CT-Scan Segmentation and Classification Using Deep Learning Techniques

Zhifei Huang, Kimi Nguyen, Sanket Srivastava, Yuqiao Tang, Kangping Wang, Quella Wang

December 2024

Abstract

Accurate segmentation of brain structures in CT images is crucial for diagnostic and surgical procedures. This paper presents two projects aimed at improving CT segmentation and classification using deep learning methods. The first project focuses on segmenting brain hemorrhages in CT images using U-Net convolutional neural networks, achieving high prediction accuracy. The second project involves utilizing classification of these hemorrhages into epidural, intraparenchymal, intraventricular, subdural, subarachnoid, and multiclass types using transfer learning of the DenseNet169 CNN model. Our methods demonstrate significant potential in enhancing the precision of brain imaging analysis.

1 Introduction

Brain imaging plays a pivotal role in diagnosing neurological conditions and planning neurosurgical procedures. Accurate analysis of brain structures from CT images is essential for tasks such as hemorrhage detection and classification. Manual analysis is time-consuming and prone to variability, highlighting the need for automated, reliable methods.

Deep learning, particularly convolutional neural networks (CNNs), has shown promise in medical image analysis tasks. In this paper, we present two projects leveraging deep learning for CT segmentation and classification:

1. **Segmentation Project**: Development of a U-Net-based CNN to segment brain hemorrhages from CT images.
2. **Classification Project**: Development of a DenseNet-based CNN to classify brain hemorrhages from CT images.

These projects aim to enhance diagnostic accuracy and surgical planning by providing precise analytical tools.

2 Segmentation Project

2.1 Data Collection

The dataset comprises various brain CT scan slices, each containing hemorrhages of different types: intraparenchymal, intraventricular, subarachnoid, subdural, epidural, and multiclass hemorrhages. The data is organized into folders corresponding to each hemorrhage type, with subfolders for different rendering types (e.g., brain window, subdural window, brain bone window). The labels are provided in accompanying CSV files.

2.2 Data Preprocessing and Augmentation

Given the limited number of samples, data augmentation techniques were employed to prevent overfitting and provide a more diverse training dataset. CT images were resized to a consistent shape, and augmentation methods such as rotations, translations, and lateral flips were applied. Downsampling was avoided as much as possible as this led to loss of critical information due to the small size of hemorrhages. Therefore, focus was placed on augmentation methods that preserved important features.

2.3 Model Architecture

We utilized a U-Net architecture for segmentation, as proposed by Ronneberger et al. [1]. The U-Net is well-suited for biomedical image segmentation due to its symmetric encoder-decoder structure, which captures both global and local context.

The network consists of:

- **Contracting path**: Repeated applications of convolutions and max pooling, doubling the number of feature channels at each step.
- **Expansive path**: Upsampling and concatenation with corresponding feature maps from the contracting path, followed by convolutions to refine the segmentation.

2.4 Training Process

The model was trained using the augmented dataset, with careful tuning of hyperparameters:

- **Loss function**: Binary cross-entropy with weighting to address class imbalance.
- **Optimizer**: Adam optimizer with an initial learning rate of.
- **Batch size**: Set to 1 due to computational constraints.
- **Epochs**: Training continued until the validation loss plateaued.

2.5 Visualization of First-Layer Weights

Visualizing the weights of the first convolutional layer provides insights into the learning process. We observed that after training, the weights exhibited discernible patterns, indicating that the network learned relevant features for segmentation. The weights were visualized by normalizing and plotting the filter values.

3 Classification Project

3.1 Data Collection

The dataset contains brain CT scan slices of different types: intraparenchymal, intraventricular, subarachnoid, subdural, epidural, normal, and multiple hemorrhages. Normal means no hemorrhage. Multiple means more than one hemorrhage. The data is organized into folders corresponding to each hemorrhage type. There are 4 subfolders (brain window, subdural window, max contrast window, and brain bone window) for each type. The labels are provided in accompanying CSV files.

