**THEORY:**

* Positive samples created from few positive images by superimposing positives on negatives at diff angles and positions
* Negative image size (100X100) should be > positive image size (50X50)
* Suppose there are 3 positive images taken and 2000 negatives - we use opencv\_createsamples separately on each positive image to generate 3x2000=6000 positive samples. Then while training we must train against 6000/2=3000 negative images
* num\_pos = num\_neg \* 2
* negative description file - path to each negative image - bg.txt
* positive description file - path to each positive image, no of objects, rectangle coordinates (x, y, w, h) – info.lst – this file is generated by first call to opencv\_createsamples
* positive vector file – postives.vec – generated by second call to opencv\_createsamples
* If you quit traincascade command, you can continue command ONLY if same parameters used the next time
* Negatives URLS:

http://image-net.org/api/text/imagenet.synset.geturls?wnid=n07942152 people

http://image-net.org/api/text/imagenet.synset.geturls?wnid=n13104059 trees

**STEPS:**

1. **mkdir workspace and cd workspace**

2. **store\_raw\_images()**

grabs negative images at url, convert to grayscale, resize, and save to folder called neg

3. Rerun previous prog for 2nd neg image url with updated pic\_num

4. Copy one broken\_link img to uglies directory. run **find\_uglies().** Deletes all uglies by comparing to main ugly

xor(ugly, question) = false if ugly and question are same. not(false) = true

5. Create desc files. – **create\_pos\_and\_neg().** In our case only bg.txt is created since we are using samples

6. **mkdir data** -- cascade xml files will go here

**mkdir info** -- pos samples will go here

7. Resize posimage to width and height less than 100x100 – resize\_pos()

8. **opencv\_createsamples -img posimage.jpg -bg bg.txt -info info/info.lst -pngoutput info**

**-maxxangle 0.5 -maxyangle 0.5 -maxzangle 0.5 -num littlelessthannumberofnegimages**

9. **opencv\_createsamples -info info/info.lst -num sameasprevcommand -w 30 -h 40 -vec positives.vec**

generates positive vector file. high width, height will cause higher training time

10. **opencv\_traincascade -data data -vec positives.vec -bg bg.txt -numPos 1800 -numNeg 600 -numStages 12 -w 40 -h 60 -featureType LBP**

numPos should have leeway of atleast 300 from total number of positive samples

numpos = 0.9 x numberofsamples

if 0-11 are done, train with numStages 12

**9/3/18**

Reached 0.009 acceptance ratio bike.jpg 40x60 12 stages 1600 pos 800 neg

Truck.jpg 20x80

**11/3/18 - Trying To Train A Face Cascade Using Merge Of Positives.Vec**

- downloaded fruits, flowers, trees, mountains negatives. resized to 100x100. cleaned uglies. total 4800 images. renumbered neg images

- 5 pictures of my face. All cropped to ~ 270x370. resized all me.jpgs to 45x60

- created bg.txt

- opencv\_createsamples -img r1.jpg -bg bg.txt -info info1/info.lst -pngoutput info1 -maxxangle 0.5 -maxyangle 0.5 -maxzangle 0.5 -num 3350

- repeat with all 5 images, changing info1 to info2 and so on

- created dir ‘vecs’

- opencv\_createsamples -info info1/info.lst -num 3350 -w 45 -h 60 -vec vecs/positives1.vec

- repeat for 2-5

- put mergevec.py in same dir. python mergevec.py -v vecs -o finalpos.vec

- to test o/p of mergevec.py, opencv\_createsamples -w 45 -h 60 -vec finalpos.vec

- opencv\_traincascade -data data -vec finalpos.vec -bg bg.txt -numPos 1800 -numNeg 3700 -numStages 12 -w 45 -h 60 -featureType LBP

- 4 hrs later, training terminated after stage 9. Ar was 0.0003. got 4 features until stage 8. Stage 9 had 3. Xml had good accuracy with (gray,1.3,30) for detect multiscale function

- trained 1 more stage. Ar 0.0001. 4 features. I think 9 was better

- to further improve accuracy, note false +ve and include hard mining. And increase image size a little (58x78) and number of samples (3700x5400).

**12/3/18**

* Bike 64x86 6 images
* Truck 80x110 3 images
* Created samples for every image with 5750 negatives
* opencv\_traincascade -data data -vec finalpos.vec -bg bg.txt -numPos 3800 -numNeg 5800 -numStages 10 -w 64 -h 86 -featureType LBP
* reached stage 9 after 12 hrs with acceptanceRatio of 0.0003

**24/3/18**

* 6 car images of our 3 toy cars, 70x70
* 6042 negative images 100x100 out of which 242 images of our prototype background
* opencv\_createsamples -img 1.jpg -bg bg.txt -info info1/info.lst -pngoutput info1 -maxxangle 0.5 -maxyangle 0.5 -maxzangle 0.5 -num 6000
* opencv\_createsamples -info info1/info.lst -num 6000 -w 70 -h 70 -vec vecs/positives1.vec
* opencv\_traincascade -data data -vec finalpos.vec -bg bg.txt -numPos 3000 -numNeg 6000 -numStages 10 -w 70 -h 70 -featureType LBP

**26/3/18**

* 6340 negative images with 149 hard mining images of road
* opencv\_createsamples -img 1.jpg -bg bg.txt -info infonew/info1/info.lst -pngoutput infonew/info1 -maxxangle 0.5 -maxyangle 0.5 -maxzangle 0.5 -num 6300
* opencv\_createsamples -info infonew/info1/info.lst -num 6300 -w 70 -h 70 -vec vecsnew/positives1.vec
* python mergevec.py -v vecsnew -o finalpos.vec
* opencv\_traincascade -data datanew -vec finalpos.vec -bg bg.txt -numPos 4000 -numNeg 6300 -numStages 16 -w 70 -h 70 -featureType LBP