aiGuard

Monitors security cameras images, uses AI to tag them and sends notifications.

Dependencies

```
# imageai dependencies
pip3 install tensorflow tensorflow-gpu numpy scipy opencv-python pillow matplotlib h5py keras
pip3 install imageai --upgrade
# custom dependencies
pip3 install watchdog piexif python-pushover
```

Running on a kvm virtual machine

As I developed this on my local machine I had no speed issues. However after installing on the final machine (a kvm virtual machine) I had a number of issues. First, when running the .py I had a `Illegal instruction (core dumped)`. This is similar to the problem described here. The first approach was to downgrade to tensorflow 1.5:

```
pip3 uninstall tensorflow
pip3 install tensorflow==1.5
```

this worked however the speed was terrible. Checking for CPU flags only gave:

```
grep flags -m1 /proc/cpuinfo | cut -d ":" -f 2 | tr '[:upper:]' '[:lower:]' | { read FLAGS; OPT="-march=native"; for flag in $FLAGS; do case "$flag" in "sse4_1" | "sse4_2" | "ssse3" | "fma" | "cx16" | "popcnt" | "avx" | "avx2") OPT+=" -m$flag";; esac; done; MODOPT=${OPT//_/\.}; echo "$MODOPT"; }
-march=native -mcx16 -mpopcnt
```

The solution was to stop the virtual machine and edit its configuration from:

```
<vcpu>1</vcpu>
```

to:

```
<cpu mode='host-passthrough' check='none'>
    <cache mode='passthrough'/>
</cpu>
<vcpu>1</vcpu>
```

after restarting the guest I had the following flags: `-march=native -mssse3 -mfma -mcx16 -msse4.1 -msse4.2 -mpopcnt -mavx -mavx2`, I was able to install the latest tensorflow version (1.14) and the speed greatly improved. (`virsh nodeinfo` shows the available number of cores you can use)