Onboard Deep Learning for Efficient Small Satellite Reflectance Retrievals and Downlink

Sarvesh Garimella and David Ryglicki
MyRadar
Portland, OR
vesh@myradar.com

myradar™

Background: MyRadar's Use Case & Mission

- Mission
  - Democratizing environmental intelligence in a changing climate
  - Nowcasting + Alerts
- HORIS CubeSat Platform
  - RGB camera, hyperspectral imager, thermal sensor
  - Low-power AI chipset
  - Constellation with < hourly revisit
  - Faster detection –> **alerting**

myradar™

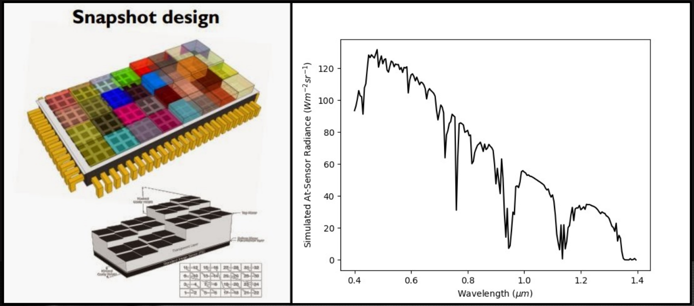
Methods: Alerting from Orbit

- Orbital sensor platforms provide rich datasets
- Data outpaces processing ability, operational cadence
- Can we apply AI techniques to overcome bottlenecks and send alerts from orbit?
    - Reflectance Retrievals
    - Detection/Segmentation
    - Compression/Downlink

myradar™

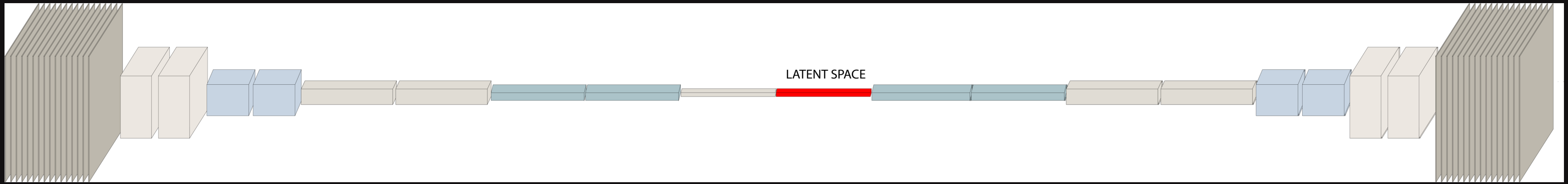
Methods: AI Reflectance Retrievals

- Comparing Empirical Line Method to RT calculations
- Atmospheric vertical structure and aerosol profiles required
- Lookup tables without downlink requirement
- Deep learning to constrain **aerosol profile** from RGB context sensor
- N-class classification of aerosol scene using transfer learning



Methods: AI Compression

- Convolutional autoencoders for compression tasks
- Self-supervised training on input and reconstructed spectra/images
- 2D and 1D networks compared
- Focus on 1D network for better memory characteristics and better literature performance (Kuester et al., 2020)



myradar™

Methods: AI Super Resolution

Model	Filters	Blocks	Size, weights	Size, serialized
EDSR	128	32	42 MB	126 MB
DSen2	64	12	3.5 MB	11 MB
Res-U-net	[32, 64, 128, 256, 512, 1024]	3 per filter group	207 MB	620 MB

- Multiple architectures designed for potential failovers
- MAE and MAE + DSSIM were used as loss functions, with a weight of 0.25 on DSSIM
    - DSSIM = 1.0 – SSIM (Wang et al. 2004)

