Detecting Vegetation Encroachment Along Power Lines with Handcrafted Features and an RBF-SVM Classifier

Anish Kumar Pal
Department of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
25m1087@iitb.ac.in

***Abstract*—Vegetation encroachment around overhead power lines and substations is a leading cause of unplanned outages, equipment damage, and wildfire ignition. Manual right-of-way (ROW) inspection is slow, expensive, and error-prone, which motivates automated remote-sensing solutions. While deep convolutional networks dominate modern image classification, they require large annotated corpora and substantial computation, limiting their use on edge and embedded inspection platforms. This study presents an interpretable pipeline that uses a Support Vector Machine (SVM) with a radial basis function (RBF) kernel and manually created color and texture descriptors to classify satellite/aerial picture patches into high- and low-density vegetation. From each patch we extract per-channel colour moments (mean, standard deviation, skewness, kurtosis), Gray-Level Co-occurrence Matrix (GLCM) descriptors, and a Local Binary Pattern (LBP) texture summary. Evaluated on a balanced corpus of 12,351 image patches, the model attains a test accuracy of 98.2%, a macro-averaged F_1 score of 0.983, and an Receiver Operating Characteristic - Area Under the Curve (ROC–AUC) of 0.998. Grouped 5-fold cross-validation yields a mean macro-F_1 of 0.983, confirming that the performance is stable and not an artefact of a single split. Permutation feature-importance analysis shows that colour moments—particularly the red and green channel means—are the dominant discriminative cues. The proposed approach is accurate, reproducible, computationally inexpensive, and interpretable, making it well suited for real-time, resource-constrained infrastructure-monitoring deployments.**

***Index Terms*—Vegetation encroachment, support vector machine, GLCM, local binary pattern, colour moments, remote sensing, power line monitoring, texture classification.**

I. Introduction

VEGETATION encroachment, the intrusion of trees, shrubs and undergrowth into the right-of-way (ROW) of electrical infrastructure such as transmission lines, distribution feeders and substations, is a persistent threat to grid reliability and public safety. Contact between conductors and vegetation can lead to flash-overs, forced outages, equipment damage and catastrophic wildfires in dry climates [1]. Hence a large portion of the maintenance budget of utilities is allocated to vegetation management and ROW inspection.

Traditional inspection relies on field crews, manned helicopter patrols, or manual review of aerial imagery. These methods are labour-intensive, expensive, slow to cover large networks, and subject to human fatigue and inconsistency. The growing availability of high-resolution satellite and unmanned-aerial-vehicle (UAV) imagery [2] has created an opportunity to automate encroachment screening through computer vision.

For image classification, deep learning—specifically, convolutional neural networks (CNNs) [6], [7]—has taken the lead. However, CNNs typically require large, carefully labelled datasets and significant computational resources for both training and inference. In many utility settings—where labelled data is scarce, models must run on drones or edge devices, and the reasoning behind an alert must be auditable—these requirements are impractical. Classical machine learning with carefully engineered features offers an attractive alternative: it is data-efficient, fast, and interpretable.

In this work we revisit Support Vector Machines (SVMs) for vegetation encroachment detection. We frame the problem as a binary classification of image patches into *high-density* and *low-density* vegetation, which can be used as a screening layer over a transmission corridor to flag high-risk segments. Our contributions are:

- A compact, interpretable feature representation combining per-channel colour moments, GLCM texture descriptors, and an LBP texture summary.
- An RBF-kernel SVM classifier trained and validated on a balanced corpus of 12,351 image patches.
- A rigorous evaluation including a held-out test set, grouped 5-fold cross-validation, ROC/PR analysis, and permutation feature-importance analysis.
- A fully reproducible, open-source implementation (https://github.com/anishpal2235/vegetation_encroachment_detection).

This is how the rest of the paper is organized. Previous research on vegetation monitoring using remote sensing is reviewed in Section II. The dataset and its class composition are described in Section III. The RBF-SVM classifier and the feature-extraction pipeline are displayed in Section IV. The experimental setup is described in Section V. VI consists of the results and their discussion. Lastly, Section VII wraps up and talks about potential future directions.

