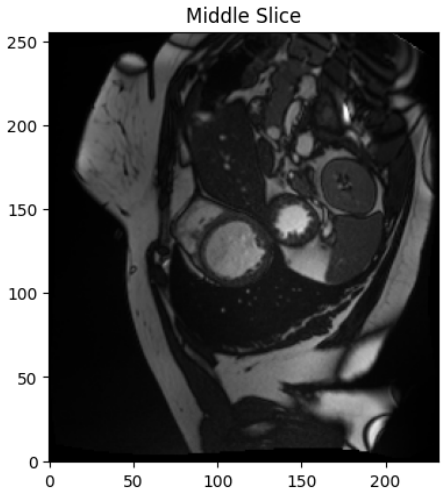
Budapest University of
Technology and Economics
BME

Faculty of Electrical Engineering and Informatics (VIK)

Major :

Artificial Intelligence and Data Science




# Deep Learning Project

# Model Ensemble for Medical Image Segmentation

*Presented by :*

AKAABOUNE Fatima Ezzahra
LI Jingxue

*Supervised by :*

Mr.AL-RADHI Mohammed Salah

*Defended on December 10, 2025*

Academic Year: 2024/2025

# CONTENTS

# 1 Introduction

In recent years, deep learning has emerged as a powerful tool for solving complex problems in medical imaging, particularly in semantic segmentation tasks. Semantic segmentation is essential in applications like cardiac MRI analysis, where accurate delineation of anatomical structures plays a critical role in diagnostics and treatment planning. However, achieving high accuracy in such tasks often requires advanced techniques to mitigate issues like model overfitting and sensitivity to variations in data distribution.

One such technique is the use of **model ensembles**, where multiple models are combined to make predictions. Ensembles are known to improve the robustness and accuracy of deep learning solutions by leveraging the diverse strengths of individual models. This approach has been widely adopted by top-performing teams in AI competitions, showcasing its effectiveness in boosting performance at the expense of increased computational costs.

This project explores the construction and application of model ensembles for semantic segmentation tasks, with a focus on cardiac MRI datasets. The primary objectives are:

1. Selecting a suitable dataset, preferably from cardiac MRI segmentation.
2. Identifying an open-source segmentation model as a baseline.
3. Training multiple models and constructing an ensemble from them.
4. Analyzing the performance improvements, computational costs, and practical benefits of ensemble methods.

To achieve these objectives, we draw insights from relevant scientific literature and open-source projects, such as DivergentNets and SenFormer. This work will evaluate the ensemble's effectiveness by comparing individual model performance against the ensemble on metrics such as segmentation accuracy, Dice coefficient, and computational throughput.

By the end of this project, we aim to provide a detailed analysis of ensemble methods in semantic segmentation, highlighting their advantages, challenges, and potential applications in medical imaging.

# 2 Dataset

The dataset utilized for this project is the **ACDC Dataset** [2], derived from the MICCAI Challenge (2017) for the Automatic Cardiac Diagnosis Challenge. This dataset focuses on the diagnostic and segmentation tasks of cardiac magnetic resonance (MR) images. Detailed information about the dataset can be accessed at the official website: `https://acdc.creatis.insa-lyon.fr/`.

The dataset contains cardiac MR images organized for diagnostic purposes and segmentation analysis, providing a valuable resource for developing and evaluating machine learning models in medical imaging. It is structured into training and testing sets, with patient-specific subfolders containing various file formats, such as:

- `Info.cfg`: Metadata files describing patient-specific details.
- `MANDATORY CITATION.md`: A file emphasizing citation requirements for using the dataset.
- `.nii.gz`: 4D and 3D MR image files, including labeled ground truth and raw image data.

This dataset is publicly available and supports reproducibility and benchmarking in cardiac MR image segmentation tasks, serving as a robust foundation for machine learning and deep learning methodologies in the medical domain.

