**OCR 개발자 가이드**

**김동환**

**신민재**

**목차**

**1. 개발 목표**

**2. 사용 기술**

**3. 개발 과정**

**3.1 tesseract-ocr 설치**

**3.2 Jtessboxeditor 글자 데이터 학습**

**3.2.1 jtessboxeditor내의 글꼴 학습**

**3.2.2 이미지 파일 학습**

**3.2.3 학습 데이터 구축중 주의 사항**

**4. 문제점 및 해결**

**1. 개발 목표**

OCR프로그램을 이용하여 한글 및 영어 이미지 파일 글자 인식

**2. 사용 기술**

1. Tesseract-OCR

- Apache License 2.0에 따라 배포되는 무료 소프트웨어

- 구글에서 개발 후원

2. 영상처리

- OCR 대상 이미지의 정확성 향상

3. Jtessboxeditor

- OCR 학습

**3. 개발 과정**

1. tesseract-ocr 설치

<https://tesseract-ocr.github.io/tessdoc/Home.html>에서 자신에게 맞는 버전다운로드

2. 글자 데이터 학습

-글자 데이터 학습을 도와줄 프로그램으로 jtessboxeditor를 사용

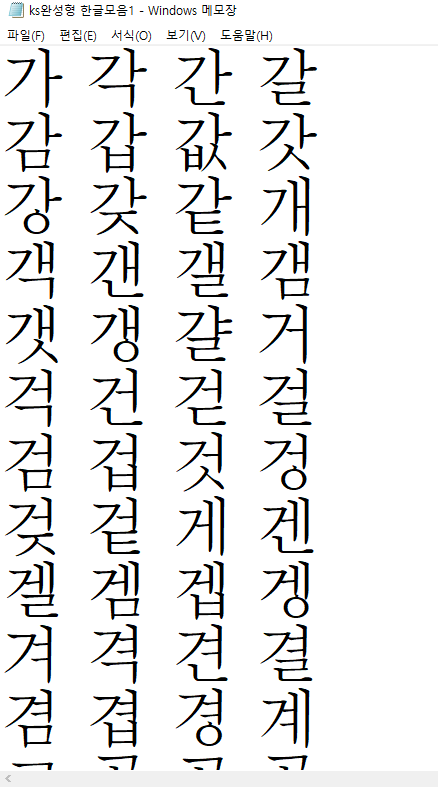
기존의 글자 학습 에디터나 명령어로 데이터 학습 시 한글에 대한 정확도가 많이 떨어진다.

Jtessboxeditor의 경우 프로그램안에 입력되어 있는 몇 가지 영어와 한글 글꼴 학습이 가능하고 박스 화 된 이미지를 가져와 프로그램내에서 수정 후 학습이 가능하다.

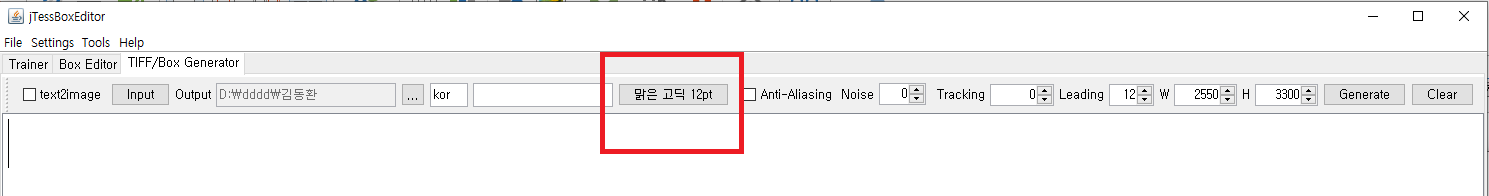
2.1 jtessboxeditor내의 글꼴 학습

1.훈련 시킬 한글 글자 모음 텍스트 파일을 준비한다.

인코딩을 UTF-8으로 해야한다.



