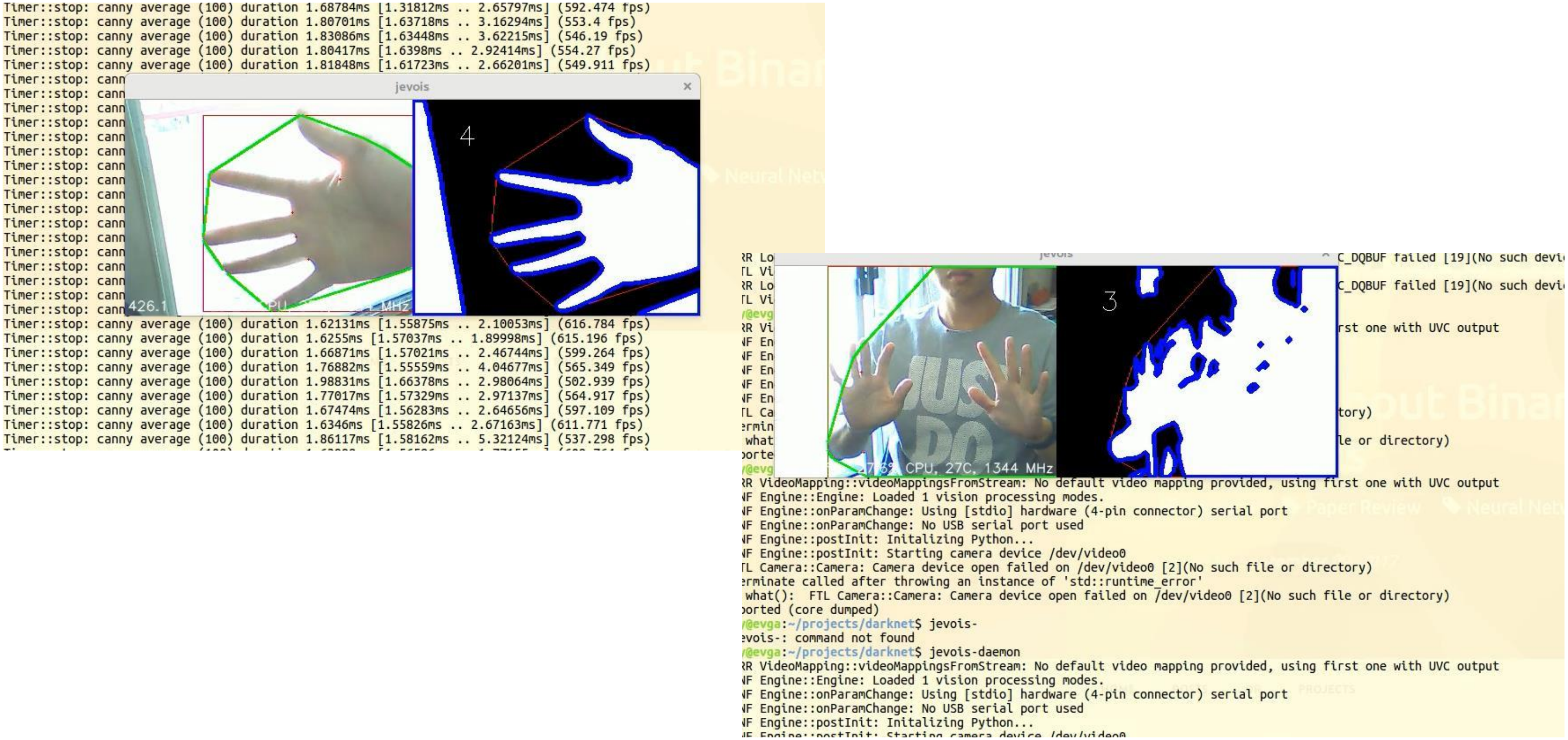
Interactive Gesture Plugin using Quantized Neural Network

Shaofan Lai

Gesture Recognition

- Traditional methods: (Filter+Threshold)
    - Sensitive to the background.
    - Hard-coded definition of gesture
- State-of-the-art Devices: (Kinect, Leap Motion)
    - Requires extra information (e.g. depth, infrared)
    - Expensive
- Deep Neural Network (Lab setting):
    - Intensive computation
    - Works with high-resolution video
    - Millions of data samples

Recognize Gesture with JeVois (OpenCV)

