International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI-Based Satellite Image Change Detection for Environmental Monitoring and Disaster Assessment Using Deep Learning and Explainable Artificial Intelligence

Yogeshchandra Puranik (Assistant Professor), **Akshay Patil, Shubham Karmilkar, Aditi Kamble and Prital Patil**
Department of Master of Computer Applications, Progressive Education Society's Modern College of Engineering, Pune, Maharashtra, India
Assistant Professor, Department of Master of Computer Applications,
Progressive Education Society's Modern College of Engineering, Pune, Maharashtra, India

Abstract— Rapid building-damage assessment from satellite imagery supports disaster response, yet manual survey is slow and many change-detection benchmarks provide only binary changed/unchanged labels. This paper presents an end-to-end pipeline for pixel-level, five-class building-damage segmentation from pre- and post-disaster image pairs in the xBD dataset. A Siamese U-Net with a shared ImageNet-pretrained ResNet34 encoder fuses post-event features with absolute feature differences at five scales and is trained with a class-weighted Focal plus Dice loss. Data are partitioned by disaster event: seven events for training, two for validation and one, Hurricane Michael, for testing. On the held-out event, the selected checkpoint reaches 87.6% pixel accuracy and a building-localisation F1 of 0.780, but damage grading remains weak: the macro F1 over the four damage classes is 0.286, with class F1 of 0.145, 0.150 and 0.235 for minor, major and destroyed damage. Training macro F1 rises to 0.595 while validation macro F1 stays below 0.179, which indicates poor transfer across events, and 59.4% of major-damage pixels are assigned to no damage. Grad-CAM overlays and a Django web interface are provided; neither has been quantitatively validated. This is a preliminary single-seed study on one test event, without baselines or ablations.

Keywords: change detection, building damage assessment, Siamese U-Net, class imbalance, Grad-CAM, xBD.

I. INTRODUCTION

Natural disasters cause large human and economic losses. The United Nations Office for Disaster Risk Reduction counts more than 7,300 major events between 2000 and 2019, affecting about 4.2 billion people at a cost of roughly USD 2.97 trillion [1]. After an event, responders must decide where to send rescue teams and which areas to prioritise. Field surveys are accurate but slow and risky, whereas very-high-resolution satellites can image affected areas within days [2].

Comparing images taken before and after an event can in principle reveal damaged buildings automatically. Deep learning dominates remote-sensing analysis [3], [4], and Siamese networks suit bi-temporal inputs because a shared encoder produces comparable features for both dates. Widely used change-detection benchmarks such as LEVIR-CD [5] and OSCD [6] supply only binary labels, so they do not teach a model to grade how severe the damage is. Safety-relevant use also needs predictions that people can inspect.

This paper documents a complete pipeline on xBD [2] and reports its first event-level held-out evaluation. Our contributions are modest and stated as such:

- a Siamese U-Net with a shared ResNet34 encoder and per-scale fusion of post-event features with absolute feature differences, for five-class pixel-level damage segmentation;
- a documented data pipeline (polygon rasterisation, 256×256 tiling, event-level partition) and an imbalance-aware training recipe (class-weighted Focal plus Dice loss);
- a held-out evaluation on an unseen disaster event with per-class metrics, a confusion-matrix analysis and an account of training dynamics, including the failure modes we observed; and
- Grad-CAM overlays and a Django interface for inference and reporting, which are descriptive components and are not evaluated here.

We do not claim state-of-the-art accuracy. Published xBD systems such as BDANet [7] and ChangeOS [8] use dedicated building-localisation stages and different evaluation protocols, so our numbers are not comparable with theirs (Sec. III-G).

A. Problem Statement and Objectives

Manual assessment of disaster damage from satellite imagery is time-consuming and hard to scale, and existing automated approaches often stop at binary change detection and do not grade structural damage. The research therefore targets bi-temporal analysis, pixel-level damage-severity classification, class-imbalance handling, interpretable predictions and a practical workflow. The project has the following objectives, each described with the purpose it serves:

1. **Bi-temporal damage segmentation.** Implement a Siamese U-Net with a shared ImageNet-pretrained

ResNet34 encoder that takes a pre-event and a post-event image and assigns every pixel to one of five classes (background, no damage, minor damage, major damage, destroyed), so that the severity of damage, and not only the presence of change, is predicted.

2. **Leakage-free data preparation.** Build a documented pipeline for the xBD dataset that rasterises building polygons, tiles scenes into 256×256 patches, discards uninformative patches and partitions the data by disaster event, so that evaluation is not inflated by scenes from the same event appearing in both training and test data.

