BOBCAT-DI:
PETS Software Description

Army Research Laboratory,
Adelphi MD

from

Laboratory for Language and Media Processing
University of Maryland, College Park, MD, USA

December 20, 2008

Table of Contents

List of Figures

PETS Software Description

1 Introduction

1.1 Background

Over the past five decades, evaluation has become increasing important as the field of document image understanding has developed. A number of independent evaluations have been run by various academic organizations, all focusing on slightly different problems. In recent years, the University of Maryland has developed a number of tools aimed at supporting generic annotation and evaluation of document (and video) data. A set of three reports is being produced from this one year BOBCAT-DI research and development projects. The three reports include

- A *Segmentation and Evaluation Survey*- designed to identify major algorithms, tools and evaluation methodologies in the community,
- The *PETS Software Description* (This report) – a toolkit to evaluate segmentation, line detection and image enhancement algorithms, based on BOBCAT-DI requirements and as a response to the state of the art, and
- *Selected Evaluations using the PETS environment* – Evaluations designed to demonstrate the capabilities of the tools and provide a framework for use in operational environments.

It is the hope that this work will lead to a generic repository for evaluation which host data, tools, algorithms and evaluation results for community wide comparison.

1.2 Project Overview

The DoD Sequoyah Foreign Language Translation Program, managed by the interim Sequoyah Transition Mgt Office (STMO) under PEO IEWS, Ft Monmouth, NJ, is intended to address critical linguist shortfalls in US warfighting and intelligence operations through automated language translation capabilities (speech and text) and to provide document image processing and OCR capabilities for cases when material to be translated is paper or document images. To support unbiased, vendor-neutral assessment of technology candidates prior to field testing and deployment, the STMO has initiated a web-accessible, distributed "Best-of Breed Configurable Active Testbed" (BOBCAT) led and operated by ARL and distributed across NRL and AFRL. Yet to be incorporated into the testbed is the capability to assess OCR and other document image processing (DIP) tools. The STMO as well as the ODNI have tasked ARL with integrating document image (DI) processing assessment into BOBCAT, creating BOBCAT-DI . BOBCAT-DI will be used to assess a variety of document image processing capabilities and tools, with a focus on Arabic and other Southwest Asian languages. The image processing and analysis metrics and methods, particularly as applied to document images obtained from cameras, scanners, etc., is needed to enable assessments that are reliable, robust, and scientifically defensible

1.3 Organization of Report

The tools necessary for the proposed web based evaluation need to be robust, efficient and configurable to adapt to the variety of different research projects currently within the community. The GEDI and PETS software described in this report are a response to the challenge of being able to provide a configurable adaptable solution.

Section 2 describes the GEDI software tool at a high level and outlines the basic enhancements that have been made during this project, both under BOBCAT funding and through contributions from other organizations.

Section 3 highlights the PETs Evaluation Metrics and Protocols including newly developed metrics using Image differencing. The evaluation for differencing is necessary to support enhancement and other pixel level analysis algorithms.

Section 4 provides a summary and conclusions.