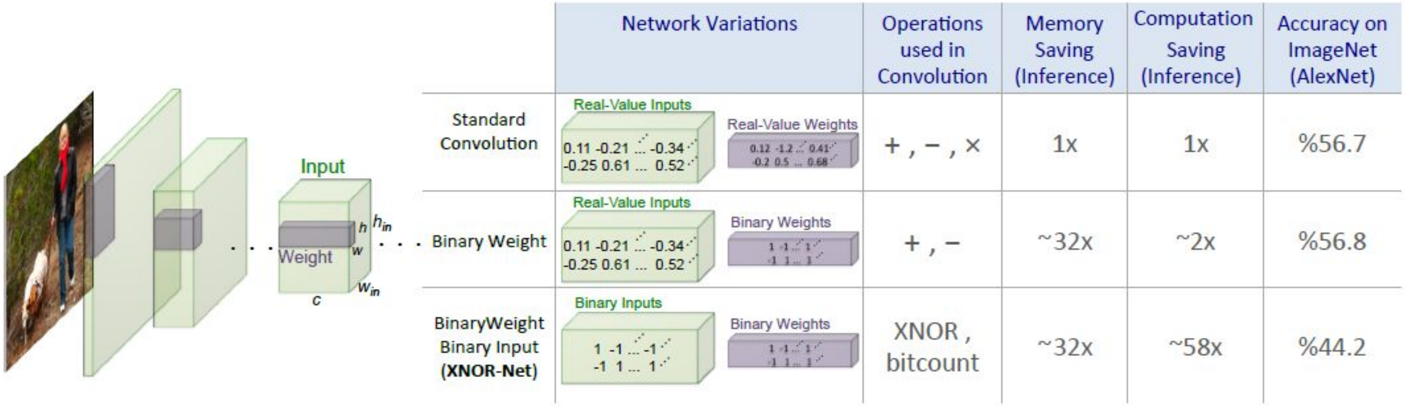


```
Timer::stop: canny average (100) duration 1.68784ms [1.31812ms .. 2.65797ms] (592.474 fps)
Timer::stop: canny average (100) duration 1.80701ms [1.63718ms .. 3.16294ms] (553.4 fps)
Timer::stop: canny average (100) duration 1.83086ms [1.63448ms .. 3.62215ms] (546.19 fps)
Timer::stop: canny average (100) duration 1.80417ms [1.6398ms .. 2.92414ms] (554.27 fps)
Timer::stop: canny average (100) duration 1.81848ms [1.61723ms .. 2.66201ms] (549.911 fps)
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: cann
Timer::stop: canny average (100) duration 1.62131ms [1.55875ms .. 2.10053ms] (616.784 fps)
Timer::stop: canny average (100) duration 1.6255ms [1.57037ms .. 1.89998ms] (615.196 fps)
Timer::stop: canny average (100) duration 1.66871ms [1.57021ms .. 2.46744ms] (599.264 fps)
Timer::stop: canny average (100) duration 1.76882ms [1.55559ms .. 4.04677ms] (565.349 fps)
Timer::stop: canny average (100) duration 1.98831ms [1.66378ms .. 2.98064ms] (502.939 fps)
Timer::stop: canny average (100) duration 1.77017ms [1.57329ms .. 2.97137ms] (564.917 fps)
Timer::stop: canny average (100) duration 1.67474ms [1.56283ms .. 2.64656ms] (597.109 fps)
Timer::stop: canny average (100) duration 1.6346ms [1.55826ms .. 2.67163ms] (611.771 fps)
Timer::stop: canny average (100) duration 1.86117ms [1.58162ms .. 5.32124ms] (537.298 fps)
```

```
RR Lo                                                                   C_DQBUF failed [19](No such devi
TL Vi
RR Lo                                                                   C_DQBUF failed [19](No such devi
TL Vi
/@evg
RR Vi                                                                   rst one with UVC output
NF En
NF En
NF En
NF En
NF En
TL Ca                                                                   tory)
ermin
 what                                                                   le or directory)
orte
/@evg
RR VideoMapping::videoMappingsFromStream: No default video mapping provided, using first one with UVC output
NF Engine::Engine: Loaded 1 vision processing modes.
NF Engine::onParamChange: Using [stdio] hardware (4-pin connector) serial port
NF Engine::onParamChange: No USB serial port used
NF Engine::postInit: Initalizing Python...
NF Engine::postInit: Starting camera device /dev/video0
TL Camera::Camera: Camera device open failed on /dev/video0 [2](No such file or directory)
erminate called after throwing an instance of 'std::runtime_error'
 what():  FTL Camera::Camera: Camera device open failed on /dev/video0 [2](No such file or directory)
orted (core dumped)
/@evga:~/projects/darknet$ jevois-
evois-: command not found
/@evga:~/projects/darknet$ jevois-daemon
RR VideoMapping::videoMappingsFromStream: No default video mapping provided, using first one with UVC output
NF Engine::Engine: Loaded 1 vision processing modes.
NF Engine::onParamChange: Using [stdio] hardware (4-pin connector) serial port
NF Engine::onParamChange: No USB serial port used
NF Engine::postInit: Initalizing Python...
NF Engine::postInit: Starting camera device /dev/video0
```