3.2 Data Preprocessing and Augmentation

We have tried two ways to preprocess the data of four windows. We first used all images in the windows. Each image was regard as an individual sample for training and validation, even though they had same file name. Another method is that we tried to stack the image from 4 windows together. Then, using the stacked images for training and validation. Given the limited number of samples, data augmentation techniques were employed to prevent overfitting and provide a more diverse training dataset. CT images were resized to a consistent shape, and augmentation methods such as rotations, translations, and lateral flips were applied. All labels were one-hot encoded as a 1x7 vector, allowing for ease of classification.

3.3 EfficientNetB4

3.3.1 Model Architecture

We first utilized the EfficientNetB4 model architecture [3] for a quick check on multi-class classification with seven output classes. The EfficientNetB4 model was chosen due to its efficient features. I could rapidly acquire the results and decide which way is better for data preprocessing. The model was customized as follows:

- **Base architecture**: A pre-trained EfficientNetB4 network (trained on ImageNet) was used as the backbone with the fully connected layers excluded (include_top=False) to allow for customization. The input size for the model was set to 380x380 pixels, as required by EfficientNetB4.
- **Layer freezing and fine-tuning**: The layers of the EfficientNetB4 backbone were frozen to leverage the pre-trained weights and prevent overfitting during initial training.
- **Custom head**: A Global Average Pooling (GAP) layer was added to reduce the spatial dimensions of the feature maps. A Dropout layer with a dropout rate of 0.5 was added to prevent overfitting. A fully connected dense layer with 256 units and a ReLU activation function was added for feature extraction. Another Dropout layer with a rate of 0.5 was added before the output layer. The final output layer consisted of a 7-class softmax function for multi-class classification.

3.3.2 Training Process

The EfficientNetB4 model was trained on our augmented CT-scan data, leveraging the following setup:

- **Loss function**: Categorical Cross-Entropy, appropriate for multi-class classification tasks.
- **Optimizer**: ADAM (Adaptive Moment Estimation) optimizer with a learning rate of 0.001.
- **Batch size**: A batch size of 32 was used for both training and validation, balancing computational efficiency with model stability.
- **Data augmentation**: Rotation (up to 30°), Width and height shifts (20%), Shear transformations (20%), Zoom (20%), Horizontal flipping. Grayscale values were rescaled to the range [0, 1].
- **Callbacks**: Early stopping can be optionally added to monitor validation loss and prevent overfitting.

3.4 DenseNet169

3.4.1 Model Architecture

For the final method, we utilized the DenseNet169 model architecture [2] for multiclass classification with seven output classes. The DenseNet169 model was chosen due to its efficient feature reuse and gradient flow properties, which are critical for robust feature extraction in biomedical image analysis. The model was customized as follows:

- **Base architecture**: A pretrained DenseNet169 network (on ImageNet) was used as the backbone, excluding fully connected layers (‘include_top=False‘) to allow for customization. The input size for the model was set to 224x224 pixels, as required.
- **Layer freezing and fine-tuning**: The initial five layers of the DenseNet backbone were unfrozen to allow gradient updates, allowing a balance between transfer learning and fine-tuning.
- **Custom head**: A Global Average Pooling (GAP) layer was added to reduce the spatial dimensions of the feature maps, followed by a fully connected dense layer with 1024 units and the ReLU activation function. In this layer, regularization of L2 ($\lambda = 0.01$) was used to prevent overfitting. A dropout layer ($p = 0.5$) was incorporated before the output layer, which utilized a 7-class softmax function for multiclass classification.

3.4.2 Training Process

The model DenseNet169 was trained on our augmented CT-scan data, leveraging the following setup:

- **Loss function**: Categorical cross-entropy, appropriate for multi-class classification.
- **Optimizer**: Adaptive Moment Estimation (ADAM) [4] optimizer with weight decay (3×10^{-5}) and a learning rate of 1×10^{-5}, to improve generalization.

- **Learning rate schedule**: A cosine decay scheduler was employed to periodically reset the learning rate, starting at 1×10^{-5} and decaying to 1×10^{-6}.
- **Batch size**: A batch size of 32 was used for training and validation, balancing computational efficiency with model stability.
- **Data augmentation**: The training data underwent real-time augmentation, including rotation (up to $20°$), width and height shifts (10%), zoom (20%), and horizontal flipping. All images were rescaled to $[0, 1]$.
- **Callbacks**: Early stopping monitored validation loss with a patience of five epochs, and the learning rate scheduler ensured adaptive learning throughout training.

