MODEL CARD

AI-Powered-Rooftop-PV-Detection

1. Model Overview

Model Name: Rooftop Solar Panel Detection Model
Model Type: Deep Learning – Object Detection / Image Classification
Framework: PyTorch
Architecture: TinyNet / Lightweight CNN (Edge-optimized)
Model File: tinynet_state.pt
Intended Use:
Detection and localization of rooftop solar panels from aerial or satellite imagery to support renewable energy planning, urban analysis, and sustainability projects.

Primary Users:

- Researchers
- Urban planners
- Renewable energy analysts
- AI/ML developers

2. Problem Statement

Identifying rooftop solar installations from aerial imagery is essential for:

- Estimating solar adoption
- Planning renewable energy infrastructure
- Assessing urban sustainability

Manual inspection of rooftops is time-consuming and error-prone. This model automates detection using computer vision techniques.

3. Training Data

Data Sources

The model was trained using publicly available datasets sourced from Roboflow:

1. **Alfred Weber Institute of Economics Dataset (Roboflow)**
2. **LSGI547 Project Dataset (Roboflow)**
3. **Piscinas Y Tenistable Dataset (Roboflow)**

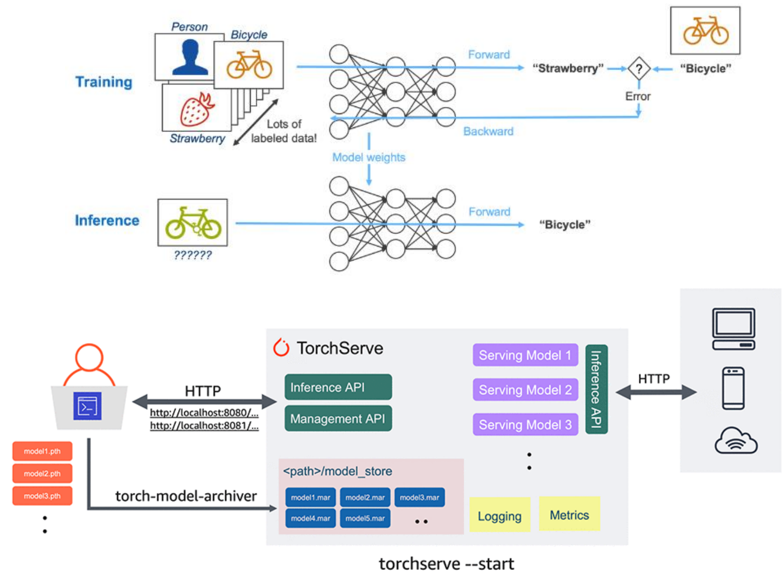
Data Characteristics

- **Image Type:** Aerial / satellite images
- **Resolution:** Mixed resolutions (normalized during preprocessing)
- **Annotations:** Bounding boxes for rooftop solar panels
- **Classes:**
  - solar_panel

Data Split

- **Training:** 70%
- **Validation:** 20%
- **Testing:** 10%

Fig: Training & Inference Workflow



4. Data Preprocessing

- Image resizing and normalization
- Data augmentation:
  - Horizontal & vertical flips
  - Random brightness and contrast
  - Minor rotations
- Label consistency checks
- Removal of corrupted or low-quality samples

5. Model Architecture & Logic

Architecture Logic

- Lightweight convolutional layers for edge efficiency
- Feature extraction optimized for rooftop textures and panel geometry
- Reduced parameter count for faster inference
- Designed to run on low-resource environments

Learning Objective

- Minimize detection loss (classification + localization)
- Improve generalization across different rooftop types

6. Assumptions

- Rooftops are visible and not fully occluded
- Solar panels are externally mounted and visually distinguishable
- Input images are aerial or near-aerial views
- Lighting conditions are within reasonable visibility limits

