## **1. Upload and Start Animation**

### **Endpoint:**

POST /predict

### **Description:**

Uploads a source image and driving video, then starts the animation process asynchronously.

### **Request:**

* **Form Data:**

### **Required Arguments:**

| **Argument** | **Type** | **Description** |
| --- | --- | --- |
| source | File | The source image or video to animate. |
| driving | File | The driving video that guides the animation. |

### **Optional Arguments:**

| **Argument** | **Type** | **Default** | **Description** |
| --- | --- | --- | --- |
| flag\_normalize\_lip | bool | False | Whether to normalize the lip state before animation. |
| flag\_relative\_motion | bool | True | Whether to use relative motion during animation. |
| flag\_do\_crop | bool | True | Whether to crop the source image or video. |
| flag\_pasteback | bool | True | Whether to paste the animated face back onto the original image. |
| flag\_stitching | bool | True | Whether to use stitching for smoother animations. |
| animation\_region | str | "all" | The region to animate (exp, pose, lip, eyes, all). |
| driving\_option | str | "expression-friendly" | Controls how the driving motion is applied (expression-friendly or pose-friendly). |
| driving\_multiplier | float | 1.0 | Multiplier for driving motion strength. |
| flag\_crop\_driving\_video | bool | False | Whether to crop the driving video. |
| scale | float | 2.3 | Scale factor for cropping the source. |
| vx\_ratio | float | 0.0 | Adjusts horizontal position of cropped face. |
| vy\_ratio | float | -0.125 | Adjusts vertical position of cropped face. |
| scale\_crop\_driving\_video | float | 2.2 | Scale factor for cropping the driving video. |
| vx\_ratio\_crop\_driving\_video | float | 0.0 | Adjusts horizontal position in the cropped driving video. |
| vy\_ratio\_crop\_driving\_video | float | -0.1 | Adjusts vertical position in the cropped driving video. |
| driving\_smooth\_observation\_variance | float | 3e-7 | Controls the smoothing of the driving motion. |

### **Response:**

* **Success (HTTP 200)**

|  |
| --- |
| {  "task\_id": "12345678-90ab-cdef-1234-567890abcdef",  "status\_url": "http://localhost:5000/task\_status/12345678-90ab-cdef-1234-567890abcdef"  } |

* **Error (HTTP 400)**

|  |
| --- |
| {  "error": "Please upload both source and driving files"  } |

## **2. Check Task Status**

### **Endpoint:**

GET /task\_status/<task\_id>

### **Description:**

Retrieves the current status of an animation task.

### **Request:**

* **Path Parameter:**
  + task\_id (string, **Required**) – The ID of the animation task.

### **Response:**

* **Success (HTTP 200)**

|  |
| --- |
| { "status": "PENDING" } |
| { "status": "FAILED", "error": "Some error message" } |
| {  "status": "COMPLETED",  "output\_file": "http://localhost:5000/download/animation.mp4"  } |

## **3. Download Output File**

### **Endpoint:**

GET /download/<filename>

### **Description:**

Allows users to download the generated animation file.

### **Request:**

* **Path Parameter:**
  + filename (string, **Required**) – The name of the generated animation file.

### **Response:**

* **Success (HTTP 200)**
  + Returns the requested animation file as a downloadable attachment.
* **Error (HTTP 404)**

|  |
| --- |
| { "error": "File not found" } |