Meteorological Super-Resolution
vs
Wind Representations

Gruppo 21
Marzia De Maina, Matteo Galiazzo, Federica Santisi

Alma Mater Studiorum - Università di Bologna

Dipartimento di Informatica - Scienza e Ingegneria (DISI)

Prof. **Fabio Merizzi**

Purpose

The purpose of the project is to study the impact of different wind representations in the context of super-resolution.

Problem explaination and theory recap

[1]

Datasets and Preprocessing

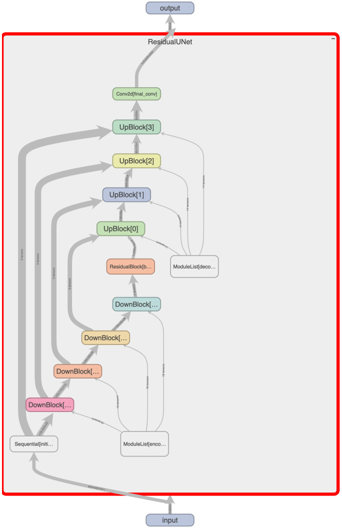
Model and hyperparameters - Introduction

Our model is based on the U-Net architecture, extended with residual connections for better gradient flow and stability. The input has 2 channels (u, v) and the output also has the same 2 channels representing the high-resolution version of the same fields. The key building blocks are:

- **Residual Block**: it's the core module of the architecture. It has two paths
  - One that applies convolutions and normalization.
  - One that does nothing (the residual application).

  The model learns how much to use either path.
- **Downsampling Block**: reduces the spatial resolution and increases the feature size. It uses stride 2 to downsample and includes a residual block.
- **Upsampling Block**: these are the mirror of down blocks. They use bilinear upsampling to increase the spatial dimension instead of transposed convolutions. Then, we concatenate the skip connections from the encoder and use a 1×1 convolution to reduce the number of channels after concatenations.

Model and hyperparameters



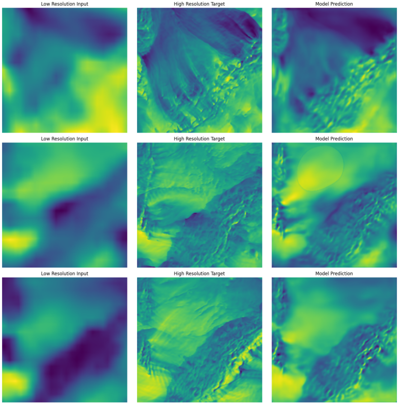
Model and hyperparameters - Transposed vs Bilinear

Network Type	Parameters Count	Training Time	Test Loss	Test SSIM
Transposed Conv	13,041,922 (13.0 M)	54:03 16.22s/it	0.197755	0.709096
Bilinear Interp	3,607,586 (3.6 M)	08:04 2.42s/it	0.171656	0.749828

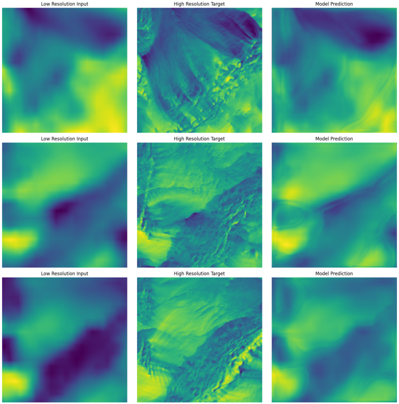
Tabella: Comparison of bilinear interpolation vs transposed convoltuion.

Model and hyperparameters - Loss

Even the best model couldn't get decent results with the suggested MSE loss, so we adopted a combined loss approach



L1 + SSIM loss. SSIM: 0.7556



MSE loss. SSIM: 0.7022

Model and hyperparameters - Loss

```
class L1SSIMLoss(nn.Module):
def __init__(self, alpha=0.85, ssim_window_size=11, ssim_data_range=1.0,
     ssim_channel=1):
    super(L1SSIMLoss, self).__init__()
    self.alpha = alpha
    self.l1_loss = nn.L1Loss() # Mean Absolute Error
    self.ssim_loss_fn = SSIMLoss(window_size=ssim_window_size, data_range=
         ssim_data_range, channel=ssim_channel)

def forward(self, y_pred, y_true):
    ssim_val_loss = self.ssim_loss_fn(y_pred, y_true)
    l1_val_loss = self.l1_loss(y_pred, y_true)

    combined_loss = self.alpha * ssim_val_loss + (1 - self.alpha) *
         l1_val_loss
    return combined_loss
```

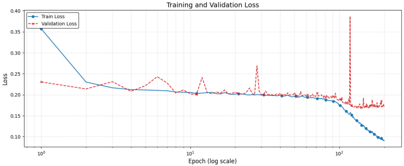
Model and hyperparameters - Hyperparameters

Channels	Epochs	Loss	Batch Size	LR	LR Scheduler	Weight Decay	Test Loss
[32, 64, 128]	200	L1 + SSIM	8	1.00E-03	NO	1.00E-05	0.19090
[16, 32, 64, 128]	200	L1 + SSIM	8	1.00E-03	NO	1.00E-05	0.17700
[16, 32, 64, 128, 256]	300	L1 + SSIM	8	1.00E-03	NO	1.00E-05	**0.16768**
[16, 32, 64, 128, 256]	300	MSE + SSIM	8	1.00E-03	NO	1.00E-05	<u>0.15464</u>
[16, 32, 64, 128, 256]	300	MSE	8	1.00E-03	NO	1.00E-05	0.39491
[16, 32, 64, 128, 256, 512]	300	L1 + SSIM	8	1.00E-03	NO	1.00E-05	0.19207
[16, 32, 64, 128, 256, 512]	300	L1 + SSIM	8	1.00E-03	YES	1.00E-05	0.19207

Tabella: Performance comparison of different network configurations. Best and runner-up models are highlighted in bold and underline, respectively.

Training

Training curves for the best model (loss and SSIM):



Loss curve

SSIM curve

Results

Final Considerations

References

[1] Fabio Merizzi, Andrea Asperti e Stefano Colamonaco. "Wind speed super-resolution and validation: from ERA5 to CERRA via diffusion models". In: *Neural Computing and Applications* 36.34 (2024), pp. 21899–21921.