3. **Class-imbalance handling.** Counter the dominance of background and undamaged pixels with a class-weighted Focal loss combined with Dice loss, so that the rare but operationally critical classes (major damage and destroyed) contribute meaningfully to training.

4. **Explainable predictions.** Provide Grad-CAM heat-maps that show which image regions influenced a predicted damage class, so that users can judge whether a prediction rests on plausible evidence.

5. **Usable workflow.** Integrate the model into a Django web application that stores results, shows class percentages, generates a narrative summary with Google Gemini (with a rule-based fallback) and exports a PDF report that non-experts can read.

6. **Event-level evaluation protocol.** Train on the training events, select the checkpoint on the validation events and evaluate it once on a held-out event, reporting per-class metrics and error structure, and specify baselines and ablations for the full-scale evaluation.

Objectives 1–5 are met at the implementation level. The held-out evaluation on one event (Sec. IV) shows that objectives 1 and 3 are only partly achieved: buildings are localised (F1 0.780) but damage grading is weak (macro F1 0.286). The baselines and ablations of objective 6 have not yet been run.

B. Research Gap

Five-class pixel-level damage grading is not new: it is established on xBD [7], [8]. This work therefore does not claim a new task; it addresses several gaps that remain around such models, each of which is tackled here only at the system level.

The first gap is explainability. To our knowledge, the xBD models cited above do not report visual explanations of their damage predictions, although this has not been surveyed systematically. We therefore add Grad-CAM overlays so that users can see which image regions influenced a predicted damage class; these explanations are qualitative only.

The second gap concerns evaluation. Random image-level or patch-level splits let highly related scenes from the same disaster appear in both training and testing, which inflates the reported scores. We assign every disaster event to exactly one partition and report results on a held-out event, but with a single test event and a single seed this evaluation measures transfer to one unseen hurricane, not to disasters in general.

The third gap is deployment and baselines. Research code often stops at reported metrics, so we provide an end-to-end web workflow whose latency and accuracy have not been benchmarked. In addition, controlled comparisons against plain Siamese and early-fusion U-Nets are specified in Sec. III-H but have not yet been performed, so the contribution of the individual design choices remains unmeasured.

II. LITERATURE REVIEW

A. Classical Change Detection

Early change-detection techniques compare co-registered images pixel by pixel. Image differencing subtracts the two images and thresholds the result, principal-component analysis (PCA) projects the temporal image stack onto components that concentrate change, and change vector analysis (CVA) describes each pixel by the magnitude and direction of its spectral change vector [9]. These methods are computationally light and need no training data, but they treat every pixel independently and are therefore sensitive to illumination and seasonal differences, sensor noise and small mis-registration, which produce many false changes. For disaster imagery, where buildings must be separated from changing vegetation, water and shadows, such handcrafted features are not sufficient.

B. Deep-Learning-Based Change Detection

Convolutional networks replaced handcrafted features with learned ones. Fully convolutional Siamese networks [10] introduced three U-Net-like designs: an early-fusion network that stacks both images as input, and two Siamese variants in which a shared encoder processes each image and the skip connections combine the two feature streams by concatenation or by absolute difference. The shared encoder with a feature-difference signal is the basis of the fusion used in this project. SNUNet-CD [11] combines a Siamese network with densely connected nested U-Net decoders and an ensemble channel attention module to preserve fine detail in very-high-resolution images.

Transformer-based models then added long-range context: BIT [12] converts CNN features into a small number of semantic tokens and models their relations with a transformer, and ChangeFormer [13] uses a hierarchical transformer encoder in a Siamese network. TransSiamUNet [14] augments a Siamese U-Net with transformer encoders and is evaluated on the OSCD benchmark, which makes it the closest architectural relative of this work; a Siamese Swin-Unet [15] replaces the convolutional blocks with a pure-transformer U-shaped design, and a survey reviews U-Net variants for change detection [16]. A controlled study [17] found that

a plain U-Net baseline, including Siamese concatenation and difference variants with a shared encoder, remains highly competitive with more elaborate architectures on standard benchmarks. This motivates the straightforward Siamese U-Net chosen here and the simple baselines in the planned evaluation. Most of these models, however, are designed and benchmarked for a binary changed/unchanged output on datasets such as LEVIR-CD [5] and OSCD [6], not for grading damage severity.

C. Building Damage Assessment on xBD

The xBD dataset [2] is a large-scale benchmark for building damage assessment from satellite imagery: it provides pre- and post-disaster images with building polygons labelled on a four-level damage scale (no damage, minor, major, destroyed) from 19 disaster events. Compared with binary benchmarks, xBD requires a model both to localise buildings and to grade damage, and its class distribution is extremely skewed towards background and undamaged buildings.