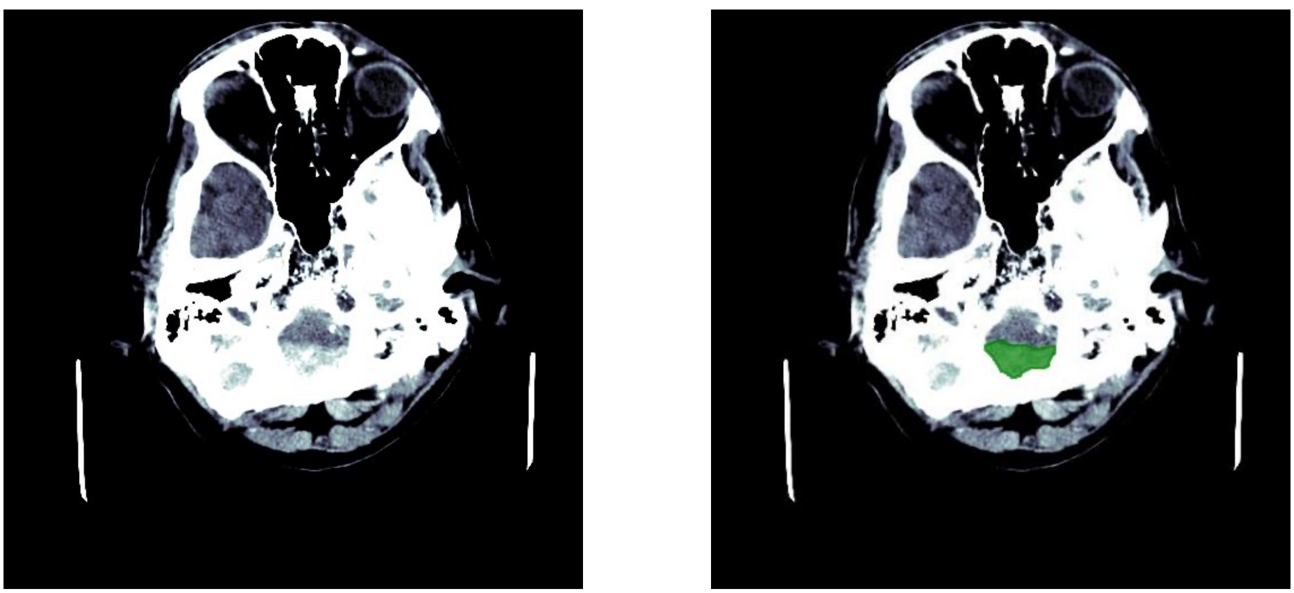
3.5 Visualization of Learning Progress

To evaluate the learning process, we monitored training and validation accuracy and loss across epochs.

4 Results

4.1 Segmentation Results

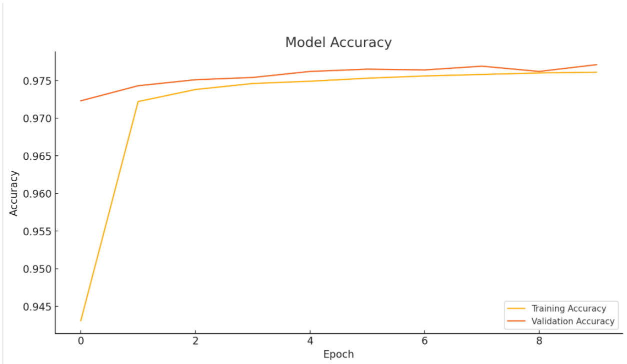
The U-Net model achieved a high prediction accuracy of 98% on the validation set. Figure 1 illustrates the segmentation results, showing close alignment between the predicted and actual ventricle positions.



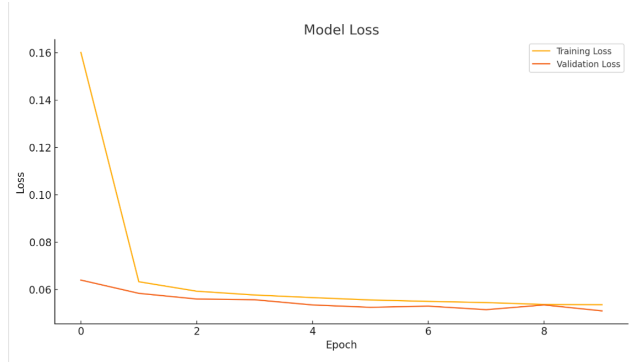
(a) Actual segmentation.