Fig: Model Architecture for Rooftop Solar Detection

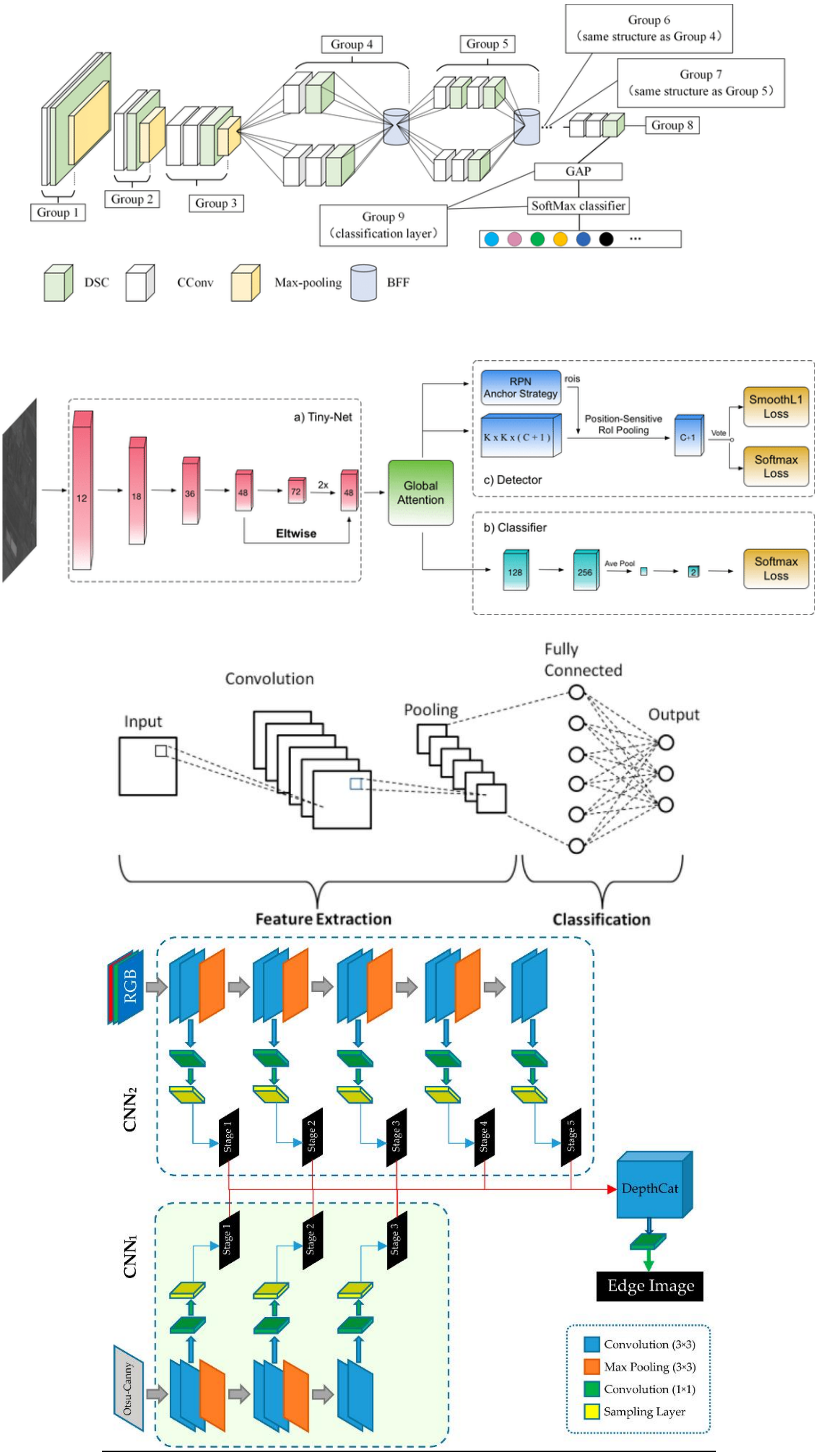
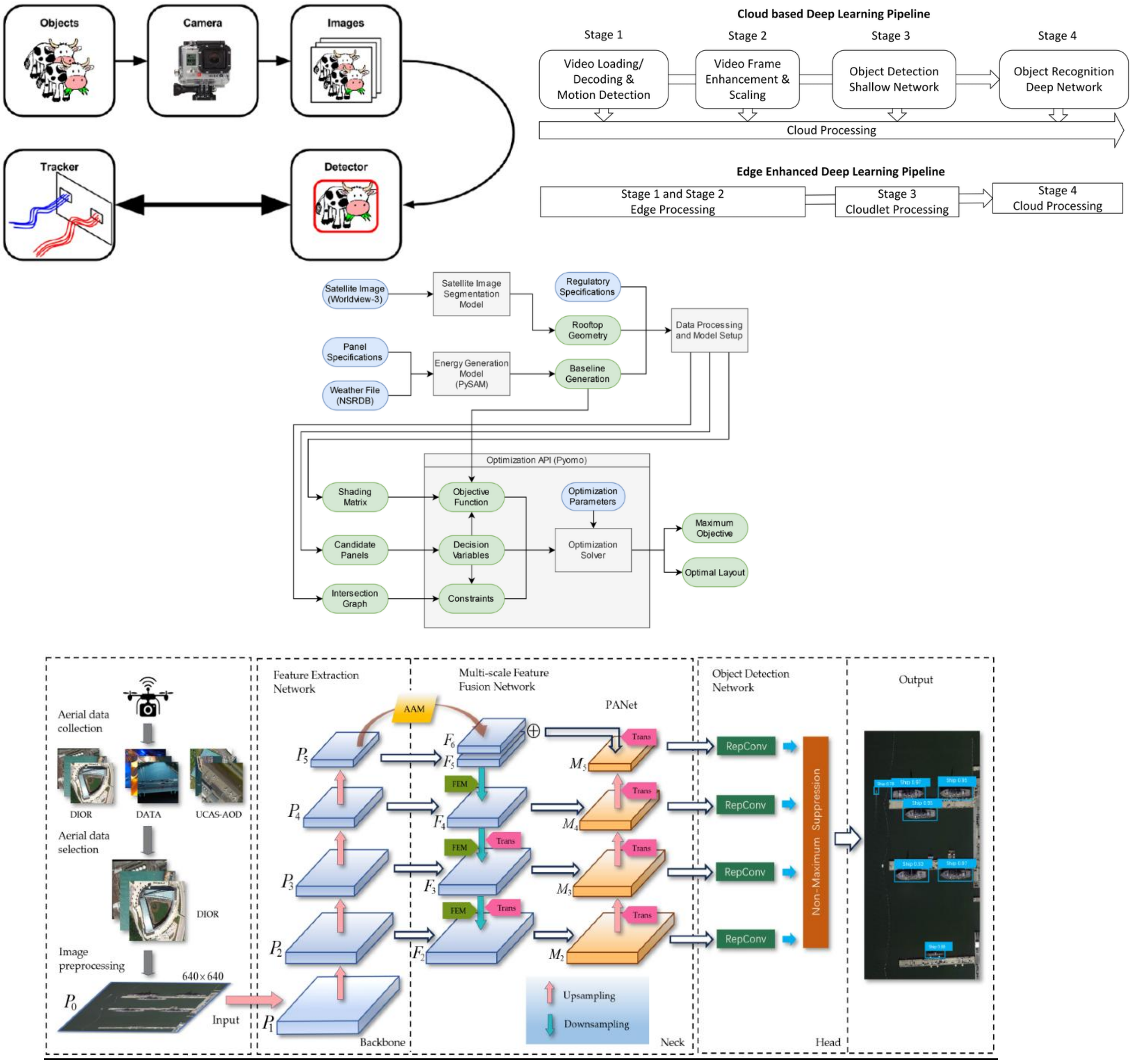