# 3 Preprocessing

The preprocessing of the dataset involves several crucial steps to prepare the medical images for analysis. First, the images, stored in NIfTI format ('.nii.gz'), are loaded using the 'nibabel' library. Each image undergoes Min-Max normalization to scale the intensity values between 0 and 1, ensuring consistency across the dataset. To handle variations in image dimensions, padding is applied to resize all images to a uniform target shape of '(192, 256, 8)', accommodating the model's input requirements. A custom PyTorch 'Dataset' class, 'MedicalDataset', is implemented to dynamically preprocess images during training. This class loads individual files, applies normalization, and pads them before converting them into tensors suitable for model input. These preprocessing steps ensure the data is standardized and ready for efficient and accurate training of the deep learning models.

# 4 Models [1]

## Feature Pyramid Network (FPN)

The architecture utilized in this study is based on the Feature Pyramid Network (FPN), a widely-adopted architecture for semantic segmentation. FPN leverages a pyramid hierarchy of feature maps to extract both high-resolution and low-resolution spatial information, making it well-suited for medical image segmentation tasks such as cardiac MRI analysis.

**Model Architecture: Feature Pyramid Network (FPN)**

FPN operates by integrating features at multiple levels of the backbone network and constructing a top-down pathway for feature aggregation. The architecture consists of the following key components:

- **Backbone:** A convolutional neural network (e.g., ResNet or EfficientNet) is used as the feature extractor. It generates feature maps at different scales.
- **Top-Down Pathway:** High-level semantic features are propagated from the top layers down to the lower-resolution layers, enabling multi-scale feature fusion.
- **Lateral Connections:** These connections combine features from the backbone with the top-down features, ensuring that both high-resolution and contextual information are preserved.
- **Prediction Layers:** The final layers produce pixel-wise predictions, typically followed by a softmax or sigmoid activation for multi-class or binary segmentation tasks, respectively.

This architecture effectively balances computational efficiency and segmentation accuracy, making it particularly effective for capturing fine-grained details in medical images.

**Performance Evaluation**

The model was trained over 10 epochs, with performance evaluated using pixel-level **Accuracy** and **Intersection over Union (IoU)**. These metrics provide insights into how well the model delineates structures in the cardiac MRI dataset. The training and validation loss trends reveal the following:

**Training and Validation Loss Trends**

- **Epochs 1-3:** The losses showed minimal improvement, with the validation loss consistently higher than the training loss, suggesting early-stage underfitting.

- **Epochs 4-6:** Substantial improvements were observed in both training and validation losses, indicating that the model was learning effectively. The best model was saved at Epoch 5, which achieved the lowest validation loss.
- **Epochs 7-8:** Overfitting began to manifest, as the validation loss increased while the training loss continued to decrease.
- **Epochs 9-10:** The training loss further declined, but the validation loss fluctuated, demonstrating stabilization with minor variability.

**Key Observations**

- **Best Model Checkpoints:** The best-performing models were saved after Epochs 4, 5, and 6, with Epoch 5 delivering the most optimal validation performance.
- **Segmentation Classes:** The model performed binary segmentation with pixel classes $\{0, 1\}$.
- **Overfitting:** Signs of overfitting were observed beyond Epoch 6, suggesting a need for regularization or early stopping mechanisms.

**Evaluation Methodology**

The `evaluate_accuracy_and_iou` function computes pixel-level accuracy and IoU by comparing the model's predicted outputs to the ground truth masks. The predictions are obtained by selecting the class with the highest probability for each pixel using the `argmax` operation. Key steps include:

- Reshaping the ground truth masks to match the predicted outputs.
- Calculating pixel-level accuracy as the ratio of correctly predicted pixels to the total number of pixels.
- Computing the IoU as the ratio of the intersection of predicted and true positive pixels to their union.

The `evaluate_model` function aggregates these metrics across all batches in the test set, providing the final accuracy and IoU values. For ensemble models, the `evaluate` function calculates the IoU score using a weighted Jaccard index across classes.