II. Related Work

Remote-sensing-based vegetation monitoring spans a wide range of techniques, from spectral indices such as the Normalised Difference Vegetation Index (NDVI) [4], [5] to LiDAR-based canopy modelling [3] and, more recently, deep learning [6].

Spectral-index methods are simple and effective when multispectral or hyperspectral bands are available, but high-resolution RGB imagery from platforms such as Google Earth Pro or consumer UAVs lacks dedicated near-infrared bands, limiting the direct use of NDVI. LiDAR provides precise three-dimensional canopy geometry and clearance measurements [3] but is costly to acquire and process at network scale, and recent CNN-based corridor-inspection methods [2] require large labelled datasets.

Classical machine-learning classifiers built on handcrafted features remain competitive for texture-rich problems. The Gray-Level Co-occurrence Matrix (GLCM) [10] and Local Binary Patterns (LBP) [11] are long-standing, robust texture descriptors, while colour moments [9] capture the chromatic signature of vegetation. Support Vector Machines [12] are particularly well suited to such moderate-dimensional feature spaces because they generalise well from limited data and offer strong margins via the kernel trick. Notably, Haroun *et al.* [8] reported that an SVM combined with statistical colour and texture features extracted from satellite imagery achieved 98.27% recall in detecting vegetation encroachment along transmission corridors, demonstrating the viability of the classical pipeline for industrial-scale monitoring. The work follows this direction by offering a fully reproducible implementation, an extended evaluation including cross-validation and feature-importance analysis.

III. Dataset

We use a publicly available vegetation-density image dataset hosted on Zenodo [25], complemented by high-resolution satellite imagery captured from Google Earth Pro over high-tension power-line corridors. The imagery is partitioned into square patches and labelled into two classes: *low-density* vegetation (label 0) and *high-density* vegetation (label 1). High-resolution scenes were exported at up to 4800×3120 pixels with overlay layers (place labels, borders, roads) disabled to reduce non-vegetation distractors before patch extraction.

Fig. 1 shows representative patches from each class. High-density patches exhibit darker, greener, more texturally complex canopy signatures, whereas low-density patches are lighter and contain bare soil, grass, or built surfaces.

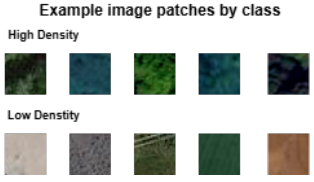


Fig. 1. Example image patches for the two classes. Top row: high-density vegetation; bottom row: low-density vegetation.

The corpus is essentially balanced, comprising 6,177 low-density and 6,174 high-density patches for a total of 12,351 samples (Fig. 2). The near-perfect class balance means that accuracy is a meaningful headline metric and that no class re-weighting or resampling is required.

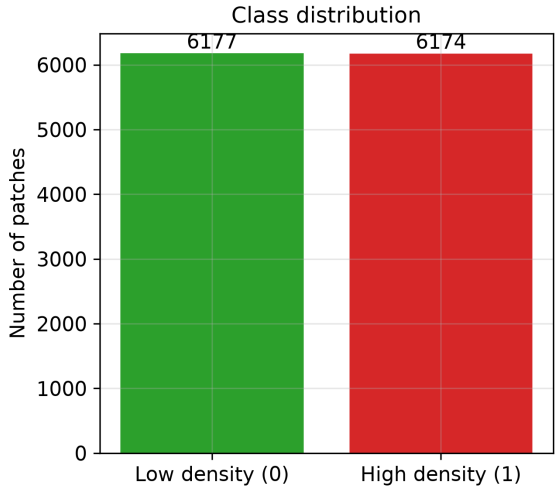


Fig. 2. Class distribution of the dataset: 6,177 low-density and 6,174 high-density patches.

IV. Methodology

The pipeline (Fig. 3), summarised in Algorithm 1, consists of four stages: (i) feature extraction, (ii) feature standardisation, (iii) SVM training, and (iv) evaluation and deployment. Each is described below.