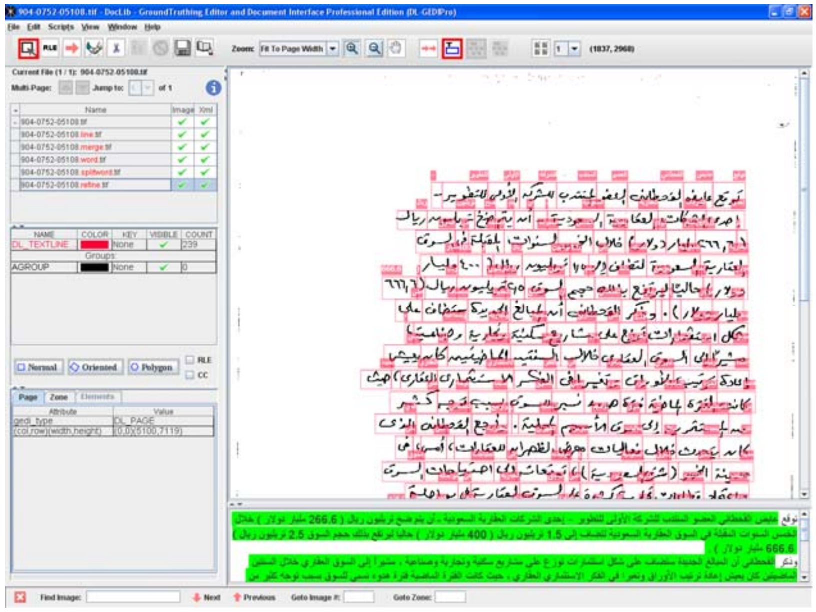


Figure 1: GEDI Tool

2 GEDI Ground Truthing Tool

GEDI is an editor that assists you in ground truthing scanned text documents. Its basic structure involves two types of files, an Image file, and a corresponding .xml file in GEDI Format (see below). A series of image files stored in the same directory (folder) can be opened simultaneously and the interface maintains a one to one correspondence with XML files of the same name, in same or another directory. When you begin ground truthing an image, you can configure the interface to allocation the creation of different types of zones, each of which may have a custom set of "attributes".

GEDI allows users to enter information at both the page level and at the zone level. Zones are typically used to mark the locations of physical regions on the page, such as text, signatures, logos, etc. Different zone types should be used to label different types of page elements. Information about a particular zone or page can be entered by setting values for attributes. Pages and zone types can have any number of attributes associated with them, and each instance of a page or zone can have its own values for the attributes. For labeling text, the pre-defined DL_TEXTLINEGT zone type can be used which manages predefined attributes for contents and character or word offsets within the zone. For specifics about any particular feature of GEDI, please select it from the table of contents to the left. Some topics have sub-topics and can be expanded by clicking on the plus sign next to them.

If you are already familiar with GEDI, you may still want to take a look at Appendix: Modification History.

2.1 GEDI Philosophy

Before beginning any annotation, it is useful to understand the GEDI Philosophy for annotation.

When the user annotates an image, GEDI stores the information in an XML file in the following format:


```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--GEDI was developed at Language and Media Processing Laboratory, University of
Maryland.-->
<GEDI xmlns="http://lamp.cfar.umd.edu/GEDI" version="1.0">
    <USER name="Elena" date="5/23/2008 17:24" dateFormat="mm/dd/yyyy hh:mm">    </USER>
    <USER name="Orri" date="6/11/2008 12:52" dateFormat="mm/dd/yyyy hh:mm">    </USER>
    <DL_DOCUMENT src="sample.tif" docTag="xml" NrOfPages="3">
         <DL_PAGE gedi_type="DL_PAGE" src="sample.tif" pageID="1" width="1728"
                  height="2292">
           <DL_ZONE gedi_type="DL_TEXTLINEGT" id="2" col="1285" row="269" width="166"
                  height="335" orientationD="16.169" contents="" offsets=""
                  segmentation="word"> </DL_ZONE>
           <DL_ZONE gedi_type="DL_TEXTLINEGT" id="3"
                  polygon="(151,253);(274,274);(610,294);(561,331);(540,375);(312,380);
                           (255,414);(109,404);(76,363);(107,349);(113,300)"
                  offsets="" segmentation="word"> </DL_ZONE>
           <DL_ZONE gedi_type="DL_TEXTLINEGT" id="4" col="208" row="500" width="245"
                  height="40" contents="" offsets="" segmentation="word"> </DL_ZONE>
           <DL_ZONE gedi_type="DL_TEXTLINEGT" id="5" col="214" row="626" width="494"
                  height="35" contents="To: Mr. A. Sadovnick" offsets=""
                  segmentation="word"> </DL_ZONE>
           <DL_ZONE gedi_type="DL_TEXTLINEGT" id="6" col="214" row="704" width="98"
                  height="40" contents="" offsets="" segmentation="word"> </DL_ZONE>
           <DL_ZONE gedi_type="DL_TEXTLINEGT" id="7" col="396" row="706" width="303"
                  height="34" contents="" offsets="" segmentation="word"> </DL_ZONE>
           <DL_ZONE gedi_type="DL_TEXTLINEGT" id="8" col="728" row="165" width="296"
                  height="259" contents="" offsets="" segmentation="word"> </DL_ZONE>
         </DL_PAGE>
         <DL_PAGE gedi_type="DL_PAGE" src="sample.tif" pageID="2" width="2592"
                  height="3300">
         </DL_PAGE>
         <DL_PAGE gedi_type="DL_PAGE" src="sample.tif" pageID="3" width="2592"
                  height="3300">
         </DL_PAGE>
    </DL_DOCUMENT>
</GEDI>
```