Dedicated models have been proposed for this task. BDANet [7] is a two-stage network: the first stage localises buildings with a U-Net, and the second stage grades damage with a two-branch multi-scale U-Net that uses cross-directional attention between pre- and post-event features. ChangeOS [8] integrates an object localisation network and a damage classification network in a single semantic change detection framework and works at the level of building objects. Both explicitly separate localisation from grading, unlike the single-stage implementation in this work, and their reported scores are not directly comparable with ours because our event partition and metrics differ (Sec. III-G). This project therefore uses a simpler single-stage model and concentrates on explanation, event-level evaluation and deployment (Sec. I-B).

D. Explainable AI

Deep networks are hard to inspect, which matters when predictions guide rescue decisions. Grad-CAM [18] produces a coarse localisation map by weighting the activation maps of a convolutional layer with the gradient of a class score, showing which image regions influenced that class. Seg-Grad-CAM [19] extends the idea to semantic segmentation by summing the class score over a chosen set of pixels, which is the formulation used in this work. SHAP [20] assigns additive feature attributions based on Shapley values and has theoretical guarantees, but it is computationally heavier for images. Such maps show where the network looked; they do not show that the prediction is correct.

E. Comparison of the Surveyed Approaches

Table I compares the surveyed approaches with the proposed system.

TABLE I. COMPARISON OF SURVEYED APPROACHES WITH THE PROPOSED SYSTEM

Approach	Task	Relation to this work
FC-Siam [10], SNUNet-CD [11]	Binary change detection	Siamese lineage
BIT [12], ChangeFormer [13]	Binary, transformer-based	Possible future encoder
TransSiamUNet [14]	Binary change detection (OSCD)	Closest architecture; not xBD
BDANet [7]	xBD damage, two-stage	Same task; adds a localisation stage and attention
ChangeOS [8]	xBD damage, object-based	Same task; object-level localisation
This work	**xBD, five-class pixel-level**	**Single stage; one held-out event; explanations and interface unvalidated**

III. METHODOLOGY

A. Dataset and Preprocessing

1) xBD Subset

xBD [2] contains sub-metre RGB image pairs of 1024×1024 pixels from 19 disaster events in its full release; our local copy contains ten events. Each building polygon in the post-disaster label files carries one of four damage labels (no damage, minor, major, destroyed). Polygons are rasterised into masks with values 0–4 (background, no damage, minor, major, destroyed); polygons labelled un-classified are mapped to background, an ignore label would be more faithful and is left for future work.

2) Masks, Tiling and Filtering

Each image pair and mask is tiled into non-overlapping 256×256 patches (a 4×4 grid, 16 patches per image). Patches that are almost entirely background are discarded, which leaves 15,845 patches in total; retained patches are cached as NumPy arrays. This filter removes uninformative tiles but changes the class distribution relative to whole scenes. The complete pre-processing pipeline is shown in Fig. 1.



Fig. 1. Data preparation and pre-processing pipeline for xBD.

3) Event-Level Partition

Scenes from one event share sensor, season, architecture and context, so an image- or patch-level split lets a model exploit event-specific appearance and inflates test scores. We therefore assign each event wholly to one partition (Table II). No event appears in two partitions. The partition differs from the official xBD split, so our scores are not directly comparable with published xView2 results. With only one test event, the evaluation measures transfer to a single unseen hurricane, not to disasters in general.

TABLE II. EVENT-LEVEL PARTITION

Split	Patches	Events
Train	7,114	guatemala-volcano, hurricane-florence, hurricane-matthew, midwest-flooding, palu-tsunami, santa-rosa-wildfire, socal-fire
Validation	5,113	hurricane-harvey, mexico-earthquake
Test	3,618	hurricane-michael

4) Class Distribution

The classes are strongly imbalanced (Table III). In the training split, background covers 81.71% of pixels and no damage 14.77%, while minor, major and destroyed damage cover only 0.94%, 0.60% and 1.98%. The validation events differ again, and the test event has yet another damage profile, so the three partitions have clearly different class balance.

TABLE III. PIXEL SHARE OF EACH CLASS IN THE THREE PARTITIONS

Class	Train (%)	Val. (%)	Test (%)
0 Background	81.71	70.84	85.35
1 No damage	14.77	22.36	8.69
2 Minor damage	0.94	1.93	3.53
3 Major damage	0.60	4.74	2.15
4 Destroyed	1.98	0.12	0.28

5) Augmentation

Training patches receive shared spatial transforms for the pre image, post image and mask: horizontal and vertical flips and random 90° rotations. Photometric colour jitter (brightness and contrast 0.8–1.2, saturation 0.9–1.1, hue ±0.05) and light Gaussian blur are applied to the images only [21], followed by ImageNet normalisation. Validation and test patches are only normalised.

