# 1 — Refined project title & short abstract

**Title:** Assessment of Urban Flood Risk in Ibadan Using Convolutional Neural Networks and Sentinel Imagery: A Comparative Study with NDWI-based Methods

**Abstract (1 paragraph):**  
This study develops and evaluates a Convolutional Neural Network (CNN) semantic-segmentation pipeline to map flood-prone areas in Ibadan from Sentinel-2 (and Sentinel-1 where available) imagery and auxiliary environmental layers (DEM-derived indices, urban density, land cover). The CNN’s performance (IoU, F1, accuracy) will be compared against a classical NDWI thresholding baseline. Feature-importance and ablation studies (Grad-CAM, permutation/occlusion) will identify the most influential environmental predictors. Model outputs will be validated against historical flood extents (notably the 2011 and 2023 Olodo events). All code, models, and processed data products will be published openly to promote reproducibility.

# 2 — Novelty & contribution (so it’s publishable)

* Combine **Sentinel-2 optical bands + DEM & urban metrics** in a CNN segmentation framework specifically focused on Ibadan (few papers target this city in depth).
* Rigorous **spatial cross-validation** (tile-based) to avoid leakage — many studies do random splits.
* Head-to-head **statistical comparison** (paired tests & bootstrapping) of CNN vs NDWI on the same held-out tiles and event dates.
* **Explainability**: Grad-CAM + permutation occlusion to rank environmental features for practical flood-management insights (useful for civil engineers and local planners).
* Public release of data-processing scripts & trained weights for reproducibility.

# 3 — Data you must collect (minimum + recommended)

Minimum (required):

* Sentinel-2 MSI Level-2A (surface reflectance) covering Ibadan for flood & non-flood dates (bands B2,B3,B4,B8,B11,B12).
* SRTM DEM (30 m) — compute elevation, slope, TWI.
* Administrative boundary of Ibadan (OSM or local shapefile).
* Historical flood extents / inventory for validation: official reports, shapefiles, and remote detection for the 2011 and 2023 Olodo floods.

Recommended (strongly improves model & publication quality):

* Sentinel-1 SAR scenes for the flood events (helps derive flood masks under clouds).
* Land use / land cover maps (ESA WorldCover or manually derived).
* Built-up / urban density (derived from classification or OSM building footprints).
* Rainfall time series (CHIRPS, ERA5, or local station data) for temporal context.

# 4 — Ground truth / label creation

1. **Event-based masks:** For each target flood event date (e.g., 2011, 2023), derive water masks:
   * Prefer Sentinel-1 (radar) if available — threshold backscatter to separate water; postprocess with morphological cleaning.
   * If using Sentinel-2, compute NDWI / MNDWI and threshold (+ manual correction), but guard against clouds / shadows.
2. **Manual digitization:** Use Google Earth or high-res imagery to correct/validate masks in critical zones (Olodo).
3. **Label quality:** Exclude ambiguous pixels (edge buffers) to reduce label noise.
4. **Splits:** Split by spatial tiles (not random pixels). Reserve at least 20% of area as held-out test; use 4-fold spatial CV on the remaining.

# 5 — Model(s) & baselines

Primary models:

* **U-Net** with ResNet34 encoder (transfer learning) — baseline segmentation.
* **DeepLabv3+** with a strong encoder (ResNet50/EfficientNet) — higher capacity.

Loss & optimizer:

* Loss: **BCE + Dice (or Jaccard)** to handle class imbalance.
* Optimizer: **Adam** (lr 1e-4 start) with ReduceLROnPlateau and early stopping.

Baselines:

* **NDWI / MNDWI thresholding** (Otsu or tuned threshold on validation tiles).
* Optional: XGBoost or RF using per-pixel features (bands + indices + DEM features) as a non-spatial ML baseline.

# 6 — Input features to test (experiment groups)

* Spectral bands only (core).
* Bands + spectral indices (NDWI, MNDWI, NDVI, NDBI).
* Bands + indices + DEM derivatives (elevation, slope, TWI).
* All of the above + urban density / built-up fraction + distance-to-river.

# 7 — Evaluation metrics & statistical tests (publication quality)

Primary segmentation metrics:

* **IoU (Jaccard)** — primary.
* **F1 (Dice)**, **Precision**, **Recall**, **Accuracy**.