- 256 x 256 subdomains extracted from Sentinel-2 20-m RED (Band 4) and NIR (Band 8a)
- Batch size of 64, shuffle size of 96
- Learning rate: 1.0e-4 exponentially decay to 1.0e-5 over 200 epochs
- Early stopping based on validation loss (10 epochs of no improvement)

myradar™

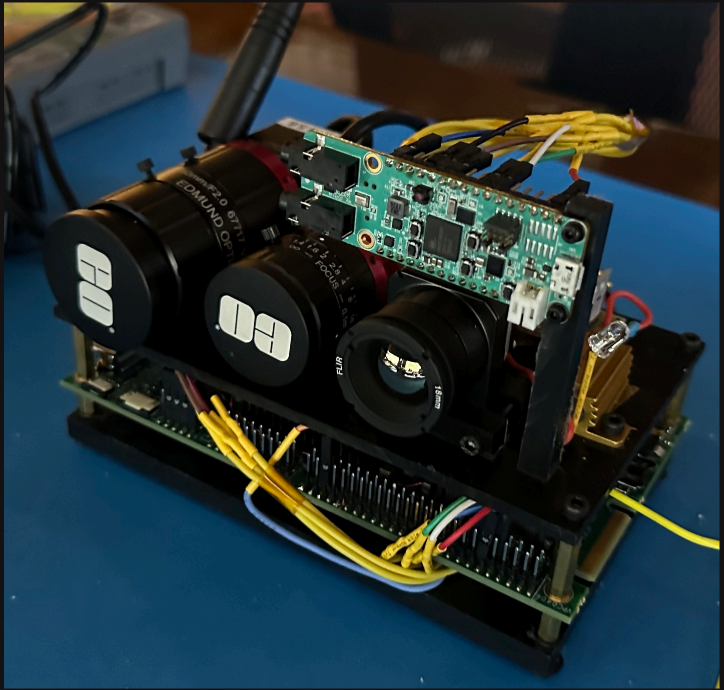
Methods: Alerting Duty Cycle

- Uplink (atmospheric profile + aerosol profile specific) lookup tables
  - Apply sensor transfer function: simulation → sensor bandwidths and QEs
- Context sensor in "scan mode" for scenes of interest
- Switch to "active mode" to collect radiance spectra + context image
  - Normalize measured spectra with simulated white reflectance spectrum
- Perform spectral classification to determine alert / no alert
- Downlink encoded bitmap/flags for detected features

myradar™

Results: “Flat Sat” Tests

- 30 ms inference time using 4 mJ
- 30 GOPS with AI core drawing 13mW maximum power for 63 ms (capture+inference), using ~1 mJ
- Total energy consumption for acquiring and processing one image ~ 7 mJ.
- 1 Hz operation for 1 hour was 25 J.



myradar™

Results: Context and Compression AI

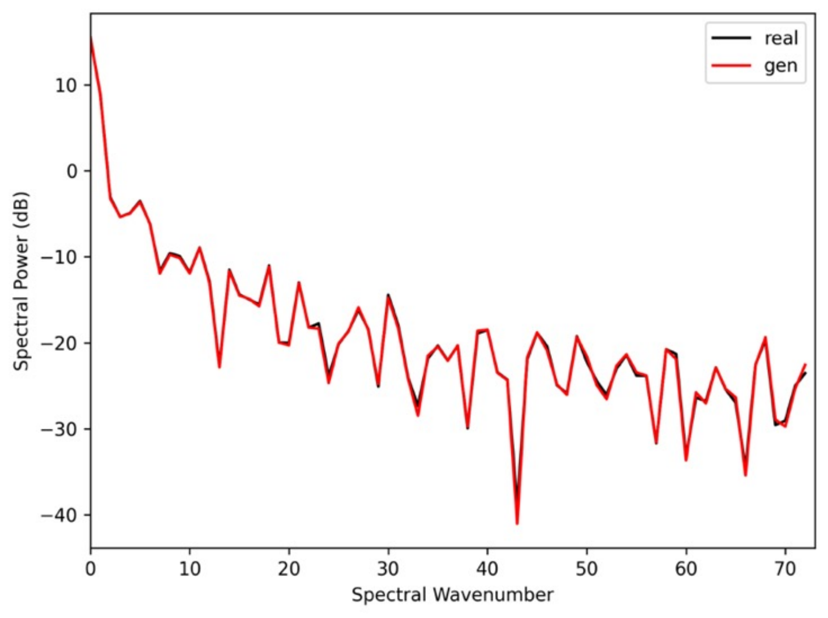
- AI-assisted aerosol profile lookup (64x64x3 pixel tiles) on low power AI chipset
  - 14-layer SimpleNet V2
    - 98% 6-class accuracy
    - 373k params, 14 MB on disk
    - 0.3 W, 9.2 s inference (slow, layer I/O)
  - 7-layer "Simpler" Net
    - 97% 6-class accuracy
    - 55k params, <1 MB on disk
    - <1 mW, 15 ms