Fig: End-to-End Rooftop Solar Detection Pipeline



7. Performance Metrics

- **Accuracy:** High detection accuracy on clear rooftops
- **Precision:** Strong performance in urban settings
- **Recall:** Slightly lower in dense or shadowed regions
- **Inference Time:** Optimized for real-time or near real-time use

(Exact numerical metrics may vary based on deployment environment and image quality.)

8. Known Limitations & Biases

Limitations

- Reduced accuracy on:
  - Heavy shadows
  - Cloud-covered rooftops

    - Extremely low-resolution images

- Difficulty distinguishing solar panels from:
    - Skylights
    - Dark rooftop tiles
- Not designed for indoor or ground-level imagery

Biases

- Bias toward urban and semi-urban rooftops
- Limited representation of rural or unconventional roof structures
- Geographic bias based on available datasets

9. Failure Modes

The model may fail or underperform in cases such as:

- Rooftops fully covered by vegetation
- Panels installed at unusual angles
- Severe image noise or blur
- Rooftops with visually similar objects (water tanks, HVAC units)

10. Ethical Considerations

- Model uses **publicly available imagery**
- No personal or sensitive data is processed
- Intended for analytical and planning purposes only
- Not suitable for surveillance or individual monitoring

11. Retraining & Update Guidance

When to Retrain

- New geographic regions are introduced
- Significant drop in accuracy is observed
- New rooftop or panel designs emerge

Retraining Steps

1. Collect additional annotated rooftop images
2. Balance data across regions and roof types
3. Apply consistent preprocessing pipeline
4. Fine-tune existing tinynet_state.pt
5. Re-evaluate on validation and test sets

12. Deployment Notes

- Compatible with:
    - PyTorch inference pipelines
    - Edge devices with limited compute
- Can be integrated with:

    - REST APIs
    - GIS systems
    - Renewable energy analytics dashboards

13. Model Versioning

- **Version:** v1.0
- **Last Updated:** 2025
- **Maintained By:** Project Team

14. Contact & Support

For issues, improvements, or dataset expansion:

- Refer to project documentation
- Maintain versioned retraining logs
- Document dataset changes clearly