

- **Accuracy Comparison:** The RCNN achieved 95.17%, outperforming U-Net (90.86%) and BP (88.83%).
- **Sensitivity and Specificity:** The RCNN demonstrated the highest sensitivity (98.42%) and specificity (89.28%) across models.
- **Segmentation Efficiency:** The IKMC algorithm identified tumor boundaries with minimal computational overhead.

9. Challenges Faced

1. **High Model Complexity:** RCNN requires substantial computational resources for training.
2. **Data Imbalance:** The dataset had significantly fewer images for certain tumor types, impacting generalizability.
3. **Noise Management:** Adaptive filtering addressed noise but required careful parameter tuning.

10. Future Directions

1. **Real-Time Processing:** Develop lightweight RCNN variants for real-time diagnosis in clinical settings.
2. **Data Augmentation:** Use synthetic data generation to balance the dataset.
3. **Transfer Learning:** Employ pre-trained RCNN models to reduce training time and improve generalizability.

11. Discussion

The RCNN-based approach represents a significant step forward in automated brain tumor classification. By leveraging advanced segmentation and classification techniques, the model balances accuracy and efficiency. However, the reliance on computationally intensive processes highlights the need for further optimization.

12. Research Gap

1. **Preprocessing Complexity:** Most approaches lack robust preprocessing techniques that integrate seamlessly with the classification model.
2. **Real-Time Deployment:** Existing methods are not optimized for real-time clinical applications.
3. **Contextual Learning:** Few models effectively utilize recurrent layers to capture contextual information, as demonstrated by RCNN.

Reference:

Paper 1: A Deep Analysis of Brain Tumour Detection from MR Images Using Deep Learning Networks

Paper 2: A distinctive approach in brain tumour detection and classification using MRI

Paper 3: A Hybrid CNN-SVM Threshold Segmentation Approach for Tumour Detection and Classification of MRI Brain Images

Paper 4: An Image Processing-based and Deep Learning Model to Classify Brain Cancer

Paper 5:Automated Brain Tumour Segmentation and Classification in MRI Using YOLO-Based Deep Learning

Paper:6 : Brain Tumour Detection Based on Deep Learning Approaches and Magnetic Resonance Imaging

Paper 7 : Brain tumor detection from images and comparison with transfer learning methods and 3-layer CNN

Paper 8: Brain tumour detection from MRI images using deep learning techniques

Paper 9: Brain tumour diagnosis from MRI based on Mobilenetv2 optimised by contracted fox optimization algorithm

Paper 10: Deep-Learning Detection of Cancer Metastases to the Brain on MRI

Paper 11: Detection of Brain Tumour based on Features Fusion and Machine Learning

Paper 12: Employing Deep Learning and Transfer Learning for Accurate Brain Tumour Detection

Paper 13: Improved Multiclass Brain Tumour Detection Using Convolutional Neural Networks and Magnetic Resonance Imaging

Paper 14 :MRI Brain Tumor Detection Using Deep Learning and Machine Learning Approaches

Paper 15 : Optimized Brain Tumor Detection: A Dual-Module Approach for MRI Image Enhancement and Tumor Classification

Paper 16: MRI-based brain tumour image detection using CNN based deep learning method.
Paper 17: Detection of tumors on brain MRI images using the hybrid convolutional neural network architecture
Paper 18: Research on Feature Extraction of Tumor Image Based on Convolutional Neural Network
Paper 19: Detection of brain tumors from MRI images base on deep learning using hybrid model CNN and NADE.
Paper 20: Brain tumor MRI images identification and classification based on the recurrent convolutional neural network