2. jtessboxeditor TIFF/BOX generator

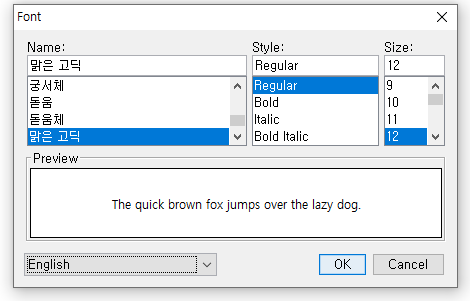


-TIFF/Box generator로 이동

-Input으로 준비한 한글 글자 모형 텍스트파일을 입력

-Output으로 파일이 생성 될 폴더 지정

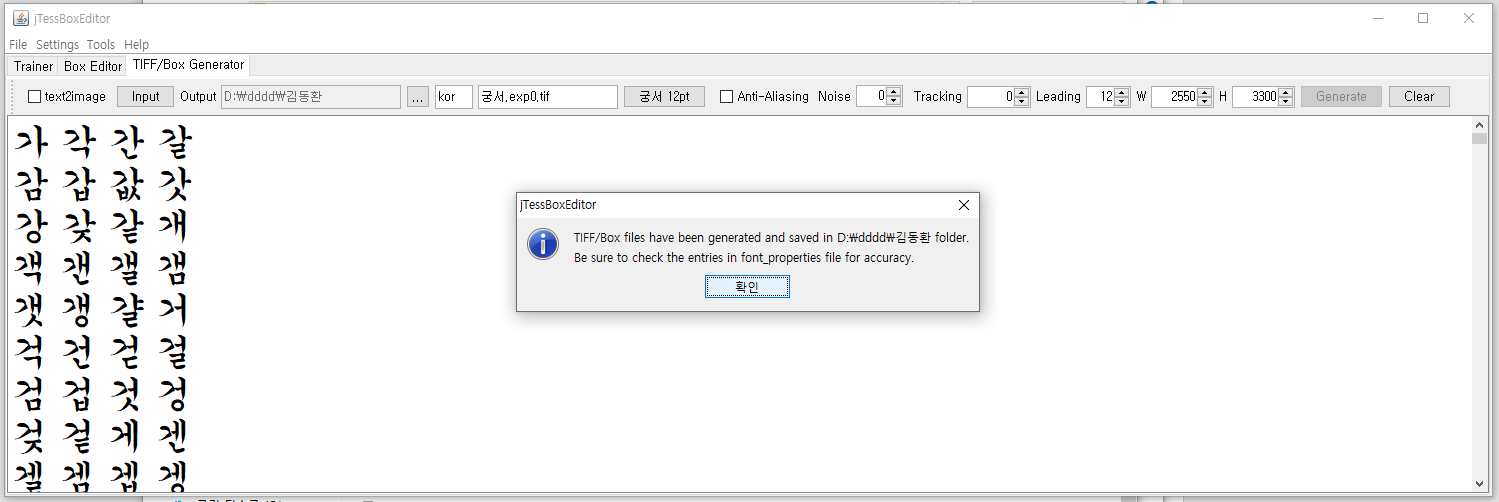
-빨간색 상자를 클릭해 글꼴 설정 후 빨간 박스 왼쪽에서 kor(영어일 경우 eng) 입력

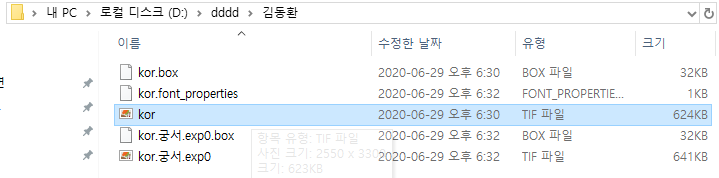


(원하는 글꼴 선택가능)

