King’s College London

Engineering Department – Faculty of Natural, Mathematics and Engineering Sciences

AI-Enabled Low-Carbon Reconstruction in Conflict Zones: Engineering Scalable Solutions for Post-War Recovery

Supplementary Material - Methods

Author: Youssef Alan Franci – youssef.a.franci@kcl.ac.uk

Supervisor: Mohit Arora, PhD – mohit.arora@kcl.ac.uk

Project Report submitted in partial fulfilment of the requirements of the MSc Individual Project Module

August 2025

Contents

List of Figures

List of Tables

Chapter 1

Damage Assessment

The damage classification model developed in this study represents a multi-modal, data-driven approach to post-conflict structural assessment. Drawing from gaps identified in recent literature—particularly the lack of spatially resolved ground-truth data, class imbalance correction, and multi-temporal consistency—this methodology integrates atmospherically corrected satellite imagery, synthetic aperture radar (SAR) metrics, and spatial-geometric modelling to produce building-level damage predictions in the Gaza Strip. This chapter outlines the design, preprocessing, and modelling workflow in detail, justifying each step based on current research limitations and articulating the novel contributions of this approach.

1.1 Methodological Framework and Rationale

As established in the literature review(**??**), existing approaches to structural damage detection suffer from fragmented data sources, static imagery interpretation, and limited integration of spatial-geometric features. This project aims to address these deficiencies by developing an automated classification pipeline that:

- Operates at the individual building level using spatially joined remote sensing data and open geographic databases;
- Combines multi-spectral optical imagery with radar-based SAR features;
- Implements feature engineering techniques informed by vegetation loss and material reflectance analysis;
- Applies advanced ensemble learning with imbalance handling;
- Supports full interpretability and model reproducibility.

This model pipeline not only integrates traditional features such as NDVI and RGB bands but also introduces radar backscatter statistics and delta band differencing—components seldom used together in a unified, building-resolved classification system. The integration of contextual geometry (from OpenStreetMap), UNOSAT multi-temporal data, and PlanetScope imagery allows for high-resolution, scalable mapping of damage in areas with sparse ground-truth data.

1.2 Data Acquisition and Preprocessing

1.2.1 Data Sources

The input data span three core modalities:

1. **Damage Point Observations:** Derived from the United Nations Satellite Centre (UNOSAT)[1][**??**], delivered in geographic database format (.gdb) and converted to GeoJSON for integration. Each point includes a numerical damage classification (1 = Destroyed, 2 = Damaged, 3 = Possibly Damaged, 4 = No Visible Damage) and up to nine temporal snapshots.

2. **Building Footprints:** Extracted from OpenStreetMap (OSM)[2] [**??**] using a 30-zone grid-based Overpass API query covering the full bounding box of the Gaza Strip. Twenty-two zones returned valid building geometries, amounting to over 360,000 structures as shown in Table 1.1 and Figure 1.1.

3. **Spectral and SAR Imagery:** Sourced from PlanetScope (8-band Surface Reflectance imagery) [3] and Sentinel-1 SAR data [4](preprocessed for amplitude and coherence backscatter analysis). The selection of these image products was based on their spectral richness, temporal availability, and atmospheric correction quality.

   Author's Note: *Due to data availability constraints, image clarity considerations, and access restrictions, August 2, 2023, and February 18, 2025, were selected as the baseline dates for the extraction of SAR and optical satellite imagery used in the pre- and post-conflict assessment, respectively.* **??**

1.2.2 Temporal Filtering of UNOSAT Observations

The UNOSAT dataset includes up to nine temporal observations per feature. To align each observation with a consistent reference window, a custom filtering routine selects the date closest to the post conflict date we have chosen (18 February 2025) (Ref 3):

$$d_{\text{closest}} = \underset{d_i \in D}{\arg\min}, |d_i - d_{\text{ref}}|, \tag{1.1}$$

where D is the set of valid observation dates and d_{ref} is the fixed reference timestamp. Records with no valid observations were discarded to maintain temporal coherence.

1.2.3 Spatial Join of Damage Points to Buildings

To create a labelled training dataset, UNOSAT damage points were spatially joined to OSM building footprints in QGIS [5, 1, 2]. The spatial join rule was defined such that each building inherits the **maximum** damage level from any intersecting point:

$$L_{\text{building}} = \max(L_{\text{intersections}}), \tag{1.2}$$

ensuring conservative classification. Buildings with no overlapping damage points were labeled as “null” and excluded from training. The resulting file (`buildings_with_damage.gpkg`) provided geometry and label inputs for supervised learning.

Zone ID	Building Count	Zone ID	Building Count
Z1	27,546	Z11	3
Z2	19,417	Z12	149
Z3	1,872	Z13	0
Z4	807	Z14	4,938
Z5	229	Z15	36,658
Z6	0	Z16	20,761
Z7	16,990	Z17	1,047
Z8	53,355	Z18	707
Z9	26,950	Z19	0
Z10	412	Z20	0

Table 1.1: Sample of building polygon counts extracted per zone

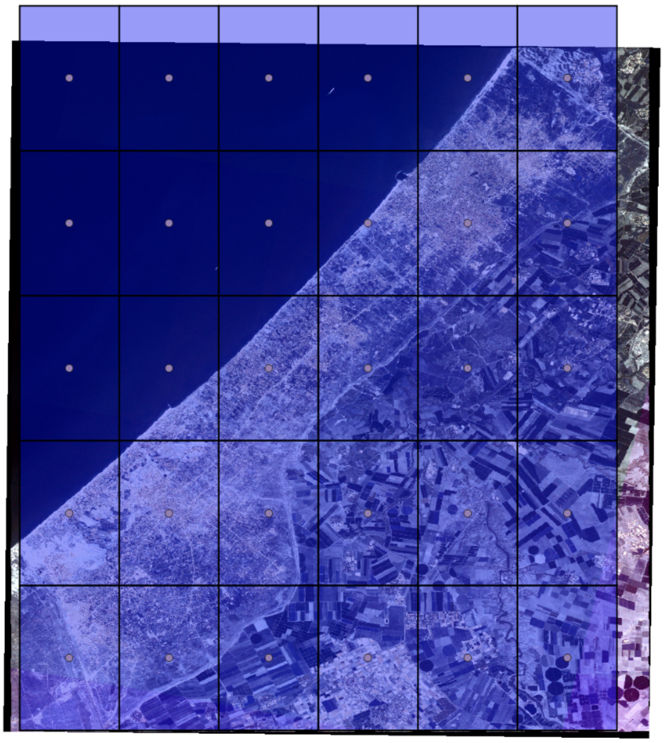

Figure 1.1: Grid-based zoning of Gaza Strip for OSM building footprint extraction

1.3 Spectral and SAR Feature Engineering

This study employs a building-centric feature engineering approach, integrating optical and radar satellite data to construct a robust damage classification model. Unlike traditional pixel-based methodologies, which process satellite imagery at the individual pixel level, this research applies self-similarity and contextual modeling principles at the building object level. Each structure, identified through OpenStreetMap (OSM) footprints, is treated as a unit of analysis, allowing for contextually grounded assessments that mirror real-world damage boundaries and reduce noise introduced by pixel-scale variation.

1.3.1 Object-Level Self-Similarity Descriptors

Building on the framework of Kahraman et al. (2016), who proposed the use of Self-Similarity Descriptors (SSD) for damage detection in dense urban scenes, this project adapts that concept from a pixel-based implementation to a building-level descriptor model [6]. Instead of comparing small pixel patches within an image, we construct a feature vector for each building that encodes how its spectral and environmental properties have changed over time. To operationalize this, pre- and post-conflict surface reflectance values were extracted for each building using the centroid of its polygon geometry, and values were aggregated from PlanetScope's 8-band Surface Reflectance imagery. The computed descriptors include:

$$\Delta B_i = B_i^{\text{post}} - B_i^{\text{pre}}, \tag{1.3}$$

where B_i represents the surface reflectance in spectral band i, including Red, Green, Blue, NIR, and SWIR bands. This temporal difference vector forms the core self-similarity descriptor for each building, capturing how its spectral signature changed as a result of conflict-induced structural or environmental alterations.