B. Siamese U-Net with Shared ResNet34 Encoder

Pre- and post-event patches I_{pre} and I_{post} (3×256×256) pass through the same ImageNet-pretrained ResNet34 [22] encoder E, so both are embedded in one feature space (Fig. 2):

$$F_{\text{pre}} = E(I_{\text{pre}}),\ F_{\text{post}} = E(I_{\text{post}}) \qquad (1)$$

Five feature maps are used: stem (64 channels), layer1 (64), layer2 (128), layer3 (256) and layer4 (512). The model has 24,374,405 parameters.

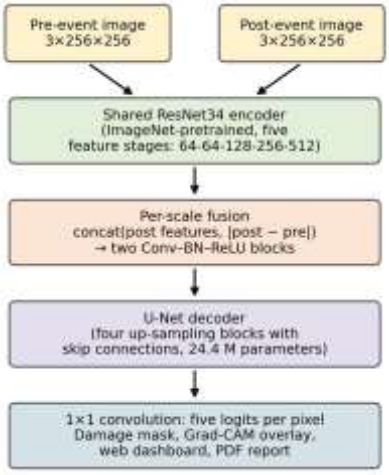


Fig. 2. Siamese U-Net with shared ResNet34 encoder, per-scale fusion and U-Net decoder.

C. Change-Feature Fusion

At each scale i the post-event features and the absolute feature difference are concatenated and passed through two Conv–BatchNorm–ReLU blocks, giving the decoder both the current appearance and an explicit change signal:

$$G_i = \text{Conv}\left(\text{Concat}\left(F_{\text{post,i}}, \left|F_{\text{post,i}} - F_{\text{pre,i}}\right|\right)\right) \qquad (2)$$

D. U-Net Decoder

Following the U-Net design [23], four decoder blocks (512→256, 256→128, 128→64, 64→32 channels) each upsample with a transposed convolution, concatenate the fused features of the matching scale and apply two Conv–BN–ReLU layers. A final upsampling step restores 256×256 resolution and a 1×1 convolution produces five logits per pixel; the predicted class is the argmax over classes.

E. Loss Function

Because background and undamaged pixels dominate, the loss combines a class-weighted Focal loss [24] with a soft Dice loss [25]:

$$L = L_{\text{focal}} + \lambda L_{\text{dice}} \qquad (3)$$

$$L_{\text{focal}} = -\alpha_t (1 - p_t)^{\gamma} \log p_t \qquad (4)$$

$$d_c = \frac{2\sum p_c g_c + s}{\sum p_c + \sum g_c + s} \qquad (5)$$

$$L_{\text{dice}} = 1 - \frac{1}{|K|}\sum_{c \in K} d_c \qquad (6)$$

with $\gamma = 2$, $\lambda = 1$ and smoothing $s = 1$; p_t is the predicted probability of the true class, p_c and g_c the predicted probability and one-hot ground truth of class c, and K the four damage classes. The Dice term is averaged over the damage classes only, so that the large background class cannot mask poor minority-class overlap. The class weights α_t were set by hand in rough inverse-frequency order and were not tuned.

F. Training Configuration

The model was implemented in PyTorch [26] and optimised with AdamW [27] under a cosine learning-rate schedule (Table IV). All 18 epochs were run, and the checkpoint with the highest validation macro F1 (epoch

14) was kept and evaluated once on the test event. Training used a single random seed.

TABLE IV. TRAINING CONFIGURATION

Parameter	Configuration
Framework	PyTorch 2.3.0, Python 3.12.10, CUDA 12.1
Backbone	ResNet34, ImageNet-pretrained
Input / batch	256×256; batch 2; gradient accumulation 4 (effective batch 8)
Optimiser	AdamW, learning rate 1×10^{-4}, weight decay 1×10^{-4}
Scheduler	Cosine annealing from 1×10^{-4} to 1×10^{-6} over 18 epochs
Stabilisation	Gradient clipping 1.0; FP32 (mixed precision disabled)
BatchNorm	Train mode; statistics from batches of 2 (not frozen)
Iterations	3,557 per epoch; median epoch time about 16 min
Model selection	Highest validation macro F1 (epoch 14 of 18); single seed
Hardware	NVIDIA GeForce GTX 1650, 4 GB memory

G. Evaluation Metrics

Metrics are accumulated in a global confusion matrix. For each class we report precision P = TP/(TP+FP), recall R = TP/(TP+FN), F1 = $2PR/(P+R)$ and IoU = TP/(TP+FP+FN). Macro averages are taken over the four damage classes only (background excluded). Pixel accuracy includes background and is dominated by it. Building-localisation F1 treats every damage class as "building" and measures separation from background. Because the application is safety-relevant, recall on major damage and destroyed is operationally the most important figure.

