

---

# OpenCV Haar Classifier

## 학습 방법

---

1. Github repository 를 클론합니다.

```
git clone https://github.com/mrnugget/opencv-haar-classifier-training
```

2. ./positive\_images 디렉토리에 positive image 들을 넣고 그 이미지들의 리스트 txt 파일을 만듭니다.

```
find ./positive_images -iname "*.jpg" > positives.txt
```

3. ./negative\_images 디렉토리에 negative image 들을 넣고 그 이미지들의 리스트 txt 파일을 만듭니다.

```
find ./negative_images -iname "*.jpg" > negatives.txt
```

4. positive samples 을 bin/createsamples.pl 스크립트로 만들고 ./samples 디렉토리에 저장합니다.

```
perl bin/createsamples.pl positives.txt negatives.txt samples 1500\  
"opencv_createsamples -bgcolor 0 -bgthresh 0 -maxxangle 1.1\  
-maxyangle 1.1 maxzangle 0.5 -maxidev 40 -w 80 -h 40"
```

5. tools/mergevec.py 로 ./samples 디렉토리 안의 samples 들을 병합합니다.

```
python ./tools/mergevec.py -v samples/ -o samples.vec
```

Note: struct.error: unpack requires a string argument of length 12

만약 위와 같은 에러가 생기면, **samples** 디렉토리로 이동한 후, length 0 의 모든 파일을 제거하세요.

6. opencv\_traincascade 로 트레이닝을 시작합니다.결과는 ./classifier 에 저장됩니다.

```
opencv_traincascade -data classifier -vec samples.vec -bg negatives.txt\  
-numStages 20 -minHitRate 0.999 -maxFalseAlarmRate 0.5 -numPos 1000\  
-numNeg 600 -w 80 -h 40 -mode ALL -precalcValBufSize 1024\  
-precalcIdxBufSize 1024
```

더 빠른 학습을 원한다면, LBP 특징 추출 타입으로 설정할 수 있습니다.

```
opencv_traincascade -data classifier -vec samples.vec -bg negatives.txt\  
-numStages 20 -minHitRate 0.999 -maxFalseAlarmRate 0.5 -numPos 1000\  
-lbp True
```

```
-numNeg 600 -w 80 -h 40 -mode ALL -precalcValBufSize 1024\
-precalcIdxBufSize 1024 -featureType LBP
```

Training 프로그램이 시작되면 Parameter 들이 출력되고 Training 이 시작됩니다. 각각의 스테이지에서 각각의 트레이닝 결과가 출력됩니다.

```
===== TRAINING 0-stage =====
<BEGIN
POS count : consumed   1000 : 1000
NEG count : acceptanceRatio   600 : 1
Precalculation time: 11
+-----+-----+-----+
|  N  |      HR      |      FA      |
+-----+-----+-----+
|   1 |           1 |           1 |
+-----+-----+-----+
|   2 |           1 |           1 |
+-----+-----+-----+
|   3 |           1 |           1 |
+-----+-----+-----+
|   4 |           1 |           1 |
+-----+-----+-----+
|   5 |           1 |           1 |
+-----+-----+-----+
|   6 |           1 |           1 |
+-----+-----+-----+
|   7 |           1 | 0.711667 |
+-----+-----+-----+
|   8 |           1 |    0.54 |
+-----+-----+-----+
|   9 |           1 |    0.305 |
+-----+-----+-----+
END>
Training until now has taken 0 days 3 hours 19 minutes 16 seconds.
```

각각의 row 는 training 된 feature 를 표현하고, HitRatio 와 FalseAlarm ratio 입니다. training 스테이지가 작은 숫자들(e.g. N = 2) 까지 형성되면, training data 는 잘못 입력된 것일 가능성이 있습니다.

각각의 classifier 단계의 끝은 파일로 저장되고 프로세스가 멈추고 다시 시작합니다. Training 속도 최적화를 위한 setting 과 같습니다.

7. 프로세스가 끝날 때까지 긴 시간동안 기다리면 됩니다.

8. 최종 완료된 학습 xml 파일을 사용하면 됩니다.

```
cd ~/opencv-2.4.9/samples/c
chmod +x build_all.sh
./build_all.sh
./facedetect --cascade="~/finished_classifier.xml"
```