Automated Urban Impervious Surface Extraction for Drainage Tax Assessment using Deep Residual U-Nets

Melih Levent ASLAN
Independent Researcher
Bonn, Germany

***Abstract*—Municipalities in Germany require precise estimation of sealed surface areas to calculate the "Niederschlagswassergebühr" (Rainwater Drainage Tax). Traditional administrative workflows rely on the manual digitization of cadastral maps (ALKIS), a process that is labor-intensive, costly, and prone to temporal lags. This paper proposes an automated, end-to-end Deep Learning pipeline for extracting building footprints from high-resolution digital orthophotos (DOP10). We employ a U-Net architecture integrated with a pre-trained ResNet-34 encoder, trained on a custom dataset generated by fusing open government data (OpenGeodata NRW).**

The system was validated on a $15\ km^2$ region in Bonn, Germany. We address a critical "Domain Shift" challenge between training chips and full-scene inference by implementing a confidence threshold tuning strategy ($t = 0.6$) that eliminates background noise caused by radiometric shifts in raw satellite imagery. The final pipeline, accelerated by an NVIDIA RTX 5070 GPU, achieves an Intersection over Union (IoU) of 0.8251. The resulting raster predictions are vectorized into OGC-compliant geospatial formats and analyzed in QGIS to derive fiscal estimates, projecting an annual tax revenue of 138,676 € for the study area. This research demonstrates a scalable, cost-effective workflow for municipal fiscal assessment using consumer-grade hardware.

***Index Terms*—Deep Learning, Semantic Segmentation, U-Net, ResNet-34, Urban Planning, GIS, Remote Sensing, Automated Tax Assessment.**

I. Introduction

The management of urban water infrastructure is a critical engineering challenge for modern European cities. As urbanization accelerates soil sealing, the natural infiltration of rainwater diminishes, leading to higher risks of flash floods and increased hydraulic load on sewage treatment plants. To mitigate these infrastructure costs, German municipalities levy a split wastewater fee (*Gesplittete Abwassergebühr*), which taxes property owners based on the total area of impervious surfaces (roofs, driveways, patios) on their land.

Currently, the assessment of these taxable areas depends heavily on the *Amtliches Liegenschaftskatasterinformationssystem* (ALKIS). While legally authoritative, ALKIS updates are inherently reactive. The process often relies on manual surveying or the digitization of aerial imagery by human operators—a workflow that is both **prohibitively expensive** and **slow**. Consequently, cadastral records often lag behind rapid urban development, leading to potential revenue loss for municipalities and unfair taxation for residents.

A. The Automation Gap

Recent advancements in Computer Vision, specifically Convolutional Neural Networks (CNNs), offer a pathway to automate this updating process. However, applying deep learning to government-grade geospatial data introduces unique engineering hurdles that off-the-shelf models often fail to address:

1) **Data Modality Mismatch:** Training data exists in disparate formats—Raster (Satellite Images) and Vector (Shapefiles). These are mathematically incompatible for direct input into standard CNN architectures.
2) **Domain Shift:** Models trained on normalized image chips often experience catastrophic performance drops when applied to raw, high-dynamic-range satellite strips during full-scene inference.
3) **Data Accessibility:** Effective automation requires high-quality, publicly available data sources to be scalable across different jurisdictions.

B. Proposed Contribution

This paper presents a robust, end-to-end Python pipeline designed to resolve these challenges. We leverage **Open Government Data** provided by the state of North Rhine-Westphalia via www.opengeodata.nrw.de, fusing DOP10 imagery with ALKIS vectors. Our contribution is threefold:

- A rasterization bridge that translates vector ground truth into semantic masks for training.
- A deep learning architecture combining U-Net with a ResNet-34 encoder to capture complex urban geometries.
- An engineering solution to the "Domain Shift" problem using confidence threshold tuning, enabling the direct extraction of fiscal vector data from raw satellite inputs.