- Lightweight (57k params) 1D fully-Convolutional Autoencoder
  - <150 ms compression/inference on single core CPU with 256x256x145 datacubes
  - MAE of 6E-4 DN
  - 18:1 dB SNR suitable for sub-pixel classification (Kuester et al., 2020)
  - 100-150 ms for acquisition of a similar datacube
  - Near real-time compression

myradar™

Results: Compression Power Spectra

- Average power spectral density for real and reconstructed spectra
  - Low wavenumber power = broad-scale features
  - High wavenumber power = to finer-scale spectral features
- Preservation of spectral power distribution



myradar™

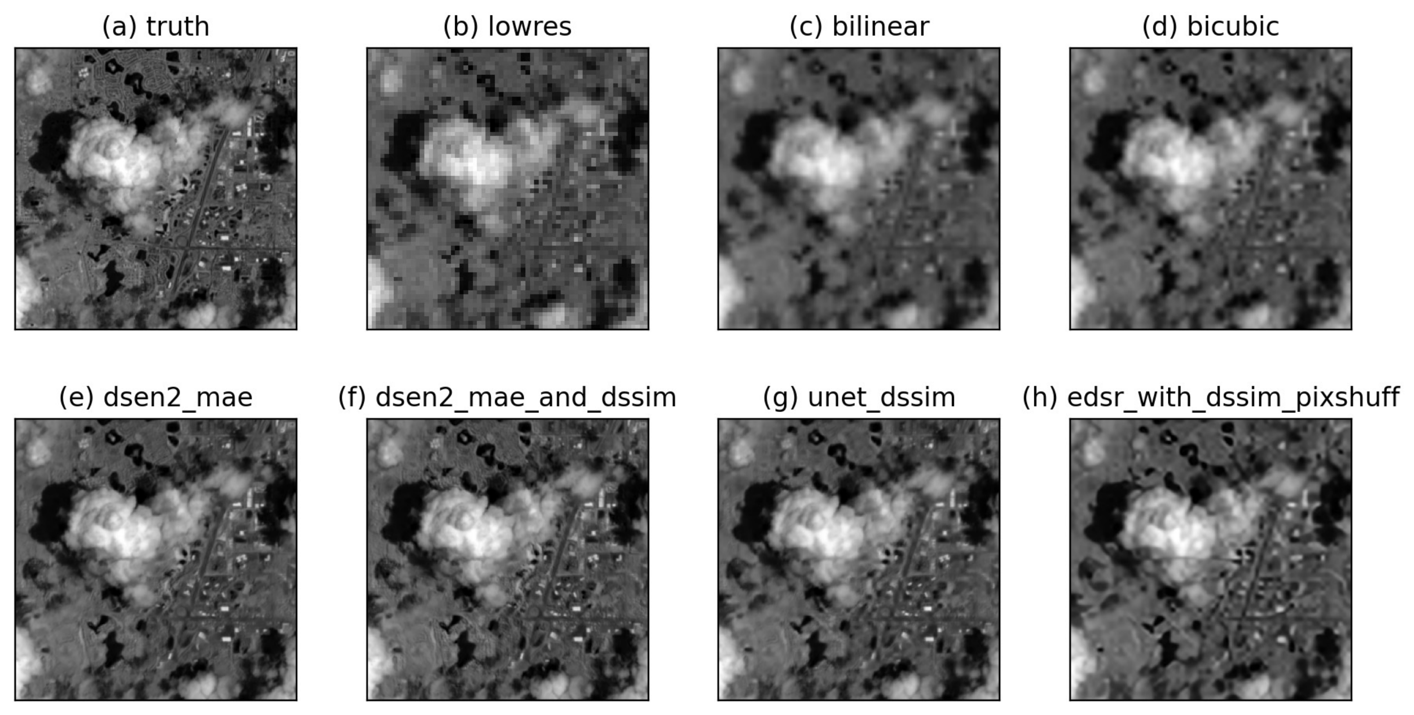
Results: Super Resolution AI

Model	MAE	MAPE	PSNR	SSIM	SRE	TVAR
EDSR, MAE	0.042	102950	27.246	0.696	19.478	2019
EDSR, MAE + DSSIM	0.041	105235	27.422	0.719	19.723	2330
DSen2, MAE	0.037	45956	28.043	0.713	20.026	2948
DSen2, MAE + DSSIM	0.038	55472	28.122	0.737	20.682	3243
U-net, MAE	0.036	42084	28.748	0.748	20.956	2295
U-net, MAE + DSSIM	0.036	52753	28.773	0.772	20.843	2770

- PSNR = *Peak Signal-to-Noise Ratio*
- SSIM = *Structural Similarity*
- SRE = *Signal to Reconstruction Error Ratio* (Lanaras et al. 2018)
- TVAR = *Total Variation*

myradar™

Results: Super Resolution AI Output



myradar™

Discussion/Conclusions

- Many use cases where improving latency / time resolution is more important than improving spatial resolution: wildfires, defense
- AI techniques can enable low-power real-time onboard processing
- Address bottlenecks for compute and downlink to allow generating alerts from remote sensing data at the edge
- Requires further laboratory studies for duty cycle determination + tests in the field and on orbit

Future Work

- Refine AI performance on flight model hardware
- Diversify training data collected
- Late 2024 launch of science pathfinders
- Orbital training data collection
- Alerting demonstration

myradar™

Acknowledgements

- SPIE, Organizers
- NOAA SBIR NA23OAR0210337
- Paul Kocyla, Pradipta Dey, Portland State University, Purdue University

myradar™

Backup Slides

Data Details: Sentinel-2

- Acquisition
  - GeoJSONs built on geojson.io
  - SentinelSat API to filter for cloud cover <1% for all dates/times in 2021
  - Georgia, Montana, South Dakota, NM-CO border, northern CA, central IN, and central FL (testing)

