Appendix A. Spatial Resolution and Zoom Levels in Google Maps

In Google Maps satellite imagery, the spatial resolution is directly related to the zoom level z. The approximate ground resolution $R(z, \phi)$ in meters per pixel at a given latitude ϕ can be computed as:

$$R(z, \phi) = \frac{156543.03392 \cdot cos\ cos\ (\phi)}{2^z}$$

This formula is derived from the Web Mercator projection used by Google Maps (Google Developers, n.d.). The constant 156543.03392 corresponds to the resolution at zoom level 0 along the equator, where the Earth's circumference (approximately 40,075 km) is mapped onto a 256-pixel-wide tile. As the zoom level increases, the resolution doubles in each direction, which is reflected in the denominator 2^z. The $cos\ cos\ (\phi)$ term accounts for the reduction in effective resolution at higher latitudes due to the projection's distortion.

In the state of Florida, USA, with an average latitude of $\phi \approx 28.5°$, we have:

$$R(14, 28.5°) \approx \frac{156543.03392 \cdot 0.879}{2^{14}} \approx 8.39\ meters/pixel.$$

Similarly, for higher zoom levels at \(\phi\approx28.5^\circ\):

$$R(15, 28.5°) \approx \frac{156543.03392 \cdot 0.879}{2^{15}} \approx 4.19\ meters/pixel$$

$$R(16, 28.5°) \approx \frac{156543.03392 \cdot 0.879}{2^{16}} \approx 2.09\ meters/pixel$$

$$R(17, 28.5°) \approx \frac{156543.03392 \cdot 0.879}{2^{17}} \approx 1.04\ meters/pixel$$

The total ground coverage of a single 256 × 256 pixels image at zoom level 14 is approximately covering an area of about 4.62 square kilometers. For other zoom levels, under the same conditions and assuming a 256 × 256-pixel image, the approximate ground coverage would be:

- At zoom level 15: about 1.16 square kilometers.
- At zoom level 16: about 0.29 square kilometers.
- At zoom level 17: about 0.07 square kilometers.

Appendix B. Semantic Segmentation Model Details

This appendix provides additional technical details on the semantic segmentation models used for feature extraction. These models are not modified or fine-tuned in this study; they are used as pre-trained feature extraction tools to convert raw images into structured feature vectors.

B.1. DeepLabV3+

DeepLabV3+ employs an encoder–decoder framework in which the encoder captures high-level semantic features and the decoder refines spatial details for boundary delineation. DeepLabV3+ uses atrous convolution to expand the receptive field while preserving spatial resolution. The atrous convolution can be written as:

$$y[i] = \sum_k \quad x[i + r \cdot k]w[k],$$

where$y[i]$ is the output feature map at position i, $x[i]$ is the input feature map, $w[k]$ is the convolution kernel, and r is the dilation rate. DeepLabV3+ employs Atrous Spatial Pyramid Pooling (ASPP) to capture multi-scale contextual information:

$$ASPP(x) = \sum_{i} \quad f_{atrous}(x, W_i, r_i),$$

where f_{atrous} represents atrous convolution with kernel weights W_i and dilation rate r_i. The model is trained using pixel-wise cross-entropy loss:

$$L = -\sum_{i=1}^{H} \quad \sum_{j=1}^{W} \quad \sum_{c=1}^{C} \quad y_{ijc}\, log\, log\, \left(p_{ijc}\right),$$

where H and W are the image height and width, C is the number of classes, y_{ijc} is the ground-truth label for pixel (i, j) and class c, and p_{ijc} is the predicted probability.

B.2. CSAILVision semantic segmentation model

For satellite-image segmentation, we use the CSAILVision semantic segmentation model with a ResNet-50 encoder and PPMdeepsup decoder. The PPMdeepsup decoder incorporates a Pyramid Pooling Module (PPM) to capture contextual information at multiple spatial scales and uses deep supervision to strengthen intermediate feature learning.

Let F denote the encoder feature map. The pooled feature at pyramid level l can be written as:

$$P_l = (F)\,, \qquad l = 1, \ldots, L,$$

The multi-scale pooled features are then upsampled and concatenated with the original feature representation for decoding. Deep supervision can be represented as:

$$L_{total} = L_{main} + \sum_{i} \quad \lambda_i L_{i,aux},$$

where L_{main} is the loss of the final output, $L_{i,aux}$ is the loss of an intermediate auxiliary output, and λ_i controls the contribution of the i-th intermediate feature learning.

Appendix C. Model Specifications and Hyperparameters

Tables 3 and 4 summarize the model specifications and input variables used in the predictive analysis. RF and GBDT were implemented using fixed hyperparameter settings rather than a tuning procedure; the settings are reported in Table 3. Table 3 describes the three predictive methods, while Table 4 lists the common sociodemographic variables and imagery-derived variables used under each imagery setting.

Table 3. Model specifications for predictive analysis

Method	Model description	Model settings
MLR	Multiple linear regression estimated separately for each commuting mode.	Input variables were standardized before estimation.
RF	Random forest regression estimated separately for each commuting mode.	Number of trees = 1,000; random seed = 42.
GBDT	Gradient boosting decision tree regression estimated separately for each mode.	Number of boosting stages = 1,000; random seed = 42.

Notes: Driving, public transit, and active modes were modeled separately. RF and GBDT used fixed settings rather than hyperparameter tuning. All models were evaluated on the held-out test set, with bootstrap confidence intervals computed for test-set R^2 values.

Table 4. Input variables used under different imagery settings

Imagery Setting	Common Variables	Imagery-Derived Variables
Street-view imagery	Sociodemographic variables[a]	Relative proportions of road, building, sky, sidewalk, plant, car, and person
Satellite imagery (zoom levels 14–17)	Sociodemographic variables[a]	Relative proportions of building, road, plant, water, earth (bare soil or undeveloped land), and bridge[b]

[a] Sociodemographic variables include educational attainment, unemployment rate, vehicles per capita, male–female ratio, median individual income, average household size, and racial composition.

[b] Satellite-image models are estimated separately for zoom levels 14, 15, 16, and 17 using the same feature categories.

Reference

Google Developers. (n.d.). Map and tile coordinates. Retrieved April 14, 2025, from https://developers.google.com/maps/documentation/javascript/coordinates