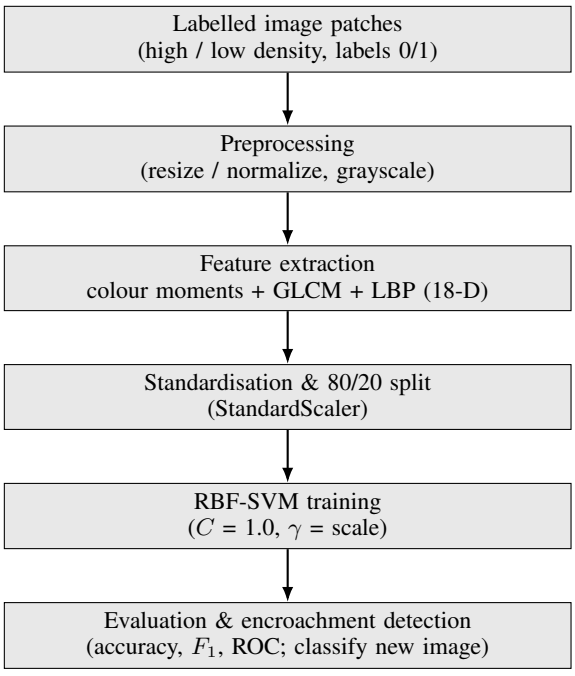


Fig. 3. Overview of the vegetation-encroachment detection pipeline, from labelled image patches through feature extraction and standardisation to RBF-SVM classification and deployment.

Algorithm 1 Vegetation Encroachment Detection Pipeline

Require: Labelled patches $\{I_n\}_{n=1}^{N}$ with labels $y_n \in \{0, 1\}$
Ensure: Trained RBF-SVM and test-set predictions
1: **for** each patch I_n **do**
2: Compute per-channel colour moments $\mu_c, \sigma_c, \gamma_c, \kappa_c$ {12 features}
3: Convert I_n to grayscale; compute GLCM $p(i, j)$ at offset 1, 0°
4: Derive Haralick descriptors: contrast, dissimilarity, homogeneity, energy, correlation {5 features}
5: Compute LBP ($P = 8$, $R = 1$) and take its aggregate {1 feature}
6: Assemble $\mathbf{f}_n \in \mathbb{R}^{18}$
7: **end for**
8: Split data $80/20$ into train/test (`random_state`$= 42$)
9: Fit StandardScaler on the training split; standardise all $\mathbf{f}_n$
10: Train RBF-SVM ($C = 1.0$, $\gamma =$ scale) on standardised training features
11: Evaluate on the held-out test set: accuracy, macro-F_1, ROC–AUC
12: **return** trained classifier and predictions

A. Feature Extraction

For each RGB patch I we compute an 18-dimensional feature vector composed of colour moments, GLCM texture descriptors, and an LBP summary.

1) Colour moments: Following the colour-moment representation of Stricker and Orengo [9], for each colour channel $c \in \{R, G, B\}$ with pixel values $x_1, \ldots, x_N$ we compute the first four statistical moments:

$$\mu_c = \frac{1}{N} \sum_{i=1}^{N} x_i, \tag{1}$$

$$\sigma_c = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu_c)^2}, \tag{2}$$

$$\gamma_c = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{x_i - \mu_c}{\sigma_c} \right)^3, \tag{3}$$

$$\kappa_c = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{x_i - \mu_c}{\sigma_c} \right)^4. \tag{4}$$

These capture brightness (μ), spread (σ), asymmetry (γ), and peakedness (κ) of the per-channel intensity distribution, yielding $3 \times 4 = 12$ colour features.

2) GLCM texture descriptors: The grayscale patch is quantised to 256 levels and a Gray-Level Co-occurrence Matrix [10] is computed for an offset of one pixel at 0°, symmetrised and normalised to probabilities $p(i, j)$. From it we derive five Haralick descriptors:

$$\text{Contrast} = \sum_{i,j} (i - j)^2 \, p(i, j), \tag{5}$$

$$\text{Dissimilarity} = \sum_{i,j} |i - j| \, p(i, j), \tag{6}$$

$$\text{Homogeneity} = \sum_{i,j} \frac{p(i, j)}{1 + (i - j)^2}, \tag{7}$$

$$\text{Energy} = \sqrt{\sum_{i,j} p(i, j)^2}, \tag{8}$$

$$\text{Correlation} = \sum_{i,j} \frac{(i - \mu_i)(j - \mu_j) \, p(i, j)}{\sigma_i \, \sigma_j}. \tag{9}$$

These quantify local intensity variation, texture coarseness, and spatial correlation, adding five texture features.