myradar™

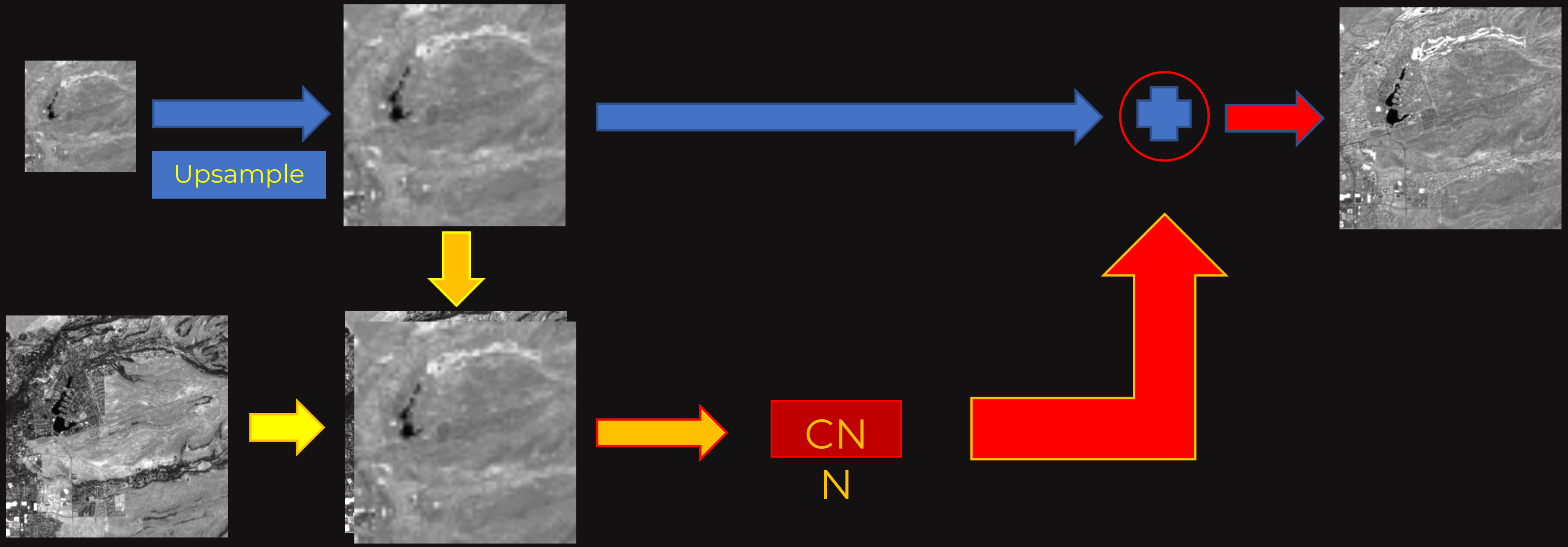
Data Details: Sentinel-2

- Super-resolution
  - 256 x 256 subdomains extracted from 20-m RED (Band 4) and NIR (Band 8a); scaled by 0.5 and 99.5 percentiles
    - Following techniques in Lanaras et al. (2018; Gaussian blurring, area-averaging) downsampled NIR 20-m resolution to 80-m resolution
      - Matches our sensors' differences (4X)
    - ~38K training pairs, ~8K validation pairs, ~4K testing pairs; training subjected to additional augmentation (rotation, flipping)

Data Details: Sentinel-2

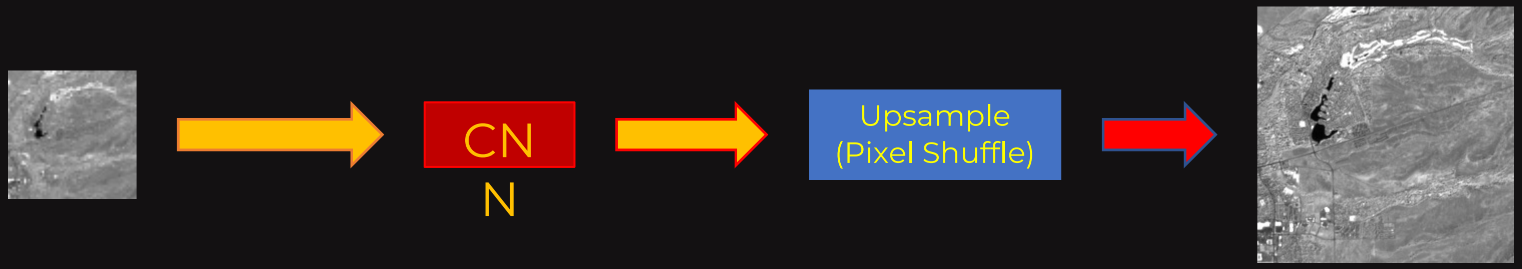
- Spectral Classification
  - ESA WorldCover 10m dataset (Zanaga et al. 2022)
    - 11 land-use categories
    - Remapped to 20-m Sentinel-2 TIFFs using GDAL (nearest neighbor)
    - 50,000 random points extracted per scene (~2 million total), limited to boreal summer
      - 80/20 training/testing split; stratified shuffle technique employed to maintain distribution
  - Additional variables derived: NDVI, GEMI, MSAVI, NDWI, NDBI, MNDWI, ANDWI, EVI
  - Four-fold stratified split grid search cross-val for determining best model, optimizing for *balanced accuracy*.