For orientation we also compute an xView2-style score, 0.3 × localisation F1 + 0.7 × the harmonic mean of the four damage-class F1 values, from pixel counts. It equals 0.378 on the test event. The official challenge metric is defined differently (on building objects and on its own partition), so this number must not be compared with leaderboard scores.

Comparability. Our partition, metric granularity (pixels, not buildings) and evaluation on a single event differ from published xBD protocols. We therefore make no comparison with BDANet [7] or ChangeOS [8], and we have not trained baseline models under our protocol.

H. Planned Evaluation, Baselines and Ablations

The first priority is a stronger evaluation: repeat training with several seeds, add leave-one-event-out or multiple test events, and run the baselines listed in Table V under the same protocol. Only experiments actually performed will be reported.

TABLE V. PLANNED BASELINES AND ABLATIONS (NOT YET RUN)

Configuration	Question answered
A. Proposed (post + |Δ| fusion; Focal + Dice)	Reference
B. Class-weighted CE only	Value of Focal + Dice
C. Focal only	Contribution of Dice
D. Dice only	Contribution of Focal
E. Concat [pre, post], no |Δ|	Value of difference signal
F. 6-channel early-fusion UNet [17]	Value of Siamese design

I. Explainable AI

Grad-CAM [18] weights each activation map A^k of a target layer by the spatially averaged gradient of the class score y^c, then applies ReLU (k indexes channels, Z is the number of spatial positions):

$$w_k^c = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \quad (7)$$

$$L^c = \mathrm{ReLU}\left(\sum_k w_k^c A^k\right) \quad (8)$$

Because the original Grad-CAM was defined for classification, we follow the Seg-Grad-CAM formulation [19]: the target score is the sum of the target-class logits over all pixels, the gradients are averaged per channel to weight the activations of a late decoder layer, and the ReLU-clipped, max-normalised map is resized to the input and overlaid on the post-event image [18]. The maps are conditioned on the image pair because the decoder activations depend on both inputs. We did not compare layers or run deletion tests, so the maps should be read as illustrations, not evidence. A SHAP-based attribution [20] is future work and is not implemented.

J. System Deployment

The trained model is served by a Django application with a MySQL back end. A user uploads a pre- and post-event pair; the system resizes it to 256×256, predicts the damage mask, overlays Grad-CAM, reports the percentage of predicted-building pixels per damage class and exports a PDF report.

Narrative summary. An optional narrative summary is generated by a Google Gemini vision model [28] from the images and statistics, with a deterministic rule-based fallback when the model is unavailable or no key is configured. The text is cached and reused in the PDF report.

Caveats. (i) The language-model narrative is not evaluated for factual accuracy and should be treated as a draft for expert review. (ii) Whole images are resized to 256×256 instead of tiled, reducing effective resolution relative to training. (iii) No calibrated per-class confidence is computed. (iv) There is no building-footprint gating, so terrain change can be reported as damage: in earlier experiments with a sanity-check checkpoint, scenes from outside xBD showed terrain change reported as building damage; this has not been re-examined with the final checkpoint and should be assumed to persist. (v) Inference latency and memory use have not been measured and the interface has not been tested with users. (vi) Uploaded imagery is sent to a third-party service for the narrative, so data-use terms and imagery licences must be checked before sensitive or licensed imagery is used.

IV. RESULTS

A. Held-Out Test Event

Table VI lists per-class results of the epoch-14 checkpoint on Hurricane Michael (3,618 patches, 237,109,248 pixels). Pixel accuracy is 87.6% and

building-localisation F1 is 0.780. The macro F1 over the damage classes is 0.286 (macro IoU 0.184, macro precision 0.301, macro recall 0.348).

TABLE VI. PER-CLASS RESULTS ON THE HELD-OUT TEST EVENT (HURRICANE MICHAEL)

Class	P	R	F1	IoU	Share (%)
Background	0.981	0.931	0.955	0.914	85.35
No damage	0.480	0.854	0.615	0.444	8.69
Minor damage	0.172	0.126	0.145	0.078	3.53
Major damage	0.364	0.094	0.150	0.081	2.15
Destroyed	0.186	0.319	0.235	0.133	0.28
Macro (4 damage cl.)	**0.301**	**0.348**	**0.286**	**0.184**	–

Intact buildings are found well (no-damage recall 0.854) but with modest precision (0.480), because many damaged pixels are also called no damage. Minor and major damage reach F1 of only 0.145 and 0.150. Major-damage recall is 0.094: roughly nine in ten major-damage pixels are missed as that class. Destroyed has the highest damage-class F1 (0.235) with recall 0.319 but precision 0.186, so the model flags destroyed buildings often but with many false alarms; this class covers only 0.28% of test pixels.

