# Beginning

Storm environment, my jar, your code is enough for running.

Example

VIDEO 15 10 my\_video\_spout=”C:\example.avi”

HOG 1 6 18 true my\_HOG\_bolt=my\_video\_spout

FOUT 1 C:\example\_HOG\_feature.txt my\_outputToFile\_bolt=my\_HOG\_bolt

Basic idea:

Every line contains exactly one function, and every function is a bolt in Storm.

Function formulation:

Function\_Name Parameters Bolt\_name=Input\_names

# Conceptual

Based on Apache Storm, specialize for video processing.

For generality, it’s important to set a specification for the data tuples between bolts.

The basic tuple look like this:

Tuple(Image/Feature/Features, Filename, Pack, Frame, Patch, Scale, sPatch,…)

The Image is the picture data in opencv\_mat formulate.

Feature is a array\_list of Double

Features is a list of feature

Filename is the original video’s filename

Pack is the serial number of pack of frames. Commonly we will divide a video into small packs of frames.

Frame is the serial number of frame in this pack.

Patch is the serial number of patch in this frame. Sometimes we will cut a big picture into small patches and then process it

Scale is for the different scale processing

sPatch is meant to be used by dense trajectory, but now it’s useless

# Functions

## Video input spout

VIDEO *frames* *packs* *name*=”*input\_file*”

*frames* is the number of frames in one pack

*packs* is the number of packs will be generated

This spout will emit tuples one per frame, not one per pack!

Between consecutive packs, there will be [*frames*/2] frames overlapped.

For example, Pack 1 will have [0,1,2,3,4,5] frame, Pack 2 will have [3,4,5,6,7,8]

Input file is recommended in avi formulate

## Optical flow bolt

LK\_OF x side sigma thres name=input1(,input2…)

X is the parallelism, i.e. the number of worker will set for this bolt.

This function actually use the opencv’s optical flow algorithm, the side and sigma parameter is useless for now.

Thres is the limit of the maximum optical flow vector length, exceed this will be considered as error

Bolts can have any numbers of inputs.

The optical flow algorithm needs consecutive 2 frames, so this bolt will buffer a number of frames and wait for their consecutive frame. Using a hashmap and information of tuple to implementing this.

## Dense trajectory bolt

DENSETRAJ x frame median\_side traj\_side name=input1(,input2…)

X is the parallelism

Frame is the number of optical flow tuples for one pack. Need this in parameter is because I haven’t figure out a good way to automatically judge a pack is completed received.

Median\_side is the length for square for calculate the displacement of point

Traj\_side is the length of square that dense trajectory will generate

Dense trajectory bolt’s input must be optical flow bolts. And it will generates lots list of small patches of optical flow field and image in this pack as output, every list contain one trajectory’s patches in a pack.

## HOG,HOF,MBH bolt

HOG x side bin is\_single name=input1(,input2…)

HOF x side bin is\_single name=input1(,input2…)

MBH x side bin is\_single name=input1(,input2…)

HOG x side bin is\_single name=input1(,input2…)

X is the parallelism

Side and bin is the parameter for hog feature

Is\_single is a Boolean value (‘true’/’false’) to set the mode of bolt. If it’s true, the hog bolt will treat the input as only one patch. Otherwise, the bolt will first divide the input into patches.

The HOG,HOF,MBH,Traj\_shape bolts have a same feature, that is: input can be a list of images. And the bolts will automatically process them and combine the output feature list into one. This feature is useful for the dense trajectory processing.

## Feature vector merge

FMERGE x n name=input1(,input2…)

X is the parallelism

N is the number of different feature

This bolt will combine different type of feature vectors(the Double list) into one big vector. The order of feature maybe different from the input but will be constant.

## sPatch merge

SPMERGE x n name=input1(,input2…)

N is the number of spatches dense trajectory will generate

This bolt will add features into one feature list. When received n different spatches, it will emit the list.

## BOW bolt

BOW x filename name=input1(,input2…)

Filename is the path of the dictionary for BOW.

BOW bolt will convert list of features into one BOW feature, with feature type “BOW\_name”

## SVM bolt

SVM x filename name=input1(,input2…)

Using libsvm.

Filename is the path of libsvm model file.

It will predict the input feature using the model.

Output needs to be determined.

# Example

VIDEO 15 4 A1="C:\testing\label\_0.avi"

VIDEO 15 4 A2="C:\testing\label\_1.avi"

LK\_OF 1 4 1.0 100.0 B=A1,A2

DENSETRAJ 1 14 3 4 C=B

HOG 1 6 18 true D=C

HOF 1 3 8 true E=C

MBH 1 3 32 true F=C

TRAJSHAPE 1 C1=C

SPMERGE 4 600 H1=D

SPMERGE 4 600 H2=E

SPMERGE 4 600 H3=F

SPMERGE 4 600 H4=C1

BOW 1 C:\testing\trainedBOW\_HOG.txt I1=H1

BOW 1 C:\testing\trainedBOW\_HOF.txt I2=H2

BOW 1 C:\testing\trainedBOW\_MBH.txt I3=H3

BOW 1 C:\testing\trainedBOW\_TRAJ.txt I4=H4

FMERGE 1 4 G=I1,I2,I3,I4

SVM 1 C:\testing\SVM.model J=G