The GEDI represents a document as consisting of pages, and on each page a set of zones, and for each zone a set of attributes. The types of zone, and the user attributes which describe them are fully configurable in the interface. Each zone does have a set of “required” attributes which include 1) Zone ID, 2) GEDI-type, and 3) the coordinates of the zone – row, col, height and width for normal zones, an additional orientation attribute for oriented zones, and a list of points for polygonal zones.

The interface itself provide many different tools to help manage and maintain this metadata including function keys, drop down lists, color coding of attributes, support of multiple languages (including bi-text) etc. Details and lessons learned are found throughout this help document. As one additional introduction, here are some details on how the data is stored in the GEDI XML format. Please refer the example above:

The main tag is a GEDI tag, and it has two types of tags within it. It has one USER tag per user that has modified it (which includes the name of the user, the data modified, and the format of the date modified [mm/dd/yyy hh:mm is the default]), and the other is the DL_DOCUMENT tag, which represents the document, and includes information such as the number of pages of the image being worked on and the system path of the image. The DL_DOCUMENT tag has within it one tag per page of the image being worked on (in most cases, one, but if you have a multipage tiff, it will be more than one), whose attributes include the height, width, page id (1 through however many pages), and the system path of the image.

Within each page tag, there are a number of DL_ZONE tags, which correspond to the zones drawn on the image in GEDI. If the user has turned on parent-child mode to draw any of the zones, some of the DL_ZONE tags will have other DL_ZONE tags within them (these are the zones drawn inside the parent zone).

If the zone is a DL_TEXTLINEGT zone, it will also include the contents, offsets, and segmentation. Of course, the user is free to add attributes to the page, or to any of the zone types. These will also be included in the xml file.

2.2 GEDI Modifications

For this project, the team has made several sets of modification, primarily related to the ability of the system to

- Annotate handwritten textline data with the additional of Polygons
- Represent pixel level details by providing the a run length encoding
- Represent Reading Order
- Direct Integration of Evaluation Capabilities via scripts

The figure below shows results, colored in different ways as a direct result of evaluation.