B. Error Structure

Fig. 3 shows the row-normalised confusion matrix. Most errors are grading errors, not detection errors: only 9.4% of major-damage pixels and 11.6% of minor-damage pixels are labelled background, but 59.4% of major and 71.0% of minor pixels are labelled no damage, and a further 17.9% of major pixels are labelled minor. Under-grading of damage is therefore the dominant failure, which is the more dangerous direction for response planning.

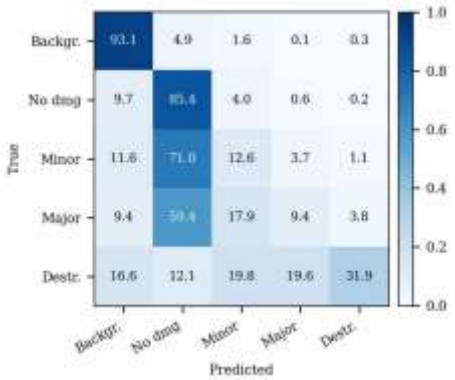


Fig. 3. Row-normalised confusion matrix on the test event (percent of true pixels; rows sum to 100).

C. Validation Versus Test, and Training Dynamics

TABLE VII. SUMMARY ON VALIDATION AND TEST EVENTS (EPOCH-14 CHECKPOINT)

Metric	Validation	Test
Pixel accuracy	0.803	0.876
Localisation F1	0.800	0.780
Macro F1 (damage)	0.178	0.286
Macro IoU (damage)	0.130	0.184
Minor / Major / Destroyed F1	0.033 / 0.021 / 0.003	0.145 / 0.150 / 0.235
xView2-style pixel score	0.246	0.378

Validation performance is much lower than test performance (macro F1 0.178 versus 0.286), mainly because major and destroyed damage are almost never recovered on the validation events (major recall 0.011, destroyed recall 0.006). The two partitions differ in hazard type and class balance (Sec. III-A), and with only two validation events and one test event we cannot say which factors explain the gap. We make no causal claim.

Fig. 4 shows that training and validation behaviour diverge. Training macro F1 rises from 0.161 to 0.595, whereas validation macro F1 fluctuates between 0.121 and 0.178, and validation loss increases from 1.53 in the first epoch to values around 1.9–2.1. The training metrics are computed on augmented training batches during the epoch, so they are not directly comparable with evaluation-mode numbers, but the gap is large enough to indicate that the model fits event-specific appearance and transfers poorly. The selected epoch 14 (validation macro F1 0.178) is only marginally better than epochs 13 (0.169) and 17 (0.171); with two validation events, checkpoint selection is therefore close to noise.

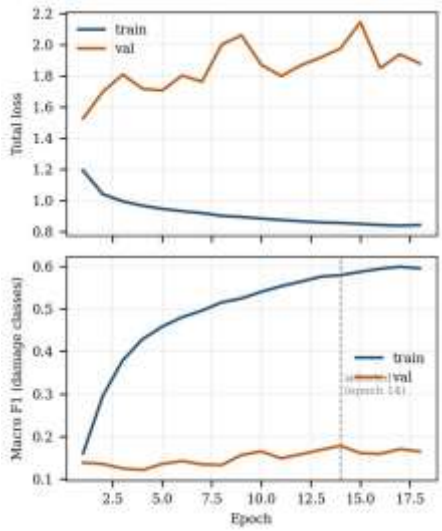


Fig. 4. Training and validation loss (top) and macro F1 over the damage classes (bottom). The dashed line marks the selected checkpoint.

D. Qualitative Results

Fig. 5 shows one successful and one failing test patch. In the first example the model separates the large intact buildings from background and the Grad-CAM map for the no-damage class highlights mainly the lower roof, with an isolated hot-spot on vegetation at the left. In the failure case, damage that is minor in the ground truth is predicted as undamaged and the Grad-CAM map spreads over the building and parts of its surroundings. These maps are qualitative aids; their faithfulness has not been tested.