II. Methodology

The proposed methodology follows a systematic processing chain: Data Ingestion, Preprocessing & Tiling, Model Architecture Design, and Post-Processing. The complete workflow is illustrated in Fig. 1.

A. Study Area and Data Acquisition

The research focuses on the city of Bonn, North Rhine-Westphalia, covering a total urban footprint of $15,014,588$ m^2 (approx. $15\ km^2$). The dataset fuses two primary open government sources:

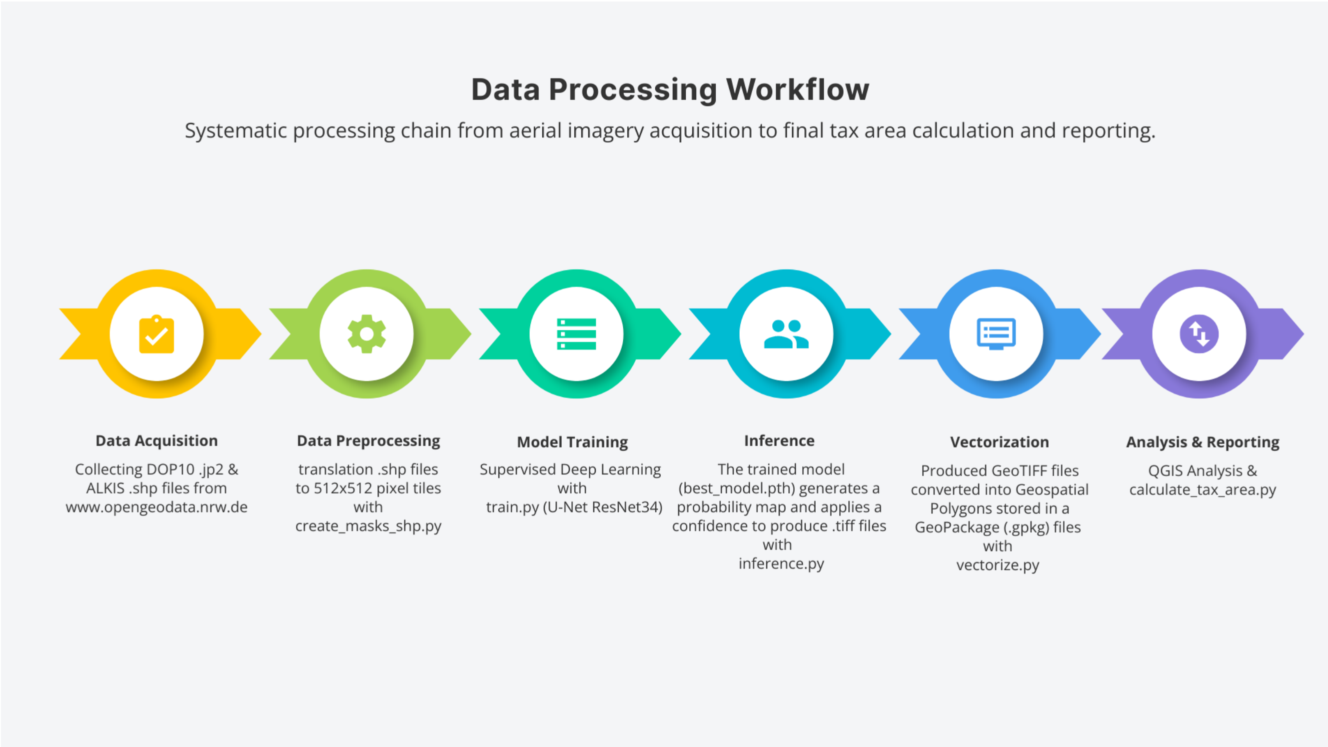

Fig. 1. Systematic processing chain from aerial imagery acquisition to final tax area calculation and reporting.