(b) Model prediction.

Figure 1: Original vs. Predicted Ventricle Segmentation. Left: Actual segmentation. Right: Model prediction.



(a) Training Accuracy.



(b) Training Loss.

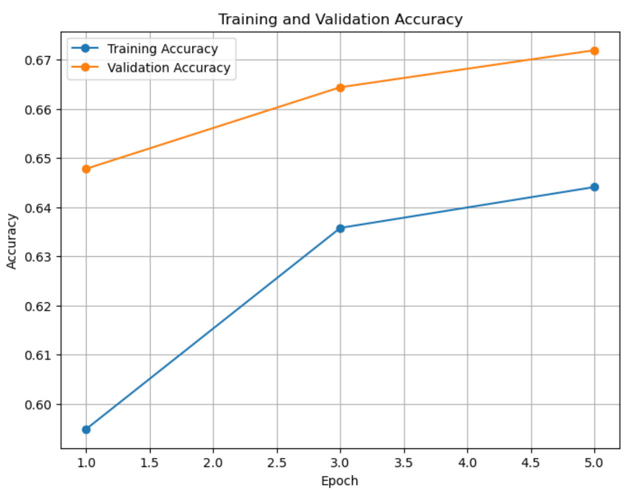
Figure 2: Training Metrics

The loss and accuracy curves (Figure 2) indicate successful training, with the model converging after a certain number of epochs.

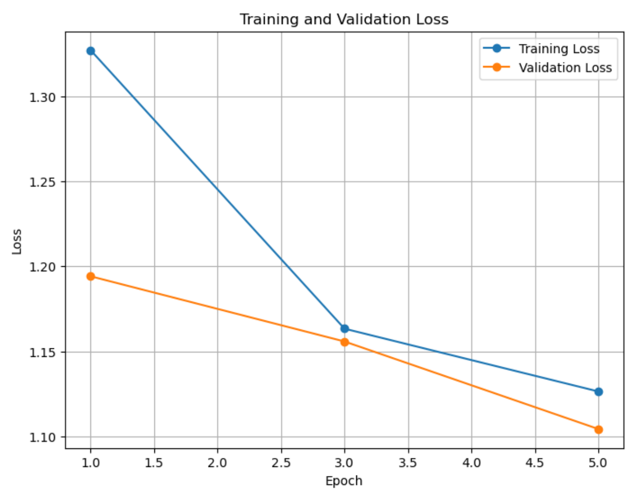
4.2 Classification Results

4.2.1 EfficientNetB4

The EfficientNetB4 model only achieved a validation accuracy of 67.7%, with a training accuracy of 64.4%. We immediately dropped this method.



(a) Training Accuracy.