3) Local Binary Pattern: Finally, a Local Binary Pattern [11] with $P = 8$ neighbours at radius $R = 1$ encodes micro-texture by thresholding each pixel's neighbourhood. We summarise the LBP response by its aggregate (sum) to obtain one additional texture feature. In total each patch is represented by an 18-dimensional vector $\mathbf{f} = [\,12 \text{ colour}, 5 \text{ GLCM}, 1 \text{ LBP}\,]$.

B. Feature Standardisation

Because SVMs with RBF kernels are sensitive to feature scale [13], every feature is standardised to zero mean and unit variance using statistics estimated on the training set only:

$$\tilde{f}_k = \frac{f_k - \mu_k^{\text{train}}}{\sigma_k^{\text{train}}}. \tag{10}$$

To avoid information leakage, the same transformation is applied to the test data.

C. SVM Classifier

We train a Support Vector Machine [12] with an RBF kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\!\left(-\gamma \, \|\mathbf{x}_i - \mathbf{x}_j\|^2\right), \tag{11}$$

which maps the feature space into a higher-dimensional space where the two vegetation classes become linearly separable. The model solves the soft-margin primal problem

$$\min_{\mathbf{w}, b, \xi} \; \tfrac{1}{2} \|\mathbf{w}\|^2 + C \sum_i \xi_i \quad \text{s.t.} \quad y_i(\mathbf{w}^\top \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \tag{12}$$

with $\xi_i \geq 0$. We use the regularisation parameter $C = 1.0$ and the heuristic $\gamma =$ scale, following common practice for RBF-SVM tuning [13]. The classifier is implemented in scikit-learn [21], which wraps the LIBSVM solver [14].

V. Experimental Setup

The full pipeline is implemented in Python using OpenCV [24], scikit-image [22], SciPy [23], and scikit-learn [21]. Features are extracted for all 12,351 patches and the data is split $80/20$ into training (9,880 patches) and testing (2,471 patches) sets with a fixed random seed (random_state $=$ 42) for reproducibility. Standardisation parameters are fit on

the training set only. The RBF-SVM is trained with $C = 1.0$ and $\gamma = \text{scale}$.

We report accuracy, per-class precision, recall, and F_1 score, the confusion matrix, and the macro-averaged F_1. To assess robustness we also perform *grouped* 5-fold cross-validation [20], where patches originating from the same source scene are kept within a single fold to prevent optimistic bias from spatially adjacent, highly correlated patches. ROC and precision–recall curves are computed from the SVM decision scores, and permutation feature importance [15] is used to quantify each feature's contribution to the macro-F_1.

VI. Results and Discussion

A. Overall Performance

On the held-out test set the RBF-SVM achieves an accuracy of 98.2%. The confusion matrix (Fig. 4) shows $1{,}213$ correctly classified low-density patches with only 23 false positives, and $1{,}213$ correctly classified high-density patches with only 22 false negatives. The errors are balanced across classes (row-normalised values of 0.98 on both diagonals), indicating no systematic bias toward either class.

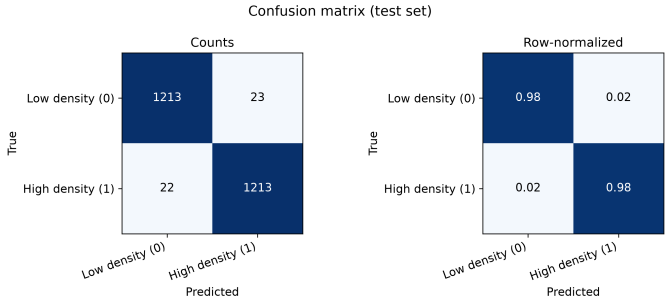


Fig. 4. Confusion matrix on the test set (counts, left; row-normalised, right). Misclassifications are rare and symmetric across classes.