**Performance Results**

**Training and Validation Loss Trends:**

- **Epochs 1-3:** Minimal improvement in both training and validation losses, with validation loss generally higher than the training loss.
- **Epochs 4-6:** Significant improvements in both training and validation losses. The best model was saved at Epoch 5, which showed the lowest validation loss.
- **Epochs 7-8:** Onset of overfitting observed, with validation loss beginning to increase despite continued decrease in training loss.
- **Epochs 9-10:** Training loss continued to decline, while validation loss stabilized with minor fluctuations.

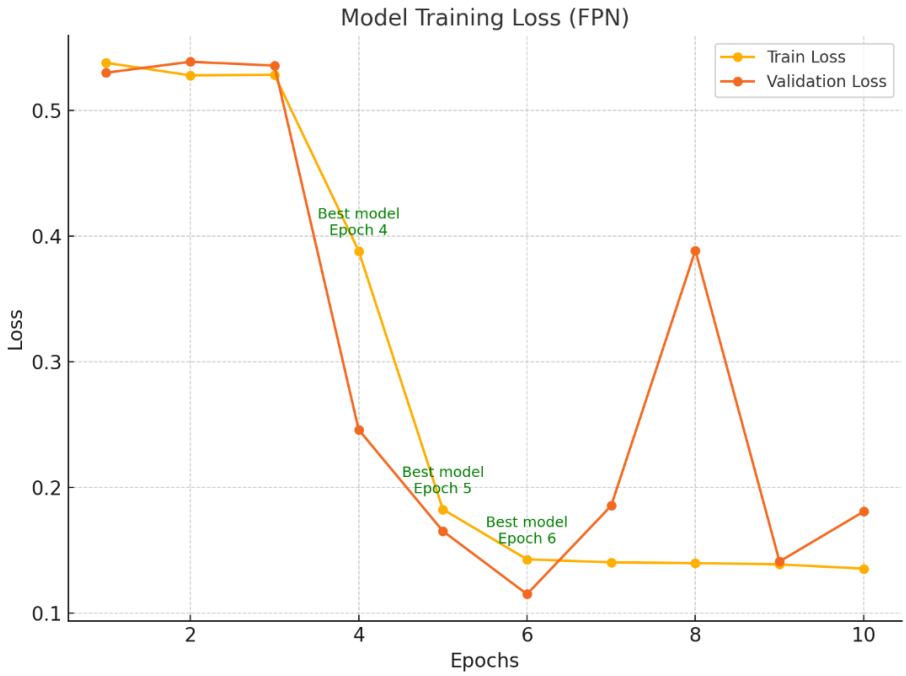



Figure 1: Model Training Loss (FPN)

**Key Observations:**

- **Best Model Checkpoints:** Saved after Epochs 4, 5, and 6, with the best validation loss occurring at Epoch 5.
- **Segmentation Classes:** The model performs binary segmentation with classes {0, 1}.
- **Overfitting:** Notable signs of overfitting were observed in later epochs, necessitating regularization techniques or early stopping.

**Conclusion:** The model demonstrated its best performance during Epochs 4-6, achieving significant improvement in both training and validation losses. While overfitting was observed in later epochs, the early checkpoints provide a robust basis for inference. The metrics of accuracy and IoU reinforce the effectiveness of the model in segmenting cardiac MRI images. Further refinement, such as hyperparameter tuning or augmentation strategies, could enhance its generalizability.

**Conclusion**

The FPN architecture demonstrated robust performance, achieving its best results during Epochs 4-6. Its ability to integrate multi-scale features made it well-suited for cardiac MRI segmentation. Despite some overfitting in later epochs, the saved checkpoints represent an effective solution for binary segmentation. Future work may explore additional architectural enhancements or regularization techniques to further improve generalization and mitigate overfitting.

## DeepLabV3Plus

### Model Architectures

1. **DeepLabV3Plus**:
    - **Architecture**: DeepLabV3Plus is a state-of-the-art semantic segmentation model designed to capture both local and global context efficiently. It uses atrous convolution to extract multi-scale features without significantly increasing computational cost and integrates them with an encoder-decoder structure for refined predictions.