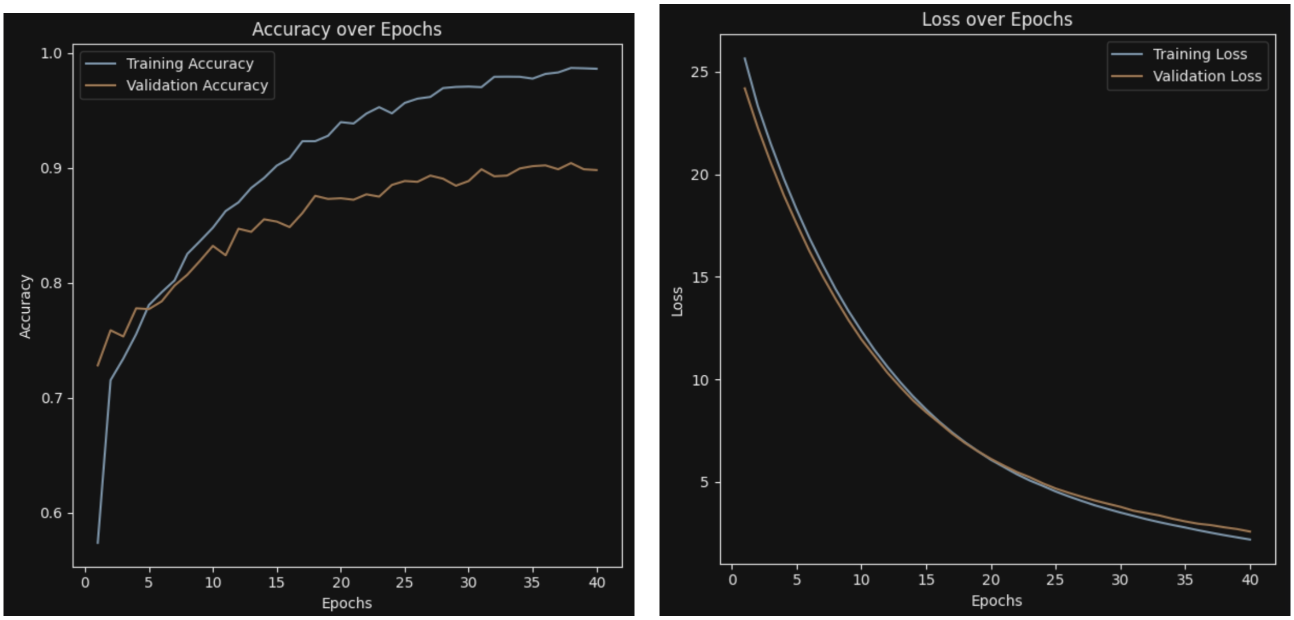


(b) Training Loss.

Figure 3: EfficientNetB4 Model

4.2.2 DenseNet169

The DenseNet169 model achieved a respectable validation accuracy of 89.8%, with a training accuracy of 98.6%.



(a) Training Accuracy.

(b) Training Loss.

Figure 4: DenseNet169 Model

Figure 4 illustrates the successful training results, showing decrease of loss over time, as well as increasing accuracy over time.

5 Discussion

The use of U-Net architectures proved effective for segmenting brain structures in CT images. The high accuracy demonstrates the potential of deep learning in medical image analysis. However, challenges remain:

- **Limited dataset size** necessitated extensive augmentation.
- **Computational constraints** restricted the complexity of models and training duration.

Future work includes expanding the dataset, exploring transfer learning techniques, and integrating the models into clinical workflows.

6 Conclusion

In this paper, we presented two deep learning-based projects aimed at enhancing the analysis of brain CT scans through accurate segmentation and classification of hemorrhages.

Segmentation Project: Utilizing the U-Net convolutional neural network architecture, we successfully segmented various types of brain hemorrhages with a high validation accuracy of 98%. The U-Net model demonstrated exceptional capability in delineating hemorrhage regions, highlighting its suitability for medical image segmentation tasks. The visualization of the first-layer weights confirmed that the network effectively learned relevant features, contributing to the model's high performance.

Classification Project: We explored two approaches for hemorrhage classification using transfer learning with EfficientNetB4 and DenseNet169 models. The EfficientNetB4 model, while efficient, underperformed with a validation accuracy of only 67.7%, leading us to discontinue its use. In contrast, the DenseNet169 model achieved a commendable validation accuracy of 89.8% and a training accuracy of 98.6%, underscoring its robustness and effectiveness in multiclass classification tasks. The DenseNet169's ability to efficiently reuse features and maintain strong gradient flow significantly contributed to its superior performance.

Challenges and Limitations: - *Dataset Size*: Both projects were constrained by the limited size of the available dataset, necessitating extensive data augmentation to mitigate overfitting. Future efforts should focus on expanding the dataset to further enhance model generalization. - *Computational Resources*: Training deep neural networks, especially 3D models like U-Net, demands substantial computational power. Leveraging cloud-based resources or more advanced hardware could facilitate training more complex models and experimenting with larger architectures. - *Model Selection*: The underperformance of EfficientNetB4 highlights the importance of selecting appropriate architectures tailored to the specific characteristics of medical imaging data.

Future Work (see #7 for details): - *Data Expansion*: Increasing the dataset size through collaboration with medical institutions will enable the training of more generalized and accurate models. - *Advanced Architectures*: Exploring other architectures, such as ResNet or attention-based models, could further improve segmentation and classification performance. - *Clinical Integration*: Integrating these models into clinical workflows will require rigorous validation and user-friendly interfaces to assist medical professionals in diagnosis and surgical planning. - *Real-Time Processing*: Optimizing models for real-time or near-real-time processing will enhance their utility in clinical settings, where timely decision-making is critical.

