

# Face Detection API Documentation

## Overview

The Face Detection API allows users to upload an image and determine whether it contains a real face. The API evaluates the image using various metrics and returns a JSON response indicating if the face is real, along with additional details about the image. It is important that the image is taken directly from a **smartphone camera**, as uploading an older or edited photo may reduce the accuracy of the detection.

## Base URL

`https://veriface.info/api_upload`

## Authentication

The API requires authentication through a Bearer token provided in the Authorization header of the request. Register at <https://veriface.info> to obtain your secret key.

- Secret Key: `your_key`

## API Endpoints

Upload Image and Detect Real Face

- Endpoint: `/api_upload`
- Method: `POST`
- Content-Type: `multipart/form-data`
- Authorization: Bearer Token (Authorization: `Bearer your_secret_key`)

## Request Parameters

- `file` (required): The image file you want to upload and analyze. It should be included in the request as a form-data file field.

Sample CURL Command

```
curl -X POST -H "Authorization: Bearer your_secret_key" -F "file=@/path/imgfile.jpg" https://veriface.info/api_upload
```

## Response

The response is a JSON object containing the result of the face detection analysis. The following fields are included in the response:

- `status` (string): Indicates the status of the detection. Possible values are:
  - + **"no face detected"**: No face was detected in the image.
  - + **"too close"**: The detected face is too close to the camera.
  - + **"too far"**: The detected face is too far from the camera.
  - + **"Fake image"**: The image does not contain a real face.
  - + **"success"**: A real face was successfully detected.
- `result` (boolean): **True** if a real face was detected; otherwise, **False**.

## Example Responses

- No Face Detected:

```
{
  "status": "no face detected",
  "result": False
}
```

- Face Too Close:

```
{
  "status": "too close",
  "result": False
}
```

- Face Too Far:

```
{
  "status": "too far",
  "result": False
}
```

- Fake Image:

```
{
  "status": "Fake image",
  "result": False
}
```

- Real Face Detected:

```
{
  "status": "success",
  "result": True
}
```

## Error Handling

- 401 Unauthorized:

+ The request is missing or has an incorrect Authorization header.

Response:

```
{
  "error": "Unauthorized access"
}
```

- 400 Bad Request:

+ The request does not contain a file or the file part is missing.

Response:

```
{
  "error": "No file part"
}
```

+ The file was uploaded but no file was selected.

Response

```
{
  "error": "No selected file"
}
```

## **Notes**

- The uploaded image file is temporarily saved to the uploads/ directory on the server. After processing, the file is automatically deleted.
- Ensure that the image is clear, well-lit, and properly framed for the most accurate detection results.

## **Troubleshooting**

- Ensure the server is running and accessible via the IP address and port specified.
- Verify that the Bearer token is correctly included in the Authorization header.
- Check that the image file is correctly formatted and not corrupted.