# Efficient Poverty Mapping from High Resolution Remote Sensing Images

Kumar Ayush, Burak Uzkent, Kumar Tanmay, Marshall Burke, David Lobell, Stefano Ermon

Conference: AAAI

# Motivation

- Accurate poverty data is crucial for governments, NGOs, and policymakers to allocate resources effectively.
- Problems with Traditional household surveys are:
  - Expensive and time-consuming.
  - Conducted infrequently (e.g., once every 5+ years in many developing nations).
  - Limited in coverage (small sample sizes, missing rural areas)
  - High-resolution imagery is expensive ($10–$20 per km²)

- **Key Question: How can we use satellite images to map poverty efficiently while minimizing costs?**

# Related Work

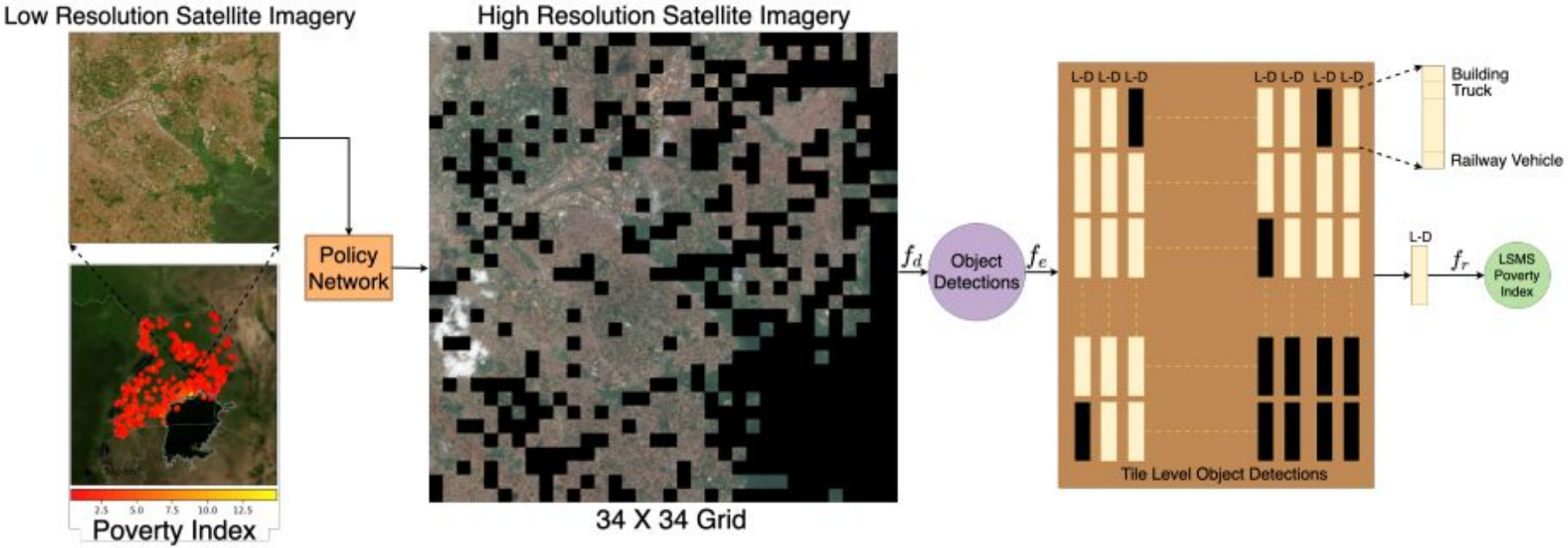
- Survey-Based Approaches (Expensive, Time Consuming).
- Nightlight-Based Approaches (fails in rural, low-income regions).
- Deep Learning on Satellite Images (Requires expensive high-resolution images for entire regions).

# The Proposed Solution

- Leverage freely available low-resolution satellite images (Sentinel-2, 10m resolution).
- Use Reinforcement Learning (RL) to selectively acquire high-resolution tiles:
- Learn where to buy high-res images, instead of acquiring everything.
  - Extract meaningful information using Object Detection (YOLOv3).
- Count objects like buildings, vehicles, and infrastructure.
  - Train a poverty prediction model using extracted object counts.

- **Result:**
  - 80% fewer high-resolution images required.
  - $2.9M estimated cost savings for Uganda.
  - Higher accuracy ($r^2 = 0.61$) than previous approaches ($r^2 = 0.53$).

# Architecture



# Fine-grained Object Detection on High-Resolution Satellite Imagery

## Why do we need object detection?

> The goal is to **extract meaningful features** from high-resolution satellite images that can help us in predicting the poverty levels.

Basically, instead of directly using the raw images, we extract count of certain classes that then act as features and act as our input in the prediction model

(**Ayush et al. 2020**) showed that **counting objects** (e.g., number of trucks, houses, etc.) can predict poverty levels well..

# Fine-grained Object Detection on High-Resolution Satellite Imagery

1. One key issue is the lack of annotated data.

2. This issue is solved by pre-training the object detector on a different dataset which in the paper was **xView**, a **large-scale satellite image dataset** with labeled objects. xView has 10 parent classes and 60 child classes.