- **Encoder**: A ResNet-34 backbone pretrained on ImageNet for robust feature extraction.
- **Decoder**: Upsampling and refinement of the features to produce high-resolution segmentation maps.
- **Objective**: Perform binary segmentation with classes $[0, 1]$.

**Training Results**

- **DeepLabV3Plus**:
  - **Training and Validation Loss Trends**:
    * The model showed steady improvements in both training and validation loss, achieving its best performance in Epoch 7, where the validation loss reached its lowest value (0.1077).
    * The best-performing model checkpoint was saved during Epochs 1, 3, 4, and 7.
  - **Performance Observations**:
    * Significant performance gains were seen early in the training process, indicating efficient learning of features.
    * Validation loss stabilizes towards the later epochs, suggesting the model has effectively generalized to unseen data.

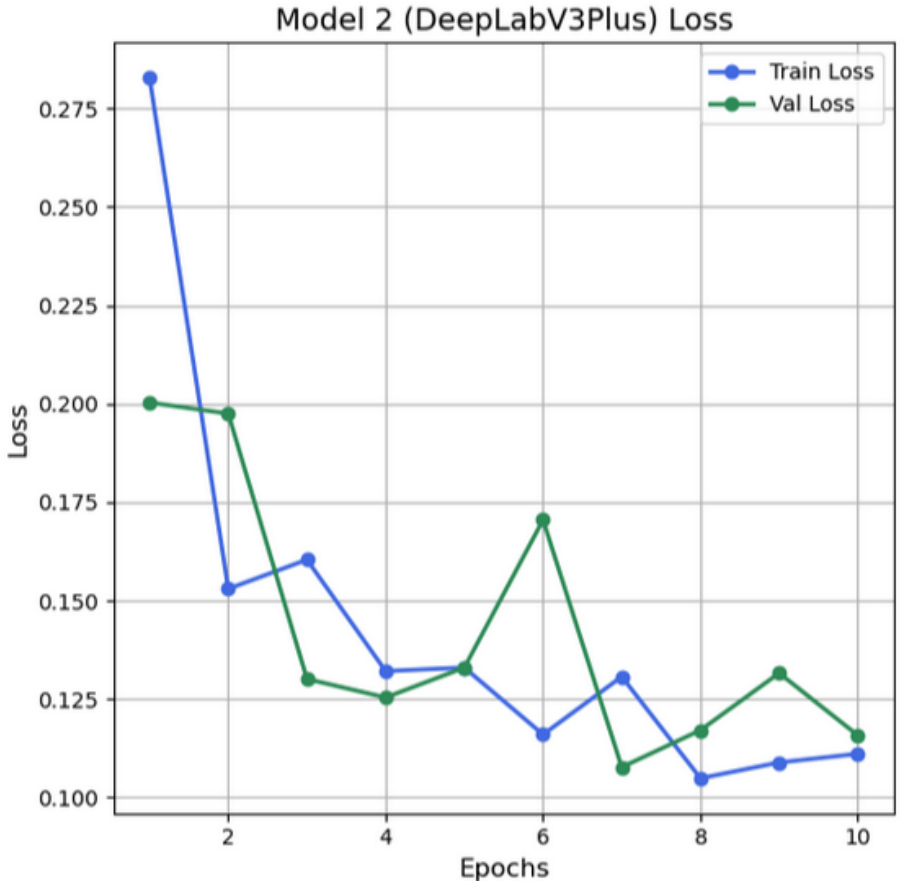



Figure 2: DeepLabV3Plus Model Loss

**Key Observations**

1. **DeepLabV3Plus** outperformed FPN in terms of both loss reduction and overall generalization on the validation set.

2. **FPN** showed stability but slower progress in training, possibly due to its pyramid architecture, which may need more fine-tuning of hyperparameters to unlock its full potential.

3. Both models performed binary segmentation with mask values $[0, 1]$, and the Dice Loss was used as the loss function for its ability to handle imbalanced datasets effectively.

**Conclusion** DeepLabV3Plus emerged as the more efficient and effective model in this training pipeline, with notable performance during Epochs 3 and 7. FPN displayed potential but may require additional tuning or longer training to match its performance.

## Ensemble Model

The third training was done using an ensemble combination of two semantic segmentation models—DeepLabV3Plus and Feature Pyramid Network (FPN).

### Model Architecture

The model architecture combines two individual models—**Model 1 (DeepLabV3Plus)** and **Model 2 (FPN)**—using an ensemble approach. The ensemble model aggregates their outputs with weighted averages, where Model 1 is given a weight of 60% and Model 2 40%.

- **Model 1 (DeepLabV3Plus)**: Utilizes atrous convolutions and an encoder-decoder structure to capture multi-scale context, generating high-resolution output ideal for semantic segmentation tasks.
- **Model 2 (FPN)**: Employs a top-down architecture with lateral connections to enhance feature extraction at multiple scales, capturing both low-level and high-level semantic information.

The ensemble model output is computed as:

$$\text{Ensemble Output} = 0.6 \times \text{Model 1 Output} + 0.4 \times \text{Model 2 Output}$$

This allows the ensemble to leverage the strengths of both individual models.

### Evaluation Pipeline

The evaluation pipeline involves the following steps for both models and the ensemble:

- **Configuration Class**: A `Config` class centralizes all hyperparameters, including device selection (GPU or CPU), model configurations (e.g., encoder type, input channels, activation function), and training parameters (e.g., learning rate, number of epochs, batch size).
- **Model Initialization**: Both models are initialized using the same encoder architecture (`ResNet-34`). For DeepLabV3Plus, no pre-trained weights were used, while FPN utilized ImageNet weights.
- **Inference**: Predictions are generated for each batch of test images, and class predictions are derived based on the highest scoring pixel for each model.
- **Accuracy Calculation**: Accuracy is computed by comparing predicted masks with ground truth:

$$\text{batch\_accuracy} = \text{torch.mean}((\text{predictions} == \text{ground truth}).\text{float32})$$

- **Results Aggregation**: Final accuracy is calculated as the mean of batch accuracies:

$$\text{final\_accuracy} = \frac{\text{total\_accuracy}}{\text{batch\_count}}$$

### Performance Evaluation and Results

The performance of the models is evaluated on 25 test batches. The results are summarized as follows:

- **Batch Accuracy**: Model 1 consistently outperforms Model 2 in each batch. The ensemble model typically performs better than Model 2 but slightly trails Model 1.
- **Final Accuracy**: The final accuracies after testing are:
  - **Model 1 (DeepLabV3Plus)**: 94.95%
  - **Model 2 (FPN)**: 88.77%
  - **Ensemble Model**: 94.90%
- **Ensemble Performance**: The ensemble provides a good compromise, improving upon Model 2's performance while maintaining high accuracy. However, it does not quite reach the performance of Model 1 due to the higher weight assigned to DeepLabV3Plus.

### Insights and Conclusion

The ensemble model leverages the strengths of both individual models and performs well compared to Model 2, while still trailing Model 1 slightly. This shows that ensemble methods can enhance performance by combining models with complementary strengths. Fine-tuning the individual models, particularly Model 2, could further improve the overall results.

# 5 Conclusion

The comparison and analysis of the results for both models yield the following insights:

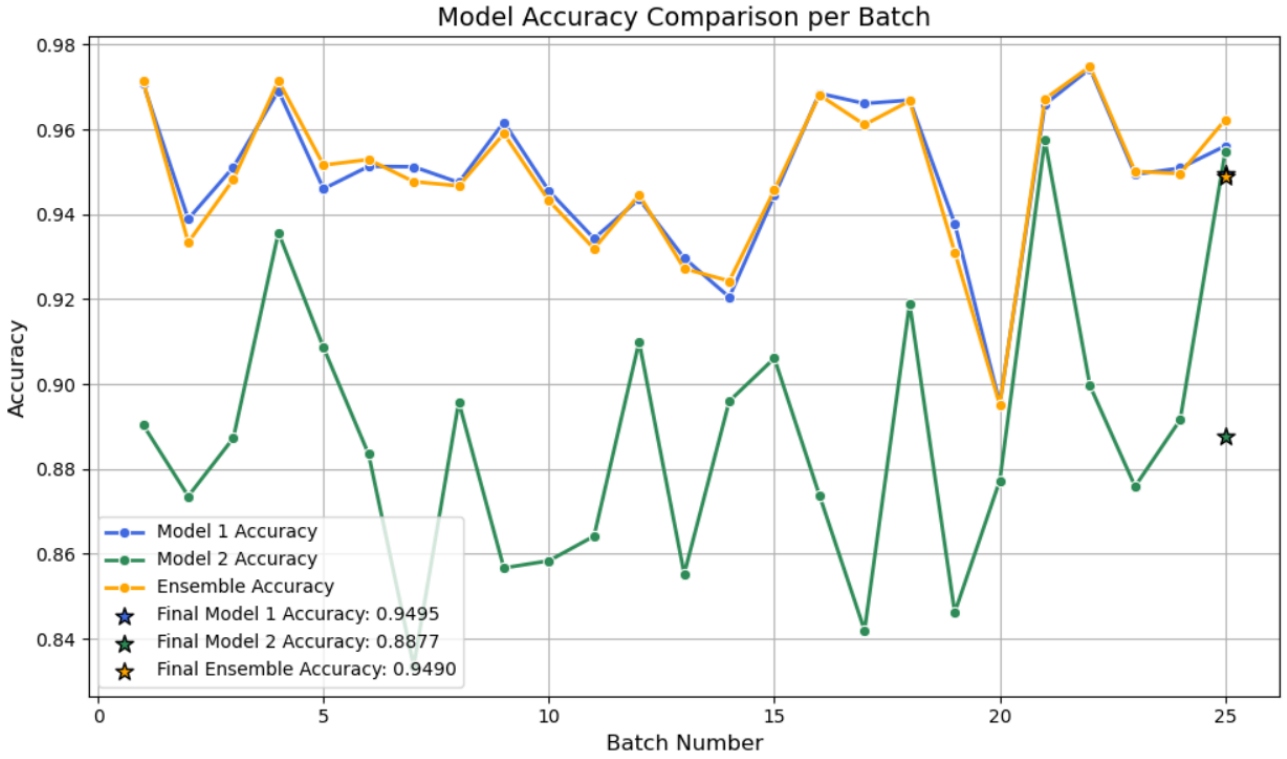



Figure 3: Model Accuracy Comparaison per Batch

- **DeepLabV3Plus** demonstrates strong and consistent performance, with an average accuracy of 0.9495. Despite slight fluctuations, the accuracy remains high, typically between 94-97%.
- **FPN** shows lower average accuracy, with a value of 0.8877. Its accuracy hovers around 85-90%, and while some batches perform well (such as Batch 21 and 22), FPN generally lags behind DeepLabV3Plus.
- **Ensemble Model** achieves an average accuracy of 0.9490, which is very close to DeepLabV3Plus's performance. The ensemble, benefiting from a weighted configuration of 0.6 for DeepLabV3Plus and 0.4 for FPN, outperforms FPN and closely matches DeepLabV3Plus.

From these observations, it is clear that:

1. DeepLabV3Plus outperforms FPN across all batches, establishing it as the stronger performer.
2. The ensemble model improves upon FPN, though it closely matches DeepLabV3Plus's performance. The combination of both models leverages their strengths, enhancing overall accuracy.
3. Accuracy fluctuations are noticeable in FPN, especially in certain batches (e.g., Batch 17), where its performance drops. The ensemble model helps mitigate these fluctuations by averaging the results.

In conclusion, the ensemble model provides a balanced and improved performance by leveraging the strengths of both models. Further fine-tuning of FPN could potentially enhance its accuracy. However, even without additional tuning, the ensemble serves as a robust solution, closely matching the performance of DeepLabV3Plus.

# BIBLIOGRAPHY

[1] divergent-nets, December 2024. [Online; accessed 8. Dec. 2024].

[2] Human Heart Project, December 2024. [Online; accessed 8. Dec. 2024].