

# Jester

## User Manual

---

### **Students**

- Paul Treanor - 16347251
- Colin Gorman - 17354073

### **Supervisor**

Dr. Paul Clarke

### **Date completed**

28/04/2021

# Table of Contents

## 1. Using the app

- [1.1 User guide](#)
- [1.2 Video demo](#)

## 2. Installing and hosting the Jester API

- [2.1 System Requirements](#)
- [2.2 Dependencies](#)
- [2.3 System Download](#)
- [2.4 API installation](#)
- [2.5 Running the server](#)

## 3. Hosting the frontend

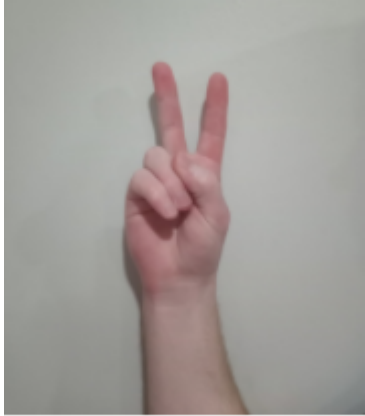


- [3.1 Download](#)
- [3.2 Running the app](#)
- [3.3 Connecting to a Jester server instance](#)
- [3.4 A note on SSL certificates](#)

## 4. Troubleshooting

- [4.1 Gesture recognition not working](#)
- [4.2 Device camera not showing up on app](#)
- [4.3 "Out of memory" error on server](#)
- [4.4 Slow response times](#)

# 1. Using the app

Jester is a web-app which allows users to control their device's camera using hand gestures. There are 3 camera functions which can be controlled by 3 separate gestures.

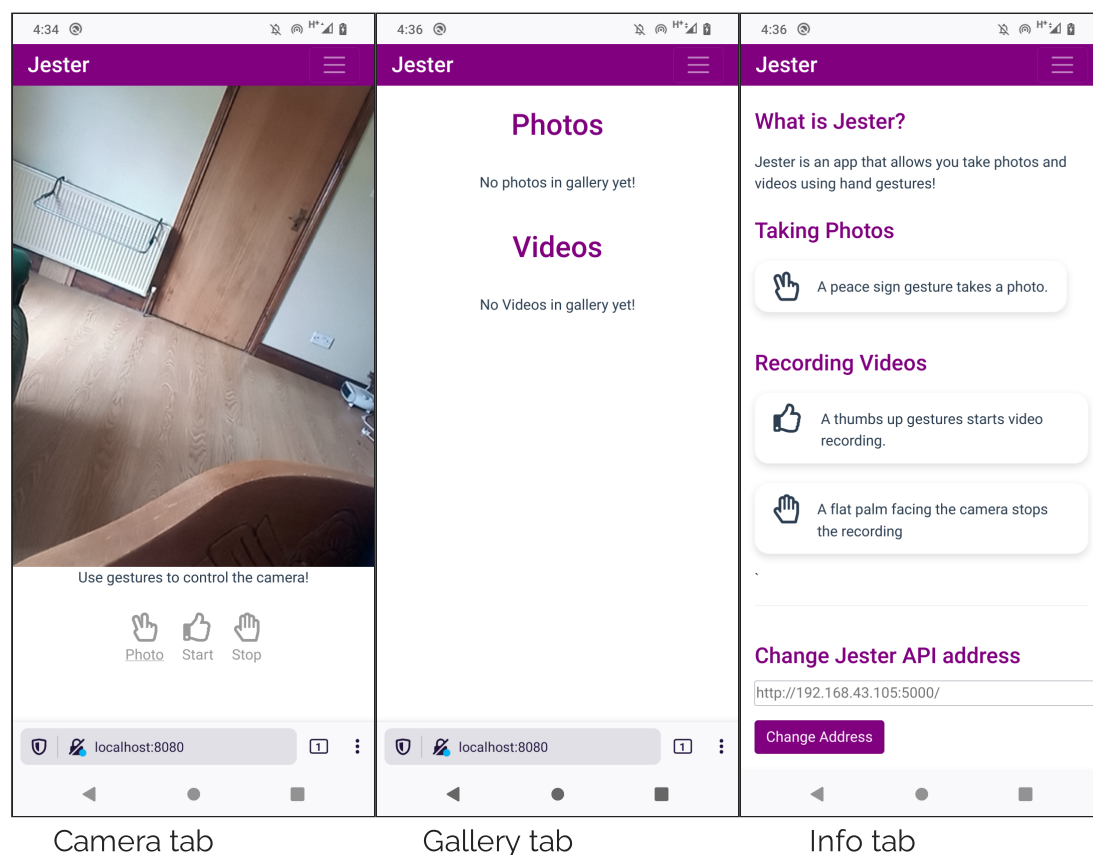
Camera function	Hand gesture	Example
Taking a photo	Peace sign	
Start video recording	Thumbs up	
Stop video recording	Flat palm	

## 1.1 User Interface

The camera, gallery, and information tabs can be reached via the navbar at the top of the app.

