1. Introduction

This project aims to predict systolic blood pressure (BP) by processing raw sensor signals and converting them into image representations. Two deep learning models were developed and evaluated:

- **Joint Model:** A multimodal architecture that fuses features from multiple image representations (scalogram, recurrence plot, Gramian Angular Field, and Markov Transition Field) using a Vision Transformer (ViT) backbone and a 3D convolutional neural network (3D-CNN).
- **Baseline Model:** A simpler CNN that uses only the scalogram images.

In addition to standard test set evaluation, uncertainty estimation via Monte Carlo dropout and K-fold cross-validation (using a small sample of the full dataset) were performed.

2. Data Preprocessing and Feature Engineering

2.1 Mounting and Loading Data

- **Google Drive Mounting:**
  The project data, stored in Google Drive (under the folder `BP_data_Cabrini2018`), was mounted using Google Colab's drive API.
- **File Loading:**
  Multiple file formats were processed:
  - **Text Files:** BeatScope Easy text files were loaded by detecting the header row.
  - **Excel Files:** Loaded via `pandas.read_excel()`.
  - **MAT Files:** Sensor signals were loaded from MATLAB files using `scipy.io.loadmat()`.

2.2 Signal Preprocessing

- **Filtering:**
  A Butterworth bandpass filter was designed and applied to each sensor signal. Filtering parameters were adapted based on the computed heart rate frequency (HRF) from the PPG signal:
  - For ECG, the filter used a range from HRF to 30 Hz.
  - For PPG and bio-impedance (BImp), the filter was set from HRF to 2.5×HRF.
- **Normalization:**
  Z-score normalization was applied to the filtered signals to standardize them.
- **Segmentation:**
  The signals were segmented into fixed-length windows (e.g., 10-second segments) to create uniform samples for further analysis.
- **HRF Computation:**
  The heart rate frequency was computed using a Fast Fourier Transform (FFT)

restricted to the frequency range corresponding roughly to 48–180 beats per minute (bpm).

2.3 Processing Sensor Data

For each subfolder in the dataset:

- **Text and Excel Files:** BP measurements were extracted .
- **MAT Files:**
  Raw signals from multiple sensors (ECG, PPG, and BImp) were accumulated, processed (filtering, normalization, segmentation), and stored in a structured format.

A summary function printed out details about each file and processed signal, facilitating data quality checks and inspection.

2.4 Image Representation Generation

Each segmented signal was transformed into several image representations:

- **Scalogram (Spectrogram):**
  Calculated via the Short-Time Fourier Transform (STFT) and log-scaled.
- **Recurrence Plot (RP):**
  Constructed using pairwise Euclidean distances with thresholding.
- **Gramian Angular Field (GAF):**
  Generated by converting the signal into polar coordinates and computing pairwise cosine sums.
- **Markov Transition Field (MTF):**
  Computed by quantizing the signal and modeling state transitions.

These representations were normalized or clipped as necessary and then stacked across sensor channels (ECG, PPG, BImp) into a single 3-channel image for each modality. This stacking produces visually distinct “patchwork” images that combine information from all sensors.

3. Model Architectures and Training

3.1 Vision Transformer (ViT) and 3D-CNN Fusion (Joint Model)

- **ViT Backbone:**
  The ViT splits an image into non-overlapping patches and projects them into an embedding space. Learnable positional embeddings and a Transformer encoder (with multi-head self-attention and MLP layers) are used to extract global features.
- **3D-CNN Fusion:**
  The 3D-CNN operates on the temporal stack of features extracted from four image representations (scalogram, RP, GAF, and MTF). Convolutional blocks reduce the spatial dimensions while preserving temporal information, followed by global pooling and a fully connected layer that outputs the BP prediction.

- **Loss Function:**
  A combined loss (mean squared error plus mean absolute error) was used to optimize model performance.
- **Training Setup:**
  The joint model was trained for 8 epochs with Adam optimizer. The best validation loss was tracked and used to update the final model.

3.2 Baseline Model

A simpler CNN model was built to process only scalogram images. It comprises two convolutional layers, followed by fully connected layers to predict the BP. Despite effective learning (as indicated by decreasing training loss), the baseline model's performance on the test set was substantially worse than the joint model.

3.3 Data Split and Evaluation

- **Dataset Splitting:**
  The dataset was divided into training (70%), validation (15%), and test (15%) sets.
- **Uncertainty Estimation:**
  Monte Carlo dropout was applied during inference by performing 50 forward passes with dropout enabled. The mean and standard deviation of the predictions provided a measure of uncertainty.
- **K-Fold Cross-Validation:**
  A 5-fold cross-validation was performed on a sample subset of the full data. The limited sample size led to high variability in the results, which is noted in the discussion.

4. Results and Explanations

4.1 Joint Model Performance

- **Test Set Evaluation:**
  - **Normalized Test Loss:** 0.1581
  - **Denormalized BP Predictions (mmHg):** Approximately `[154.52, 149.72, 148.16, 153.03]`
  - **Regression Metrics:**
    - **MAE:** 3.10 mmHg
    - **RMSE:** 3.15 mmHg

Explanation:
The joint model shows very high accuracy on the test set. The low MAE and RMSE indicate that the multimodal fusion (combining four distinct image representations) captures rich information about the physiological signals, enabling precise BP prediction.

4.2 Uncertainty Estimation

- **Monte Carlo Dropout:**

    - **Mean Predictions (mmHg):** Approximately `[116.53, 117.03, 117.52, 124.00]`
    - **Prediction Uncertainty (std, mmHg):** Approximately `[14.82, 14.86, 14.15, 16.28]`

Explanation:
By enabling dropout during inference, the model produces a distribution of predictions. The standard deviations represent the uncertainty in the model's predictions, which is valuable for assessing confidence in the estimated BP values.

4.3 K-Fold Cross-Validation

- **Setup:**
  A 5-fold cross-validation was performed using a sample subset of the full dataset.
- **Results per Fold (MAE / RMSE):**
    - Fold 1: 125.10 / 125.20 mmHg
    - Fold 2: 417.02 / 417.12 mmHg
    - Fold 3: 77.19 / 77.29 mmHg
    - Fold 4: 216.24 / 217.00 mmHg
    - Fold 5: 574.59 / 574.60 mmHg
- **Overall Averages:**
    - **MAE:** 282.03 ± 186.94 mmHg
    - **RMSE:** 282.24 ± 186.86 mmHg

Explanation:
The large variability across folds is attributed to using only a sample of the full dataset for cross-validation. The limited sample size in each fold leads to unstable performance metrics. In a full-scale experiment, more data would likely yield more consistent and reliable estimates.

4.4 Baseline Model Performance

- **Training Dynamics:**
  The baseline CNN trained on scalogram images showed decreasing loss over 8 epochs.
- **Test Set Evaluation:**
    - **MAE:** 29.54 mmHg
    - **RMSE:** 29.55 mmHg

Explanation:
The baseline model, which uses only a single modality (scalogram), underperforms compared to the joint model. The significantly higher errors demonstrate that relying on a single image representation is insufficient for capturing the complex dynamics inherent in the sensor signals, reinforcing the advantage of the multimodal approach.

5. Conclusion

- **Joint Model Advantages:**
  The multimodal joint model combining a ViT backbone and 3D-CNN fusion demonstrates excellent performance, achieving MAE and RMSE around 3 mmHg. Its ability to provide uncertainty estimates further strengthens its utility for reliable BP prediction.
- **Importance of Multimodality:**
  The superior performance of the joint model compared to the baseline CNN (which had errors of approximately 29.5 mmHg) highlights the benefit of integrating diverse image representations derived from raw sensor signals.
- **Data Sample Considerations:**
  The cross-validation results, while variable due to the use of a limited sample subset, underscore the need for using the full dataset to obtain robust and consistent performance metrics.

Future work should focus on expanding the dataset for cross-validation.