Code modified from: github.com/vipul-sharma20/gesture-opencv

Quantized (Binarized) Neural Network



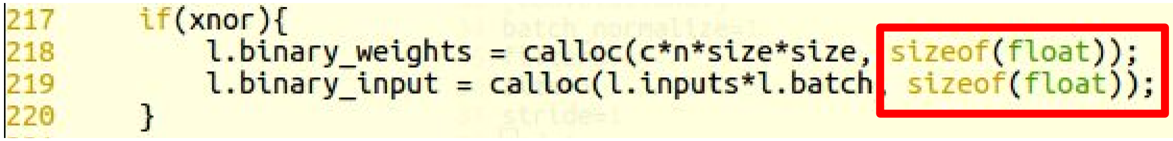
	Network Variations		Operations used in Convolution	Memory Saving (Inference)	Computation Saving (Inference)	Accuracy on ImageNet (AlexNet)
Standard Convolution	Real-Value Inputs 0.11 -0.21 ... -0.34 -0.25 0.61 ... 0.52	Real-Value Weights 0.12 -1.2 ... 0.41 -0.2 0.5 ... 0.68	+ , − , ×	1x	1x	%56.7
Binary Weight	Real-Value Inputs 0.11 -0.21 ... -0.34 -0.25 0.61 ... 0.52	Binary Weights 1 -1 ... 1 -1 1 ... 1	+ , −	~32x	~2x	%56.8
BinaryWeight Binary Input **(XNOR-Net)**	Binary Inputs 1 -1 ... -1 -1 1 ... 1	Binary Weights 1 -1 ... 1 -1 1 ... 1	XNOR , bitcount	~32x	~58x	%44.2

Fig. 1: We propose two efficient variations of convolutional neural networks. **Binary-Weight-Networks**, when the weight filters contains binary values. **XNOR-Networks**, when both weigh and input have binary values. These networks are very efficient in terms of memory and computation, while being very accurate in natural image classification. This offers the possibility of using accurate vision techniques in portable devices with limited resources.

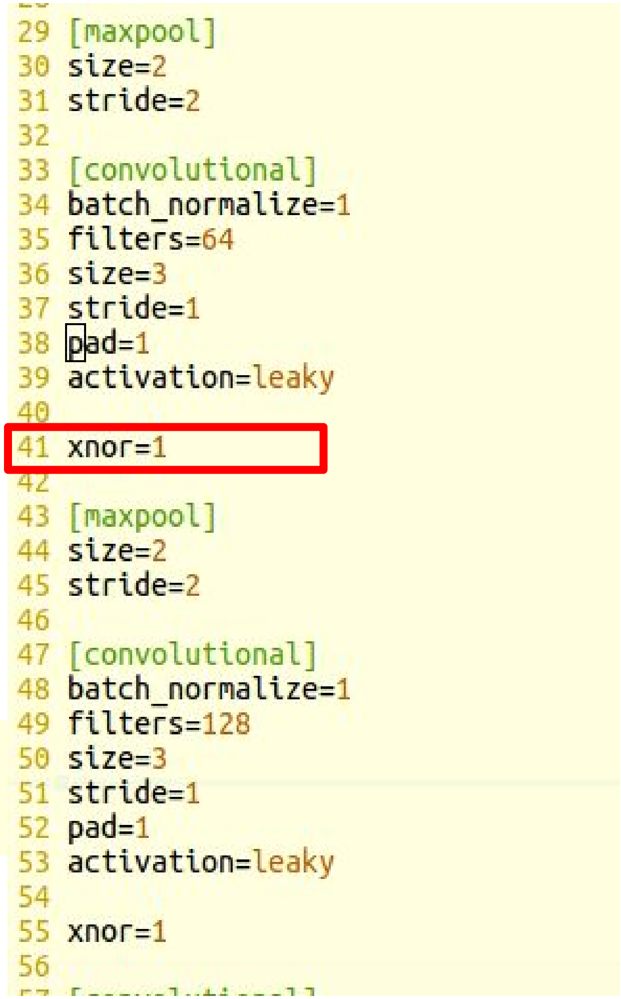
More details can be found on: https://shaofanlai.com/post/47

XNOR-Network with Darknet

- On CIFAR-10
  - Standard Network
    - top 1: 0.831600, top 5: 0.992000
  - Xnor Network
    - top 1: 0.684600, top 5: 0.976100
- Drawback
  - No time advantage
    - GPU (GTX 1080 Ti)
      - Standard~2.872s vs Bin~5.340s
    - CPU (Intel 7700K)
      - Standard~32.908s vs Bin~33.316s
  - No memory/storage advantage



```
    if(xnor){
        l.binary_weights = calloc(c*n*size*size, sizeof(float));
        l.binary_input = calloc(l.inputs*l.batch, sizeof(float));
    }
```



```
[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
activation=leaky

xnor=1

[maxpool]
size=2
stride=2

[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky

xnor=1

[convolutional]
```

My Proposal

Using quantized neural network to recognize the gestures in my workplace:

- Using only JeVois Camera(s)
- Relatively consistent background
- Customized gestures
- Interacting with the system in real time

Challenge

- Training and employment of the gesture model
    - Data (Collected in workplace? Transfered from other gesture datasets?)
    - Model (Attention-base? Detection-based? Tracking-based?)
    - Training (Unsupervised (with saliency)? Supervised (with labels)? Reinforcement Learning (with auxiliary sensors)?
- Implementation of the Quantized Neural Network
    - Memory usage improvement
    - Computation acceleration
- Hacking GNOME system plugin
    - Communication with JeVois
    - User Interface design

References

- *Binarized Neural Networks: Training Neural Networks with Weights and Activations Constrained to +1 or -1*
- *Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations*
- *XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks*