Velocity estimation from Doppler radar

Supervisor: Mark Henney Group: 61, Kazi (s230039), Saad (s240376), Ava (s242184) & Joshua (s241656)

02456 Deep learning, Fall 2024 Technical University of Denmark DTU

Introduction

The use of Doppler radar for measuring velocities utilizes the principle of a frequency range shift when the object under observation is in motion. We have a basic model already (which goes by the name of SpectrVelCNNRegr); however, the underlying task is to propose a more accurate model that does not incur additional complexity. The intention, in this case, is to minimize the error rate referred to as RMSE and/or reduce inference time. [1, 2, 3].

Data

- The dataset features spectrogram stacks ($\mathbf{6 \times 74 \times 918}$) from Doppler radar, targeting radial velocity ($\mathbf{[0, -50]}$ m/s) for regression tasks.
- It is divided into 1234 training, 83 validation, and 739 test observations, with power spectrograms normalized to $\mathbf{[0, 1]}$ and phase spectrograms to $\mathbf{[-\pi, \pi]}$.
- All spectrograms are resized to a consistent $\mathbf{74 \times 918}$ using bilinear interpolation.

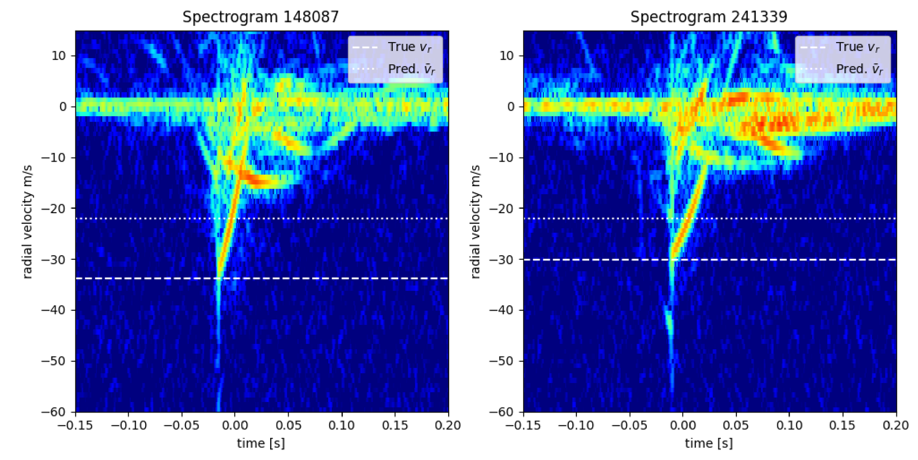

Figure 1: Visualization of multiple spectrograms with true and predicted radial velocities annotated.

Methodology

- **Iterative approach:** Adjusted the model incrementally through systematic experiments. Focused on parameters (e.g., activation functions, kernel sizes, dilation convolution, global average pooling).
- **Guided by insights:** Adjustments based on prior experiment results.
- **Goal:** Reduce RMSE and/or inference time while retaining total parameter count.

We experimented with six CNN architectures:

- **Base Model:** Standard CNN for baseline comparison.
- **GAPNet (II):** Adds global average pooling to reduce parameters.
- **DilateNetGAP (III):** Combines dilation and GAP for efficiency.
- **DilateNet (II):** Uses dilated convolutions for multi-scale features.
- **KernelNet (III):** Optimizes kernel sizes with dropout for learning.
- **SiLUNet (II):** Features SiLU activation for better gradient flow.

Model Architecture

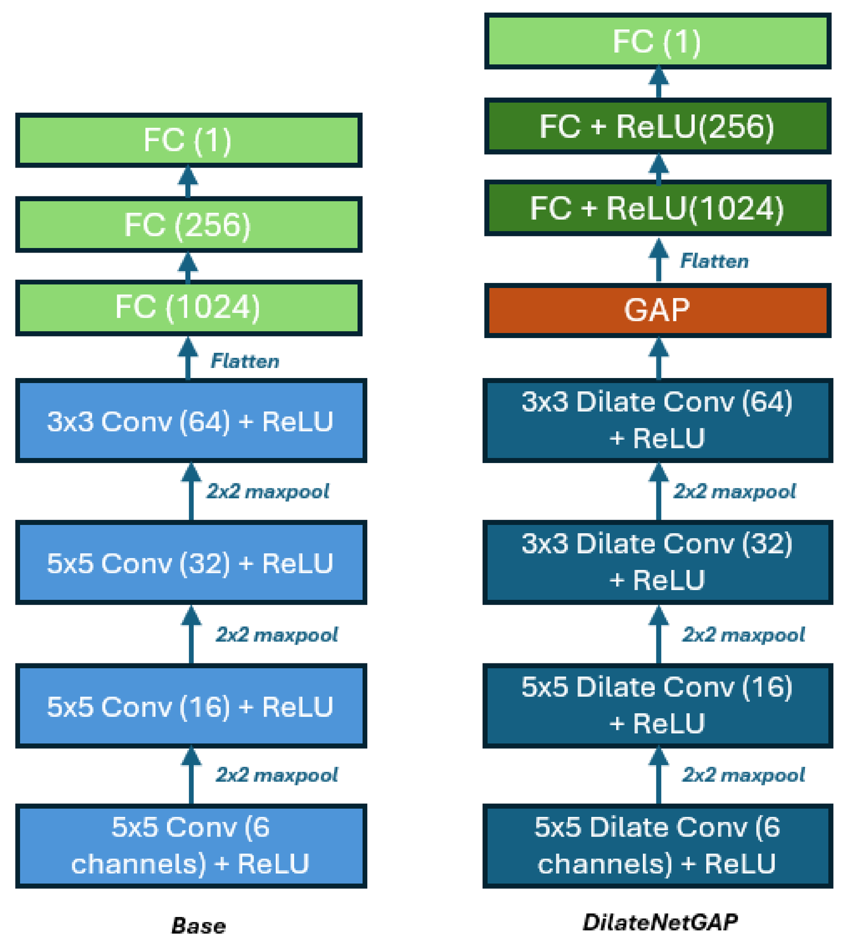

Figure 2: Base and **DilateNetGAP (III)** model architecture representations.

Results

All models were trained for 300 epochs, evaluating test loss, RMSE, and parameter count. DilateNetGAP achieved a favorable balance between performance and complexity, with a test RMSE of 1.38857 and only 0.502 million parameters. In contrast, SiLUNet attained a lower test RMSE of 0.62656 but required 38.4 million parameters, indicating higher complexity. DilateNet offered strong performance with a test RMSE of 0.95263 and a moderate parameter count of 22.4 million. Both training and test data informed modifications to optimize trade-offs.

Model	Parameters(M)	Training Loss	Test Loss	Training RMSE	Test RMSE
Base Model	38.4	0.25834	2.43421	0.50827	1.56020
GAPNet	0.535	2.86976	3.77800	1.69404	1.94371
DilateNetGAP	**0.502**	**1.92812**	**2.58312**	**1.38857**	**1.60721**
KernelNet	30.2	0.76844	2.80730	0.87661	1.67550
SiLUNet	38.4	0.39258	2.30997	0.62656	1.51986
DilateNet	22.4	0.90750	2.46732	0.95263	1.57077

Table 1: Performance Analysis of CNN Models on Training and Test Datasets

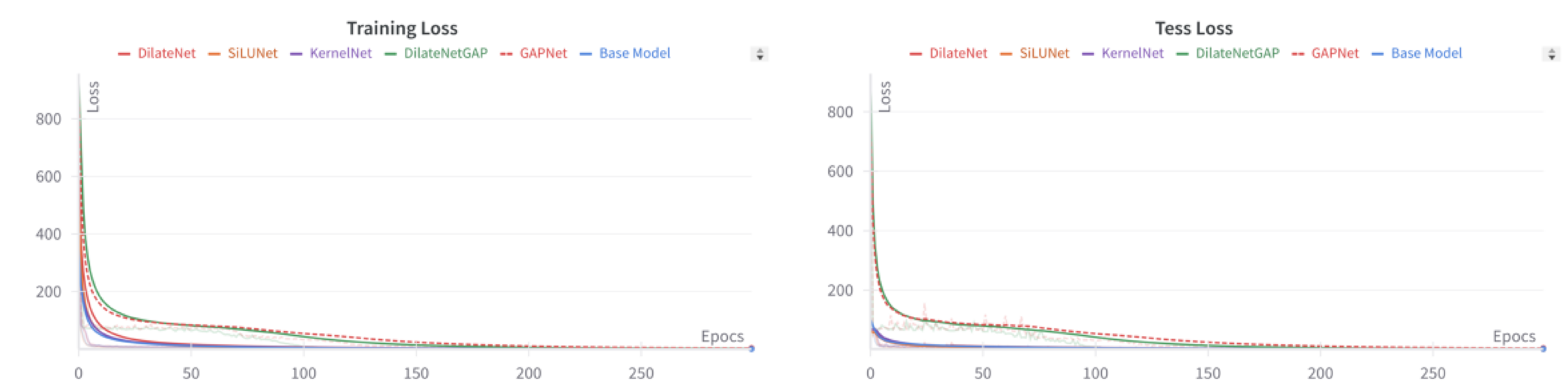

Figure 3: Training and test loss curves for models over 300 epochs. **DilateNetGAP** balances performance and complexity, while **SiLUNet** achieves lower losses at higher complexity.