The precision, recall, and F_1 scores per class are all 0.98 (Fig. 5), and Table I summarises the headline results. The macro-averaged F_1 of 0.98 confirms uniformly strong performance on both classes.

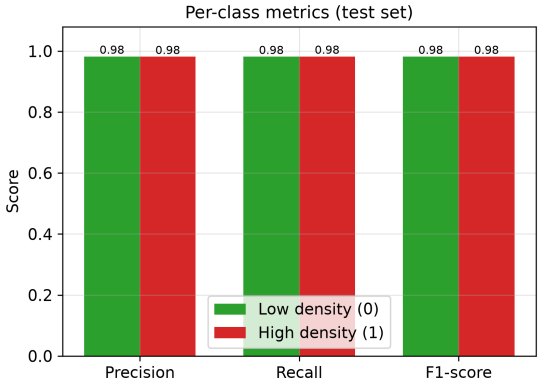


Fig. 5. Precision, recall, and F_1 per class on the test set, uniformly at 0.98.

TABLE I
Summary of Test-Set Performance

Metric	Value
Accuracy	0.982
Macro precision	0.98
Macro recall	0.98
Macro F_1	0.98
ROC–AUC	0.998
Average precision (PR–AUC)	0.998
5-fold CV mean macro-F_1	0.983

B. Discrimination Quality

The ROC curve (Fig. 6, left) yields an area under the curve of 0.998, and the precision–recall curve (right) gives an average precision of 0.998. Both indicate that the decision scores separate the two classes almost perfectly across the full range of operating thresholds, allowing an operator to tune the recall/precision trade-off without sacrificing much performance—useful when the cost of a missed encroachment differs from the cost of a false alarm.

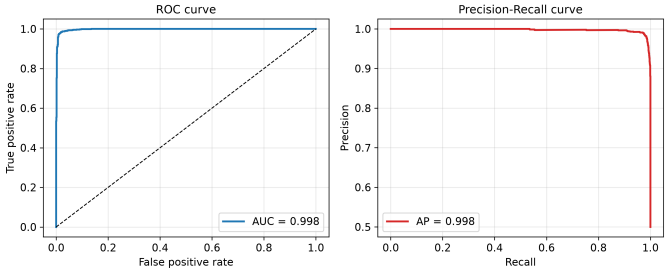


Fig. 6. Threshold-independent performance: the ROC curve (AUC $= 0.998$, left) and precision–recall curve (AP $= 0.998$, right) confirm strong, balanced discrimination between the two classes.

C. Cross-Validation Stability

Grouped 5-fold cross-validation (Fig. 7) produces a mean macro-F_1 of 0.983 with negligible variance across folds. This demonstrates that the model generalizes consistently across various corpus subsets and that the reported accuracy is not a result of a favorable train/test split.

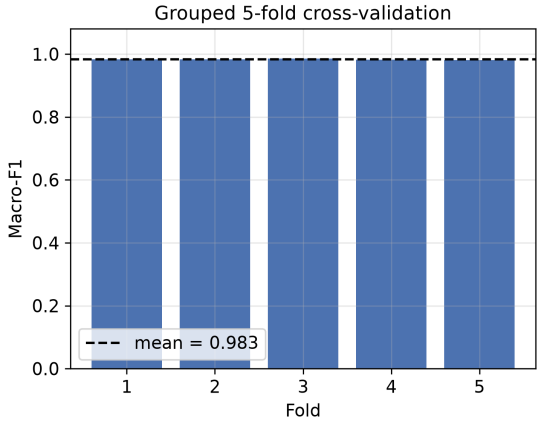


Fig. 7. Grouped 5-fold cross-validation macro-F_1 scores (mean $= 0.983$).

