System Manual

All code can be found at the following GitHub repository: https://github.com/ckoppula199/UCL-Microsoft-IXN-Final-Year-Project.

Video Indexer Script

In order to use the Video Indexer script you will need to create a Video Indexer account. This can be done by following this Microsoft documentation:

https://docs.microsoft.com/en-us/azure/media-services/video-indexer/video-indexer-use-apis

This documentation also shows you how to find your Video Indexer account ID, Video Indexer API key and your Video Indexer Region which you will need to provide to the accompanying config file.

You will also need an Azure blob storage account to which videos will be uploaded. To do this follow this Microsoft documentation:

https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blob-create-account-block-blob?tabs=azure-portal

You should then create a container within the blob storage account. Take note of the container name, storage account name and the storage account key (can be found by clicking on keys on the navigation tab on the left of the Azure portal page) as these will need to be provided to the accompanying config file.

Next upload a video to the container and take note of the name of the file being uploaded.

You will then need to install Python 3.8.x and PiP which can be done from https://www.python.org/.

Open a command prompt and navigate to the directory with the code, config file and requirements file in it. The type the following command:

```
pip install -r requirements.txt
```

This will install the required python modules.

Finally fill out the config file with the required details and you are now ready to use the Video Indexer script.

Machine Learning Models

Note: Only the image classification models have code since the object detection models were made using the cloud.

The ML models can be opened and viewed in two ways. The first is using Jupyter notebooks with python and installing all the required modules yourself. The easier way is to use Google Colab. Navigate to https://colab.research.google.com/.

Open up one of the notebook files using Colab and you can now see all the code used to make the ML models. If you wish to run the code, you will need to upload the Animals-10 from Kaggle by using the following link:

https://www.kaggle.com/alessiocorrado99/animals10.

Then upload the dataset to your google drive account. You will need to replace the following lines in the code with the paths to the folders in your google drive accounts.

```
!7z x "/content/drive/MyDrive/Data.zip"
!7z x "/content/drive/MyDrive/test.zip"
```

Now when you run the code it will ask you to login to your google drove account, it will then unzip the datasets and begin running the training and testing code.

Camera Trap Simulator

To run the camera trap simulator, first install Python 3.8.x and PiP from https://www.python.org/. Then open a command prompt and enter the following:

pip install -r requirements.txt

This will install all the required python modules. If you wish to use a pre-recorded video or a TensorFlow model with the simulator then ensure they are located somewhere on the same machine on which you are running the simulator code from.

You will also need to set up an Azure IoT Hub resource. This can be done by following the Microsoft Documentation:

https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-create-through-portal.

You should then follow this documentation to generate a connection string that will allow you to send data from the simulator straight to Azure IoT Hub. Add this connection string to the accompanying config file.

https://docs.microsoft.com/en-us/azure/iot-hub/quickstart-send-telemetry-python.

You are now ready to use the camera trap simulator.