-generate 클릭

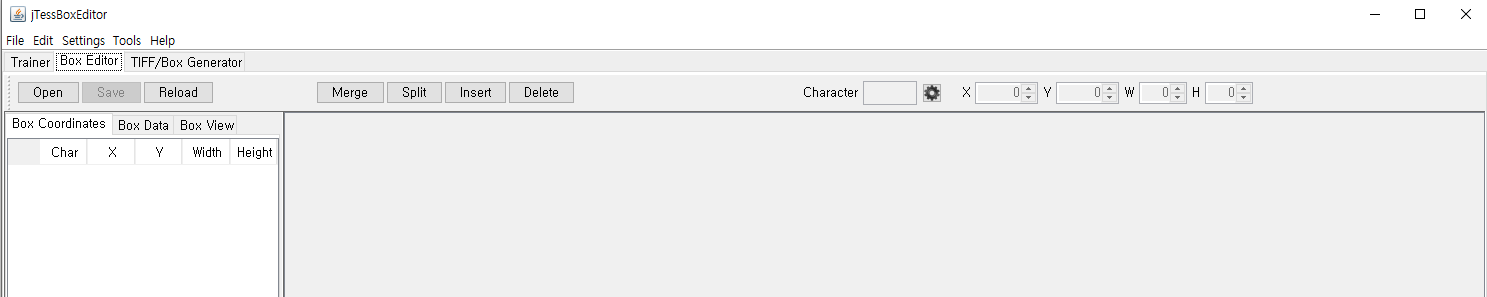
Generate된 후의 화면



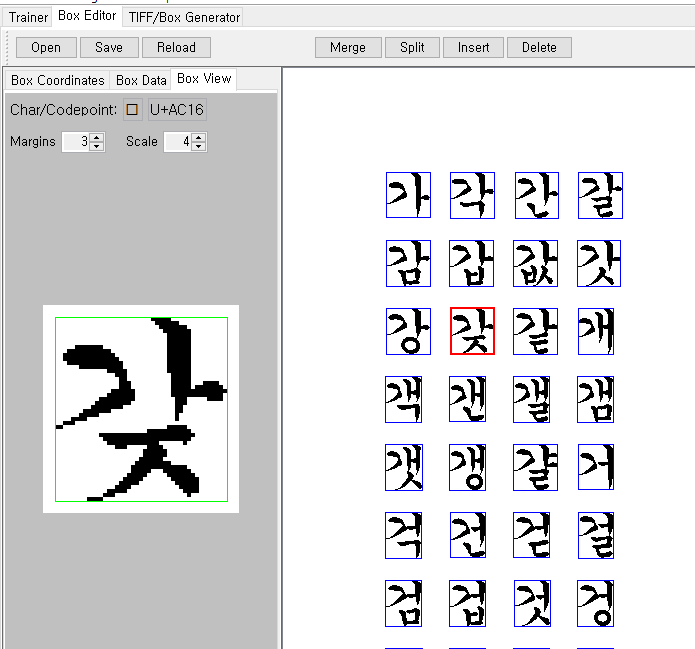


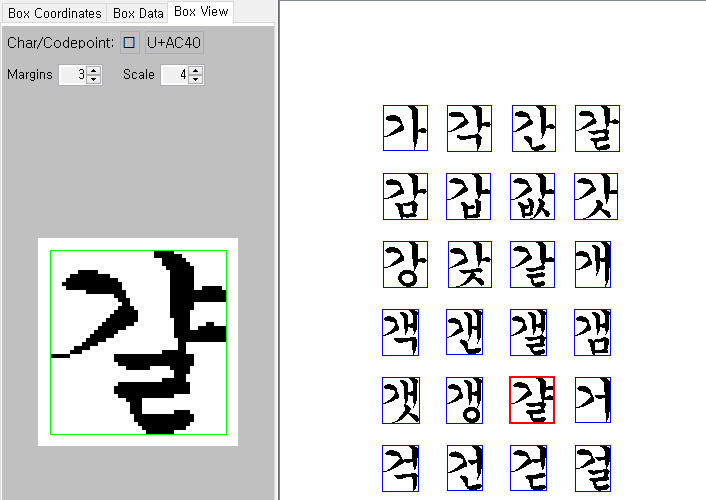
-Output에서 지정한 폴더에 생성된 tif파일 확인

3 Box Editor로 설정



-open으로 위에서 생성된 kor.궁서exp0.tif 파일을 열어준다.