In addition to band differencing, the Normalized Difference Vegetation Index (NDVI) and its temporal change were calculated to reflect vegetation loss:

$$\text{NDVI} = \frac{\text{NIR} - \text{Red}}{\text{NIR} + \text{Red}}, \quad \Delta\text{NDVI} = \text{NDVI}^{\text{post}} - \text{NDVI}^{\text{pre}}. \tag{1.4}$$

Vegetation loss or transformation—especially in residential areas—can signify rubble accumulation, fire damage, or heavy structural collapse.

Each building's full SSD vector thus takes the form:

$$\text{f}_{SSD} = [\Delta B_1, \Delta B_2, \Delta\text{NDVI}, \text{Zone, Governate, Building Area,...}] \tag{1.5}$$

combining spectral, environmental, and contextual features to form a multi-dimensional representation of structural change.

1.3.2 Synthetic Aperture Radar (SAR) Features

To improve damage detection in visually ambiguous or obstructed areas (e.g., due to debris, smoke, or shadowing), SAR data from Sentinel-1 was integrated. SAR's sensitivity to dielectric properties and surface roughness offers a complementary modality to optical reflectance. The following building-level radar features were extracted from pre-processed VV polarization imagery:

$$\Delta\text{Amp}_{\text{SAR}} = \frac{A^{\text{post}} - A^{\text{pre}}}{A^{\text{post}} + A^{\text{pre}} + \epsilon}, \tag{1.6}$$

where A is the average radar amplitude (intensity) over the building footprint or centroid region, and ϵ is a small smoothing constant to prevent instability. This normalized change captures structural disruption as loss of surface coherence or reflectivity.

1.3.3 Building-Level Contextual Smoothing via Markov Random Field Logic

While no formal pixel-based Markov Random Field (MRF) was implemented, we operationalized the core principle of spatial dependency through building-level contextual aggregation. In the original MRF formulation, damage classification is refined using a joint energy minimization function that penalizes abrupt changes in label assignment between neighboring pixels. We adapted this idea by integrating spatial and administrative metadata at the building level:

- **Zone and Municipality Codes:** These identifiers implicitly capture urban density, built environment patterns, and conflict targeting patterns.
- **Neighborhood Density Measures:** Using GIS buffering techniques, local building density and damage clustering statistics were calculated.

This approach approximates MRF-style smoothing by promoting damage label consistency across contiguous buildings, especially those falling within the same conflict-affected zone. The classification model is thus encouraged to treat each building not in isolation, but in relation to its geographic and administrative context.

Conceptually, the smoothed loss function for label prediction can be described as:

$$E(\mathbf{L}) = \sum_{b\in B} \phi_b(l_b) + \sum_{(b_i,b_j)\in\mathcal{N}} \psi(l_{b_i}, l_{b_j}), \tag{1.7}$$

where:

- B is the set of all buildings;
- l_b is the predicted damage label for building b;
- $\phi_b(l_b)$ is the unary cost (from classifier prediction);
- $\psi(l_{b_i}, b_{b_j})$ is the pairwise penalty for inconsistent labels between neighboring buildings b_i and b_j.

Although this function is not explicitly minimized via a graphical model, it is approximated in the ensemble learner's decision boundaries through structured features that reflect spatial correlation.

1.3.4 Cleaning, Encoding, and Dataset Finalization

After feature computation, the data was cleaned and standardized:

- Columns with 90% missing data were dropped.
- Numerical features were filled with zero.
- Categorical variables (e.g., governorate) were imputed as "unknown" and encoded using `LabelEncoder`.

- All features were cast to numeric types using:

  ```
  gdf[col] = pd.to_numeric(gdf[col], errors='coerce')
  ```

This produced a building-level feature matrix combining SSD-style spectral and SAR differences with MRF-inspired contextual data, ready for supervised classification.

1.4 Damage Classification Model: Architecture, Class Balancing, and Evaluation

1.4.1 Model Selection: Extreme Gradient Boosting (XGBoost)

To classify building-level damage severity, the final feature matrix was modeled using the Extreme Gradient Boosting (XGBoost) algorithm. XGBoost is a high-performance, tree-based ensemble method known for its speed, scalability, and resilience to noise and class imbalance. It has demonstrated superior performance in remote sensing classification tasks, including post-disaster infrastructure evaluation [7].

The model was configured for multi-class classification, using the `multi:softmax` objective function to predict a single integer-encoded damage label for each structure. Four final damage classes were defined, based on UNOSAT taxonomy:

- `0`: No Visible Damage
- `1`: Possibly Damaged
- `2`: Damaged
- `3`: Destroyed

Sparse or under-represented classes (e.g., `5` in some variations of the dataset) were merged into the nearest semantic category (e.g., `4` → `3`) to simplify the class structure and reduce over fitting.

1.4.2 Addressing Class Imbalance: Weighting and Oversampling

Consistent with findings from prior literature, class imbalance posed a significant challenge—most buildings in the training set exhibited either possible damage or destruction. To ensure the model performed well across all severity levels, two imbalance mitigation techniques were applied:

1. Class Weighting Sample weights were computed based on inverse class frequency using Scikit-learn's `compute_class_weight('balanced')` function:

$$w_i = \frac{n}{k \cdot n_i}, \tag{1.8}$$

where:

- n is the total number of training samples,
- k is the number of unique classes,
- n_i is the number of samples in class i.

These weights were applied to the loss function during training, forcing the model to penalize errors in minority classes more heavily.

2. SMOTE Oversampling A data augmentation strategy used was SMOTE (Synthetic Minority Oversampling Technique), which synthetically generates new feature vectors for under-represented classes using k-nearest neighbour interpolation [8]. This method was toggleable based on a configuration flag:

```
if apply_smote:
X_train, y_train = SMOTE().fit_resample(X_train, y_train)
```

1.4.3 Train-Test Splitting and Stratification

The dataset was split into training and testing sets using an 80/20 split, ensuring proportional representation of all damage classes via stratified sampling:

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42
)
```

This preserves class balance across both subsets and ensures unbiased performance evaluation.

1.4.4 Hyperparameter Optimization

Model tuning was performed using `RandomizedSearchCV` over a predefined distribution of hyperparameters, including:

- Learning rate (`eta`)
- Maximum tree depth

- Column sampling rate
- Subsample rate
- Regularization terms (λ, α)

Each configuration was evaluated via 3-fold cross-validation on the training set, using accuracy as the scoring metric:

```
grid = RandomizedSearchCV(
    estimator=clf,
    param_distributions=param_dist,
    n_iter=10,
    scoring="accuracy",
    cv=3,
    verbose=1,
    n_jobs=-1
)
grid.fit(X_train, y_train, sample_weight=train_weights)
```

The best-performing estimator was retained for final evaluation.

1.4.5 Evaluation Metrics and Confusion Analysis

Model performance was assessed using the following metrics:

- **Accuracy:** Overall percentage of correct predictions.
- **Precision, Recall, F1-Score:** Calculated per class, with macro and weighted averages.
- **Confusion Matrix:** Visualized via Seaborn heatmap to identify class-specific misclassification patterns.

These metrics provide a multi-faceted view of model behavior, particularly in imbalanced scenarios. For example, high recall in the “Destroyed” class indicates strong sensitivity to catastrophic damage, while high precision suggests few false positives in this critical category.

1.5 Validation, Interpretability, and Integration into the Reconstruction Pipeline

1.5.1 Model Validation and Generalization

To ensure the classifier's robustness and applicability in real-world post-conflict scenarios, validation was conducted on a held-out test set representing 20% of the total labeled data. Stratified sampling ensured that damage categories were proportionally represented in both training and test partitions, thus allowing for consistent comparison.

The primary performance metrics—accuracy, macro-averaged F1-score, and per-class precision and recall—were evaluated against ground-truth UNOSAT labels. Additional insights were derived from confusion matrix visualizations, which helped diagnose misclassification patterns, particularly between adjacent categories such as "Possibly Damaged" and "Damaged." These adjacent-class errors reflect inherent ambiguity in satellite imagery and were treated as expected performance noise rather than systemic failure.

1.5.2 Explainability and Feature Attribution

Given the high-stakes nature of post-conflict reconstruction and potential downstream decision-making implications, model interpretability was prioritized. XGBoost's tree-based structure permits straightforward feature attribution through built-in methods such as `feature_importances_`,which ranks inputs based on their relative contribution to decision splits.

This interpretability was essential for:

- Verifying that spectral deltas and SAR features were key drivers in predicting high-damage labels.
- Understanding contextual bias introduced by administrative metadata (e.g., certain municipalities systematically showing higher damage levels).
- Ensuring transparency for humanitarian and urban planning stakeholders who require accountability in AI-driven damage assessments.

In future iterations, SHAP (SHapley Additive exPlanations) values could further enhance feature-level interpretability by quantifying the contribution of each feature to a specific prediction.

1.5.3 Integration into the Reconstruction Pipeline

The damage classifier developed in this study serves as the entry point to a larger AI-enabled reconstruction framework. Once buildings are labelled according to their damage severity, these classifications inform debris volume estimation, cost estimation and sustainability modelling.

This integration positions the damage classifier not as an isolated tool, but as a critical upstream module in a comprehensive, data-driven, and environmentally conscious reconstruction pipeline.

1.5.4 Contribution to the State of the Art

This methodology advances the current state of damage classification in several key ways:

- **Object-Level SSD Implementation:** By shifting SSD calculations from pixels to buildings, the model reflects real-world structural boundaries and reduces noise.
- **Context-Aware MRF Approximation:** Instead of relying on pixel adjacency, spatial smoothing is embedded at the building level via administrative metadata and zone-aware modelling.
- **Multi-Modal Feature Fusion:** Integration of spectral deltas, NDVI, SAR backscatter changes, and geospatial metadata enables the classifier to operate under both optical clarity and occlusion conditions.
- **Reproducibility and Scalability:** All data processing and modelling steps were implemented in open-source frameworks (GeoPandas, Scikit-learn, XGBoost), with compatibility for cloud-based workflows (Google Colab, GEE).

These contributions position the model as a novel, reproducible, and extensible solution for post-conflict damage detection, with immediate applicability in humanitarian and reconstruction contexts.

Conclusion

This chapter has presented the complete methodology underlying the building-level damage classification system developed for the Gaza Strip. Drawing on the limitations observed in current literature—particularly the lack of spatial grounding, temporal consistency, and multimodal feature integration—this approach offers a scalable, interpretable, and context-aware solution to damage detection in post-conflict environments. By combining atmospherically corrected spectral data, radar-derived backscatter features, and spatially joined administrative metadata, the model transcends pixel-level noise and instead operates on semantically meaningful urban objects. The classifier not only achieves robust performance across imbalanced damage categories but also seamlessly integrates into a broader AI-enabled reconstruction pipeline. In doing so, it lays the technical foundation for accurate debris estimation, cost modelling, and sustainable material planning, thereby directly supporting both emergency response and long-term recovery strategies.

Chapter 2

Debris Analysis

This chapter presents the technical workflow developed for per-building debris quantification and material usability classification, leveraging pre- and post-event Sentinel-1 SAR data. The method extends the dataset and damage pipeline described in the preceding part (Ref: Part **??**), utilizing robust, data-driven thresholds, and integrates open-source geospatial analysis tools to maximize reproducibility and rigour.

2.1 Data Assembly and Preprocessing

2.1.1 Geospatial Dataset Structure

All analysis utilizes the existing master GeoPackage (`damage_features_buildings_updated.gpkg`) containing $N = 335,000+$ individual building polygons for the Gaza Strip, each annotated with:

- Geometry (EPSG:32636/UTM Zone 36N)
- Building attributes (area, metadata)
- Model-based damage class prediction
- Per-building mean SAR backscatter (pre- and post-event, VV polarization, log scale)

Buildings lacking valid geometry or SAR features are excluded *ab initio.*

2.1.2 SAR Feature Extraction

For each building polygon i:

$$\mu^{(i)}_{\mathrm{VV,pre}} = \frac{1}{|\Omega_i|} \sum_{p \in \Omega_i} \log_{10}(I_{\mathrm{VV,pre}}(p)) \tag{2.1}$$

$$\mu^{(i)}_{\mathrm{VV,post}} = \frac{1}{|\Omega_i|} \sum_{p \in \Omega_i} \log_{10}(I_{\mathrm{VV,post}}(p)) \tag{2.2}$$

where Ω_i is the set of pixels within building i and $I_{\mathrm{VV},*}$ is the calibrated backscatter intensity from Sentinel-1 GRD imagery. Zonal statistics are performed using `rasterio` and `geopandas` in Python.

2.1.3 Data Quality and Masking

Non-numeric or missing SAR values are imputed as NaN. Buildings with less than 10 valid pixels in either pre- or post-event scene are masked. Histograms of $\mu_{\mathrm{VV,pre}}$ and $\mu_{\mathrm{VV,post}}$ are inspected to verify lack of systematic bias or sensor error (see Fig. 2.1 and Fig. 2.2).

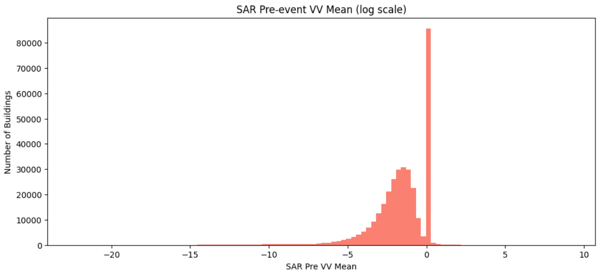

Figure 2.1: Histogram of mean VV-band SAR intensities over building footprints in the post-event image, computed in the logarithmic domain.

2.2 Feature Engineering

2.2.1 SAR Intensity Drop (Primary Change Proxy)

The per-building SAR intensity drop is defined as:

$$\Delta\sigma^{(i)} = \mu^{(i)}_{\mathrm{VV,post}} - \mu^{(i)}_{\mathrm{VV,pre}} \tag{2.3}$$

This is the principal change metric, serving as a proxy for physical change, with negative values indicating loss of reflective material (collapse/clearance) and positive values indicating stable or new reflective surfaces.

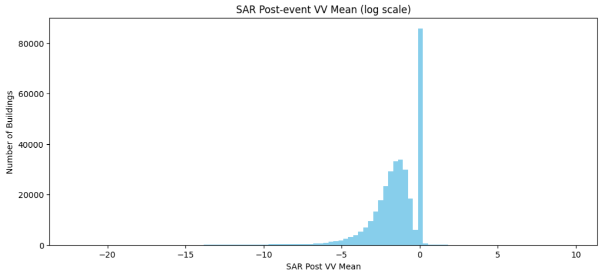

Figure 2.2: This histogram displays the distribution of mean VV-polarized SAR backscatter for all buildings following the conflict. Its primary purpose is to reveal the empirical structure of material reflectivity in Gaza, supporting both threshold calibration and class interpretation. Multi-modality or long tails in this distribution may indicate subpopulations (e.g., dense concrete districts vs. zones of metallic debris), justifying the use of region-specific rather than literature-based thresholds. Annotated vertical lines indicate the boundaries used for classifying material type. By confirming that class boundaries are anchored in data, this plot underpins the methodological validity of the subsequent classification.

2.2.2 Statistical Summary and Threshold Calibration

The empirical distribution of both $\mu_{\mathrm{VV,post}}$ and $\Delta\sigma$ is analyzed across all buildings. Quantiles (see Table 2.1) are computed for robust thresholding, avoiding the use of arbitrary or literature-derived cutoffs.

Table 2.1: Key empirical quantiles for SAR features, Gaza dataset.

Quantile	$\mu_{\mathrm{VV,post}}$	$\Delta\sigma$
0.01	-8.65	-2.22
0.05	-4.38	-1.04
0.25	-2.21	-0.14
0.50	-1.32	0.00
0.75	0.00	0.42
0.95	0.00	1.42
0.99	0.00	2.72

2.3 Building-Level Classification Rules

2.3.1 Material Type Assignment

Material type is assigned based on the post-event mean VV SAR value, using Gaza-specific thresholds:

$$\text{MaterialType}(i) = \begin{cases} \text{“steel/metal roof or new debris”}, & \mu_{\text{VV,post}}^{(i)} \geq 0 \\ \text{“concrete/masonry”}, & -2 < \mu_{\text{VV,post}}^{(i)} < 0 \\ \text{“light material/possible debris or empty”}, & \mu_{\text{VV,post}}^{(i)} \leq -2 \end{cases}$$

These thresholds are justified by the observed multimodal structure of the SAR distribution (see Fig. 2.3) and are validated by inspection of urban zones with known material differences.

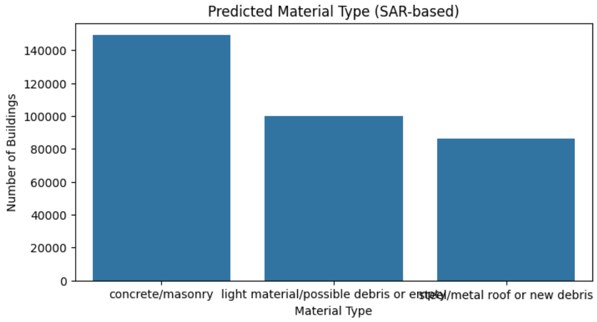

Figure 2.3: Histogram showing predicted material types relative to the number of buildings using SAR data.

2.3.2 Debris Usability Score

Debris usability is classified from the SAR intensity drop:

$$\text{Usability}(i) = \begin{cases} \text{“not usable”}, & \Delta\sigma^{(i)} < -1 \\ \text{“likely usable”}, & \Delta\sigma^{(i)} > -0.1 \\ \text{“uncertain”}, & \text{otherwise} \end{cases}$$

This rule is empirically tuned to the 5th and 25th percentiles (see Table 2.1), capturing the most destructive changes as “not usable” and stable or improved reflectors as “likely usable.”

2.3.3 Debris Amount Proxy

A relative, quantitative estimate of debris amount is computed per building:

$$D^{(i)}_{\text{proxy}} = |\Delta\sigma^{(i)}|$$

This proxy allows mapping of the spatial intensity of debris generation, supporting priority setting in clearance logistics, but is not a direct volumetric estimate.

2.4 Algorithmic Implementation

2.4.1 Efficient Vectorized Operations

All classification logic is implemented in Python, using the `pandas.DataFrame.apply` method for row-wise operations. For clarity and reproducibility, the core material type function is:

```
def sar_material_type(row):
    if row['sar_post_band1_mean'] >= 0:
        return 'steel/metal roof or new debris'
    elif row['sar_post_band1_mean'] > -2:
        return 'concrete/masonry'
    else:
        return 'light material/possible debris or empty'
```

Similarly, debris usability is assigned:

```
def sar_debris_usability(row):
    if row['sar_intensity_drop'] < -1:
        return 'not usable'
    elif row['sar_intensity_drop'] > -0.1:
        return 'likely usable'
    else:
        return 'uncertain'
```

2.4.2 Workflow Summary

1. Extract zonal SAR means for each building using pre-aligned, radiometrically calibrated Sentinel-1 scenes.

2. Compute intensity drop $\Delta\sigma$ and absolute value proxy.

3. Assign material type and debris usability per above rules.
4. Aggregate outputs in a GIS-enabled table for further visualization and downstream planning.

2.5 Assumptions and Limitations

- SAR backscatter is assumed radiometrically comparable across scenes due to standardized preprocessing.
- Thresholds are region-specific, based on Gaza's empirical SAR distributions; some ambiguity may persist near boundaries.
- Proxy debris metric is relative, not a direct measure of tonnage or volume.
- Usability classes are a heuristic mapping, with some uncertainty for intermediate values.

Chapter 3

Multi-Criteria Spatial Planning

This chapter presents a comprehensive methodology for analysing post-conflict urban damage, estimating debris and resource needs, and generating actionable reconstruction strategies in the Gaza Strip. The approach leverages advanced spatial analysis techniques, machine learning, and sustainable material modelling. The process begins with data acquisition and spatial preprocessing, continues through gridded spatial damage assessment and debris estimation, and concludes with enriched per-building cost and carbon analyses.

3.1 Geospatial Data Acquisition and Preprocessing

The analytical workflow begins with a geospatial dataset comprising over 340,000 building polygons in Gaza. Each building footprint is enriched with metadata including predicted damage classification (`model_pred_class`), inferred typology (e.g., apartment, house, mosque), and geometrical attributes. The dataset is ingested from a GeoPackage (`buildings_with_predictions.gpkg`) and requires projection into a metrically consistent coordinate reference system to ensure valid spatial operations.

To enable accurate distance and area-based computations, the data are reprojected to EPSG:32636, the UTM Zone 36N projection applicable to Palestine. The following Python snippet ensures projection consistency:

```
import geopandas as gpd

gdf = gpd.read_file('buildings_with_predictions.gpkg')
if gdf.crs is None or gdf.crs.to_epsg() != 32636:
    gdf = gdf.to_crs('EPSG:32636', allow_override=True)
```

Following re-projection, descriptive statistics are generated to verify the distribution of typologies and damage classifications. This includes basic plots of damage severity and structure type across the region.

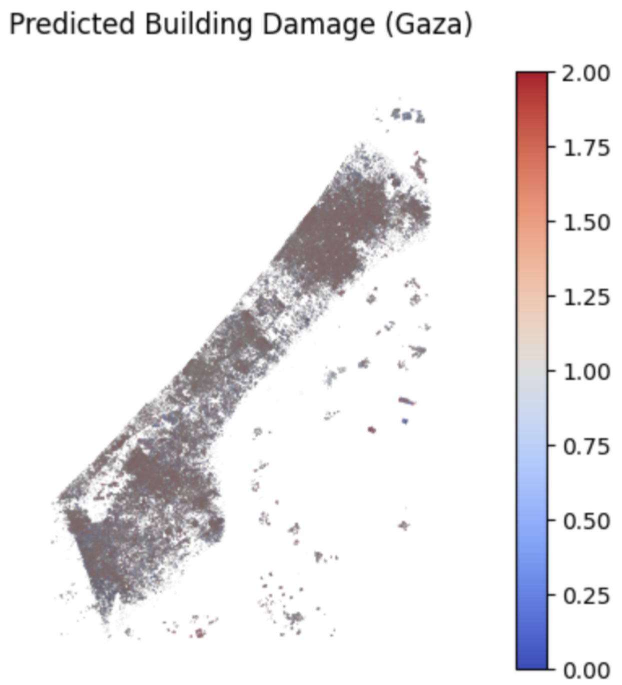

Figure 3.1: Predicted building damage distribution across Gaza.

3.2 Spatial Grid Construction and Building Aggregation

To move from building-scale predictions to neighbourhood-scale spatial insights, a uniform rectangular grid is overlaid across the region. Each cell in the grid measures 250 m × 250 m, balancing spatial resolution and computational tractability [9, 10].

The bounding box of the building dataset is extracted, and cells are generated programmatically:

```
from shapely.geometry import box
import numpy as np

def create_grid(gdf, grid_size=250):
    xmin, ymin, xmax, ymax = gdf.total_bounds
    cols = int(np.ceil((xmax - xmin) / grid_size))
    rows = int(np.ceil((ymax - ymin) / grid_size))
    polygons = [
        box(xmin + i * grid_size, ymin + j * grid_size,
            xmin + (i + 1) * grid_size, ymin + (j + 1) * grid_size)
        for i in range(cols) for j in range(rows)
    ]
    return gpd.GeoDataFrame({'geometry': polygons}, crs=gdf.crs)
```

Each building is assigned to a grid cell based on centroid containment using `GeoPandas.sjoin()`. This enables the aggregation of building-level metrics to the grid scale.

3.3 Computation of Grid-Level Destruction Metrics

For each cell, we calculate key spatial metrics including the number of buildings ($n_{\text{buildings}}$), total built-up area (A_{built}), and proportion of destroyed structures ($P_{\text{destroyed}}$). The formula for the destruction rate is:

$$P_{\text{destroyed}} = \frac{n_{\text{destroyed}}}{n_{\text{buildings}}} \tag{3.1}$$

This destruction metric supports spatial prioritization. The result is visualized using a choropleth map.

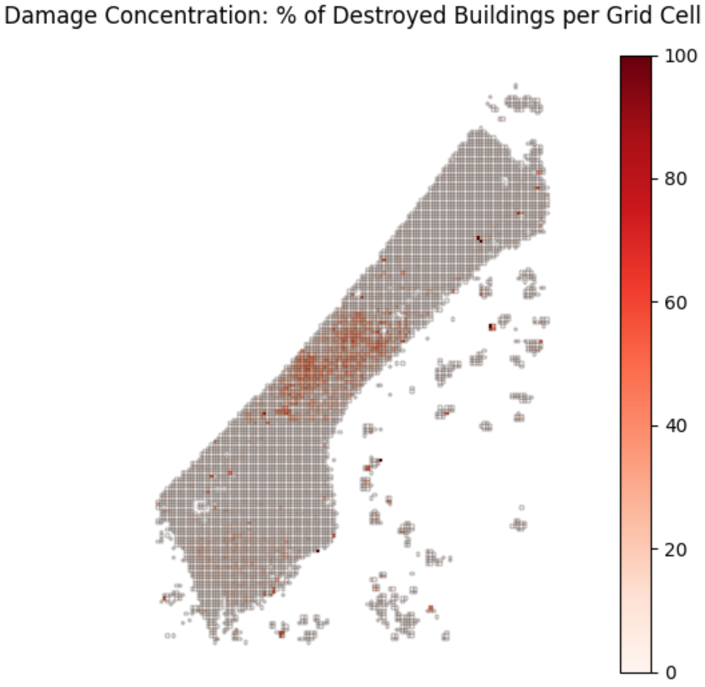
Damage Concentration: % of Destroyed Buildings per Grid Cell
100
80
60
40
20
0

Figure 3.2: Grid-level percentage of destroyed buildings.

3.4 Assignment of Reconstruction Strategy

Using the proportion of destruction in each cell, we assign one of three reconstruction strategies: `repair`, `partial_rebuild`, or `full_rebuild`. This follows the rule-based classification proposed by ESCWA and Fiol et al. [10]:

$$\begin{aligned} \text{repair:} \quad & 0 \leq P_{\text{destroyed}} < 0.2 \\ \text{partial\_rebuild:} \quad & 0.2 \leq P_{\text{destroyed}} < 0.5 \\ \text{full\_rebuild:} \quad & 0.5 \leq P_{\text{destroyed}} \leq 1.0 \end{aligned}$$

The assigned strategies are rendered spatially as shown in Figure 3.3.

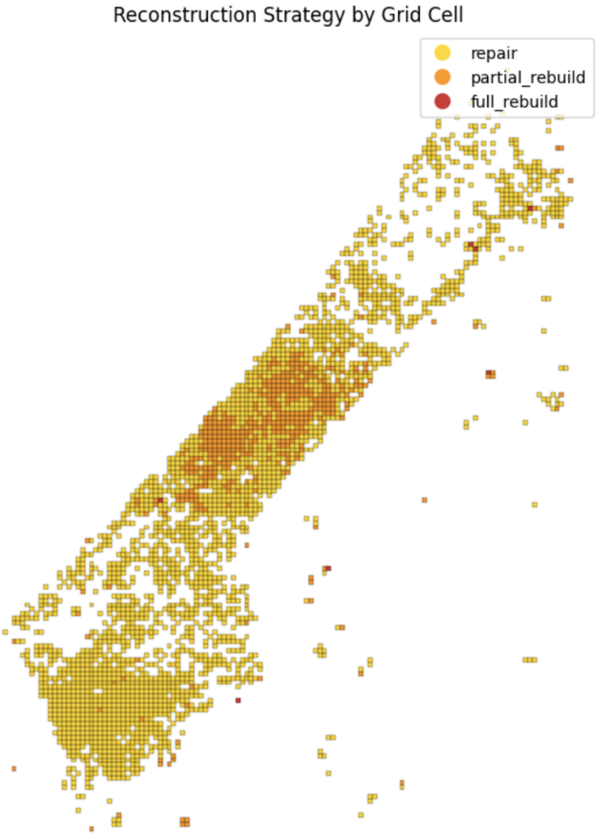

Figure 3.3: Reconstruction strategy assignment by grid cell. Yellow cells represent areas predominantly eligible for repair, while red denotes zones requiring full structural reconstruction.

3.5 Quantification of Debris and Circularity Potential

We estimate the volume and mass of debris generated by destroyed buildings using footprint area and an assumed height of 10 m for mid-rise Gaza structures. The volume is calculated as:

$$V_{\text{bldg}} = A_{\text{footprint}} \cdot h \tag{3.2}$$

For debris mass:

$$M_{\text{debris}} = V_{\text{bldg}} \cdot \rho \quad \text{where} \quad \rho = 2.1\,\text{t/m}^3 \text{ [11]} \tag{3.3}$$

To estimate usable recycled materials:

$$M_{\text{usable}} = M_{\text{debris}} \cdot 0.65 \text{ [12]} \tag{3.4}$$

A cell-level circularity score is computed as:

$$\text{circularity\_score} = \frac{M_{\text{usable}}}{M_{\text{debris}}} \in [0, 1] \tag{3.5}$$

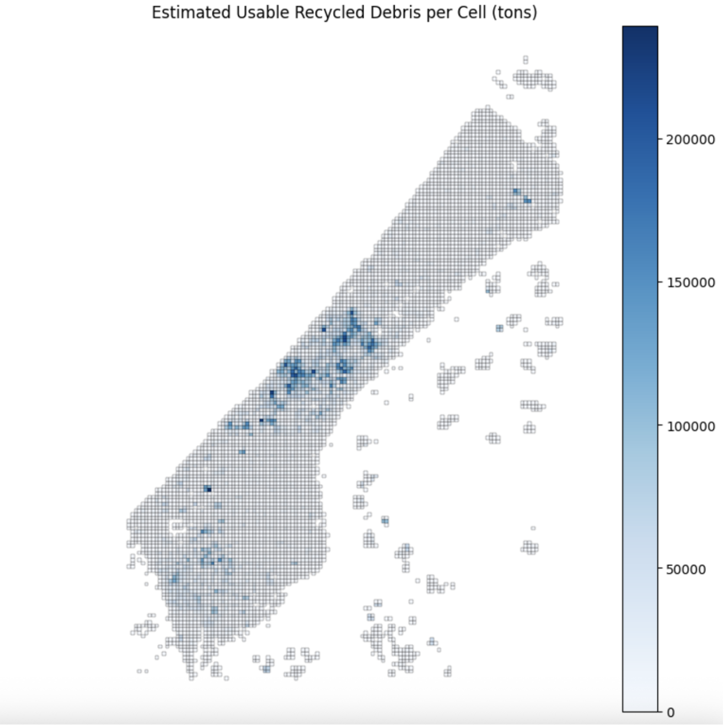

Figure 3.4: Estimated usable recycled debris per grid cell (in tons). High concentrations are co-located with destruction zones, suggesting strong potential for in-situ recycling.

3.6 Construction of the Material Lookup Table

A critical component of the reconstruction cost and carbon estimation pipeline is the `material_lookup.csv` table. This structured dataset encodes regionally calibrated values for material intensity (kg/m^2), unit cost (USD/ton), and embodied carbon (kg CO$_2$/m^2) across five key typologies and four material categories. Its development followed a multi-source triangulation methodology designed to ensure scientific rigor and regional validity.

3.6.1 Typology Framework

We adopted five dominant building typologies reflective of Gaza's built environment and broader MENA construction archetypes:

- **APT**: Apartment blocks and multi-family housing
- **SFH**: Single-family houses
- **EDU**: Educational buildings (schools, colleges)
- **COM**: Commercial and public buildings (markets, mosques)
- **IND**: Industrial and warehouse structures

These typologies were assigned based on OpenStreetMap attributes, field classification, and validated against previous reconstruction surveys and reports.

3.6.2 Material Categories

Four core material groups were included based on structural relevance and cost significance in regional construction:

- Concrete
- Steel
- Brick/block masonry
- Sand and stone (aggregates, screed, basefill)

3.6.3 Data Sources and Cross-Validation

Quantitative parameters for each `typology--material` pair were sourced from a minimum of two of the following references:

- **Akin et al. (2025)**: Life Cycle Assessment (LCA) and material flow benchmarks for Western Asia and North Africa (WANA) [13]
- **RASMI (2023)**: Empirical studies of reconstruction in the Middle East, including Palestine, Lebanon and Egypt [11]
- **Heeren and Fishman (2019)**: Global building material intensity benchmarks with MENA coverage [11]

- **Dai et al. (2024)**: UK-based intensities used for calibration of educational typologies
- **UNESCO and World Bank Guidelines (2022–2024)**: Public and institutional building material specifications

Where multiple sources were available, values were selected based on convergence (mean or mode), or, when necessary, the most conservative estimate was used to avoid underestimation. Example selection rationale includes:

- Concrete intensity for APT was set to 1150 kg/m^2 based on agreement between Akin et al., Fishman, and regional design guidelines.
- Embodied carbon for steel in IND buildings was derived from Fishman.
- Sand/stone values were determined from local Gaza and Egypt price lists referenced by World Bank procurement reports.

3.6.4 Unit Cost Standardization

Material costs were normalized to USD per metric ton, accounting for:

- Border surcharges and import constraints (notably on steel and cement)
- Gaza-Egypt market prices (2022–2024)
- Exchange rates at the time of compilation

3.6.5 Embodied Carbon Assignment

Embodied carbon intensities (kg CO_2/m^2) were selected to represent typical practices in the region, including:

- Type of cement used in concrete (e.g., blended or OPC)
- Degree of steel recycling
- Transport and energy sources based on regional grid emissions

3.6.6 Example Entry

The resulting `material_lookup.csv` table forms the backbone of the cost and carbon model described previously.

typology	material	intensity_kg	unit_cost_us	carbon_kg_p
APT	concrete	1150	85	500

Table 3.1: Example entry extracted from the `material_lookup.csv`

3.7 Building-Level Re-Attribution and Material Estimation

Each building is then enriched with its parent grid cell's attributes using spatial intersection. This includes the assigned strategy, estimated debris volumes, and circularity score. This allows cost and carbon impact calculations to be contextualized locally.

The regionally calibrated material lookup table (`material_lookup.csv`) provides values for material intensity (kg/m^2), unit cost (USD/ton), and embodied carbon (kg CO$_2$/m^2) for each typology and material [13, 11, 14].

For example, concrete material consumption is estimated as:

$$M_{\text{concrete}} = I^{T}_{\text{concrete}} \cdot A_{\text{GFA}} \quad \text{and cost} \quad C_{\text{concrete}} = \frac{M_{\text{concrete}}}{1000} \cdot C_{\text{unit}} \tag{3.6}$$

Where I^{T}_{concrete} is the intensity for typology T, and C_{unit} is the unit cost. These calculations are extended to other materials: steel, bricks, and sand/stone. Embodied carbon is modeled similarly [11].

3.8 All-In Cost Modelling and Circular Adjustments

To account for infrastructure, labour, and design overheads, an uplift is applied:

$$C_{\text{all-in}} = C_{\text{materials}} \cdot (1 + s + c) + A_{\text{GFA}} \cdot I_{\text{infra}} \tag{3.7}$$

where $s = 15\%$ (soft cost), $c = 35\%$ (contingency), and $I_{\text{infra}} = \$200/\text{m}^2$. For buildings in full_rebuild zones, a 25% deduction is applied to concrete and carbon metrics to reflect circular practices.

3.9 Summary

This methodology builds an end-to-end spatial analysis and material estimation pipeline rooted in open-source geospatial tools, empirical thresholds, and regionally adapted engineering assumptions. It supports scalable damage analytics, localized reconstruction planning, and sustainability assessments in fragile post-conflict contexts.

In the subsequent chapter, we present the results generated by this pipeline and provide visual and tabular insights to substantiate key spatial, economic, and material patterns across the Gaza Strip.

Chapter 4

Urban Layout Optimization

This chapter presents a reproducible pipeline for urban layout optimization and circular reconstruction in post-conflict Gaza. The workflow consists of five key stages:

1. Data Loading and Preprocessing
2. Feature Engineering: Typology, Elements, and Debris Circularity
3. Scenario Generation: Land Use and Typology Assignment
4. Metaheuristic Optimization: Genetic Algorithms and ML Surrogates
5. Citywide Analysis: Cost, Carbon, and Accessibility Metrics

Each step is implemented using open-source Python libraries (GeoPandas, Pandas, scikit-learn, XGBoost) and geospatial data in standard formats (GeoPackage, CSV). The following subsections provide technical detail, code snippets, and mathematical expressions for full reproducibility.

4.1 Data Loading and Preprocessing

4.1.1 Geospatial Data Sources

The pipeline uses spatial and tabular datasets extracted from prior damage mapping, field surveys, and open data repositories:

- `buildings_with_predictions.gpkg`: Buildings, geometry, predicted damage class, and materials
- `full_rebuild_blocks_carbon_cost.gpkg`: Urban blocks with carbon/cost estimates
- `park_candidate_zones_refined.gpkg`: Candidate open/green spaces

- `material_lookup.csv`: Material archetypes for the Middle East (intensity, cost, carbon)
- `reconstruction_summary_by_typology_strategy.csv`: Summary statistics by scenario

4.1.2 Data Loading (Python Example)