Area/object metrics:

* Flood area difference (%), boundary Hausdorff distance (if polygons available).

Statistical testing:

* Compute per-tile IoU for CNN and NDWI. Use **paired Wilcoxon signed-rank test** (non-parametric) or paired t-test if normality holds to test if CNN outperforms NDWI.
* **Bootstrap** 95% CIs on mean IoU and F1 (resample tiles).
* Report effect sizes (e.g., mean IoU difference + CI).

Significance & robustness:

* Run ablation: remove feature groups and report metric drop with CIs.
* Perform sensitivity to NDWI threshold (report best and worst reasonable thresholds).

# 8 — Explainability & feature importance

* **Grad-CAM / Grad-CAM++** on encoder features to show which spatial regions drive predictions.
* **Permutation importance / occlusion**: zero out (or shuffle) each auxiliary raster globally or patchwise and measure metric change. This quantifies the importance of TWI, urban density, etc.
* Optional: train a surrogate tree-based model on patch-level CNN outputs to get interpretable feature importances.

# 9 — Validation with Olodo 2011 & 2023 floods

* Collect Sentinel-1 for each event date (preferred) and/or best-available optical imagery.
* Generate event masks (as above).
* Compare model predictions for those exact dates (temporal alignment important): compute per-event IoU, F1, area overlap and produce visual overlays for Olodo.
* Provide case-study writeups: show false positives/negatives and possible reasons (e.g., drainage, built environment changes since 2011).

# 10 — Reproducible workflow & software

* **Core stack:** Python, PyTorch (or TensorFlow/Keras), rasterio, geopandas, scikit-learn, albumentations, segmentation-models-pytorch (optional).
* **Data ops:** Use Google Earth Engine (GEE) for bulk Sentinel downloads/composites OR use SentinelHub / Copernicus APIs. Document exact queries and export parameters.
* **Experiment tracking:** Weights & Biases or MLflow for metrics & artifacts.
* **Version control:** GitHub repo with code, environment.yml, and README. Include example commands to reproduce training and evaluation.

Suggested repo structure:

/project-root

├─ data/ # raw and small processed samples (NOT huge arrays)

├─ src/

│ ├─ preprocessing.py

│ ├─ features.py

│ ├─ dataset.py

│ ├─ models.py

│ ├─ train.py

│ ├─ evaluate.py

│ └─ utils.py

├─ notebooks/ # EDA and demo notebooks

├─ experiments/ # saved checkpoints + config files

├─ results/ # maps, plots, metrics

├─ environment.yml

└─ README.md

Include a run.sh or Makefile that executes the full pipeline on a small sample.

# 11 — Writing & structuring the thesis / paper

Thesis chapters:

1. Introduction (motivation, research questions)
2. Literature review (highlight gaps and how you differ)
3. Study area & data (Ibadan — describe climate, Olodo flood events, data sources)
4. Methods — preprocessing, feature engineering, labeling, model architectures, training details
5. Experiments — ablation, baselines, CV, statistical tests
6. Results — metrics, maps, case studies (Olodo), explainability
7. Discussion — interpretation, limitations, transferability to other cities
8. Conclusion & recommendations (policy, future work)
9. Appendices — code snippets, hyperparameters, extra figures

Paper checklist for submission:

* Clear statements of novelty & limitations.
* Reproducibility: public repo + DOI (Zenodo) for code and small processed dataset.
* Include uncertainty quantification (CIs, bootstrap).
* At least 3 high-quality figures: ROC/PR, per-tile IoU boxplots vs NDWI, maps for Olodo case study.

Suggested target journals/conferences (good fits): Remote Sensing, International Journal of Applied Earth Observation and Geoinformation, Natural Hazards, Journal of Hydrology, or regional conferences. (Pick one that matches length and aim.)

# 12 — Ethical & practical considerations

* Avoid revealing precise private addresses or individual-level impacts.
* Provide uncertainty and disclaimers for decision-making (don’t claim exact flood depths unless hydraulic modelling supports it).
* Seek local data / stakeholder feedback if possible — include in limitations if not.

# 13 — Risk mitigation & common pitfalls

* **Clouds & seasonal mismatch:** use Sentinel-1 or carefully select cloud-free composites.
* **Label noise:** keep only high-confidence pixels for training; document label generation steps.
* **Spatial leakage:** enforce tile-based splits.
* **Class imbalance:** use Dice loss, oversample flooded patches, monitor false positives.