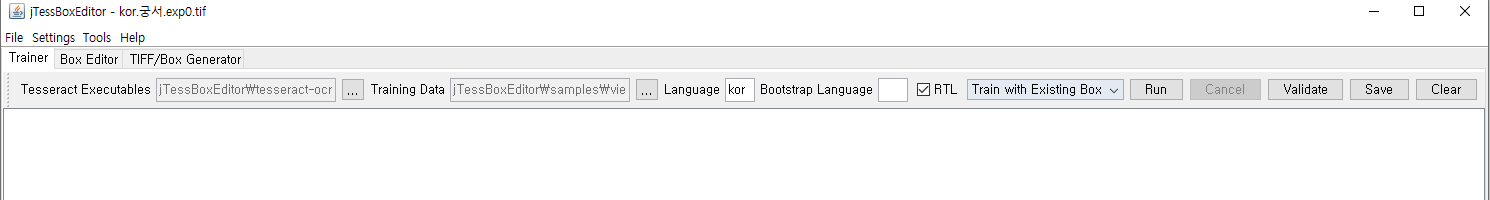




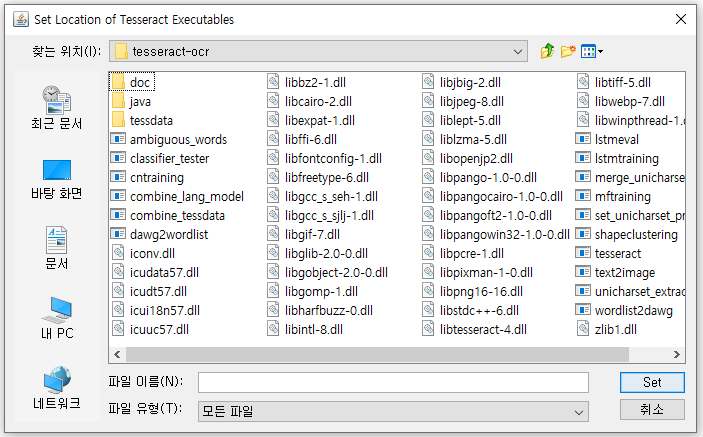
빨간 박스 안의 글자가 왼쪽 화면(box view)에 나와있다.

이미지가 박스 모양으로 나누어져 있고 원하는 글자 클릭 시 모양을 볼 수 있으며 글자가 틀렸을 경우 수정 가능하고 수정후 save를 눌러준다.

4. Trainer에서 OCR 훈련



Tesseract executables에서 학습 시킬 수 있는 파일 경로 선택

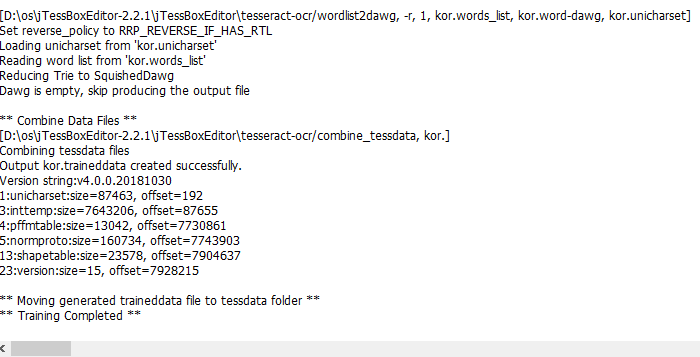


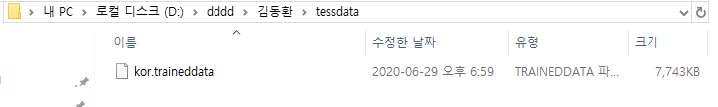
(해당 경로 내의 아무 dll파일 선택 후 set)

-Training Data에서 BOX에디터로 수정 된 kor.궁서exp0.tif 파일 선택

-Langauge에서 kor입력, RTL체크, Train with Existing Box 선택 후 run클릭해서 트레이닝 시작

Training 완료 화면





-생성된 학습 데이터 확인

-기존의 tesseract-ocr파일에 있는 한글 학습 파일을 지우고 새로운 학습 파일을 넣어준다.

**2.2 이미지 파일을 학습**

1. tif나 tiff의 이미지파일 준비

D:\dddd\alala\exp0.tiff

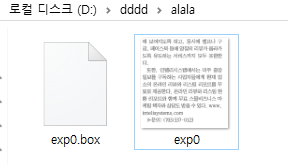
위와 같은 exp0.tiff 라는 이미지 파일을 준비하였다.

2. 이미지파일을 박스 파일로 변환

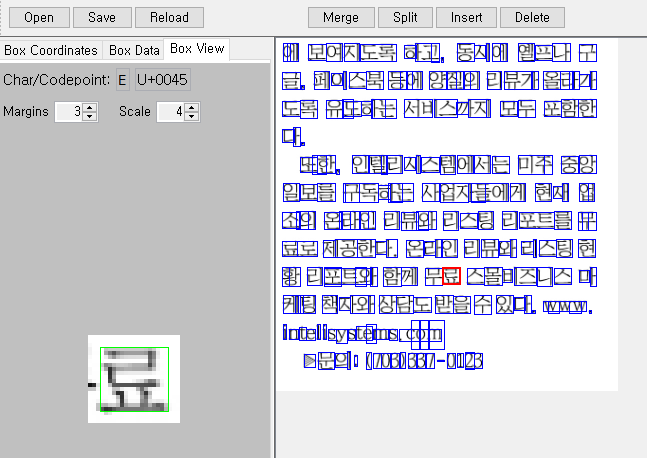
- cmd 명령어: [파일명].tif [파일명] batch.nochop makebox