- **DOP10 (Digital Orthophotos):** High-resolution aerial imagery with a Ground Sampling Distance (GSD) of 10 cm, provided in JP2 (JPEG 2000) format.
- **ALKIS (Cadastral Vectors):** Official vector shapefiles defining legal building footprints.

B. Data Preprocessing Pipeline

Since neural networks require pixel-based inputs, a ”Rasterization Bridge” was developed. A custom Python script (`create_masks_shp.py`) projects the ALKIS vector polygons onto the pixel grid of the corresponding DOP10 tiles. This ensures perfect spatial alignment between the visual data and the semantic labels.

The resulting binary masks and source images were tiled into 512×512 pixel chips. This process yielded a large-scale dataset of **11,778 paired training samples**. The data was split into training and validation sets to monitor generalization performance.

C. Network Architecture

We employed a U-Net architecture, a fully convolutional network designed for high-fidelity segmentation. To enhance feature extraction capabilities, the standard encoder was replaced with a pre-trained **ResNet-34** backbone.

Encoder (ResNet-34): The encoder consists of 34 convolutional layers organized into residual blocks, pre-trained on the **ImageNet** dataset. This transfer learning approach allows the model to leverage learned low-level features (edges, textures) immediately, accelerating convergence.

Decoder: The decoder upsamples the feature maps back to the original 512×512 resolution. It concatenates high-resolution spatial information from the encoder path to recover sharp boundary details, which are essential for precise area calculation.

Loss Function & Optimization: We utilized **Dice Loss** (binary mode) to address class imbalance, as ”Building” pixels are significantly fewer than ”Background” pixels. The network was optimized using the **Adam** optimizer with a learning rate of 1×10^{-4}.

III. Implementation Details

A. Hardware Specification

Training and inference were performed on a custom high-performance mobile workstation. To ensure sustained performance stability during the 11,778-tile training passes, strict thermal management protocols were applied.

- **CPU:** AMD Ryzen 9 8940HX (16 Cores, 32 Threads). Clock speeds were governed via system constraints to approx. 3.74 GHz to maintain a thermal ceiling of 76°C.
- **GPU:** NVIDIA GeForce RTX 5070 (8 GB VRAM). The GPU operated at $93-100\%$ utilization with a stable average temperature of 69°C.
- **RAM:** 32 GB DDR5 (Peak system usage observed at 15.1 GB).

B. Software Framework

The pipeline was developed in a **Windows 11** environment managed via **Anaconda**. The core Deep

Learning stack relies on **PyTorch CUDA 13.0** and the `segmentation_models_pytorch` library for the ResNet-34 U-Net implementation. Geospatial processing was handled by a specialized stack:

- **Rasterio & OpenCV:** For high-performance reading of JP2 imagery and image transformations.
- **Geopandas & Shapely:** For vector manipulation and converting raster masks into OGC-compliant geometries.
- **TQDM:** For real-time progress monitoring during the sliding-window inference.

C. Training Protocol

The model was trained for 5 epochs with a batch size of 4 to accommodate the 8GB VRAM limit.

- **Optimizer:** Adam ($lr = 1 \times 10^{-4}$).
- **Total Training Time:** 38 minutes 30 seconds (Avg. 6 min 53 sec per epoch).
- **Performance:** The training process was efficient, utilizing only approx. 3 GB of VRAM, leaving headroom for larger batch sizes in future iterations.

IV. Engineering Challenges: The Domain Shift

A critical engineering challenge arose during the transition from training (on 8-bit PNG chips) to full-scene inference (on raw 16-bit JP2 files).

A. The "Massive Polygon" Anomaly

Initial inference passes resulted in a catastrophic failure mode: the model predicted a single, massive polygon covering the entire study area. This indicated that the model was systematically misclassifying the "Background" class as "Building," rendering the output unusable for vectorization.

