SignBridge AI ASL Recognition API Documentation

This document describes two APIs provided by SignBridge AI for American Sign Language (ASL) recognition:

- **RESTful API**: Suitable for batch or single-shot recognition requests.
- **WebSocket API**: Suitable for real-time streaming interactions.

RESTful API

Endpoint

```
POST https://api.signbridgeai.com/recognize_sign
```

Request

- **HTTP Method**: POST
- **Content-Type**: `application/json`

Request Parameters

Field	Type	Required	Description
payload	string	Yes	Base64 encoded video data.
api_key	string	Yes	API key for authentication.
single_recognition_mode	boolean	No	Single-shot recognition mode (default false).
hint	string	No	Optional hint to assist recognition.

Example Request

```
{
  "payload": "<BASE64_ENCODED_VIDEO>",
  "single_recognition_mode": true,
  "hint": "Optional recognition hint",
```

```
  "api_key": "<YOUR_API_KEY>"
}
```

Response

Successful Response:

```
{
  "prediction": [
    {
      "prediction": "Hello",
      "confidence": 0.95
    }
  ],
  "final_packet": true
}
```

Field	Type	Description
prediction	Array	Array containing recognized text and confidence.
final_packet	boolean	Indicates if this response is the final recognition.
— prediction	string	Recognized text from the video.
— confidence	number	Confidence score (original scale, converted from log).

Error Responses:

- Invalid API key:

```
{
  "detail": "Invalid API key"
}
```

- Internal server error:

```
{
  "error": "Internal Server Error",
  "status_code": 500
}
```

WebSocket API (Real-time Streaming)

Endpoint

```
wss://api.signbridgeai.com/recognize_sign_ws
```

Handshake (Initial Connection)

Upon WebSocket connection establishment, the client must immediately send the following JSON message for handshake:

Field	Type	Required	Description
api_key	string	Yes	API key for authentication.
protocol_version	string	Yes	Communication protocol version ("1.0").
stream_type	string	Yes	Stream type, must be "video".
mode	string	Yes	Recognition mode, "single" or "multiple".

Example Handshake Request

```
{
  "api_key": "<YOUR_API_KEY>",
  "protocol_version": "1.0",
  "stream_type": "video",
  "mode": "single"
}
```

Successful Handshake Response:

```
{
  "status": "success",
  "message": "Handshake successful"
}
```

Failed Handshake Response:

```
{
  "status": "error",
  "message": "Missing required fields"
}
```

Data Streaming

Client-to-Server:

- **Binary Data**: Video stream data chunks (recommended every 500ms).
- **Control Commands**: Only "NEXT" and "DONE" as text messages.

Server-to-Client:

The server returns recognition results as JSON in real-time:

```
{
  "prediction": [
    {
      "prediction": "Hello",
      "confidence": 0.93
    }
  ],
  "final_packet": false
}
```

Ending the Session:

Client sends "DONE" to terminate processing and close the connection.

Error Handling

- **Handshake Failure:** Sends error message and closes connection.
- **Invalid Messages:** Ignored and logged, connection continues.
- **Server Exceptions:** Logged internally and connection closed gracefully.

Front-end Usage Example (React)

```
const ws = new WebSocket('wss://api.signbridgeai.com/recognize_sign_ws');

ws.addEventListener('open', () => {
  ws.send(JSON.stringify({
    api_key: process.env.REACT_APP_API_KEY,
    protocol_version: "1.0",
    stream_type: "video",
    mode: "single"
  }));
});

mediaRecorderRef.current.ondataavailable = (event) => {
  const reader = new FileReader();
```

```
  reader.onloadend = () => ws.send(reader.result);
  reader.readAsArrayBuffer(new Blob([event.data], { type: mediaRecorderRef.current.mimeType })
};

mediaRecorderRef.current.start(500);

// Stop sending data
ws.send('DONE');
```

API Key Management

- Valid API keys are predefined server-side.
- Invalid API keys trigger an error response and logging.

Unified Error Handling Strategy

- Invalid API key responses use HTTP status 403.
- Other exceptions use HTTP status 500 uniformly.

```
{
  "error": "Internal Server Error",
  "status_code": 500
}
```

This document details the SignBridge AI ASL Recognition API interfaces. Implementation strictly adheres to this specification.