3. The authors used YOLOv3 for object detection.

# YOLOv3 - You Only Look Once

- **Bounding boxes** : where the objects are located?
- **Object classes** : what objects are present ?
- **Confidence scores :** how certain the model is ?.

# Object Detection

It scans **1000×1000 pixel** tiles from high-resolution images and detects objects

Each tile gives an **L-dimensional vector** where L = 10 corresponding to the number of parent classes and it tells us the count of the objects that belong to that class.

A village is covered by **34×34 = 1156** high-resolution image tiles (each 1000×1000 pixels).

For each village, they **sum** object counts across all tiles to get a single feature vector.

This vector is then used in a **regression model** to predict poverty levels.

# Adaptive Tile Selection

GOAL : **Minimize high-res image acquisition** while **maintaining prediction accuracy**.

Key idea : **REINFORCEMENT LEARNING!**

# Adaptive Tile Selection

The high res image is too large for direct RL processing. We divide it into T non-overlapping tiles.

We try to infer $H_i$ from $L_i$ as a latent variable. Each tile of $H_i$ is associated with a L-dimensional feature of class counts.

These T tiles are further divided into S disjoint sub-tiles.

Now our task is to define a policy network to select optimal sub-tiles.

# Two-Step MDP

Step 1:

We input the low-res tile and the agent outputs a binary action array for the sub-tiles which is then converted into a probability distribution.

$$\pi(\mathbf{a}_i^j | l_i^j; \theta_p) = p(\mathbf{a}_i^j | l_i^j; \theta_p)$$

Step 2:

We then do the object detection part on the selected sub-tiles

# Cost Function

$$\max_{\theta_p} J(\theta_p, \theta_d) = \mathbb{E}_p[R(\mathbf{a}_i^j, \hat{\mathbf{v}}_i^j, \mathbf{v}_i^j)],$$

$$R = R_{acc}(\hat{\mathbf{v}}_i^j, \mathbf{v}_i^j) + R_{cost}(\mathbf{a}_i^j)$$

$$R_{acc} = -||\mathbf{v}_i^j - \hat{\mathbf{v}}_i^j||_1$$

$$R_{cost} = \lambda(1 - ||\mathbf{a}_i^j||_1/S)$$

# Optimization of the Policy Network

Our objective function is not differentiable wrt to the policy network parameters because of the discrete actions. Authors use **Policy Gradient (REINFORCE)** to update policy network parameters $\theta_p$.

$$\nabla_{\theta_p} J = \mathbb{E}\left[R(\mathbf{a}_i^j, \hat{\mathbf{v}}_i^j, \mathbf{v}_i^j) \nabla_{\theta_p} \log \pi_{\theta_p}(\mathbf{a}_i^j | l_i^j)\right]$$

The authors used mini-batch Monte-Carlo sampling to approximate the expectation

# Feasibility Analysis

What do we need?

Datasets:

Low resolution images(Sentinel-2) - Freely available

**Poverty Data (LSMS Survey for Uganda)** → Available from the **World Bank**.

High resolution images - This is the tricky part, we will need to make this dataset ourselves as getting it would be expensive.!

# Feasibility Analysis

Code availability:

https://github.com/kayush95/efficientPovertyPrediction

This repository contains the necessary codes that are YOLOv3 for object detection, RL based tile selection and the regression model for prediction.

With the datasets and the available code, it will be possible to implement and replicate this paper to a plausible extent.

# Feasibility analysis

**Implementation strategy:**

- **Phase 1:** Data collection & preprocessing.
- **Phase 2:** Implement **YOLOv3 object detection**.
- **Phase 3:** Train/test **adaptive tile selection model**.
- **Phase 4:** Train **poverty prediction model** & evaluate results.

# Experiments

The Study conducted two primary experiments.

1. Policy Network Validation on xView Dataset:
   - Our downstream dataset  does not contain object bounding boxes.
   - We train our policy network on the xView dataset.
   - Trained on 1,249 points and tested on 200 points using 2000×2000px HR images
   - The policy network achieved higher precision while using less than half the HR data and nearly halving processing time.

| | mAP | mAR | HR | Run-time |
|---|---|---|---|---|
| **No Dropping** | 24.3% | 42.5% | 100.0% | 2890 ms |
| **RL Method** | 26.3% | 41.1% | 42.3% | 1510 ms |

Table 1: Results on the xView test set.

# Experiments(Cont.)

## 2. Uganda Poverty Prediction

- In this we train and test the policy network on Uganda dataset where we have only cluster-level poverty labels.
- Using our adaptive method, we get $\hat{\mathbf{m}}_i = \sum_{j=1}^{T} \hat{\mathbf{v}}_i^j$
- We are using Gradient Boosting Decision Trees as the regression model to estimate the poverty index, yi.
- Used 320 clusters split into 80% train / 20% test.