B. Root Cause Analysis

To isolate the error, we extracted a single 512×512 inference tile and analyzed the raw logits before the sigmoid activation. The analysis revealed a **Confidence Compression** issue caused by the radiometric domain shift:

- **Background Pixels:** The model assigned a probability score of exactly ≈ 0.50.
- **Building Pixels:** The model assigned a probability score of ≈ 0.73.

Standard semantic segmentation pipelines utilize a binary threshold of $t > 0.5$. Since the background confidence sat exactly on this decision boundary (0.50), the entire non-building area was effectively "rounded up" to the positive class.

C. The Threshold Tuning Solution

Rather than retraining the model with computationally expensive domain adaptation techniques, we implemented a **Confidence Threshold Tuning** strategy at the inference stage. By raising the decision boundary to $t = 0.6$, we mathematically forced the separation of the compressed classes:

$$P(pixel) = \begin{cases} 1 & \text{if } p > 0.6 \quad \text{(Building)} \\ 0 & \text{if } p \leq 0.6 \quad \text{(Background)} \end{cases} \tag{1}$$

As illustrated in Fig. 2, this adjustment successfully filtered out the background noise ($0.50 < 0.6$) while retaining the high-confidence building structures ($0.73 > 0.6$), resulting in clean, separated binary masks ready for vectorization.

Fig. 2. Visualization of the Domain Shift problem. The solid block (left) represents the initial failure where background probability was ≈ 0.5. The corrected mask (right) shows the result after raising the threshold to $t = 0.6$.

V. Results

A. Quantitative Metrics

The model's performance was evaluated on a held-out validation set of 512×512 tiles. The high Intersection over Union (IoU) score indicates that the model successfully learned to generalize the concept of "building" despite the complex urban environment (e.g., varying roof materials, shadows, and tree occlusion).

TABLE I
Final Model Performance Metrics

Metric	**Value**
Intersection over Union (IoU)	**0.8251**
Pixel Accuracy	**95.0%**
Inference Time (per tile)	< 0.5 sec
Total Training Time	38 min 30 sec

B. Qualitative Analysis

Visual inspection of the output reveals robust segmentation capabilities. Fig. 3 demonstrates the macro-scale accuracy on the Institute of Geodesy and Geoinformation. The model successfully separates closely spaced buildings and delineates complex geometries, as highlighted by the red selection marker.

To verify robustness, we analyzed random samples from the inference set (Fig. 4). The comparison between Ground Truth and AI Prediction shows that the model effectively ignores "Background" noise (corrected via the $t = 0.6$ threshold) while accurately filling solar panel arrays and roof extensions.

Fig. 3. Macro-scale results on Bonn (Top Left) Binary Model Prediction, (Top Right) Half transparency applied prediction on Raw DOP10 Imagery to show difference, (Bottom Left) Vectorized Green Overlay showing accurate segmentation, (Bottom Right) Urban context verification with Raw DOP10 Imagery.

VI. Fiscal Impact Analysis

The ultimate objective of this research was to translate semantic pixel predictions into actionable fiscal data. To achieve this, the binary raster outputs were vectorized into OGC-standard **GeoPackage (.gpkg)** format using the `rasterio` and `shapely` libraries.

A. Spatial Reference System

Accurate area calculation requires a metric coordinate system that minimizes distortion. We projected all vector outputs to **ETRS89 / UTM zone 32N (EPSG:25832)**. This projection provides high-precision metric measurements specific to the North Rhine-Westphalia region, ensuring that the derived area values (m^2) are sufficiently accurate for cadastral assessment.

B. Automated vs. Manual Validation

We employed a dual-validation strategy to ensure the reliability of the tax estimates:

1) **Automated Bulk Calculation:** A custom Python script (`calculate_tax_area.py`) processed the entire vectorized dataset, summing the area attributes of all identified structures within the 15 km^2 study region instantly.
2) **Manual Spot Check (QGIS):** To verify geometric fidelity, the dataset was inspected in QGIS. We performed a manual spot check on the Institute of Geodesy and Geoinformation (highlighted in red in Fig. 3). Using the QGIS Field Calculator function `$area`, we confirmed that the automated Python extraction matched the interactive GIS measurement, validating the spatial alignment of the vectors.

