

User Manual

Video Indexer Script

Follow the Video Indexer Script section of the System Manual first to have everything set up. Ensure that the config file is filled out with the required details.

Also ensure you have uploaded a video file to your Azure storage container so the script has something to analyse.

Once this has all be done, open up a command prompt and enter:

```
python videoindexer.py
```

This will run the script and it will begin to download the video from Azure, send it to Video Indexer for analysis, and return the labels of objects it sees in the video in a JSON format.

Camera Trap Simulator

Follow the camera trap simulator section of the System Manual to ensure you have everything set up correctly. Once this is done decide how you want to use the simulator. It has four options:

- Motion Detection with Live Video Feed
- Motion Detection with Pre-recorded Video Feed
- Machine Learning Model Based Detection with Live Video Feed
- Machine Learning Model Based Detection with Pre-recorded Video Feed

If you plan to use the Machine Learning based detection option, you will need to have a TensorFlow model on the same machine that the simulator will be run on. The path to this model will have to be provided to the simulator through a command line argument when it is run.

If you plan to use a pre-recorded video feed then ensure the video is on the same machine that the simulator will be run on. The path to the video will have to be provided to the simulator through a command line argument when it is run.

If using a live video feed, ensure your camera is not being used by any other application before running.

Before running the simulator also ensure you have set all the values in the config file.

To run the simulator type the following into a command prompt:

```
python CameraTrapSimulator.py
```

The command line parameters you need to provide depend on the video feed and detection options you wish to use. The below table explains how to provide the command line arguments:

<no command line arguments>	Runs with live video feed and motion based detection.
-v <path to pre-recorded video file>	Runs with pre-recorded video feed and motion based detection.
-m <path to machine learning model>	Runs with live video feed and ML based detection
-m <path to machine learning model> -v <path to pre-recorded video file> (order does not matter)	Runs with pre-recorded video feed and ML based detection.