The 3 icons at the bottom of the camera tab represent each hand gesture. When a hand gesture is recognised the respective icon lights up. Camera actions are also triggered by pressing these buttons. Note that the stop video gesture (flat palm) will not be recognised unless a video is already being recorded.

Once a photo has been taken or a video has been recorded, they can be viewed and downloaded in the app's gallery.



## 1.2 Privacy

Upon opening the web app users will be prompted by their browser to confirm that they wish to give Jester access to their device's camera. If the user denies this, the gesture recognition functionality of the app will not work.

Users should be aware that while Jester does not store their images, the images are still being sent across a network to a server.

## 2.3 Video demo

Here is a brief video walkthrough showing how to use the system.

<https://youtu.be/QSDZ-cdyGXw>

## 2. Installing and hosting the Jester API

### 2.1 System Requirements

- Windows 10 64-bit
- NVidia GPU with at least 2GB of RAM

### 2.2 Dependencies

#### Python

- Supports version 3.6 or later
- Download from <https://www.python.org>.

### 2.3 System download

Download and extract the project's Gitlab repository from the following link:

<https://gitlab.com/computing.dcu.ie/treanop5/2021-ca400-ptreanor-cgorman>

### 2.4 API installation

To install the classification API run the following command from within the repository's /res directory:

```
python server_install.py
```

This will run a script which downloads and installs OpenPose in the correct location, and install required python modules.

### 2.5 Running the server

To start an Jester API server run the following command from the classifier folder:

```
python flask_server.py
```

## 3. Hosting the frontend

### 3.1 Download

The HTML, CSS and JavaScript files of the web app can be downloaded from the following Google Drive folder:

[https://drive.google.com/drive/folders/1zxs\\_RWOuMV7DQyo6X3XRd\\_wHXReXRkxL?usp=sharing](https://drive.google.com/drive/folders/1zxs_RWOuMV7DQyo6X3XRd_wHXReXRkxL?usp=sharing)

### 3.2 Running the web app

The web app can be served via HTTP or HTTPS. Jester cannot be run by opening the index.html file in a web browser.

The easiest way to host the app is with the npm http-server module, which requires Node.js to be installed on the machine. Run the following commands from the webapp folder to install the module and start a server running the web app.

```
npm install http-server -g
http-server
```

By default the server will run on <http://localhost:8080>

### 3.3 Connecting the web app to a backend server instance

The app will need to connect to an instance of the Jester API for gesture recognition to work. The address of the server you want to connect to can be edited from within the app's info tab.

1. Navigate to the web app's info tab and find the "Change Jester API address" input area.
2. Enter the server address in the form "http://<server IP address>:<port number>".

### 3.4 A note on SSL certificates

To access the device camera, the Jester web app must be hosted on a server with a valid SSL certificate. The Jester API server also requires an SSL certificate or else POST requests will fail on most web browsers.

## 4. Troubleshooting

### 4.1 The API and web app are running, but the gesture recognition is not working.

1. Ensure that the Jester API address the app is connecting to is correct. This address can be found and adjusted from within the app, via the info tab.
2. Ensure that the server running the Jester API has a valid SSL certificate.

### 4.2 The camera is not showing up on the web app.

1. Ensure that you are using a modern web browser that supports cameras. Firefox or chrome are recommended.
2. Ensure that the web app is being served from a server with a valid SSL certificate.

### 4.3 The server throw an "out of memory" error when accepting images.

This error is thrown by OpenPose and means that the server's GPU has run out of RAM while processing the image. OpenPose can be temperamental, and reinstalling [OpenPose version 1.6](#) into the project's /build folder can potentially fix the error.

While the Jester API has worked on old and low powered GPUs, the most reliable fix for this issue is to fit the server with GPU that has more RAM.

### 4.4 The response times of the gesture camera controls are very slow.

The primary performance bottleneck in the system is OpenPose (the Jester algorithms are comparatively very fast!), which is limited by the performance of the server's GPU. With that in mind running Jester on a machine with a more powerful GPU should improve the classifier's performance, although the author's have not been able to confirm this.

Network latency should also be considered when investigating response times, as many images are sent across the network each minute.

---

[Back to top](#)