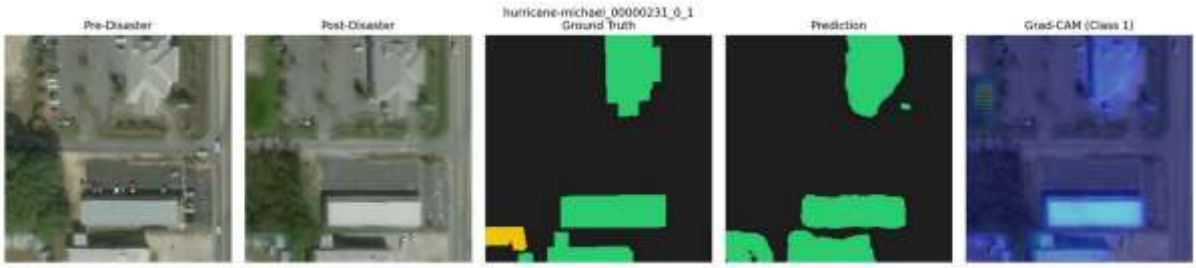


Fig. 5a. Example from the test event. Columns: pre-event image, post-event image, ground truth, prediction, Grad-CAM overlay for the no-damage class. Green marks no-damage buildings and yellow marks minor damage. The large no-damage buildings are found; the small minor-damage building at the lower left is predicted as no damage.



Fig. 5b. Failure case selected as one of the lowest-scoring test patches (same layout). The ground truth marks the building as minor damage (yellow); the model predicts no damage (green). The pre-event image appears to show the site during construction, so the pair differs in ways unrelated to storm damage; this is our visual reading of the image, not verified.

V. DISCUSSION

A. Interpretation of Results

The model detects buildings reasonably (localisation F1 0.78) but grades damage poorly, and most of the loss is under-grading: damaged pixels end up in the no-damage class. Heavy class imbalance, subtle visual differences between adjacent damage grades and a single-stage design that must localise and grade at once are plausible contributors, as is the limited number of training events, but we have not isolated any of them. The widening gap between training and validation scores suggests that the model memorises event-specific appearance; stronger augmentation, more events, or pre-training on unlabelled imagery may help.

The Focal plus Dice recipe did raise recall for the rarest class on the test event (destroyed recall 0.319), but at low precision, and without ablations we cannot attribute this to any single design choice. Results from one test event and one seed are also sensitive to chance, so the numbers in Table VI should be read as indicative.

B. Limitations

- **Evidence level.** One test event, one seed, no confidence intervals, no baselines or ablations. No claim of generalisation to unseen disasters can be made.
- **Validation design.** Checkpoint selection used two validation events, and validation scores fluctuate by about ±0.02 between epochs.
- **Comparability.** The partition, the pixel-level metrics and the retained patches (almost-empty patches were discarded) differ from official xBD protocols. Evaluation on unfiltered tiles was not performed.
- **Label handling.** The treatment of un-classified buildings follows the pre-processing pipeline and was not studied.
- **Deployment.** Whole images are resized to 256×256 rather than tiled; no calibrated confidence is produced; building-footprint gating is absent. In earlier experiments with a sanity-check checkpoint, scenes from outside xBD showed terrain change reported as building damage; this behaviour has not been re-examined with the final checkpoint and should be assumed to persist. The Gemini narrative and Grad-CAM maps are unvalidated. The system must not be used for operational decisions.
- **Scope.** Although the title refers to environmental monitoring, the pipeline is trained and evaluated only for building-damage assessment on xBD; no evaluation on other environmental changes (for example vegetation, water or land cover) is reported.
- **Data bias.** xBD over-represents certain regions and building types; transfer to other regions, including India, is untested. The model uses RGB imagery and needs a co-registered, cloud-free pre-event image.

VI. CONCLUSION

We presented a documented pipeline for five-class building-damage segmentation from bi-temporal imagery and evaluated it on a disaster event unseen during training. The model localises buildings with an F1 of 0.780, but its damage grading is weak (macro F1 0.286) and mostly errs towards under-grading, and training and validation behaviour diverge. The results are a starting

point, not a demonstration of generalisation: they come from one test event and one seed without baselines. Until multi-event evaluation, ablations, footprint gating and expert validation are done, the system should not inform operational decisions.

A. Future Work

The most important next step is a stronger evaluation: repeat training with several seeds, add leave-one-event-out or multiple test events, and run the baselines and ablations listed in Table V under the same protocol. Only this evaluation can show whether the proposed fusion and loss design generalise to unseen disasters.