- [파일명].box 파일 생성

> teseract.exe exp0.tiff exp0 batch.nochop makebox

C:\Users\사용자\Desktop\box.png  
  
3. 생성된 박스 파일 확인

4. jtessboxeditor에서 BoxEditor에 들어간 후 open으로 방금 생성된 박스 파일 오픈



5. 박스 파일 확인 후 수정 및 저장

6. 수정한 BOX파일을 이용하여 Tesseract OCR엔진을 학습한다. 아래 명령으로 tr파일과 text파일이 생성이된다. (jtessboxeditor 내의 글꼴로 수정 된 tif이미지도 적용가능)

-cmd tesseract.exe [수정된 박스 파일] [학습 후 이름] notbatch box.train

> tesseract.exe exp0.tiff exp0 nobatch box.train

7. unicharset 파일을 생성한다.

Tesseract가 출력하는 가능한 모든 문자들의 집합을 생성하는 과정이다.

-cmd unicharset\_extract.exe [box이름]

> unicharset\_extract.exe exp0.box

8. font\_properties 파일을 생성

Tesseract가 해당 폰트를 인식하였을 때, 출력할 폰트 스타일 정보를 포함하고 있다.

- cmd 명령어 – 각 옵션에 0 또는 1

- [폰트명] <italic> <bold> <fixed> <serif> <fraktur>

- 예) myfont 0 0 0 0 0

9. 클러스터링을 수행

문자의 윤곽 특징을 추출 후 prototype을 생성하기위해 클러스터링 실행

 - cmd 명령어: mftraining.exe –F font\_properties –U unicharset [언어 문자열]. unicharset [파일명].tr

> mftraining.exe -F font\_properties -U unicharset -O exp0.unicharset exp0.tr

> cntraining.exe.exp0.tr

10. 생성된 파일명을 변경한다.

- unicharset inttemp pffmtable normproto 파일명 앞에 언어 문자열 붙이기

- 예) exp0.unicharset

11. 단어 사전 등록

등록할 단어 리스트 파일 word\_list를 만들고 dawg를 생성해야한다.

> Wordlist2dawg word\_list exp.word-dawg exp.unicharset

-수 천, 수 만개의 등록된 단어를 빠른 시간내에 검색하기위해 단어를 DAWG(directed acyclic word graphs)의 자료형태로 저장한다.

-단어가 오인식이 된 경우 단어가 존재하지 않으면 인식과정에서 높은 점수를 갖는 후보문자로 선택된 단어로 대체한다.

해당 단어의 일치도가 만족될 때까지 다른 위치의 문자를 시도해 본다.

12. 학습 데이터 파일 생성

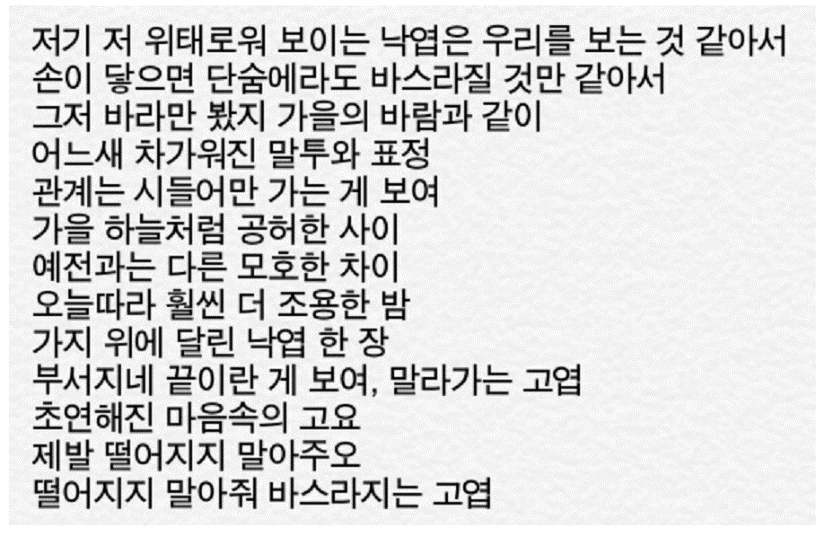
- cmd 명령어: combine\_tessdata.exe [언어 문자열].

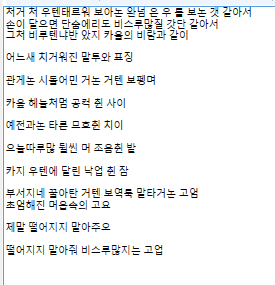
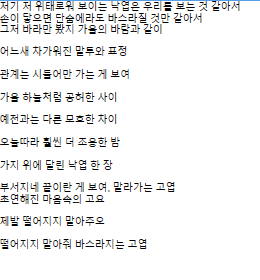
> combine\_tessdata.exe exp0.