Methodology: AI architecture, SISR (cond.)



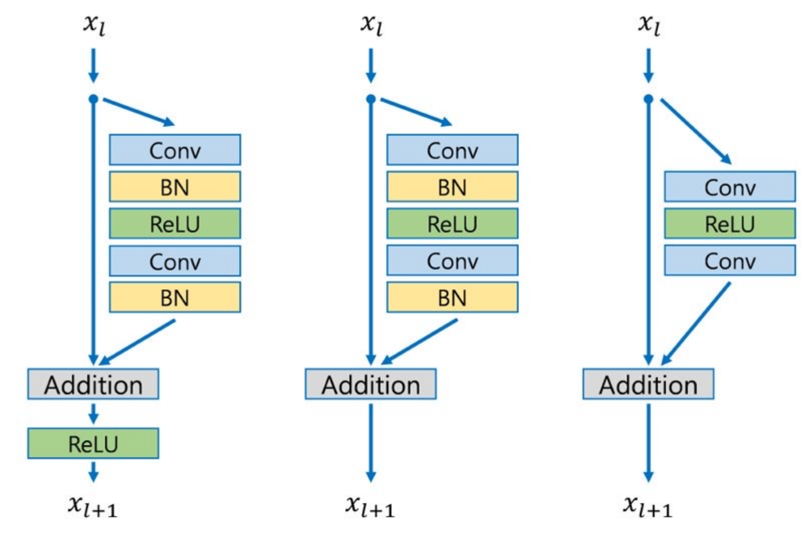
myradar™

Methodology: AI architecture, SISR (uncond.)



Methodology: SISR details

- Three models (1 uncond., 2 cond.)
  - Enhanced Deep Residual Network (EDSR; Lim et al. 2017)
    - Unconditioned
    - Uses (c) Residual Block
  - DSen2 (Lanaras et al. 2018)
    - Conditioned
    - Derivative of EDSR
    - Uses (c) block with additional Scaling parameter
  - U-Net (Ronneberger et al. 2015)
    - Conditioned
    - Res-U-net, where convolutions are replaced with (a) residual blocks
    - Strided down; nearest-neighbor up