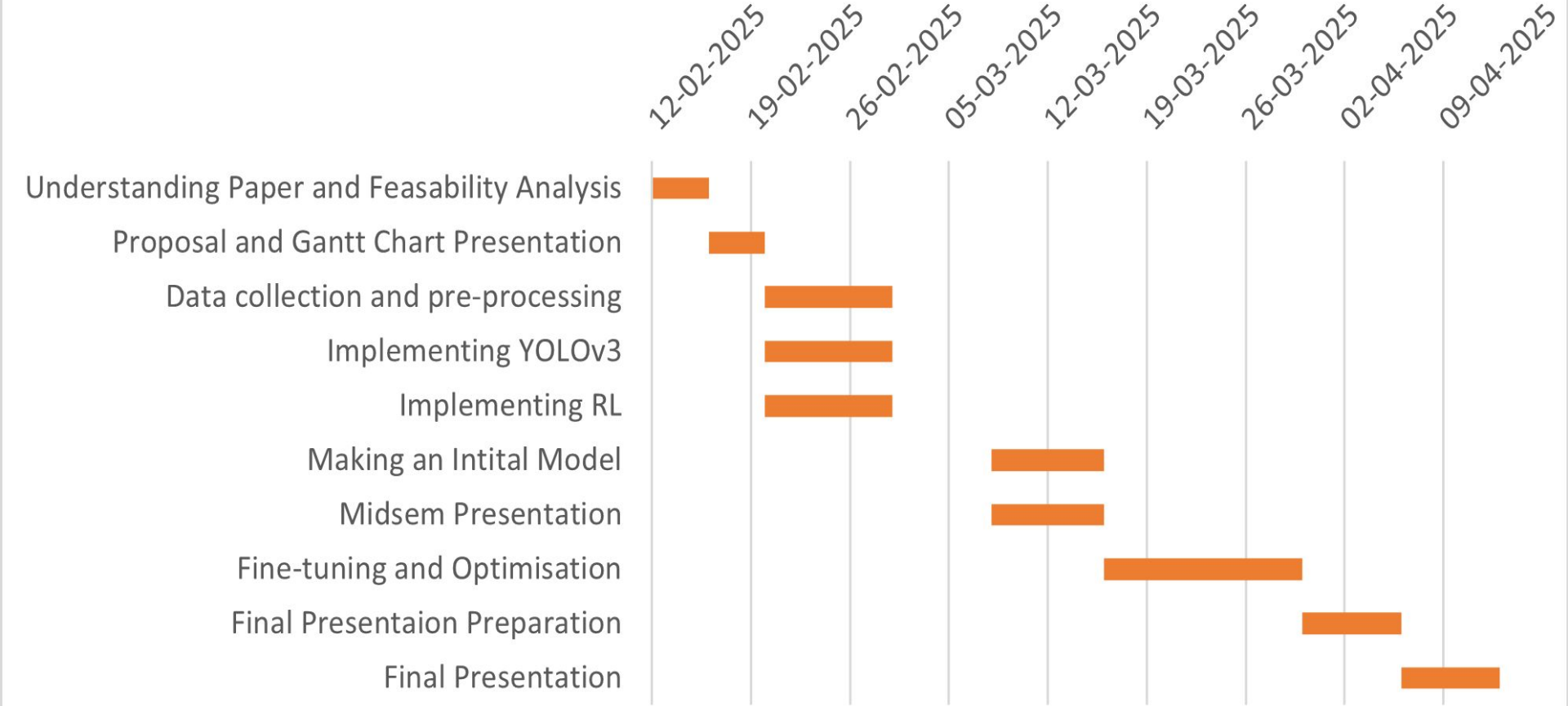
# Comparison

We compare our method with the following state of the art baseline models.

- Key metric: Pearson's $r^2$

| | No Dropping | Fixed-18 | Random-25 | Stochastic-25 | Green | Counts Pred. | Sett. Layer | Nightlights | Ours (Dry sea.) | Ours (Wet sea.) |
|---|---|---|---|---|---|---|---|---|---|---|
| $r^2$ | 0.53 | 0.43 | 0.34 | 0.26 | 0.33 | 0.49 | 0.45 | 0.45 | $0.51 \pm 0.01$ | $\mathbf{0.61 \pm 0.01}$ |
| MSE | 1.86 | 2.20 | 2.67 | 3.13 | 2.56 | 1.91 | 2.16 | 2.17 | $1.89 \pm 0.02$ | $\mathbf{1.46 \pm 0.02}$ |
| Explained Variance | 0.54 | 0.43 | 0.33 | 0.27 | 0.36 | 0.48 | 0.46 | 0.45 | $0.50 \pm 0.01$ | $\mathbf{0.63 \pm 0.02}$ |
| HR Acquisition. | 1.0 | 0.18 | 0.25 | 0.25 | 0.19 | 0.19 | 0.19 | 0.12 | 0.19 | **0.19** |

# Gantt Chart


12-02-2025
19-02-2025
26-02-2025
05-03-2025
12-03-2025
19-03-2025
26-03-2025
02-04-2025
09-04-2025
Understanding Paper and Feasability Analysis
Proposal and Gantt Chart Presentation
Data collection and pre-processing
Implementing YOLOv3
Implementing RL
Making an Intital Model
Midsem Presentation
Fine-tuning and Optimisation
Final Presentaion Preparation
Final Presentation