
foxmask

Release

Aug 02, 2017

Contents

1	Getting Started	3
1.1	Installation	3
1.2	Usage	5
1.3	Contribution	5
1.4	Code documentation	6

You will find on these pages information related to the installation and usage of FoxMask. It is intended to be read by users as well as developers.

- [Source code on GitHub](#)

Installation

We provide two methods for installing and using FoxMask:

- *Standalone installation on Linux*
- *Installation through a virtual machine* (Any platform)

Standalone installation on Linux

This is the recommended method for production.

This method will install FoxMask on your Linux computer. We provide an automated install script, which will take care of installing all the dependencies needed to run FoxMask as well as the software itself. It is made for **Ubuntu 16.04**. The file is located in `install-scripts/install-ubuntu16.sh` of the FoxMask repository. You can also download the file directly `sh` and run it. To execute the script, simply run the following command as a standard user. Note that you will need to have `sudo` rights.

```
$ sh install-ubuntu16.sh
```

Note: You can issue the install command from any directory, the script will download all the needed softwares in your home directory. Please do **not** run the script prefixed by `sudo`.

Once the installation is finished, you will have a `foxmask` directory in your home folder where you will find the software and a set of images to test the software. Congratulations, you are now ready to start using FoxMask on your Linux Box ! Consult our [Usage](#) page to get started.

This installation has been successfully tested on Ubuntu 16.04, on bare metal computers as well as on GCE (Google Cloud Engine) and VirtualBox 5.0.

Installation through a virtual machine

We do not recommend this installation for production.

We provide a complete virtual environment to run and use FoxMask. The main advantages of this installation is that you can rapidly get FoxMask running on any platform. However, there are important drawbacks to consider:

- Poor performances compared to standalone installation
- A bit heavy to manage (Vagrant + VirtualBox + share usb)

Despite theses drawbacks, we consider that using our FoxMask box can be very useful for first time users wanting to test the software, or for development purposes.

Note: Again, let me emphasize that this method is suitable to rapidly get FoxMask up and running on another platform than Ubuntu. However, anyone serious about implementing an automated image analysis pipeline with FoxMask should consider installing it on bare metal hardware running a Linux OS, or on a virtual infrastructure running *libvirt*. We provide a fully automated installation script, successfully tested on *Ubuntu 16.04*. Please see [Standalone installation on Linux](#) to install FoxMask on you Linux machine.

To run the FoxMask virtual machine on your Linux, Mac or Windows computer, you will need the following three open source softwares:

- [VirtualBox](#)
- [Vagrant](#)
- [Git](#)

Follow the installation procedure for your platform. Once theses softwares are installed, you are ready to install FoxMask by issuing the following command:

```
git clone https://github.com/edevost/foxmask.git
```

This will fetch all needed components to run the virtual machine. Once the clone command is completed, you can start the virtual machine by issuing the following commands:

```
cd foxmask
vagrant up
```

This will boot the virtual machine. You can also manage you virtual machine through VirtualBox software, and start or stop your machine from there. The start process will bring you to the Ubuntu desktop. The default user and password on this virtual environment is `vagrant` `vagrant`. Once the machine is ready, you can proceed to our [Usage](#) section to launch FoxMask on the provided set of images.

Testing on your own set of images

To test FoxMask on your own set of images, you will have to make your images available to the virtual machine. To do so, we recommend saving your image set on an usb drive, and make the drive available to VirtualBox. There are many good tutorials on the web explaining the process:

- <http://www.dedoimedo.com/computers/virtualbox-usb.html>
- <https://www.groovypost.com/howto/mount-usb-drive-virtualbox/>
- <https://techtooltip.wordpress.com/2008/09/22/how-to-use-host-usb-device-from-guest-in-virtual-box/>

Making this work will demand a bit of work and reading, but nothing out of reach of any computer users with minimal computer knowledge. The main steps to get your FoxMask virtual machine to have access to an usb drive plugged on your host computer are the following:

- Shutdown your FoxMask virtual machine if it is running
- Plug your usb drive on your computer (host)
- Install [Virtual Box 5.0](#) extension pack
- Activate usb controller on VirtualBox
- Add your drive

Once your drive is added, boot your FoxMask virtual machine and check if there is a link to your drive on the desktop. If everything went well, you will have direct access to your usb drive on your virtual machine ! You can then proceed to our [Usage](#) section to start using FoxMask on your set of images.

Usage

FoxMask program is shipped with a sample images directory (images/example/) where you will find 50 jpeg images of Arctic fox. We encourage you to test FoxMask on this sample of images.

To run FoxMask on the default provided set of images, you simply have to run the following command, from the foxmask directory:

```
python2 foxmask.py
```

This will launch the analysis process. You will see images being analyzed and mask created. At the end of the process, a CSV file is exported with the image names and the presence of detected objects (0 or 1). If required, masks and images with detected objects are also copied in subfolders.

Configuration

Configuration of where the images sets are as well as where to write results are all confined in the file `parameters.py`. In this file you will find configurable variables used when the program is executed:

imagesDir The directory where the images to analyze are located. This directory can be anywhere (shared folder, usb drive, local disk), it just need to be accessible by your machine running FoxMask. It can consist of one single folder or a set of folders. FoxMask will iterate through the folders and analyze them all.

outputDir The location where the results of the analysis will be written

maxgap The maximum time gap (in seconds) between two consecutive images to build the sequence.

minsize Minimum size of moving objects to be considered as an animal

Contribution

Contributions to the FoxMask project is highly welcome and encouraged ! You can contribute in many ways. Note that you need a GitHub account to contribute to the project.

Report issues

You can report any issues with the project in the [source code repository](#) on GitHub. These issues can be of any kind, wrong documentation, issues with installation or usage of FoxMaks. Please write as much information as possible when submitting your issue.

Contribute to the documentation

You can visit the [documentation pages](#) directly on GitHub and edit the documentation page you wish modify. You can then submit your modification.

Contribute to the code

You can easily contribute to the code by cloning the repository, checkout to a new branch, work on your branch, and push your branch to FoxMask repository. If you get a permission denied on push, please contact a FoxMask developer to grant you access. Example workflow:

```
git clone https://github.com/edevost/foxmask.git
git checkout -b my-new-feature
[work on your feature]
git add *your modified files*
git commit -m 'meaningfull message concerning your modification'
git push origin my-new-feature
```

You can then submit a pull request on GitHub. Thank you for contributing !

Code documentation

Setup

The following class will do all the preliminary setup required to run the code and clean temporary files and directories. One variable residing in `parameters.py` can be parametrised: **rmmasks**, defining if resulting masks should be kept. This could be useful, for example, for visual evaluation of the analysis.

Images to analyze and metadata

The following class will gather all images in the folder being analysed, and search in each image metadata for the `DateTimeOriginal`, which is the time the picture was taken. Based on the latter, groups of images are made.