Impact: The successful implementation of the U-Net and DenseNet169 models demonstrates the significant potential of deep learning in improving the accuracy and efficiency of brain hemorrhage analysis. These advancements can lead to better diagnostic tools, more precise surgical planning, and ultimately, improved patient outcomes.

In conclusion, our projects underscore the transformative role of deep learning in medical imaging. Continued research, coupled with interdisciplinary collaboration, will be essential in overcoming existing challenges and fully realizing the benefits of these technologies in healthcare.

7 Future Improvements and Suggestions

While the current projects have demonstrated significant advancements in CT scan segmentation and classification using deep learning techniques, several areas offer opportunities for further enhancement and development. Addressing these areas can lead to more robust, accurate, and clinically applicable models.

7.1 Data Expansion and Enhancement

- **Increase Dataset Size**: Collaborate with medical institutions to acquire a larger and more diverse set of annotated CT scans. A more extensive dataset will improve the model's ability to generalize to different populations and imaging conditions.
- **Multicenter Data Collection**: Incorporate data from multiple centers to capture variability in imaging protocols, scanner types, and patient demographics, enhancing the model's robustness.
- **Synthetic Data Generation**: Utilize generative models such as Generative Adversarial Networks (GANs) to create synthetic CT images, thereby augmenting the dataset and addressing class imbalances.

7.2 Advanced Model Architectures

- **3D U-Net Implementation**: Transition from 2D to 3D U-Net architectures to leverage the volumetric nature of CT scans, capturing spatial relationships across slices for more accurate segmentation.
- **Attention Mechanisms**: Integrate attention layers into the U-Net and DenseNet architectures to allow the model to focus on relevant regions of the image, improving segmentation and classification accuracy.
- **Ensemble Learning**: Combine predictions from multiple models to reduce variance and enhance overall performance. Techniques such as model averaging or stacking can be employed to create a more reliable predictive system.
- **Hybrid Models**: Explore hybrid architectures that combine the strengths of different neural network types, such as combining CNNs with recurrent layers for capturing temporal information in sequential CT slices.

7.3 Enhanced Data Augmentation Techniques

- **Elastic Deformations**: Apply elastic transformations to simulate realistic anatomical variations and improve the model's ability to generalize to unseen data.
- **Intensity Variations**: Introduce variations in image intensity, contrast, and brightness to make the model more robust to different imaging conditions.
- **Random Cropping and Scaling**: Implement random cropping and scaling to help the model learn to identify hemorrhages of varying sizes and positions within the brain.

7.4 Optimization of Training Processes

- **Hyperparameter Tuning**: Conduct comprehensive hyperparameter optimization using techniques such as grid search, random search, or Bayesian optimization to identify the best-performing configurations.
- **Transfer Learning Enhancements**: Fine-tune pre-trained models more extensively by unfreezing additional layers and experimenting with different learning rates to better adapt to the specific characteristics of CT scan data.
- **Regularization Techniques**: Incorporate advanced regularization methods such as dropout, batch normalization, and weight decay to prevent overfitting and improve model generalization.

7.5 Integration with Clinical Workflows

- **User-Friendly Interfaces**: Develop intuitive interfaces that allow medical professionals to interact with the models seamlessly, facilitating easy access to segmentation and classification results.
- **Real-Time Processing**: Optimize models for real-time or near-real-time inference to provide timely assistance during clinical decision-making and surgical planning.
- **Interoperability with Medical Systems**: Ensure compatibility with existing medical imaging software and hospital information systems (HIS) to enable smooth integration and data exchange.

7.6 External Validation and Robustness Testing

- **Cross-Institutional Validation**: Test the models on external datasets from different institutions to evaluate their performance and generalizability across various clinical settings.
- **Robustness to Noise and Artifacts**: Assess the models' resilience to common imaging artifacts and noise, ensuring reliable performance in real-world scenarios where data quality may vary.
- **Comprehensive Evaluation Metrics**: Utilize a broader set of evaluation metrics, including Dice coefficient, Intersection over Union (IoU), precision, recall, and F1-score, to gain a more nuanced understanding of model performance.

