Supplementary Materials

Multiscale Feature Extraction with Wavelet Scattering Transform for Remote Sensing Vegetation Classification via Machine Learning

Mattia Bruscia William Nardin Xiaoxu Guo
Limin Sun Giulia Franchi

1 Synthetic Noise Conditions

Gaussian Noise (Additive White Gaussian Noise, AWGN)

Characteristics:

- Type: additive, signal-independent
- Parameter: standard deviation σ
- Mathematical model: $I_{\text{noisy}}(x, y) = I(x, y) + N(0, \sigma^2)$
  - $N(0, \sigma^2) \sim$ normal distribution with mean 0 and variance σ^2

Tested intensity levels:

- $\sigma = 10$ (low): `dataset_rgb_gaussian_10` – minimal thermal noise
- $\sigma = 30$ (moderate): `dataset_rgb_gaussian_30` – standard sensor degradation
- $\sigma = 50$ (high): `dataset_rgb_gaussian_50` – high electronic noise conditions

Rationale: Simulates thermal noise from CCD/CMOS sensors, electronic interference from the acquisition circuit, and random disturbances during analog-to-digital conversion. It represents the most common noise type in photographic sensors.

Poisson Noise (Shot Noise)
Characteristics:

- Type: signal-dependent
- Parameter: scaling factor λ
- Mathematical model: $I_{\text{noisy}} \sim \text{Poisson}(I \times \lambda)/\lambda$
  - Noise variance is proportional to signal intensity

Tested intensity levels:

- $\lambda = 40$ (low): `dataset_rgb_poisson_40` – adequate lighting conditions
- $\lambda = 60$ (moderate): `dataset_rgb_poisson_60` – reduced lighting
- $\lambda = 100$ (high): `dataset_rgb_poisson_100` – low-light conditions

Rationale: Simulates shot noise caused by the quantized nature of photons. Particularly relevant in low-light conditions where photon count is limited. Poisson noise is intrinsic to the light detection process and cannot be completely eliminated.

Salt & Pepper Noise (Impulse Noise)
Characteristics:

- Type: impulsive, binary
- Parameter: percentage of affected pixels (p)
- Mathematical model:
  - With probability $p/2$: pixel $\rightarrow 0$ (pepper, black)
  - With probability $p/2$: pixel $\rightarrow 255$ (salt, white)
  - With probability $1 - p$: pixel unchanged

Tested intensity levels:

- $p = 5\%$ (low): `dataset_rgb_salt_and_pepper_5` – sporadic corruption
- $p = 15\%$ (moderate): `dataset_rgb_salt_and_pepper_15` – visible degradation

- $p = 25\%$ (high): `dataset_rgb_salt_and_pepper_25` – severe corruption

Rationale: Simulates data transmission errors, dead/hot pixels in the sensor, impulsive electromagnetic interference, and memory errors during acquisition or storage. Common in wireless transmission systems and defective storage devices.

Speckle Noise (Multiplicative Noise)

Characteristics:

- Type: multiplicative (proportional to local intensity)
- Parameter: variance $\sigma^2_{\text{speckle}}$
- Mathematical model: $I_{\text{noisy}} = I + I \times N(0, \sigma^2_{\text{speckle}})$
  - Noise is multiplied by pixel intensity

Tested intensity levels:

- $\sigma^2 = 15$ (low): `dataset_rgb_speckle_15` – subtle speckle pattern
- $\sigma^2 = 35$ (moderate): `dataset_rgb_speckle_35` – visible granular texture
- $\sigma^2 = 55$ (high): `dataset_rgb_speckle_55` – significant texture degradation

Rationale: Simulates speckle noise typical of coherent imaging systems such as Synthetic Aperture Radar (SAR), laser imaging, and ultrasound. Although less common in RGB optical imaging, it tests the robustness of methods to texture-dependent degradations that may occur under particular atmospheric conditions (fog, haze).