# 14 — Detailed 16-week schedule (deliverable-oriented)

This schedule assumes a single semester ~16 weeks. Each week lists main tasks and deliverables.

**Week 0 (prep) — before term starts**

* Set up GitHub repo, conda env, cloud/GEE access.  
  Deliverable: repo skeleton, environment.yml.

**Week 1 — Study area & literature**

* Finalize study area polygon for Ibadan; collect literature and write intro + research gap.  
  Deliverable: Intro + lit review draft (2 pages).

**Week 2 — Data collection**

* Download Sentinel-2 (several dates), DEM, OSM; locate flood reports for 2011 & 2023.  
  Deliverable: Raw data inventory + sample images.

**Week 3 — Preprocessing**

* Cloud mask, resample bands to common resolution, compute NDWI, MNDWI, NDVI. Compute DEM derivatives (slope, TWI).  
  Deliverable: Processed raster stack for a sample tile.

**Week 4 — Label creation**

* Produce flood masks for target events (2011 if possible, 2023). Manual correction of a few tiles (Olodo).  
  Deliverable: Binary masks for at least 2 events.

**Week 5 — Dataset & patches**

* Tile the study area (256×256 or 512×512), create train/val/test split (spatial), implement PyTorch Dataset.  
  Deliverable: Patch dataset ready.

**Week 6 — Baseline model**

* Implement U-Net baseline (bands only), minimal training loop, simple augmentation.  
  Deliverable: First trained model + validation metrics.

**Week 7 — Enhanced inputs**

* Add spectral indices + DEM inputs, retrain U-Net.  
  Deliverable: Model trained with indices + DEM.

**Week 8 — Stronger models & tuning**

* Train transfer U-Net (ResNet encoder) & DeepLabv3+ (if compute allows). Hyperparameter tuning.  
  Deliverable: Best model checkpoint.

**Week 9 — NDWI baseline & comparison**

* Implement NDWI thresholding baseline tuned on val tiles. Compute metrics for NDWI and CNN on validation.  
  Deliverable: Baseline results table.

**Week 10 — Spatial cross-validation & statistics**

* Run 4-fold spatial CV for main model variants. Compute per-tile IoU, bootstrap CIs.  
  Deliverable: CV results & stats.

**Week 11 — Ablation & feature importance**

* Remove feature groups (indices, DEM, urban) and measure metric drops. Run permutation/occlusion importance.  
  Deliverable: Ablation results & plots.

**Week 12 — Explainability + case studies**

* Grad-CAM for selected tiles; prepare Olodo 2011 & 2023 case studies (maps + error analysis).  
  Deliverable: Olodo case study figures + explanations.

**Week 13 — Final evaluations & robustness**

* Sensitivity analyses (NDWI thresholds, different tile sizes), area difference metrics, uncertainty estimates.  
  Deliverable: Robustness section results.

**Week 14 — Write results & discussion**

* Compile results, write Results & Discussion chapters. Polish figures & tables.  
  Deliverable: Draft Results + Discussion.

**Week 15 — Final write up**

* Complete Introduction, Methods, Conclusion. Prepare slides. Clean repo, create README and DOI deposit (Zenodo).  
  Deliverable: Full thesis draft + presentation slides.

**Week 16 — Submission & polish**

* Proofreading, references, produce final PDF, prepare GitHub release. Submit and prepare for viva.  
  Deliverable: Final thesis + GitHub link.

# 15 — Immediate next steps (what you can do right now — day 1 checklist)

1. Create GitHub repo and push skeleton (use structure above).
2. Acquire and clip the Ibadan boundary (OSM) and download one Sentinel-2 composite for a recent wet season date.
3. Compute NDWI and a sample flood mask for one event (use Sentinel-1 if available).
4. Implement the PyTorch Dataset that loads raster patches and masks.
5. Run a quick U-Net training on a small subset to ensure pipeline works.  
   If you want, I can now generate:

* a runnable **Google Earth Engine script** to export Sentinel-2 composites and NDWI for Ibadan; **or**
* a **starter PyTorch notebook** (U-Net + data loader + BCE+Dice loss + sample training loop) ready to run on Colab/GPU.