When using the Google Street View API, it can be challenging to get an exact match for a specific address, as the API may sometimes provide a slightly deviated image due to a few limitations. Here are some tips to improve the accuracy of your Street View image requests:

**1. Use Exact Latitude and Longitude Coordinates**

Instead of using an address string, try converting the address to exact latitude and longitude coordinates with the Google Geocoding API. Then, use these precise coordinates in your Street View API request. This typically provides a more accurate location than an address string alone.

python

Copy code

https://maps.googleapis.com/maps/api/streetview?location=LATITUDE,LONGITUDE&size=600x400&key=YOUR\_API\_KEY

**2. Adjust the heading Parameter**

The heading parameter lets you specify the compass direction of the image. When you use the Google Maps website, it often rotates the image to face the most relevant point of interest. You can try to match this by setting a specific heading in your Street View request. For example:

python

Copy code

https://maps.googleapis.com/maps/api/streetview?location=LATITUDE,LONGITUDE&size=600x400&heading=45&key=YOUR\_API\_KEY

To determine the optimal heading, you may need to experiment based on the location and the imagery you’re receiving or attempt to estimate based on the nearby roads and points of interest.

**3. Use the fov Parameter**

Adjusting the fov (field of view) parameter can also help optimize the view to match what the user sees on Google Maps. The default is 90 degrees, but narrowing it down may help focus on a specific target area:

python

Copy code

https://maps.googleapis.com/maps/api/streetview?location=LATITUDE,LONGITUDE&size=600x400&fov=80&key=YOUR\_API\_KEY

**4. Combine with Google Places API**

Use the Google Places API to get more detailed information about the place, including its precise location and the orientation of the building. If the place has been tagged, this information can help you fine-tune your Street View API request by refining heading and fov based on the specific location of interest.

**5. Try Alternate Zoom Levels**

Experimenting with different zoom levels can sometimes improve the accuracy of the image. Although zoom is not a direct parameter in the Street View API, changing fov effectively adjusts zoom.

**6. Use source=outdoor Parameter for Outdoor Locations**

If you’re looking for images of outdoor locations, you can specify source=outdoor to filter out images from inside buildings and increase the chances of getting accurate street-level imagery.

python

Copy code

https://maps.googleapis.com/maps/api/streetview?location=LATITUDE,LONGITUDE&size=600x400&source=outdoor&key=YOUR\_API\_KEY

**Example Workflow**

1. **Convert Address to Lat/Long** using Geocoding API.
2. **Fetch Additional Place Information** (optional) using Places API to determine relevant features like orientation.
3. **Set Location and Tune heading, fov, source Parameters** in the Street View API request.

By tuning these parameters, you can generally get closer to the exact view a user would see on Google Maps.

**Context**:  
"This is a KYC document verification request for a bank. The document provided should be analyzed to verify content presence, readability, compliance, and quality standards."

**Prompt**: "Perform the following checks on the attached document:

* **Blank Content**: Indicate if the document is blank (Yes/No).
* **Confidence Score**: Provide an overall confidence score for each check.
* **Readability Score**: Rate text clarity.
* **Noise Level**: Identify presence of stray or garbled characters.
* **Compliance Check**: Confirm absence of CSAM or abusive content.
* **Document Quality Score**: Rate for automated processing.
* **Suggested Action**: Recommend 'Accept,' 'Review,' or 'Re-upload.'

Return a concise summary for each item."

**SAMPLE**:

Based on the analysis of the uploaded PDF document, here are the results for each specified check:

1. **Blank Content**: No – The document contains text and images relevant to a boarding pass, so it is not blank.
2. **Confidence Score**: 95% – High confidence in the document's readability and content checks based on the extracted data.
3. **Readability Score**: 90% – The text is clear and readable, with structured information suitable for KYC review.
4. **Noise Level**: Low – There are no significant stray or garbled characters detected; the document's text is well-structured.
5. **Compliance Check**: Pass – The content is free from any abusive or inappropriate material, adhering to compliance standards.
6. **Document Quality Score**: 88% – The document's clarity, structure, and format are suitable for automated processing.
7. **Suggested Action**: Accept – The document meets quality and compliance criteria for KYC purposes.

**Prompt:**

"Validate extracted document content for KYC with these checks:

1. **Data Validity**: Confirm names, dates, addresses, and ID numbers follow standard formats and logical sequences.
2. **Authenticity**: Check for correct document structure, watermarks, barcodes, and font consistency.
3. **Database Cross-Check**: Match data with external databases if available.
4. **Face Match**: If available, compare document photo with a live selfie for identity confirmation.
5. **Geolocation**: Verify document origin aligns with uploader's IP location.
6. **Fraud Detection**: Detect any signs of tampering, synthetic content, or metadata inconsistencies.

Provide a summary with confidence scores and flag any concerns."

**Prompt:**

"Compare the extracted address from the document with addresses from third-party data sources, internal databases, and Google Maps geocoding. Calculate a **similarity score** based on:

1. **Exact Match**: Check for a full match with addresses in internal databases and third-party sources.
2. **Partial Match**: Identify partial matches (e.g., street name, city, postal code) and calculate a similarity score based on aligned elements.
3. **Geolocation Accuracy**: Cross-reference the geocoded location with the provided address to confirm proximity.
4. **Standardization**: Account for common address variations (e.g., abbreviations, format differences).

Return a similarity score (0-100%) and a brief summary of matched elements or discrepancies."