Uniform Noise (Uniform Random Noise)

Characteristics:

- Type: additive, uniform distribution
- Parameter: variation range $[-\text{r}, +\text{r}]$
- Mathematical model: $I_{\text{noisy}} = I + U(-\text{r}, +\text{r})$
  - $U(-\text{r}, +\text{r}) \sim$ uniform distribution in the interval $[-\text{r}, +\text{r}]$

Tested intensity levels:

- $r = 10$ (low): `dataset_rgb_uniform_10` – minimal variation
- $r = 25$ (moderate): `dataset_rgb_uniform_25` – visible fluctuation
- $r = 40$ (high): `dataset_rgb_uniform_40` – marked degradation

Rationale: Simulates quantization errors, uniform sampling noise, temporal jitter during acquisition, and generic non-Gaussian disturbances. Represents a conservative noise model that does not favor any specific intensity.

2 Mathematical Background: Wavelet Scattering Transform

The Wavelet Scattering Transform (WST) represents a mathematically principled approach to multi-scale feature extraction that extends classical wavelet decompositions through cascaded convolutions with wavelet filters followed by modulus non-linearities. This architecture generates representations with provable stability and invariance properties, making it particularly suitable for texture analysis and pattern recognition under signal perturbations.

2.1 Theoretical Foundation

The construction begins with a family of complex-valued Morlet wavelets, defined as modulated Gaussian functions:

$$\psi(t) = e^{-t^2/(2\sigma^2)} \cdot e^{i\omega_0 t}$$

where σ controls the spatial support and ω_0 determines the central frequency. These wavelets are then scaled and rotated to form a filter bank $\{\psi_\lambda\}_{\lambda\in\Lambda}$, where each filter is parameterized by:

$$\psi_\lambda(x) = 2^{-2j}\psi(2^{-j} r_\theta x)$$

with $\lambda = (j, \theta)$, where $j \in \{0, 1, \ldots, J\}$ denotes the scale (corresponding to frequencies 2^{-j}) and $\theta \in \{0, \pi/L, \ldots, (L-1)\pi/L\}$ represents the orientation angle. The parameter J determines the maximum scale (capturing structures up to size 2^J pixels), while L controls the angular resolution.

2.2 Cascading Scattering Architecture

The scattering transform is constructed recursively through iterated convolutions and complex modulus operations. For an input signal $x(u)$, the zero-th order coefficient captures the low-frequency average:

$$S_0 x = x \star \varphi_J$$

where $\varphi_J(u) = 2^{-2J}\varphi(2^{-J}u)$ is a low-pass averaging filter at scale J.

The first-order scattering coefficients are obtained by convolving with wavelets, applying the modulus non-linearity to ensure positivity and stability, and then averaging:

$$S_1 x(\lambda_1) = |x \star \psi_{\lambda_1}| \star \varphi_J$$

This operation decomposes the signal into directional components at different scales, where the modulus ensures translation invariance up to scale 2^J.

Higher-order coefficients capture residual variations by iterating this process on the modulus of the first-order coefficients:

$$S_2 x(\lambda_1, \lambda_2) = \big||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}\big| \star \varphi_J$$

The second-order coefficients are particularly important for capturing texture regularity and local orientation patterns that are not accessible from first-order statistics alone.

The complete scattering representation is the concatenation:

$$Sx = \{S_0 x, \{S_1 x(\lambda_1)\}_{\lambda_1}, \{S_2 x(\lambda_1, \lambda_2)\}_{\lambda_1, \lambda_2}\}$$

For RGB images, this process is applied independently to each channel, and both mean and standard deviation statistics are computed over spatial locations, resulting in approximately 486 features per image in our implementation (using $J = 3$ and $L = 8$).

2.3 Invariance and Stability Properties

The theoretical appeal of WST derives from its provable invariance and stability guarantees.

Translation Invariance: The averaging operator φ_J provides translation invariance up to scale 2^J. Formally, for any translation τ:

$$\|Sx(\cdot - \tau) - Sx(\cdot)\| \leq C \cdot 2^{-J} \|\nabla x\|$$

where C is a constant and ∇x denotes the spatial gradient. This ensures that small spatial shifts do not significantly alter the feature representation.

Deformation Stability: More generally, the scattering transform is Lipschitz-continuous with respect to diffeomorphic deformations. For a smooth deformation field $\tau(u)$ and the deformed signal $x_\tau(u) = x(u - \tau(u))$:

$$\|Sx_\tau - Sx\| \leq C\|x\| \sup_u \|\nabla \tau(u)\|$$

This property ensures that moderate geometric distortions—such as those induced by viewpoint changes, object articulation, or sensor motion—produce bounded variations in the feature space.