- **Reliability of the output.** Building-footprint gating and a two-stage design as in [7], [8] would stop the system from reporting damage where there are no buildings, and uncertainty estimation would tell users how far each prediction can be trusted.
- **Full-resolution inference.** Tiled inference would let the system analyse whole 1024×1024 scenes without shrinking them to 256×256, and latency should be measured so that the deployment claims can be checked.
- **Stronger models and inputs.** Transformer encoders could capture longer-range context, and multispectral or SAR data could help in cloudy conditions and for materials that are hard to separate in RGB images.
- **Learning with less labelled data.** Self-supervised pre-training on unlabelled imagery may reduce the memorisation of event-specific appearance observed here.
- **Better explanations.** Quantitative faithfulness tests, such as deletion experiments, and SHAP-based attributions [20] would show whether the Grad-CAM maps reflect the evidence used by the model.
- **Integration and validation.** Evaluation on non-xBD disasters, including events in India, would test whether the approach transfers to other regions and building types, and expert validation is needed before any operational use.

DECLARATIONS

Data availability. The xBD dataset is publicly available from its authors [2]. The event-level partition used in this study is listed in Table II.

Use of AI tools. The web application described in Sec. III-J uses a Google Gemini model [28] to generate an optional narrative summary; this narrative has not been evaluated for factual accuracy.

REFERENCES

[1] United Nations Office for Disaster Risk Reduction, *Human Cost of Disasters: An Overview of the Last 20 Years (2000–2019)*. Geneva, Switzerland, 2020.

[2] R. Gupta *et al.*, "Creating xBD: A dataset for assessing building damage from satellite imagery," in *Proc. CVPR Workshops*, 2019.

[3] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, 2017.

[4] L. Ma *et al.*, "Deep learning in remote sensing applications: A meta-analysis and review," *ISPRS J. Photogramm. Remote Sens.*, vol. 152, pp. 166–177, 2019.

[5] H. Chen and Z. Shi, "A spatial-temporal attention-based method and a new dataset for remote sensing image change detection," *Remote Sens.*, vol. 12, no. 10, art. 1662, 2020.

[6] R. C. Daudt, B. Le Saux, A. Boulch, and Y. Gousseau, "Urban change detection for multispectral earth observation using convolutional neural networks," in *Proc. IEEE IGARSS*, 2018.

[7] Y. Shen *et al.*, "BDANet: Multiscale convolutional neural network with cross-directional attention for building damage assessment from satellite images," arXiv:2105.07364, 2021.

[8] Z. Zheng, Y. Zhong, J. Wang, A. Ma, and L. Zhang, "Building damage assessment for rapid disaster response with a deep object-based semantic change detection framework: From natural disasters to man-made disasters," *Remote Sens. Environ.*, vol. 265, art. 112636, 2021.

[9] A. Singh, "Digital change detection techniques using remotely-sensed data," *Int. J. Remote Sens.*, vol. 10, no. 6, pp. 989–1003, 1989.

[10] R. C. Daudt, B. Le Saux, and A. Boulch, "Fully convolutional Siamese networks for change detection," in *Proc. IEEE ICIP*, 2018.

[11] S. Fang *et al.*, "SNUNet-CD: A densely connected Siamese network for change detection of VHR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022.

[12] H. Chen, Z. Qi, and Z. Shi, "Remote sensing image change detection with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022.

[13] W. G. C. Bandara and V. M. Patel, "A transformer-based Siamese network for change detection," in *Proc. IEEE IGARSS*, 2022.

[14] F. Ali, S. S. Labib, A. Mahmoud, and I. E. Fattoh, "TransSiamUNet: Transformer-augmented Siamese U-Net for precise change detection in satellite imagery," *Sci. Rep.*, vol. 16, art. 11689, 2026.

[15] Y. Tang, Z. Cao, N. Guo, and M. Jiang, "A Siamese Swin-Unet for image change detection," *Sci. Rep.*, vol. 14, 2024, doi: 10.1038/s41598-024-54096-8.

[16] Y. Amin, N. S. Trivedi, and R. Bhattad, "A comparative study of U-Net architectures for change detection in satellite images," arXiv:2506.07925, 2025.

[17] I. Corley, C. Robinson, and A. Ortiz, "A change detection reality check," in *ICLR 2024 Workshop on Machine Learning for Remote Sensing*; arXiv:2402.06994.

[18] R. R. Selvaraju *et al.*, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE ICCV*, 2017, pp. 618–626.

[19] K. Vinogradova, A. Dibrov, and G. Myers, "Towards interpretable semantic segmentation via gradient-weighted class activation mapping (student abstract)," in *Proc. AAAI*, vol. 34, no. 10, 2020, pp. 13943–13944.

[20] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. NeurIPS*, vol. 30, 2017.

[21] A. Buslaev *et al.*, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, art. 125, 2020.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*, 2015.

[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE ICCV*, 2017, pp. 2980–2988.

[25] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 3DV*, 2016.

[26] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NeurIPS*, vol. 32, 2019.

[27] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.

[28] Gemini Team, Google, "Gemini: A family of highly capable multimodal models," arXiv:2312.11805, 2023.