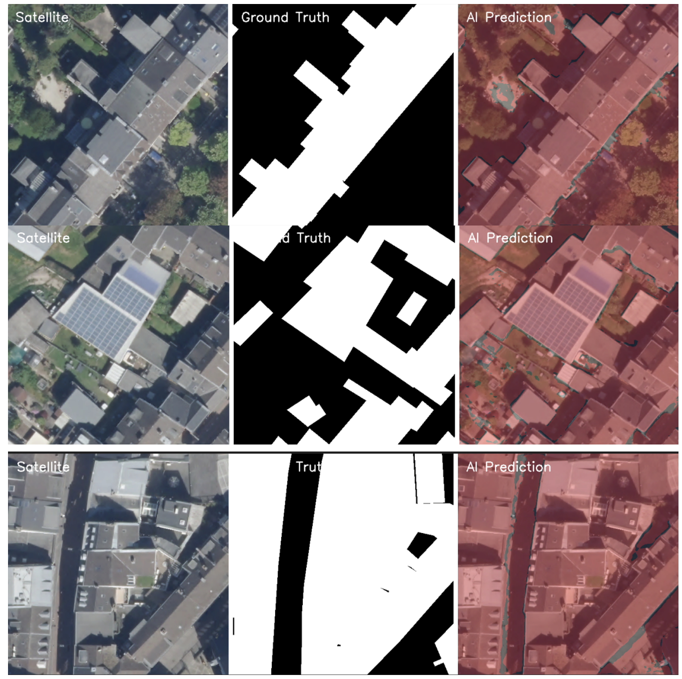

Fig. 4. Random inference samples from the test set. Columns from left to right: Raw Satellite Input, Ground Truth Mask, and AI Prediction. The results demonstrate the model's ability to generalize across different building types and lighting conditions.

C. Revenue Estimation

Based on the automated analysis, we applied a hypothetical split wastewater fee rate of **0.85 €/**m^2. The system derived the following fiscal projections:

- **Total Sealed Surface Detected:** $163,148\ m^2$
- **Estimated Annual Tax Revenue: 138,676 €**

This result demonstrates that the pipeline provides rapid, city-scale revenue estimations, allowing municipal planners to identify discrepancies between registered ALKIS records and actual physical impervious surfaces (e.g., undeclared extensions).

VII. Conclusion

This study validates that high-precision cadastral mapping can be automated using Deep Learning without reliance on industrial-scale supercomputing clusters. We successfully deployed a ResNet-34 encoded U-Net on a consumer-grade workstation to extract urban building footprints from open government data (DOP10) with an IoU of 0.8251.

The core contribution of this work lies in the engineering solution to the **Domain Shift** problem. We demonstrated that standard off-the-shelf inference often fails on raw geospatial data due to radiometric compression, and that a targeted confidence threshold tuning strategy ($t = 0.6$) is sufficient to recover precise geometries without expensive retraining.

By integrating this neural network into an end-to-end pipeline—from rasterization to vectorization and fiscal analysis—we reduced the workflow time from weeks of manual digitization to less than one hour of computation. The estimated tax revenue of **138,676 €** highlights the immediate administrative utility of such systems. Ultimately, this research confirms that modern AI architectures have matured sufficiently to handle legal-grade municipal tasks using accessible hardware.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[4] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[5] X. Zhu, D. Tuia, L. Mou, G. S. Xia, and L. Zhang, "Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[6] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 645–657, 2017.

[7] V. Iglovikov and A. Shvets, "TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation," *arXiv preprint arXiv:1801.05746*, 2018.

[8] V. Mnih, "Machine Learning for Aerial Image Labeling," Ph.D. dissertation, University of Toronto, 2013.