Noise Stability: Additive noise perturbations are also controlled. For $x_{\text{noisy}} = x + \epsilon$:

$$\|S(x + \epsilon) - Sx\| \leq \|\epsilon\|$$

The non-expansive nature of the scattering operator ensures that noise amplification does not occur across the cascaded layers, unlike in some learned representations where instability can arise.

2.4 Comparison with Convolutional Neural Networks

While CNNs and scattering networks share a similar architectural motif—alternating convolutions and non-linearities—they differ fundamentally in their construction and interpretation:

- **Filter Learning:** CNN filters are learned from data through back-propagation, whereas WST filters are predefined Morlet wavelets with fixed geometric structure. This makes WST parameter-free but less adaptable to specific domains.

- **Theoretical Guarantees:** WST provides formal stability bounds, whereas CNN stability is typically empirical and task-dependent.

- **Data Requirements:** WST does not require training data for filter design, making it suitable for scenarios with limited labeled samples—a key advantage demonstrated in our data scarcity experiments.
- **Interpretability:** Each WST coefficient corresponds to a specific scale and orientation, facilitating physical interpretation. CNN activations, while powerful, are often less interpretable.

2.5 Implementation Parameters in Kymatio

Our implementation uses the Kymatio library, which provides GPU-accelerated scattering transforms. The key hyperparameters are:

- J **(maximum scale):** Set to 3, corresponding to maximum structure size $2^J = 8$ pixels. Larger J captures coarser patterns but increases computational cost.
- Q **(quality factor):** Number of wavelets per octave, set to 1. Higher Q provides finer frequency resolution.
- L **(number of orientations):** Set to 8, uniformly distributed over $[0, \pi)$. Captures directional texture information.
- **Order:** Maximum scattering order set to 2. Third-order coefficients are typically negligible for natural images.

The total number of scattering coefficients scales as $O(J \cdot L + J^2 \cdot L^2)$, which for our parameters yields approximately 81 coefficients per channel. Computing both mean and standard deviation over spatial positions doubles this count, resulting in ~ 162 features per channel, or ~ 486 features for RGB.

3 Statistical Testing Framework: Detailed Methodology

The comparative evaluation of feature extraction methods in this study relies on a rigorous statistical framework that combines normality assessment, non-parametric hypothesis testing, multiple comparison correction, and effect size quantification. This section provides complete methodological details and worked examples using data from our experiments.

3.1 Normality Assessment: Shapiro-Wilk Test

The Shapiro-Wilk test evaluates whether a sample of observations is drawn from a normal distribution. The test statistic W is defined as:

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

where $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$ are the ordered sample values (order statistics), $\bar{x}$ is the sample mean, and $\{a_i\}_{i=1}^{n}$ are weights derived from the expected values, variances, and covariances of the order statistics of independent standard normal random variables.

The statistic W lies in the interval $[0, 1]$, with values close to 1 indicating normality. The null hypothesis H_0 : data $\sim \mathcal{N}(\mu, \sigma^2)$ is rejected for small values of W. Critical values and p-values are obtained via Monte Carlo simulation or tabulated distributions.

Application to Our Data: For the pairwise deltas $\Delta_i = \text{Macro-F1}_{\text{WST}}(i) - \text{Macro-F1}_{\text{AdvStats}}(i)$ computed across 168 configurations:

- Sample size: $n = 168$
- Computed $W = 0.9826$
- Corresponding p-value: 0.0213

Since $p = 0.0213 < 0.05$, we reject the null hypothesis of normality at the $\alpha = 0.05$ level. The distribution exhibits slight skewness and heavier tails than a Gaussian, likely due to heterogeneity in noise effects across different experimental conditions. This violation of normality necessitates the use of non-parametric tests for subsequent inference.

Similarly, for the Hybrid vs. AdvStats comparison:

- $n = 168$, $W = 0.9786$, $p = 0.0087 < 0.05 \Rightarrow$ reject normality

3.2 Wilcoxon Signed-Rank Test: Step-by-Step Procedure

The Wilcoxon signed-rank test is a non-parametric alternative to the paired t-test. It tests whether the median of paired differences differs significantly from zero, without assuming normality.