```
import geopandas as gpd
import pandas as pd

buildings = gpd.read_file('buildings_with_predictions.gpkg')
blocks = gpd.read_file('full_rebuild_blocks_carbon_cost.gpkg')
parks = gpd.read_file('park_candidate_zones_refined.gpkg')
lookup = pd.read_csv('material_lookup.csv')
```

4.1.3 Basic Filtering and Attribute Assignment

Buildings with valid geometry and typology are retained. Missing typologies are imputed based on use-class (e.g., `apartments` → APT).

```
map_dict = {'apartments': 'APT', 'yes': 'APT',
'house': 'SFH', 'school': 'EDU', ...}
buildings['typology'] = buildings['building'].map(map_dict).fillna('APT')
```

Floor counts are assigned using archetypal values if not directly available:

$$\text{levels}_i = \begin{cases} \text{given value} & \text{if present} \\ \text{regional default for typology} & \text{otherwise} \end{cases}$$

4.2 Feature Engineering: Typology, Elements, and Debris Circularity

4.2.1 Floor Area and Component Assignment

Each building's gross floor area is computed as:

$$\text{GFA}_i = A_{\text{footprint},i} \times \text{levels}_i \tag{4.1}$$

Window and door counts are mapped by typology, ensuring traceability for embodied material and cost.

```
window_defaults = {'APT': 8, 'SFH': 10, ...}
buildings['window_count'] = buildings['typology'].map(window_defaults).fillna(8)
```

4.2.2 Debris and Circularity Calculation

Usable recycled mass is calculated from predicted debris for each building or zone. The proportion of material needs met by recycled debris (the reuse ratio r) is capped at the available debris fraction:

$$r_i = \min(\text{RecyclingEfficiency} \times \text{debris}_i, 1.0) \tag{4.2}$$

The remaining need is met with new material.

4.2.3 Material, Cost, and Carbon Calculations

For each building, the total new and recycled material costs and embodied carbon are calculated as:

$$\text{Total Cost}_i = \text{Cost}_{\text{new},i} \times (1 - r_i) + \text{Cost}_{\text{recycled},i} \times r_i \tag{4.3}$$

$$\text{Total Carbon}_i = \text{Carbon}_{\text{new},i} \times (1 - r_i) + \text{Carbon}_{\text{recycled},i} \times r_i \tag{4.4}$$

Where recycled material is assumed to have lower unit cost and carbon intensity (e.g., -60% cost, -80% carbon vs. new).