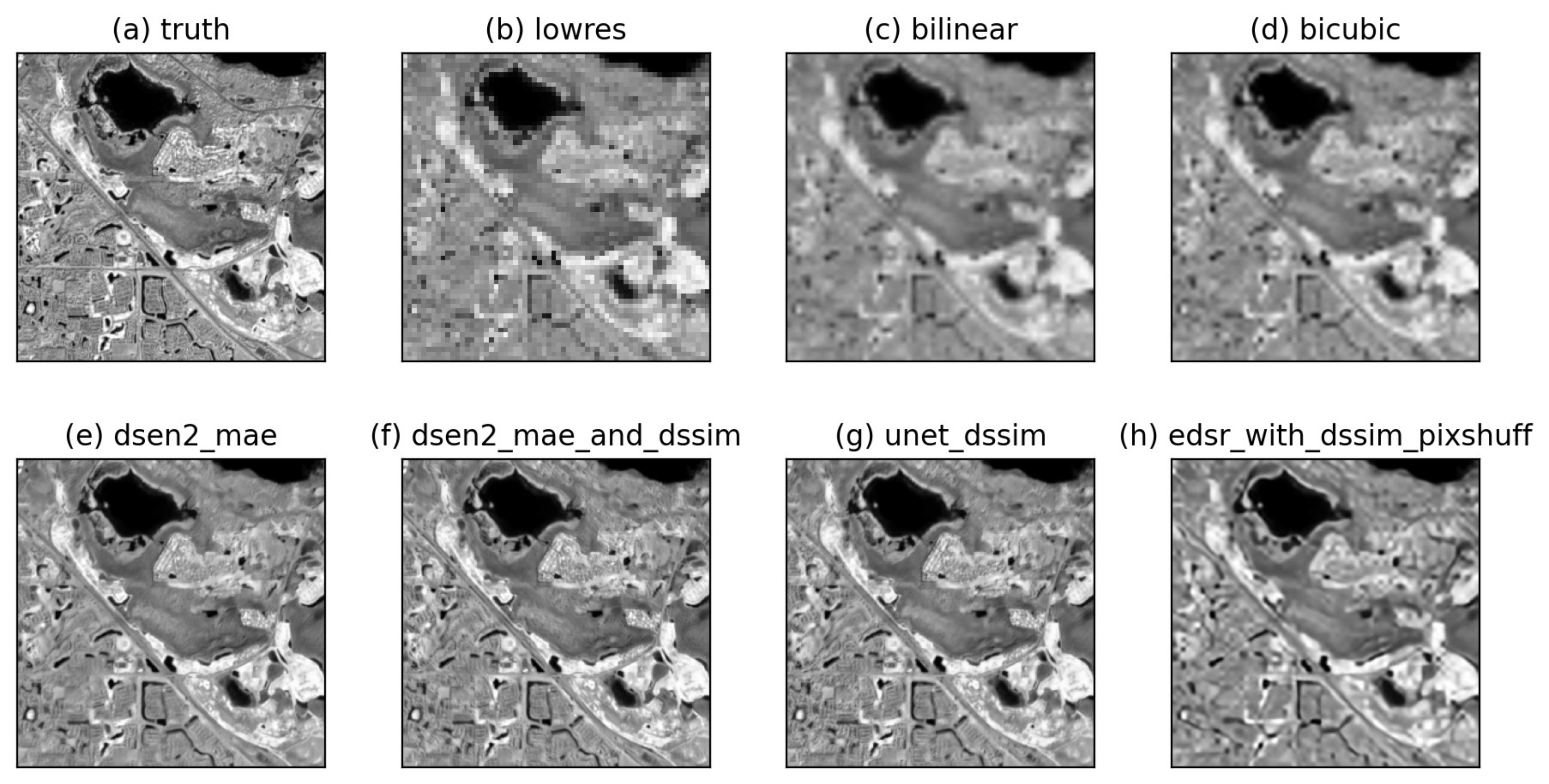
(a) Original (b) SRResNet (c) Proposed

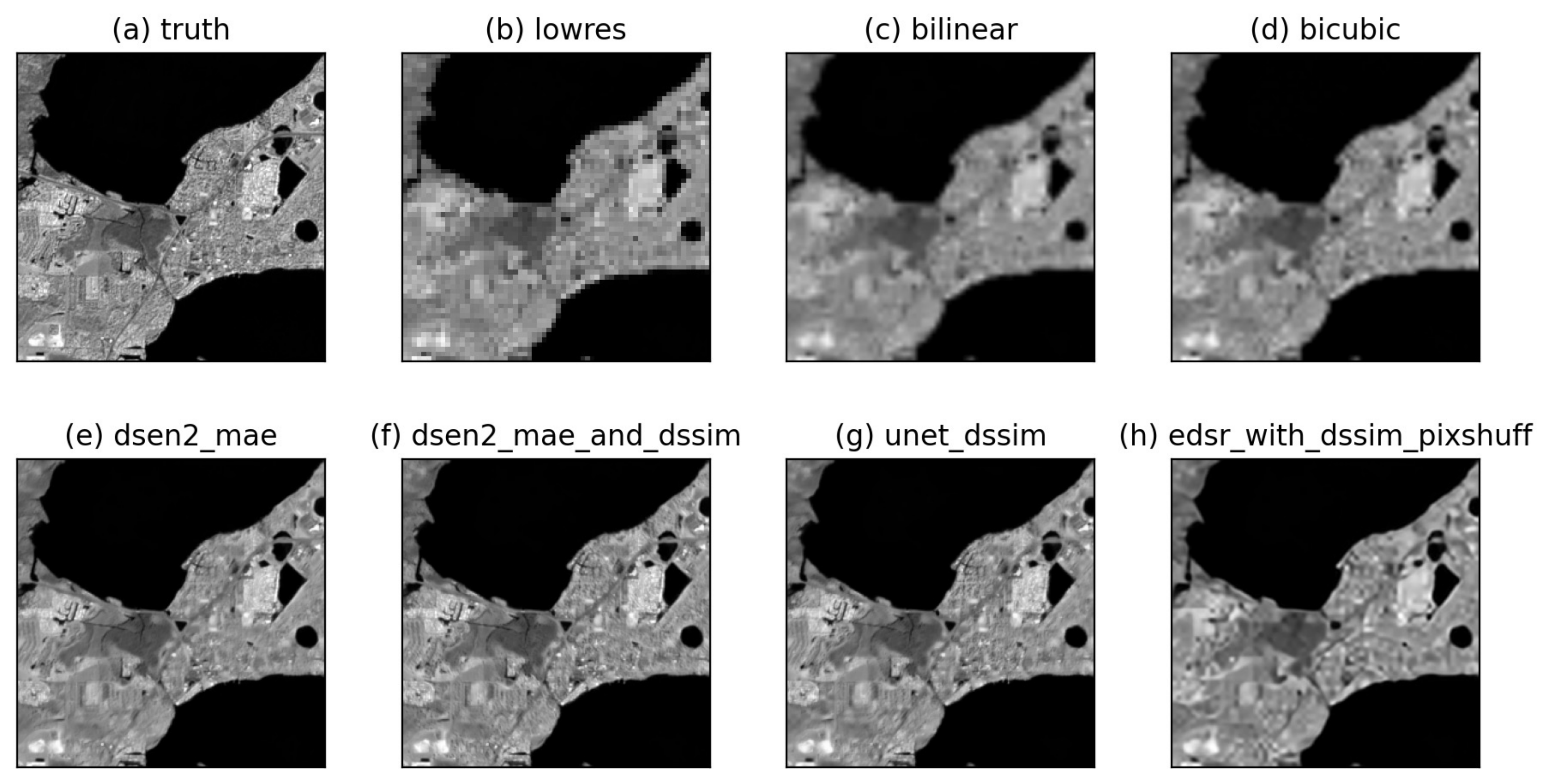
myradar™

Methodology: SISR model comparison

Model	Filters	Blocks	Size, weights	Size, serialized
EDSR	128	32	42 MB	126 MB
DSen2	64	12	3.5 MB	11 MB
Res-U-net	[32, 64, 128, 256, 512, 1024]	3 per filter group	207 MB	620 MB

- Multiple architectures designed for potential failovers
- MAE and MAE + DSSIM were used as loss functions, with a weight of 0.25 on DSSIM
    - DSSIM = 1.0 – SSIM (Wang et al. 2004)
- Batch size of 64, shuffle size of 96
- Learning rate: 1.0e-4 exponentially decay to 1.0e-5 over 200 epochs
- Early stopping based on validation loss (10 epochs of no improvement)


(a) truth
(b) lowres
(c) bilinear
(d) bicubic
(e) dsen2_mae
(f) dsen2_mae_and_dssim
(g) unet_dssim
(h) edsr_with_dssim_pixshuff



(a) truth
(b) lowres
(c) bilinear
(d) bicubic
(e) dsen2_mae
(f) dsen2_mae_and_dssim
(g) unet_dssim
(h) edsr_with_dssim_pixshuff