Algorithm:

1. **Compute differences:** For each configuration i, calculate $\Delta_i = \text{Method}_i - \text{Baseline}_i$
2. **Remove zeros:** Exclude any $\Delta_i = 0$ (reduce sample size to n')
3. **Rank absolute values:** Assign ranks $1, 2, \ldots, n'$ to $|\Delta_1|, |\Delta_2|, \ldots, |\Delta_{n'}|$ in ascending order. Ties receive average ranks.
4. **Attach signs:** Assign each rank the sign of the original Δ_i
5. **Compute rank sums:**

$$R^+ = \sum_{\{\Delta_i > 0\}} \text{rank}(|\Delta_i|), \quad R^- = \sum_{\{\Delta_i < 0\}} \text{rank}(|\Delta_i|)$$

6. **Test statistic:** $W = \min(R^+, R^-)$
7. p**-value:** Under H_0 (median$(\Delta) = 0$), W follows a known distribution. For large n ($n \geq 10$), a normal approximation is used:

$$Z = \frac{W - \mu_W}{\sigma_W}, \quad \mu_W = \frac{n'(n'+1)}{4}, \quad \sigma_W = \sqrt{\frac{n'(n'+1)(2n'+1)}{24}}$$

Illustrative Example (First 10 Deltas): For this subset: $W = \min(40, 15) = 15$. With $n' = 10$, expected $\mu_W = 10 \cdot 11/4 = 27.5$, $\sigma_W \approx 9.8$, yielding $Z \approx -1.27$ (not significant at $\alpha = 0.05$).

Full Sample Results: For the complete WST vs. AdvStats comparison ($n = 168$):

- $R^+ = 6543$, $R^- = 7629$
- $W = 6543$
- Expected $\mu_W = 168 \cdot 169/4 = 7098$
- $\sigma_W \approx 746.8$
- $Z = (6543 - 7098)/746.8 \approx -0.74$

Table 1: Wilcoxon procedure illustration (subset of data)

i	Δ_i	$\lvert\Delta_i\rvert$	Rank	Signed Rank
1	-0.016	0.016	3	-3
2	-0.003	0.003	1	-1
3	+0.040	0.040	7	+7
4	+0.018	0.018	4	+4
5	+0.078	0.078	10	+10
6	+0.038	0.038	6	+6
7	-0.006	0.006	2	-2
8	+0.033	0.033	5	+5
9	+0.040	0.040	8	+8
10	-0.027	0.027	9	-9
$R^+ = 7 + 4 + 10 + 6 + 5 + 8 = 40$			$R^- = 3 + 1 + 2 + 9 = 15$	

- Two-tailed p-value: 0.0825

Since $p = 0.0825 > 0.05$, we fail to reject H_0, concluding that the median difference is not statistically significant.

For Hybrid vs. AdvStats:

- $W = 5921$, $p = 0.0195 < 0.05 \Rightarrow$ reject H_0, significant improvement

Table 2: Wilcoxon signed-rank vs. paired t-test

Property	Wilcoxon	Paired t-test
Distributional assumption	None (rank-based)	Normality of differences
Test focus	Median difference	Mean difference
Robustness to outliers	High	Low
Statistical power (normal data)	$\sim 95\%$ of t-test	100% (optimal)
Applicability to our data	Valid (non-normal)	Invalid

Comparison with Paired t-Test: Given the rejection of normality (Shapiro-Wilk $p < 0.05$), the Wilcoxon test is the appropriate choice.

3.3 Multiple Comparison Correction: Benjamini-Hochberg FDR

When conducting k simultaneous hypothesis tests, the probability of making at least one Type I error (false positive) increases. The Family-Wise Error Rate is:

$$\text{FWER} = 1 - (1 - \alpha)^k$$

For our $k = 2$ comparisons at $\alpha = 0.05$: FWER $= 1 - 0.95^2 \approx 0.0975$ (9.75% chance of at least one false rejection).

The Bonferroni correction controls FWER by testing each hypothesis at level α/k, but is conservative and reduces statistical power. The Benjamini-Hochberg procedure instead controls the False Discovery Rate (FDR)—the expected proportion of false positives among all rejections—offering greater power.