D. Feature Space and Importance

A two-dimensional PCA [16] projection of the standardised feature space (Fig. 8) shows the two classes forming largely separable clusters, with the first two principal components explaining 34.6% and 17.3% of the variance (cumulatively 51.9%). Since the remaining variance lies in higher dimensions, this 2-D view is used only for visualisation, while the SVM is trained on the full 18-D feature vector. The partial overlap corresponds to transitional patches—sparse or mixed vegetation and edge regions—and explains the small residual error. This confirms that the handcrafted features carry strong but not trivially linear discriminative information, justifying the non-linear RBF kernel.

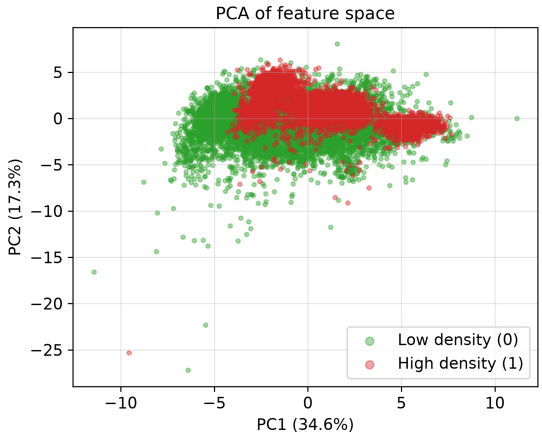


Fig. 8. PCA projection of the feature space. The classes form largely separable clusters with some boundary overlap.

Permutation feature importance [15] (Fig. 9) reveals that colour moments dominate: the red-channel mean (`R_mean`) is by far the most important feature, followed by the green-channel mean (`G_mean`) and green-channel standard deviation (`G_std`). GLCM correlation and homogeneity contribute moderately, while several LBP bins contribute little. This matches domain intuition: dense vegetation is darker and greener, so chromatic statistics are the strongest cues, with texture descriptors refining the boundary. The result also suggests that the feature set could be pruned for even faster inference with minimal accuracy loss.

We use permutation importance [17], which is model-agnostic: each feature in the held-out set is shuffled and the drop in macro-F_1 is measured, so a large drop marks a feature the model truly relies on [18]. The dominance of colour features fits remote-sensing intuition—dense canopy absorbs red and reflects green, so green-based RGB indices are classic proxies for vegetation cover [4], [19]—and shows that this cue is recoverable from plain RGB, without a near-infrared band. The fact that only a few colour moments matter also implies redundancy in the 18-D descriptor, so feature selection [18] could shrink it for faster, lighter inference with little accuracy loss.

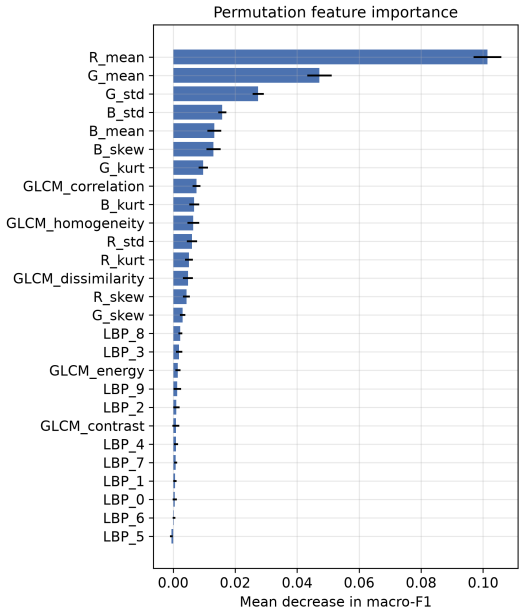


Fig. 9. Permutation feature importance (mean decrease in macro-F_1). Colour-mean features dominate the decision.