- [언어 문자열].traineddata 파일 생성

13. 기존의 tesseract-ocr파일에 있는 한글 학습 파일을 지우고 새로운 학습 파일을 넣어준다.

**원본이미지**



**학습 전 학습 후   
** 

학습 전 이미지를 제대로 인식하지 못했지만 학습 후 인식가능해 졌다.

**2.3 학습 데이터 구축중 주의 사항**

학습 이미지의 문자 사이 간격이나 글자의 크기가 작을 경우 같은 박스로 설정 되는 경우가 있기 때문에 스페이스바 간격을 기준으로 두고 20pt이상의 글씨를 기준으로 둔다.

**3. 문제점 및 해결**

**1.두줄 인식**

영어의 경우한줄로만 인식하게 되지만 한글의 경우 받침이 있어 두줄로 인식하게 된다.

>block에서 라인을 구분하는 부분에서 강제로 싱글라인으로 인식하게 만든다.

<textord.cpp TextordPage> 메소드를 보면  
if (PSM\_LINE\_FIND\_ENABLED(pageseg\_mode)) {  
    gradient = make\_rows(page\_tr\_, to\_blocks);  
  } else if (!PSM\_SPARSE(pageseg\_mode)) {  
    // SINGLE\_LINE, SINGLE\_WORD and SINGLE\_CHAR all need a single row.  
    gradient = make\_single\_row(page\_tr\_, to\_block, to\_blocks);  
  }  
  // Now fit baselines. For now only old mode is available.  
  fit\_rows(gradient, page\_tr\_, to\_blocks);  
  // Now make the words in the lines.  
  if (PSM\_WORD\_FIND\_ENABLED(pageseg\_mode)) {  
    // SINGLE\_LINE uses the old word maker on the single line.  
    make\_words(this, page\_tr\_, gradient, blocks, to\_blocks);  
  } else {  
    // SINGLE\_WORD and SINGLE\_CHAR cram all the blobs into a  
    // single word, and in SINGLE\_CHAR mode, all the outlines  
    // go in a single blob.  
    TO\_BLOCK\* to\_block = to\_block\_it.data();  
    make\_single\_word(pageseg\_mode == PSM\_SINGLE\_CHAR,  
                     to\_block->get\_rows(), to\_block->block->row\_list());  
  }

이 부분을 주석처리하고,  
gradient = make\_single\_row(page\_tr\_, to\_block, to\_blocks);  
   fit\_rows(gradient, page\_tr\_, to\_blocks);  
    make\_single\_word(pageseg\_mode == PSM\_SINGLE\_CHAR,  
                     to\_block->get\_rows(), to\_block->block->row\_list());

**2. noise에 매우취약**

조그마한 노이즈나 티끌에 인식의 오류가생긴다

해결 : Blob내의 outline을 확인하여 noise성분을 제거하는 방법

이미지 전체의 blob에서 hole을 제외한 outlihne의 합에서 평균을 낸 후 noise성분만 제거 할 수 있을 정도의 값인 4로 나누었다.

<chopper.cpp chop\_word\_main>메소드 수정

long o\_l\_threshold=0, count=0;

for (blob = word->chopped\_word->blobs, index = 0;

blob != NULL; blob = blob->next, index++) {

for(TESSLINE \*o\_l=blob->outlines ; o\_l != NULL ; o\_l = o\_l->next)

{

if(!(o\_l->is\_hole))

{

o\_l\_threshold += o\_l->BBArea();

count++;

}

}

}

o\_l\_threshold = (o\_l\_threshold/count)>>2;

printf("%d\n", o\_l\_threshold);

...

int cnt=0;

for(TESSLINE \*o\_l=blob->outlines, \*o\_tmp ; o\_l != NULL ; o\_l = o\_l->next, cnt++)

{

if(o\_l->BBArea() < o\_l\_threshold && !(o\_l->is\_hole))

{

if(cnt==0){

blob->outlines = o\_l->next;

cnt--;

}

else

o\_tmp->next = o\_l->next;

o\_l->Clear();

continue

}

o\_tmp = o\_l;

}

if(blob->outlines == NULL)

{

if(index==0)

word->chopped\_word->blobs = blob->next;

else

blob\_tmp->next = blob->next;

//blob->Clear();

index--;

continue

}

blob\_tmp = blob;