Model Performance and Visualizations

We evaluated six models based on parameters, inference time, error metrics, and true vs. predicted values. The visualizations highlight model efficiency, error trends, and prediction accuracy, offering insights into trade-offs between complexity, speed, and accuracy.

Model	Inference Time (ms)	RMSE	MAE	R^2
Base Model	0.035116	1.455111	1.075233	0.964852
GAPNet	0.015214	1.682724	1.306023	0.952996
DilateNet	0.030518	1.264666	0.914189	0.973450
DilateNetGAP	**0.010509**	**1.411116**	**0.969049**	**0.966945**
KernelNet	0.008181	1.595109	1.225366	0.957764
SiLUNet	0.039998	1.343386	1.014406	0.970042

Table 2: Performance metrics for six CNN models, with **DilateNetGAP** balancing accuracy and speed.

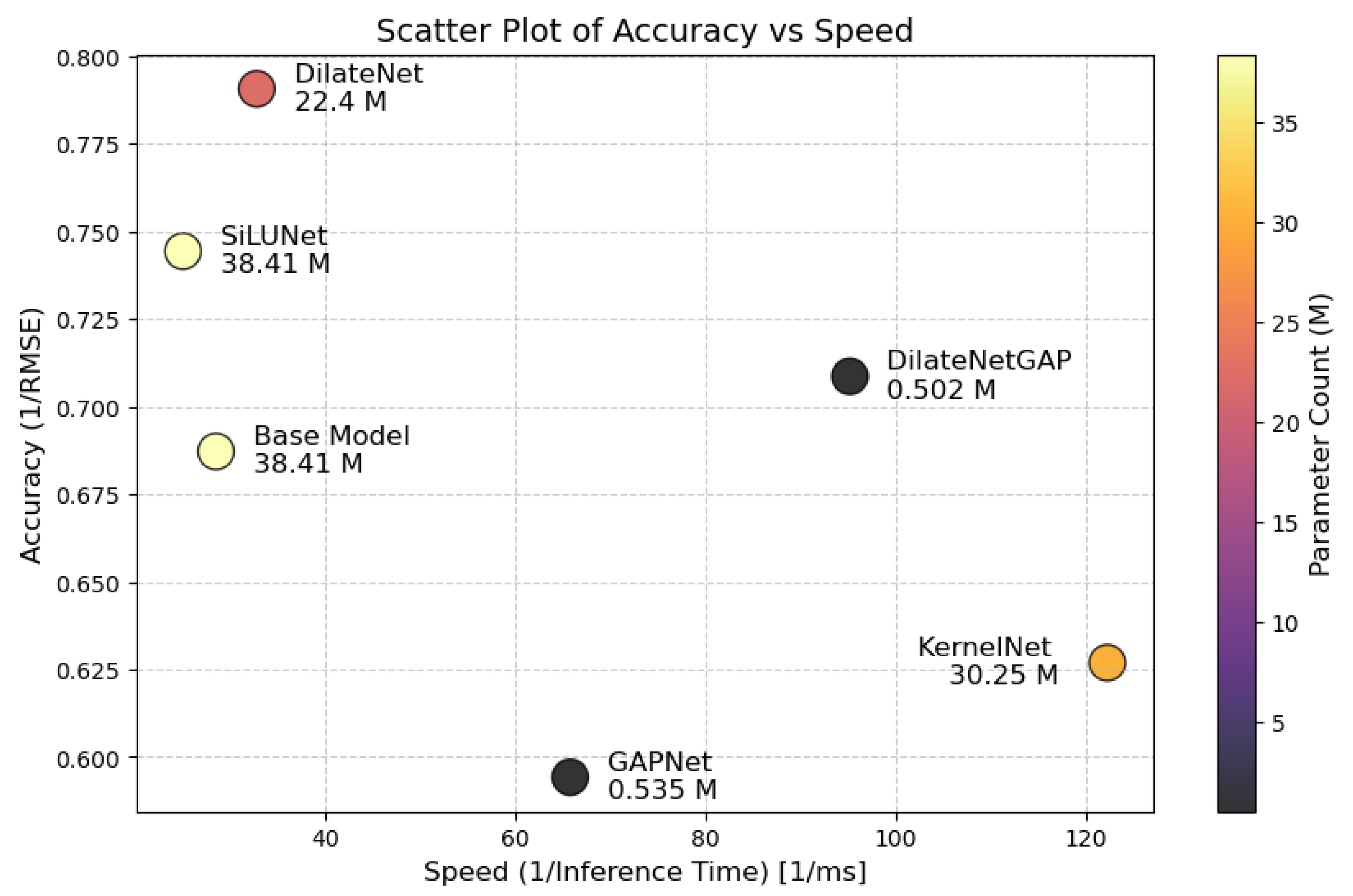

Figure 4: Scatter plot of RMSE vs. Inverse Inference Time. **DilateNetGAP** demonstrates an optimal balance between accuracy and speed.

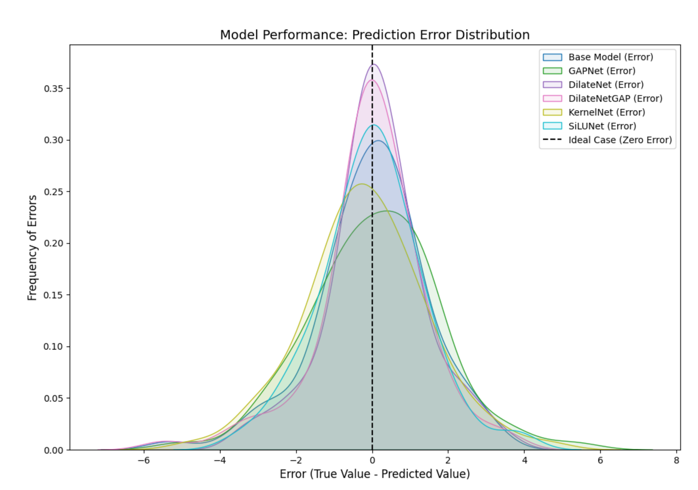

Figure 5: Prediction error distribution for all models.

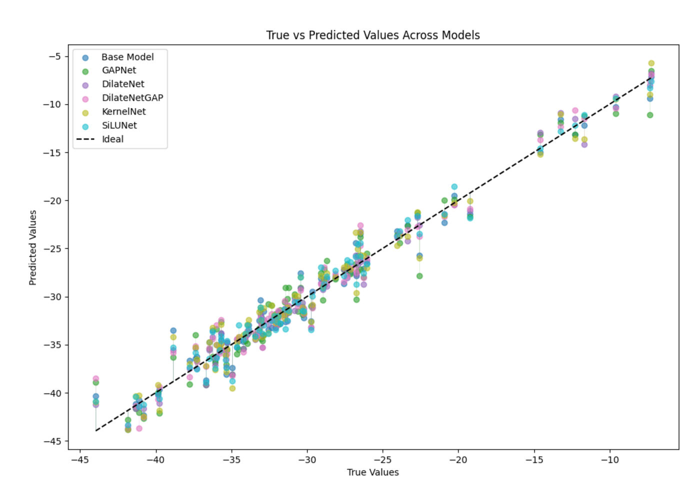

Figure 6: True vs. Predicted Values. The dashed line indicates perfect prediction.

Discussion

- **DilateNetGAP's Efficiency:** Achieves a test RMSE of 1.38857 with only 0.502M parameters, making it ideal for real-time applications.
- **Model Comparison:** SiLUNet achieves the lowest RMSE but with high complexity, while DilateNet offers a good balance of performance and resources.
- **Future Work:** Explore hybrid architectures, such as integrating SiLU activations into DilateNetGAP, combining dilation with attention mechanisms, and using advanced regularization techniques like dropout variations and loss function optimization.

References

[1] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. URL `https://www.deeplearningbook.org/`.

[2] OpenAI. Chatgpt, 2021. We would like to acknowledge the use of ChatGPT which assisted in generating and refining code, debugging, and enhancing our documentation.

[3] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. 2016.

s242184dtu.dk, s241656@dtu.dk, s230039@dtu.dk, s240376@dtu.dk