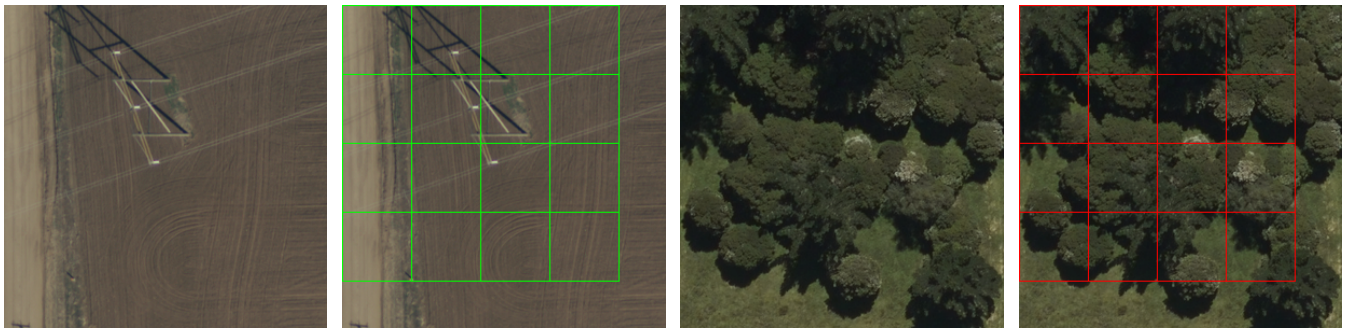
E. Qualitative Results: Inputs and Predicted Outputs

To illustrate behaviour at deployment, the trained classifier is applied as a screening layer: an input scene is tiled into fixed-size patches, every patch is standardised with the stored training statistics and classified, and the patch grid is rendered back over the image and colour-coded by predicted class (green = low-density, red = high-density vegetation flagged as potential encroachment).

Fig. 10 pairs four representative scenes, each shown as the raw RGB input (a) and the corresponding classifier output (b). The three low-density examples—cropland, a lattice transmission tower over grass, and a tower with conductors over bare soil—are all correctly assigned to the low-density class (green grid). Importantly, the man-made tower and conductors do *not* trigger false high-density alarms, because the colour and texture statistics of each patch remain dominated by the surrounding open ground. By contrast, the dense forest scene is correctly flagged as high-density: every grid cell turns red as the closed canopy fills the field of view. This binary behaviour is exactly what a corridor-screening layer requires—open ROW and infrastructure are passed, while encroaching canopy is highlighted.

Fig. 11 extends the same screening procedure to two complete corridor scenes. Open, mown, or cleared ground and built surfaces (roads, car parks, sports fields) are consistently passed as low-density, whereas tree lines, woodland belts, and dense canopy bordering the right-of-way are flagged as high-density. The predicted boundary tracks the visible forest edge closely, yielding a compact, interpretable map of high-risk segments that an operator can prioritise for inspection rather than reviewing the entire corridor manually. Together, these qualitative results confirm that the patch-level accuracy reported above translates into coherent scene-level encroachment maps.

(a) cropland (input) (b) cropland (output) (a) tower/grass (input) (b) tower/grass (output)



(a) tower/soil (input) (b) tower/soil (output) (a) forest (input) (b) forest (output)

Fig. 10. Example inputs and predicted outputs. For each scene, (a) is the raw RGB input patch and (b) is the classifier output, where the patch grid is coloured by predicted class (green = low-density, red = high-density vegetation flagged as potential encroachment). The three low-density scenes—cropland, a transmission tower over grass, and a transmission structure over bare soil—are correctly passed (green), with towers and conductors causing no false alarms, while the dense forest scene is correctly flagged (red).

(a)

(b)

Fig. 11. Full-scene screening showing encroachment risk (green = low-density, red = high-density canopy) across (a) grassland–woodland and (b) mixed recreational scenes. Imagery: Google Earth Pro [26] © Maxar Technologies.

F. Discussion

The results demonstrate that a carefully engineered classical pipeline can match the accuracy typically attributed to deep models on this task, while offering three practical advantages. First, *efficiency*: feature extraction and SVM inference are inexpensive and run comfortably on CPUs, UAVs, and edge devices. Second, *data efficiency*: strong performance is obtained without the massive labelled corpora that CNNs demand. Third, *interpretability*: feature-importance analysis provides a transparent, auditable explanation for each decision—an important property for safety- and compliance-critical utility operations. The main limitations are the reliance on RGB-only imagery (no near-infrared band) and patch-level (rather than pixel-level) labelling, both of which are addressed in future work.

