api_client.py – Google Maps Satellite Imagery Client

Overview

This module handles the fetching of satellite imagery from the Google Maps Static API. It downloads high-resolution satellite images for given geographic coordinates.

Logic

Class: GoogleMapsClient

Method	Purpose
__init__()	Initialize client with API credentials and image settings
download_satellite_image()	Fetch satellite image for given lat/lon coordinates
_resize_if_needed()	Ensure downloaded image matches target dimensions
_resize_and_filter()	Testing utility - adds darkness filter for QC testing

Key Constants

```
BASE_URL = "https://maps.googleapis.com/maps/api/staticmap"
```

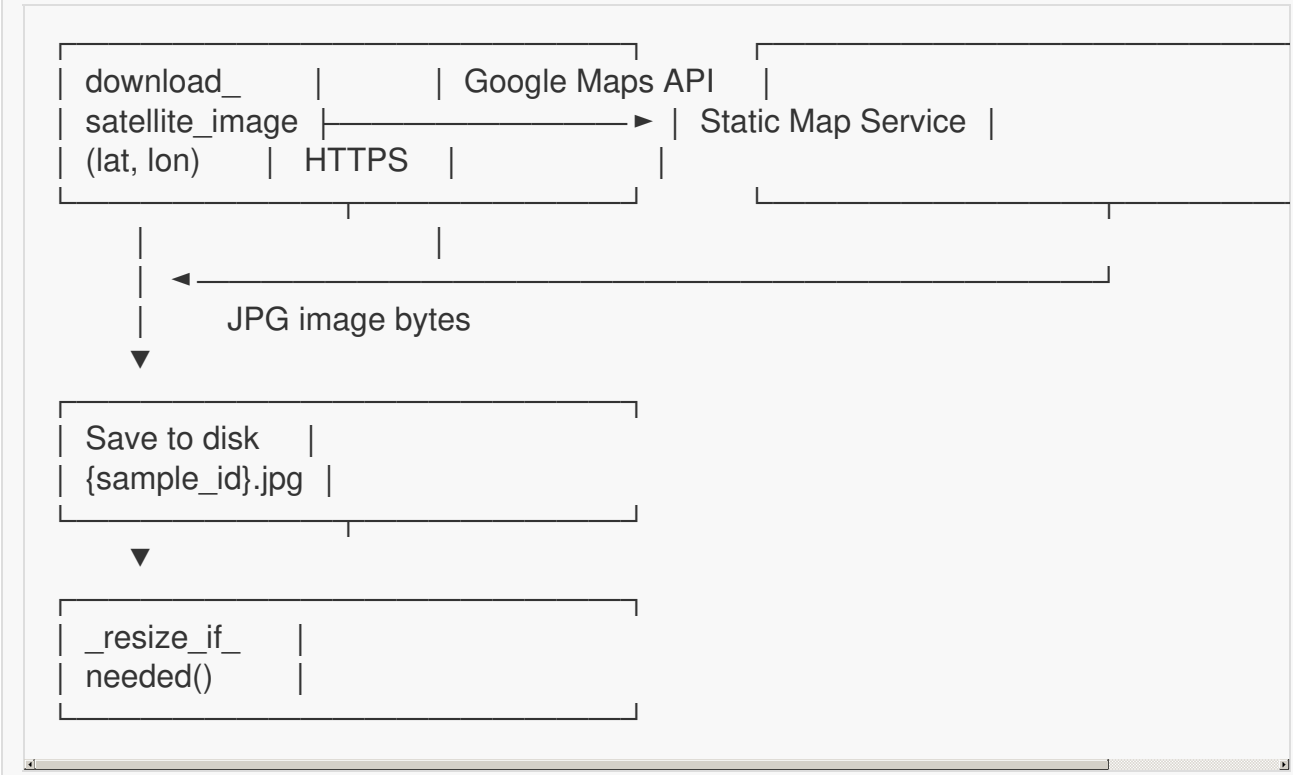
How It Works

1. Initialization

```
GoogleMapsClient(
    api_key="YOUR_KEY",
    zoom_level=20,        # Street-level detail
    image_size=1024,      # Target output size
    map_scale=2           # High DPI scaling
)
```

The request_size is calculated as image_size // map_scale because the Google API scale parameter multiplies the output dimensions.

2. Image Download Flow



3. API Request Parameters

Parameter	Value	Description
center	{lat},{lon}	Geographic coordinates
zoom	20	Maximum detail zoom level
size	512x512	Requested image dimensions
scale	2	High-DPI scaling (doubles output)
maptype	satellite	Satellite imagery type
key	API key	Authentication

Why It Works

Scale Factor Logic

Google's Static Maps API limits the base image size but applies `scale` as a multiplier:

- Request 512x512 with `scale=2` → Receive 1024x1024
- This bypasses the 640px single-request limit
- Provides higher resolution imagery for detection

Error Handling

The module handles:

- **HTTP errors** - Status code checks
- **Timeouts** - 15-second request timeout
- **Download failures** - Returns `None` for graceful pipeline continuation

Resize Safety Net

Downloaded images may not exactly match expected dimensions due to:

- API quirks at certain zoom levels
- Edge cases near map boundaries

The `_resize_if_needed()` method ensures consistent 1024x1024 input for the detector.

Usage in Main Pipeline

```python
# In pipeline.py
self.maps_client = GoogleMapsClient(
    api_key=Config.GOOGLE_MAPS_API_KEY,
    zoom_level=Config.ZOOM_LEVEL,
    image_size=Config.FINAL_IMAGE_SIZE,
    map_scale=Config.MAP_SCALE
)

# During sample processing
image_path = self.maps_client.download_satellite_image(
    lat, lon, sample_id, sample_folder
)
```

Pipeline Integration Flow

1. **Pipeline receives** coordinates from input Excel file
2. **Calls** download_satellite_image() with sample metadata
3. **Stores** image in sample-specific output folder
4. **Passes** image path to quality checker and detector

Error Handling in Pipeline

```python
if not image_path:
    logger.warning(f"Skipped sample {sample_id} - download failed")
    return None
```

Failed downloads result in skipped samples, allowing the pipeline to continue processing other locations.