Submit a detailed report that includes:
Problem Statement: Briefly revisit your problem, motivation, and significance.
Objectives: Highlight measurable goals and progress made toward them.
Technical Details: Describe the deep learning or Llm model reduction using your own data.
Triangle Model: Describe the implementation and current status of the frontend, data repository, and Ai/Ml component.
Challenges & Solutions: Document any challenges faced and how your team addressed them.
Next Steps: Outline the steps planned for the project moving forward.
Submission: Post the technical report to your team's GitHub repository.

Technical Report

Problem Statement

Stroke is a medical emergency where timely recognition and intervention can drastically improve survival rates and reduce long-term disability. However, the majority of individuals fail to receive timely medical care due to delays in symptom recognition and access to emergency services. Many stroke symptoms can be subtle, easily overlooked, or misattributed to less severe conditions. As a result, early detection of stroke remains a critical issue in healthcare. To address this, we propose developing an artificial intelligence (AI) and machine learning (ML)-based platform capable of detecting early signs of stroke through facial image analysis, offering a potential tool for early intervention and real-time response.

Objectives

The primary objective is to create a robust AI-driven platform that accurately detects early signs of stroke from facial images and integrates seamlessly with real-time emergency services. The following measurable objectives have been outlined:

1. **High Classification Accuracy:** Achieve an accuracy rate of 99% or higher in stroke classification using state-of-the-art deep learning models.
2. **Model Optimization:** Develop an optimized model capable of running efficiently on consumer devices with limited computational resources without sacrificing significant accuracy.
3. **Real-Time Interaction:** Implement live AI chatbot functionality for emergency support, including location-based services for nearby hospitals and emergency contact integration.

To date, an initial classification accuracy of 99.65% has been achieved using a ResNet50 model, with ongoing development aimed at model optimization and full platform integration.

Technical Details

Data

- **Dataset:** A total of 3,770 facial images were collected, representing both stroke and non-stroke conditions. These images serve as the foundation for training the deep learning model.
- **Preprocessing:** Data augmentation techniques—such as flipping, rotation, and zooming—were employed to increase dataset diversity and improve the model's generalization capabilities. This ensures the model can recognize stroke symptoms across a wider range of facial characteristics and lighting conditions.

Model

- **Architecture:** The core model used for stroke detection is a ResNet50 deep convolutional neural network, known for its effectiveness in image classification tasks. ResNet50's ability to extract deep features from images makes it well-suited for identifying subtle stroke-related abnormalities in facial expressions.
- **Training:** The model was trained using backpropagation with a cross-entropy loss function, optimizing its weights over several epochs to maximize classification accuracy.
- **Performance:** The ResNet50 model achieved an accuracy of 99.65% on the stroke vs. non-stroke classification task. This high performance demonstrates the model's effectiveness at distinguishing stroke-related facial characteristics.

Model Reduction

ResNet50, while highly accurate, is computationally intensive, making it unsuitable for real-time applications on devices with limited processing power. To address this, efforts are currently focused on model reduction through methods such as pruning, quantization, and distillation. These techniques aim to reduce the size of the model while maintaining a balance between performance and computational efficiency. Lightweight architectures like MobileNet are also under consideration to further enhance the platform's real-time capabilities.

Triangle Model

The platform's architecture follows a triangular model consisting of the frontend, data repository, and AI/ML component.

Frontend

- **Status:** The basic structure of the platform's frontend has been developed, enabling users to upload facial images for analysis.

- **Next Steps:** A live camera feed feature will be implemented, allowing real-time image capture and stroke detection on-the-fly.

Data Repository

- **Status:** The dataset is securely stored in the cloud, with augmentation strategies in place to ensure robustness during training.
- **Next Steps:** The dataset will be expanded to include more diverse facial images from various demographics, ensuring improved model performance across all populations. Additionally, APIs will be developed for seamless data access and storage.

AI/ML Component

- **Status:** A working prototype using ResNet50 has demonstrated high classification accuracy, achieving an accuracy of 99.65%.
- **Next Steps:** Real-time inference capabilities will be implemented, and the model will be optimized for deployment in production environments, allowing stroke detection in milliseconds on consumer-grade devices.

Challenges & Solutions

Challenge 1: Model Complexity

The ResNet50 model, while accurate, is computationally demanding, making real-time deployment challenging on devices with limited resources.

Solution: Ongoing research into model pruning, quantization, and distillation techniques aims to reduce model complexity without sacrificing significant accuracy. Additionally, lightweight models such as MobileNet are being explored as alternatives.

Challenge 2: Dataset Limitations

The current dataset may not represent all variations in stroke symptoms, especially across different ethnicities, ages, and facial characteristics.

Solution: We plan to collaborate with medical institutions to acquire a more diverse dataset that reflects a broader range of stroke symptoms and demographic variations. Advanced data augmentation techniques will also be applied to simulate more diverse facial expressions and conditions.

Challenge 3: Emergency Response Integration

Integrating the AI-based stroke detection system with emergency services such as 911 presents both technical and legal challenges.

Solution: We are collaborating with local authorities to ensure compliance with relevant regulations. The platform will provide emergency contact information and location-based services for nearby hospitals while following appropriate legal guidelines for emergency interventions.

Next Steps

1. **Model Optimization:** Continue optimizing the deep learning model to reduce its size and computational requirements, ensuring fast and accurate inference on mobile and desktop devices.
2. **Frontend Development:** Implement the live camera feed functionality, allowing users to upload real-time images for instant stroke detection.
3. **AI Chatbot Integration:** Develop a real-time chatbot powered by a language model to interact with users, provide stroke detection results, and offer emergency assistance if necessary.
4. **Emergency Response Integration:** Set up threshold-based alerts based on stroke probability scores (e.g., 90%-95%), with options to initiate a 911 call or provide nearby hospital locations.
5. **Testing & Validation:** Perform extensive testing on a more comprehensive dataset to validate the model's performance across various demographic groups and facial characteristics, ensuring reliability and fairness in detection outcomes.

This project, combining AI-powered stroke detection with real-time emergency response, holds significant potential for improving the early diagnosis and treatment of strokes. Through continuous refinement and optimization, we aim to deliver a reliable, efficient, and accessible platform that could make a substantial difference in stroke management and patient outcomes.

Submission

The technical report, including all code, models, and relevant documentation, will be submitted to the team's GitHub repository for review and further collaboration.