```
def assign_reused_material(building_row, debris_col='debris_amount_proxy'):
    debris = building_row[debris_col]
    reuse_ratio = min(RECYCLING_EFFICIENCY * debris, 1.0)
    new_ratio = 1.0 - reuse_ratio
    total_material_cost = building_row['total_material_cost_usd'] * new_ratio
    total_embodied_carbon = building_row['total_embodied_carbon_kg'] * new_ratio
    recycled_cost = building_row['total_material_cost_usd'] * reuse_ratio * 0.4
    recycled_carbon = building_row['total_embodied_carbon_kg'] * reuse_ratio * 0.2
    blended_cost = total_material_cost + recycled_cost
    blended_carbon = total_embodied_carbon + recycled_carbon
    ...
    return pd.Series({...})
```

4.3 Scenario Generation: Land Use and Typology Assignment

4.3.1 Automated Typology Mapping

Building typology is mapped from both footprint area and use-class, with robust logic:

```
def auto_typology(row):
    if area < 60: return 'SFH'
    elif area < 250: return 'APT'
    elif area < 600: return 'MIDRISE'
    ...
```

Archetype mapping consolidates these into a reduced set for optimization (e.g., all mid/high-rise → APT).

4.3.2 Proportional Land Use Enforcement

District-level land use quotas (e.g., 80% residential, 2–3% education, 1% healthcare) are enforced based on demographic needs, ensuring rebuilt neighborhoods reflect actual population structure.

4.4 Metaheuristic Optimization: Genetic Algorithms and ML Surrogates

4.4.1 Candidate Generation with Constraints

For each building or block, multiple candidates are generated, varying:

- Typology and floor count
- Debris reuse ratio r
- Proximity to parks and services (walkability)
- Rebuild strategy (repair, partial/full rebuild)

4.4.2 Objective Function Formulation

Each candidate is scored with a composite objective:

$$\text{Score}_i = \text{Cost}_i + \alpha \times \text{Carbon}_i + \beta \times \text{WalkPenalty}_i \qquad (4.5)$$

Where α and β weight carbon and walkability relative to cost.

4.4.3 Genetic Algorithm Framework

A standard GA framework (selection, crossover, mutation) is used for global optimization, subject to spatial constraints (e.g., clustering penalty for services, minimum green space per capita).

4.4.4 ML Surrogate Models (XGBoost)

Surrogate models are trained to accelerate evaluation:

```
import xgboost as xgb
cost_model = xgb.XGBRegressor(...)
cost_model.fit(X_train, y_cost_train)
```

Prediction enables rapid citywide scoring of millions of candidate configurations.

4.5 Citywide Analysis: Cost, Carbon, and Accessibility Metrics

4.5.1 Urban-Scale Summaries

Aggregate metrics are computed for the optimal assignment:

- Total and per-capita cost, embodied carbon, and operational carbon
- Distribution of walk distances to nearest park or service
- Typology mix and reuse ratio histogram

4.5.2 Accessibility (Walkability) Calculation

Distance to nearest park or amenity is calculated using KD-tree or BallTree nearest-neighbor search:

```
from sklearn.neighbors import BallTree
parks_xy = np.array([geom.centroid.coords[0] for geom in parks.geometry])
building_centroids = np.array([geom.centroid.coords[0]
for geom in buildings.geometry])
parks_tree = BallTree(parks_xy, metric='euclidean')
dist, _ = parks_tree.query(building_centroids, k=1)
buildings['nearest_park_m'] = dist
```

4.5.3 Circularity and Net Zero Assessment

Total urban carbon (embodied + operational) is summed:

$$C_{\text{urban}} = \sum_i (\text{blended\_embodied\_carbon}_i + \text{operational\_carbon}_i) \tag{4.6}$$

Net zero is assessed against offset scenarios.

4.6 Output and Reproducibility

Final optimized layouts, material assignments, and scenario results are exported as GeoPackage and CSV for transparency and further analysis:

```
buildings_optimized.to_file('buildings_urban_optimized_ml_xgb.gpkg', driver='GPKG')
buildings_optimized[csv_cols_to_keep].to_csv('buildings_urban_optimized_ml_xgb.csv',
index=False)
```

4.7 Summary

This integrated methodology enables urban planners and policymakers to optimize the layout and reconstruction of Gaza or similar post-conflict cities, dynamically allocating resources for maximum social, economic, and environmental benefit. The full pipeline is reproducible, extensible, and open for adaptation to other crisis-affected urban contexts.

Chapter 5

Economic and Environmental Cost Estimation

This chapter details the integrated, automated methodology for estimating per-building and citywide reconstruction costs in Gaza and the MENA region. The pipeline combines cleaned geospatial building data, a custom-compiled archetype material cost database, regional construction cost indices, and AI-driven material take-off and debris modeling. The process is fully scripted for reproducibility, enabling rapid updates and robust scenario analysis.

5.1 Data Sources and Preprocessing

5.1.1 Building Geometry and Attribute Data

The pipeline begins with ML-optimized datasets of building footprints, heights, typologies, and damage attributes for Gaza and related MENA cities, including:

- Building polygons (GeoPackage/CSV), each with unique `id`, typology, optimized area (`area_opt`), door/window counts, and environmental/damage metrics.
- Debris and recycling fields: estimated debris mass, usable recycled tons, and reuse ratio.
- Rebuilding strategy: `"repair"`, `"partial_rebuild"`, `"full_rebuild"`, or assigned as `"full_rebuild` if missing.

5.1.2 Material and Cost Archetype Compilation

A central innovation is the custom archetype material cost table, compiled by:

1. Gathering national and regional MI and cost datasets for Gaza, UAE, Saudi Arabia, Kuwait, Bahrain, Qatar, Egypt, Jordan, and Lebanon.

2. Harmonizing typology and material definitions to a unified schema (e.g., residential: `APT`, `SFH`, non-residential: `COM`, etc.; materials: `cement`, `steel`, `glass`, etc.).

3. Extracting MI (Material Intensity, tons/m^2), local and regional unit prices (USD/-ton), inflation-adjusted prices, and source/markup data from PDFs, government tables, and published literature (RASMI [11], Haberl [9], Akin [15], Dai [14], World Bank).

4. Averaging or range-normalizing MI and cost values for each (country, typology, material) combination; computing markups based on documented post-2023 Gaza market conditions.

5. Merging all rows into a master archetype table (`archetype_material_costs_master_full.csv`), retaining fields: `typology`, `material`, `mi_ton_per_m2`, `unit_price_usd_per_ton`, `markup`, `price_usd_per_ton_inflated`, and `source`.

Code Example: Aggregating Archetypes (Python)

```
import pandas as pd
archetypes = pd.read_csv('archetype_material_costs_master_full.csv')
# Group by typology and material, compute rounded mean MI and unit prices
typ_mat = archetypes.groupby(['typology', 'material']).agg({
    'mi_ton_per_m2': 'mean',
    'unit_price_usd_per_ton': 'mean',
    'markup': 'mean',
    'price_usd_per_ton_inflated': 'mean'
}).reset_index()
```

This robust, regionally-calibrated table forms the backbone of all material take-off and cost computations.

5.2 Data Cleaning and Building Attribute Harmonization

To ensure one unique record per building geometry, the following deduplication logic is applied:

- Group by `id`; if multiple geometries, assign suffixes; for true duplicates, retain the row with the least missing data.

- Standardize fields: set `area` from `area_opt` if present, else `area`.
- Fill missing `door_count`, `window_count`, `recycled_tons`, and `reuse_ratio` as zero.
- Assign `"full_rebuild"` to any building missing a `rebuild_strategy`.

5.3 Material Take-Off and Debris Recycling Adjustment

5.3.1 Material Take-Off Calculation

For each building and material, required tons are calculated as:

$$\text{Tons}_{\text{needed}} = \text{area} \times \text{MI}_{\text{ton/m}^2} \tag{5.1}$$

where `area` is in m^2, and MI is the archetype value for that typology and material.

5.3.2 Debris Recycling Adjustment

Net new material demand is then:

$$\text{Net New Tons} = \max(\text{Tons}_{\text{needed}} - \text{recycled\_tons}, 0) \tag{5.2}$$

Debris adjustment is only applied if the building's `reuse_ratio` > 0.

5.3.3 Material Take-Off and Debris Integration