VII. Conclusion and Future Work

This work has presented an accurate, efficient, and interpretable pipeline for detecting vegetation encroachment along power transmission corridors. Combining per-channel colour moments, GLCM texture descriptors, and an LBP summary with an RBF-kernel SVM, the model achieves 98.2% accuracy, a macro-F_1 of 0.983, and an ROC–AUC of 0.998 on a balanced corpus of $12{,}351$ patches, with a 5-fold cross-validated mean macro-F_1 of 0.983. Permutation analysis identifies colour-mean features as the principal discriminative cues.

The approach is well suited to real-time, resource-constrained infrastructure-monitoring deployments.

Future work will (i) integrate the detector with live drone and satellite feeds for continuous corridor monitoring; (ii) explore semi-supervised labelling to reduce annotation cost; (iii) incorporate multispectral bands (e.g., NIR/NDVI) where available; (iv) extend from patch-level classification to pixel-level segmentation for precise clearance estimation; and (v) optimise and quantise the pipeline for embedded edge devices with real-time visual alerting.

Reproducibility

The complete source code, including the feature-extraction routine, training script, and evaluation utilities, is available at https://github.com/anishpal2235/vegetation_encroachment_detection. The dataset is available at https://zenodo.org/records/7800234.

References

[1] J. W. Mitchell, "Power line failures and catastrophic wildfires under extreme weather conditions," *Eng. Fail. Anal.*, vol. 35, pp. 726–735, 2013.

[2] V. N. Nguyen, R. Jenssen, and D. Roverso, "Automatic autonomous vision-based power line inspection: A review of current status and the potential role of deep learning," *Int. J. Electr. Power Energy Syst.*, vol. 99, pp. 107–120, 2018.

[3] L. Matikainen *et al.*, "Remote sensing methods for power line corridor surveys," *ISPRS J. Photogramm. Remote Sens.*, vol. 119, pp. 10–31, 2016.

[4] J. W. Rouse, R. H. Haas, J. A. Schell, and D. W. Deering, "Monitoring vegetation systems in the Great Plains with ERTS," in *Proc. 3rd ERTS Symp.*, 1974, pp. 309–317.

[5] N. Pettorelli *et al.*, "Using the satellite-derived NDVI to assess ecological responses to environmental change," *Trends Ecol. Evol.*, vol. 20, no. 9, pp. 503–510, 2005.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NeurIPS*, 2012, pp. 1097–1105.

[8] F. M. E. Haroun, S. N. M. Deros, and N. Md. Din, "Detection of vegetation encroachment in power transmission line corridor from satellite imagery using support vector machine: A features analysis approach," *Energies*, vol. 14, no. 12, p. 3393, 2021.

[9] M. Stricker and M. Orengo, "Similarity of color images," in *Proc. SPIE Storage and Retrieval for Image and Video Databases III*, vol. 2420, 1995, pp. 381–392.

[10] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, pp. 610–621, 1973.

[11] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002.

[12] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[13] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," Dept. Comput. Sci., Nat. Taiwan Univ., Tech. Rep., 2003.

[14] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

[15] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[16] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed. New York, NY, USA: Springer, 2002.

[17] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: A corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010.

[18] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[19] D. M. Woebbecke, G. E. Meyer, K. Von Bargen, and D. A. Mortensen, "Color indices for weed identification under various soil, residue, and lighting conditions," *Trans. ASAE*, vol. 38, no. 1, pp. 259–269, 1995.

[20] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. IJCAI*, 1995, pp. 1137–1143.

[21] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[22] S. van der Walt *et al.*, "scikit-image: Image processing in Python," *PeerJ*, vol. 2, p. e453, 2014.

[23] P. Virtanen *et al.*, "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[24] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.

[25] "Vegetation density image dataset," Zenodo, 2023. [Online]. Available: https://zenodo.org/records/7800234

[26] Google Earth Pro, version 7.3, "Satellite imagery of power transmission corridors, various locations across Australia," Maxar Technologies, accessed Jun. 2026. [Online]. Available: https://earth.google.com