Benjamini-Hochberg Procedure:

1. Order the k raw p-values: $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(k)}$
2. For each $i = 1, \ldots, k$, compute adjusted p-value:
$$p_{\text{FDR}}(i) = \min\left(p_{(i)} \times \frac{k}{i}, 1\right)$$
3. Find the largest i^* such that $p_{\text{FDR}}(i^*) \leq \alpha$
4. Reject H_0 for all $j \leq i^*$

Table 3: FDR correction: worked example

Comparison	Raw p	Rank	k/i	p_{FDR}	Decision ($\alpha = 0.05$)
Hybrid vs. Adv	0.0195	1	$2/1 = 2.0$	$0.0195 \times 2 = 0.0390$	Reject H_0
WST vs. Adv	0.0825	2	$2/2 = 1.0$	$0.0825 \times 1 = 0.0825$	Fail to reject

Application to Our Data ($k = 2$): The Hybrid method shows a statistically significant improvement ($p_{\text{FDR}} = 0.0390 < 0.05$), while WST does not ($p_{\text{FDR}} = 0.0825 > 0.05$).

Table 4: Bonferroni vs. Benjamini-Hochberg

Method	Controls	Threshold	Power	Our Result
Bonferroni	FWER	$\alpha/k = 0.025$	Lower	Hybrid: $0.0195 < 0.025$ (✓) WST: $0.0825 > 0.025$ ($\times$)
Benjamini-Hochberg	FDR	Adaptive	Higher	Hybrid: $0.0390 < 0.05$ (✓) WST: $0.0825 > 0.05$ ($\times$)

Comparison with Bonferroni: Both methods agree on the conclusions in this case, but FDR generally provides greater power when k is large.

3.4 Effect Size: Cohen's d Interpretation

Statistical significance ($p < 0.05$) indicates that an observed difference is unlikely under the null hypothesis, but does not quantify the magnitude of the effect. Cohen's d provides a standardized measure of effect size:

$$d = \frac{\mu(\Delta)}{\sigma(\Delta)}$$

where $\mu(\Delta)$ is the mean difference and $\sigma(\Delta)$ is the standard deviation of the differences (computed with `ddof=0` for population estimate).

Table 5: Cohen's d interpretation guide with examples from our study

Range	Label	Interpretation	Our Data
$\lvert d\rvert < 0.2$	Small / Negligible	Minimal practical impact	WST: $d = -0.075$
$0.2 \leq \lvert d\rvert < 0.5$	Medium	Noticeable effect	Hybrid: $d = 0.156$
$0.5 \leq \lvert d\rvert < 0.8$	Medium-Large	Substantial effect	–
$\lvert d\rvert \geq 0.8$	Large	Major impact	–

Interpretation Thresholds:

3.5 Computational Implementation

The experiments were carried out under a controlled computational setup to ensure reproducibility, scalability, and efficient execution. Table 6 summa-

Table 6: Hardware and software configuration

Component	**Details**
CPU	Multi-core processor for Random Forest parallelization
GPU	NVIDIA CUDA-compatible (optional, used for WST acceleration)
RAM	16 GB (sufficient for full in-memory dataset processing)
Operating System	Linux (Ubuntu / WSL2)
Language	Python 3.12
Virtual Environment	`venv` (dependency isolation)

Table 7: Parallelization strategy

Task	**Implementation**
Random Forest	`n_jobs = -1` (uses all available CPU cores)
Batch execution	Sequential to ensure I/O stability
Cross-validation	Internally parallelized by scikit-learn

rizes the hardware and software environment adopted for running the full experimental pipeline, while Table 7 outlines the strategies used to manage computation and parallel execution.

Practical Interpretation for Our Study:

- **WST vs. AdvStats:** $d = -0.075$ (small negative). This corresponds to a mean difference of -0.0106 with $\sigma = 0.1410$, indicating that WST performs negligibly worse than AdvStats on average. The effect is not statistically significant and is practically irrelevant.

- **Hybrid vs. AdvStats:** $d = 0.156$ (small positive). With mean $+0.0171$ and $\sigma = 0.1092$, this represents a shift of approximately one-sixth of a standard deviation. While statistically significant ($p_{\text{FDR}} = 0.039$), the effect is modest in absolute terms (+1.71 percentage points in Macro-F1). Whether this justifies the increased computational cost depends on application requirements.

Contextualizing Effect Sizes: With $n = 168$ paired observations, even small effects can achieve statistical significance due to high statistical power. This underscores the importance of reporting effect sizes alongside p-values: our findings indicate that the Hybrid method offers a real but modest advantage, which may or may not be practically meaningful depending on the operational context (see Recommendation Matrix in Section S4).