```
for mat in materials:
    mi = buildings['typology'].apply(lambda typ:
    typ_mi_dict.get((typ, mat), 0))
    tons_needed = buildings['area'] * mi
    recycled = np.where(buildings['reuse_ratio'] > 0,
    buildings['recycled_tons'], 0)
    net_new_tons = np.maximum(tons_needed - recycled, 0)
    buildings[f"{mat}_tons"] = net_new_tons
```

5.4 Material and Component Cost Calculation

5.4.1 Per-Material Costs

For each material and building:

$$\text{Cost}_{\text{pre}} = \text{Net New Tons} \times \text{Unit Price}_{\text{USD/ton}} \quad (5.3)$$

$$\text{Cost}_{\text{infl}} = \text{Net New Tons} \times \text{Inflated Price}_{\text{USD/ton}} \quad (5.4)$$

Where prices and markups are from the archetype table, based on the latest Gaza/post-conflict market benchmarks.

5.4.2 Doors and Windows

Counts of doors/windows are drawn from building attributes (`door_count`, `window_count`). Gaza-specific market prices and markups are applied:

$$\text{Door Unit Cost (post-2023, inflated)} = \$150 \times 5 = \$750/\text{door}$$

$$\text{Window Unit Cost (post-2023, inflated)} = \$90 \times 6 = \$540/\text{window}$$

5.4.3 Total Building Cost

Total material cost (pre- and post-inflation):

$$\text{Total Material Cost}_{\text{pre/infl}} = \sum_{\text{material}} \text{Cost}_{\text{pre/infl}} \quad (5.5)$$

Total building cost (pre- and post-inflation):

$$\text{Total Cost}_{\text{pre/infl}} = \text{Total Material Cost}_{\text{pre/infl}} + \text{Door Cost}_{\text{pre/infl}} + \text{Window Cost}_{\text{pre/infl}} \quad (5.6)$$

5.4.4 Total Cost Calculation

```
buildings['door_cost_pre'] = buildings['door_count'] * 150
buildings['door_cost_infl'] = buildings['door_count'] * 150 * 5
buildings['window_cost_pre'] = buildings['window_count'] * 90
buildings['window_cost_infl'] = buildings['window_count'] * 90 * 6
mat_pre = [f"{m}_cost_pre" for m in materials]
mat_infl = [f"{m}_cost_infl" for m in materials]
buildings['total_material_cost_pre'] = buildings[mat_pre].sum(axis=1)
```

```
buildings['total_material_cost_infl'] = buildings[mat_infl].sum(axis=1)
buildings['total_cost_pre'] = buildings['total_material_cost_pre']
+ buildings['door_cost_pre'] +
buildings['window_cost_pre']
buildings['total_cost_infl'] = buildings['total_material_cost_infl'] +
buildings['door_cost_infl'] +
buildings['window_cost_infl']
```

5.5 Labor Cost Integration

Labor costs are integrated using the most recent Construction Cost Index [16] data:

$$\text{Labour Cost (NIS)} = \text{area} \times \text{Labour Cost Index (NIS/m}^2) \tag{5.7}$$

$$\text{Labour Cost (USD)} = \text{Labour Cost (NIS)} \times \text{NIS to USD conversion rate} \tag{5.8}$$

Alternatively, labor can be applied as a percentage of total material cost (typical: 40–60%, per World Bank). Final total cost:

$$\text{Total Cost with Labour} = \text{Total Cost}_{\text{pre/infl}} + \text{Labour Cost (USD)} \tag{5.9}$$

5.6 Output Generation and Aggregation

The pipeline outputs:

- Per-building CSV with all costs, attributes, and geometry ID.
- GeoPackage and GeoJSON for GIS visualization.
- Per-typology and per-strategy summaries: total new/recycled tons, costs, and recycled share.
- Scenario analysis tables, showing impact of recycling, inflation, and labour on cost structures.

5.6.1 Typology/Strategy Aggregation

```
summary = []
for (typ, strat) in buildings.groupby(['typology', 'rebuild_strategy']).groups:
    bldgs_typ = buildings[(buildings['typology'] == typ)
    & (buildings['rebuild_strategy'] == strat)]
    for mat in materials:
        tons_col = f"{mat}_tons"
        pre_col = f"{mat}_cost_pre"
        infl_col = f"{mat}_cost_infl"
        summary.append({
            'typology': typ,
            'rebuild_strategy': strat,
            'material': mat,
            'total_new_material_tons': bldgs_typ[tons_col].sum(),
            'total_recycled_material_tons': bldgs_typ['recycled_tons'].sum(),
            'cost_pre': bldgs_typ[pre_col].sum(),
            'cost_infl': bldgs_typ[infl_col].sum()
        })
summary_df = pd.DataFrame(summary)
```

5.7 Reproducibility and Scenario Analysis

The pipeline is fully scripted (Python/Pandas/NumPy), and all parameters (MI, unit prices, inflation rates, labour indices) can be updated as new data emerges. Scenario analysis (e.g., testing different recycling rates, inflation, labour markups) is performed by adjusting relevant columns and re-running the aggregation scripts.

5.8 Summary

This methodology combines regional MI/cost archetypes, AI-driven building attribute extraction, robust data cleaning, and detailed material and cost modelling with reproducible code and outputs. It directly enables transparent, scenario-ready cost estimation for large-scale, circular reconstruction planning in Gaza and the MENA region.

Bibliography

[1] UNOSAT, "Unosat gaza strip comprehensive damage assessment," 2025.

[2] O. S. Map, "Gaza update 2024 - openstreetmap wiki," 2024.

[3] P. labs PBC, "Planet application program interface: In space for life on earth," 2025.

[4] C. D. S. Ecosystem, "Copernicus data space ecosystem," 2025.

[5] QGIS.org, "Qgis geographic information system," 2025.

[6] F. Kahraman, M. İmamoğlu, and H. F. Ates, "Battle damage assessment based on self-similarity and contextual modeling of buildings in dense urban areas," 07 2016.

[7] Y. Alzubi, A. Aljaafreh, and A. Khatatbeh, "Application of machine learning techniques in estimating the construction cost of residential buildings in the middle east region," *International Journal of Construction Management*, pp. 1–13, 07 2023.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 06 2002.

[9] H. Haberl, A. Baumgart, J. Zeidler, F. Schug, D. Frantz, D. Palacios-Lopez, T. Fishman, Y. Peled, B. Cai, D. Virág, P. Hostert, D. Wiedenhofer, and T. Esch, "Weighing the global built environment: High-resolution mapping and quantification of material stocks in buildings," *Journal of Industrial Ecology*, 12 2024.

[10] M. Fiol, M. Abualhayja, W. Aboul Hosn, and J. Van Den Hoek, "Assessment of physical damage caused to buildings by the war on gaza," 2023.

[11] T. Fishman, A. Mastrucci, Y. Peled, S. Saxe, and B. van Ruijven, "Rasmi: Global ranges of building material intensities differentiated by region, structure, and function," *Scientific data*, vol. 11, 04 2024.

[12] E. B. of the United Nations Human Settlements Programme, "Preliminary report on the status of the development of the efforts to reconstruct the human settlements in the gaza strip," 2024.

[13] S. Akin, A. Eghbali, C. Nwagwu, and E. Hertwich, "Western asian and northern african residential building stocks: archetype analysis," *Buildings and Cities*, vol. 6, 05 2025.

[14] M. Dai, J. Jurszyk, C. Gillott, K. Sun, M. Lanau, G. Liu, and D. D. Tingley, "Modeling interior component stocks of uk housing using exterior features and machine learning techniques," *Journal of Industrial Ecology*, 06 2025.

[15] S. Akin, C. Chrysogonus Nwagwu, N. Heeren, and E. Hertwich, "Archetype-based energy and material use estimation for the residential buildings in arab gulf countries," *Energy and Buildings*, vol. 298, p. 113537, 11 2023.

[16] P. B. of Statistics, "Pcbs — monthly construction cost indices and percent changes by major groups in the west bank for the years 2024 and 2023," 2024.