**Megadetector install on raspberry pi 4b 8gb. Testing on ubuntu jammy aarch64 and raspios bullseye aarch64.**

**RUNNING OK on raspberry pi4b 8gb ram raspios bullseye**

FIRST NEED TO INSTALL GIVEN MINICONDA VERSION. ANACONDA DOES NOT INSTALL NEITHER LATEST MINICONDA

<https://raspberrypi.stackexchange.com/questions/137618/can-not-install-miniconda-on-raspberry-pi-4-b>

you need to download Miniconda3-py39\_4.9.2-Linux-aarch64.sh.

sudo wget http://repo.continuum.io/miniconda/Miniconda3-py39\_4.9.2-Linux-aarch64.sh

sudo /bin/bash Miniconda3-py39\_4.9.2-Linux-aarch64.sh

**This installs miniconda for the root user. To install for the current user download and:**

/bin/bash Miniconda3-py39\_4.9.2-Linux-aarch64.sh

Do not upgrade.

SET UP CONDA ENVIRONMENT FOR MEGADETECTOR AND INSTALL SUPPORT GIT REPOSITORIES

Instructions at

<https://github.com/microsoft/CameraTraps/blob/main/megadetector.md>

<https://github.com/Microsoft/cameratraps>

**Download into the home directory the megadetector version 5 a or b frozen model (runs on yolov and pytorch, do not use version 4 that runs on tensorflow and is very slow). Version 5a:**

<https://github.com/microsoft/CameraTraps/releases/download/v5.0/md_v5a.0.0.pt>

**Next download and install the git repositories**

mkdir ~/git

cd ~/git

git clone https://github.com/ecologize/yolov5/

git clone https://github.com/Microsoft/cameratraps

git clone https://github.com/Microsoft/ai4eutils

cd ~/git/cameratraps

NOTE THAT yolov had some bugs as per this link and they use other repository

<https://github.com/ultralytics/yolov5/issues/6948#issuecomment-1065213047>

**Need to create a conda env and install python packages. Instructions suggest doing automatically using:**

conda env create --file environment-detector.yml

In practice this command freezes because of unresolved packages. Have to open the file

<https://github.com/microsoft/CameraTraps/blob/main/environment-detector.yml>

Open this file with a text editor and do all the steps manually starting with creating a python 3.8 environment named cameratraps-detector:

**Enter first the ~/git/cameratraps directory, then issue the commands:**

cd ~/git/cameratraps

conda create -n cameratraps-detector python=3.8

**Then activate this environment**

conda activate cameratraps-detector

**And install the packages listed in the environment-detector.yml file**

Note that version numbers for cudatoolkit and cuddn can be current version not necessary to have those listed. pytorch and torchvision also ok with current versions from conda forge

ALL PACKAGES ARE IN CONDA FORGE but if necessary SEARCH IN ANACONDA and PYTORCH REPOSITORY

name: cameratraps-detector

channels:

- conda-forge

- pytorch

dependencies:

- python=3.8

# We pin Pillow to make it as likely as possible that images are loaded via a loader that's identical to the training environment

- Pillow=9.1.0

- nb\_conda\_kernels

- ipykernel

- tqdm

- jsonpickle

- humanfriendly

- numpy

- matplotlib

- opencv

- requests

# for running MegaDetector v4

# - tensorflow>=2.0

# for running MegaDetector v5

- pandas

- seaborn>=0.11.0

- PyYAML>=5.3.1

- pytorch::pytorch=1.10.1

- pytorch::torchvision=0.11.2

- conda-forge::cudatoolkit=11.3

- conda-forge::cudnn=8.1

**CONDA FORGE COMMANDS TO INSTALL THESE. NOTE THAT LATEST PACKAGES OF pytorch 1.13.1, torchvision 0.14.1, cudatoolkit 11.8.0 and cudnn 8.4.1.50 tested and worked ok. Installed pytorch and torchvision from conda forge and not pytorch because they had aarch64 build, running with raspios bullseye on rpi4b with 8gb ram :**

conda install -c conda-forge pillow=9.1.0

conda install -c conda-forge nb\_conda\_kernels

conda install -c conda-forge ipykernel

conda install -c conda-forge tqdm

conda install -c conda-forge jsonpickle

conda install -c conda-forge humanfriendly

conda install -c conda-forge numpy

conda install -c conda-forge matplotlib

conda install -c conda-forge opencv

conda install -c conda-forge requests

conda install -c conda-forge pandas

conda install -c conda-forge seaborn

conda install -c conda-forge pyyaml

conda install -c conda-forge pytorch

conda install -c conda-forge torchvision

conda install -c conda-forge cudatoolkit

conda install -c conda-forge cudnn

**THERE IS A BUG IN torch module upsampling.py that needs to be fixed as per**

[**https://stackoverflow.com/questions/72297590/attributeerror-upsample-object-has-no-attribute-recompute-scale-factor**](https://stackoverflow.com/questions/72297590/attributeerror-upsample-object-has-no-attribute-recompute-scale-factor)

In miniconda3/envs/cameratraps-detector/lib/python3.8/site-packages/torch/nn/modules/upsampling.py in line 153-154:

Change:

return F.interpolate(input, self.size, self.scale\_factor, self.mode, self.align\_corners,

recompute\_scale\_factor=self.recompute\_scale\_factor)

To:

return F.interpolate(input, self.size, self.scale\_factor, self.mode, self.align\_corners)

# recompute\_scale\_factor=self.recompute\_scale\_factor)

or

return F.interpolate(input, self.size, self.scale\_factor, self.mode, self.align\_corners,

# recompute\_scale\_factor=self.recompute\_scale\_factor

)

**AFTER INSTALL TEST THE MEGADETECTOR**

cd ~/git/cameratraps

conda activate cameratraps-detector

export PYTHONPATH="$PYTHONPATH:$HOME/git/cameratraps:$HOME/git/ai4eutils:$HOME/git/yolov5"

***Then you can run the script like this:***

python detection/run\_detector.py "$HOME/megadetector/md\_v5a.0.0.pt" --image\_file "some\_image\_file.jpg" --threshold 0.1

**Don't forget to change "some\_image\_file.jpg" to point to a real image on your computer and to put the correct path for the megadetector model - in this example the model is in the megadetector folder under the home directory.**

**To apply this model to larger image sets on a single machine, we recommend a different script, run\_detector\_batch.py**. This outputs data in the same format as our batch processing API, so you can leverage all of our post-processing tools. The format that this script produces is also compatible with Timelapse.

**To use this script on Linux/Mac, when you open a new Anaconda prompt, don't forget to do this:**

cd ~/git/cameratraps

conda activate cameratraps-detector

export PYTHONPATH="$PYTHONPATH:$HOME/git/cameratraps:$HOME/git/ai4eutils:$HOME/git/yolov5"

**Then you can run the script like this:**

python detection/run\_detector\_batch.py "$HOME/megadetector/md\_v5a.0.0.pt" "/some/image/folder" "$HOME/megadetector/test\_output.json" --output\_relative\_filenames --recursive --checkpoint\_frequency 10000

**THAT'S IT**