7.7 Explainability and Interpretability

- **Saliency Maps and Grad-CAM**: Implement techniques like Grad-CAM to visualize the regions of the image that the model focuses on during prediction, enhancing transparency and trustworthiness.
- **Feature Importance Analysis**: Analyze which features or image regions contribute most to the model's decisions, providing insights into the underlying decision-making process.
- **Model Documentation**: Provide comprehensive documentation and visualization tools that allow clinicians to understand and interpret the model's outputs effectively.

7.8 Multimodal Data Integration

- **Combining CT with Other Modalities**: Integrate data from other imaging modalities, such as MRI or PET scans, to provide a more comprehensive analysis and improve segmentation and classification performance.
- **Incorporating Clinical Metadata**: Utilize patient-specific metadata, such as age, medical history, and laboratory results, to enhance the predictive capabilities of the classification model.

7.9 Continuous Learning and Model Updating

- **Active Learning**: Implement active learning strategies where the model can query for annotations on uncertain or difficult cases, thereby improving its performance iteratively.
- **Automated Retraining Pipelines**: Develop automated pipelines that regularly update the model with new data, ensuring that it remains up-to-date with the latest clinical practices and data distributions.
- **Feedback Mechanisms**: Incorporate feedback from medical professionals to continuously refine and improve the model based on real-world usage and performance.

7.10 Ethical and Regulatory Considerations

- **Bias Mitigation**: Ensure that the models are free from biases related to demographics, imaging equipment, or other factors by conducting fairness assessments and implementing mitigation strategies.
- **Compliance with Medical Regulations**: Adhere to relevant medical device regulations and standards to facilitate the clinical deployment and acceptance of the models.
- **Patient Privacy and Data Security**: Implement robust data privacy and security measures to protect patient information during data collection, storage, and processing.

7.11 Exploring Alternative Learning Paradigms

- **Semi-Supervised and Unsupervised Learning**: Investigate semi-supervised or unsupervised learning approaches to leverage unlabeled data, reducing the reliance on extensive manual annotations.
- **Few-Shot Learning**: Explore few-shot learning techniques to enable the model to generalize from a limited number of labeled examples, which is particularly useful in medical imaging where labeled data is scarce.

Implementing these improvements and suggestions will not only enhance the technical performance of the models but also ensure their practical applicability and reliability in clinical environments. Continuous collaboration with medical professionals, adherence to ethical standards, and embracing innovative methodologies will be essential in advancing the field of medical image analysis using deep learning.

References

[1] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention* (pp. 234-241). Springer, Cham.

[2] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 2261-2269, doi: 10.1109/CVPR.2017.243

[3] Tan, Mingxing and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." ArXiv abs/1905.11946 (2019): n. pag.

[4] Kingma, Diederik P. and Jimmy Ba. “Adam: A Method for Stochastic Optimization.” CoRR abs/1412.6980 (2014): n. pag.

[5] Balafar, MȦ; Ramli, A. R., Saripan, M. I., & Mashohor, S. (2010). Review of brain MRI image segmentation methods. *Artificial Intelligence Review*, 33(3), 261-274.

[6] Joseph, R. P., Singh, C. S., & Manikandan, M. (2010). Brain Tumor MRI Image Segmentation and Detection in Image Processing. *International Journal of Research in Engineering and Technology*, 3(5), 1-5..

A Code Snippets

All of our code is reproducible and can be found on GitHub: `https://github.com/qquella/Brain-CT-image-hemorrhage-seg` `git`

Below is a selected code snippet demonstrating the model definition:

```python
def unet_model(input_size=(80, 120, 120, 1)):
    inputs = Input(input_size)
    # Contracting path
    conv1 = Conv3D(64, (3, 3, 3), activation='relu', padding='
  same')(inputs)
    #     additional layers
    # Expansive path
    up8 = UpSampling3D(size=(2, 2, 2))(conv7)
    #     additional layers
    outputs = Conv3D(1, (1, 1, 1), activation='sigmoid')(conv10)
    model = Model(inputs=[inputs], outputs=[outputs])
    return model
```

Listing 1: U-Net Model Definition