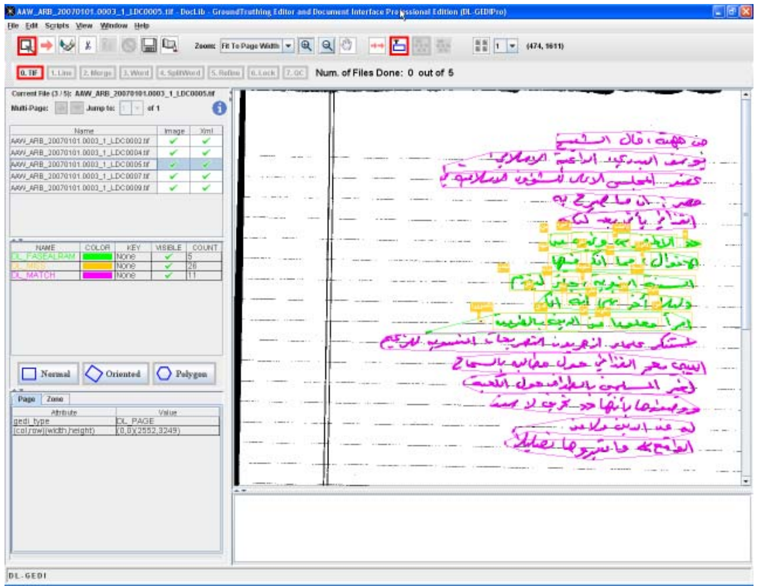


Figure 2: GEDI Showing Annotated Results

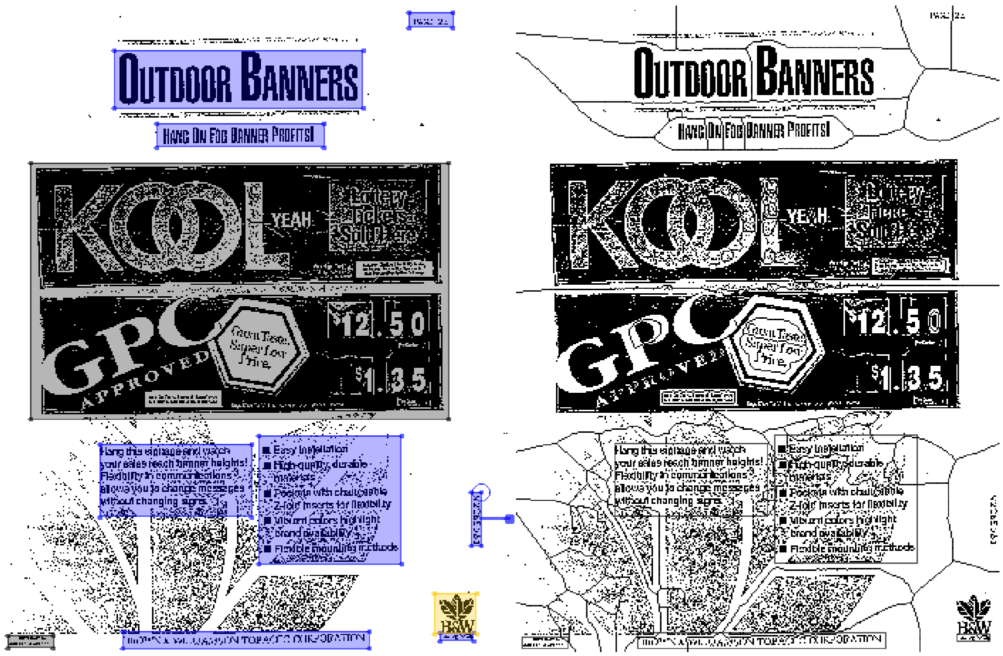


Figure 3: Ground Truth and Algorithm Results for Evaluation

3 Evaluation Metrics and Protocols

3.1 Overview

Many zone segmentation and zone classification algorithms have been proposed recently and effective and standard evaluation of such algorithms is getting considered important. This Section introduces the PETS software which was developed to evaluate zone segmentation and zone classification algorithms effectively and introduce the structures and algorithms which are used in PETS. PETS was initially designed to evaluate zone based algorithms for segmentation and classification.

3.2 PETS: Performance Evaluation Tools

3.2.1 Matching score

In the geometrical zone matching, the overlap ratio of two zones is the measure used to determine the matching score. The performance evaluation method is based on counting the number of matches between the result zones of algorithm and ground truth. We use pixel level precision and recall and get the matching score which is used to compute an F1 score of.

Let G be the ground truth and g_n be the n th ground truth zone, then $G=\{g_1, g_2, \cdots, g_n\}$. Let R be the result and r_n be the n th result zone, then $R=\{r_1, r_2, \cdots, r_n\}$. Let p_i be the i th pixel in the document. In the pixel level precision and recall,

true positive(TP) is $TP_{mn}=\{p_i | p_i \in g_m \wedge p_i \in r_n\}$,

false positive(FP) is $FP_{mn}=\{p_i | p_i \notin g_m \wedge p_i \in r_n\}$, false negative(FN) is $FN_{mn} = \{ p_i | p_i \in g_m \wedge p_i \notin r_n \}$.

Figure 2 shows a example of pixel level definition. $n(R)$ the function that counts the pixels in R set. Using these terms, we compute precision and recall as follows.

$$Precision_{mn} = \frac{n(TP_{mn})}{n(TP_{mn}) + n(FP_m)}$$

$$Recall_{mn} = \frac{n(TP_{mn})}{n(TP_{mn}) + n(FN_n)}$$

$$F1_{mn} = \frac{2 \times Precision_{mn} \times Recall_{mn}}{Precision_{mn} + Recall_{mn}}$$

Using this metric, we construct a matching score table (MST) between result zones and ground truth zones.

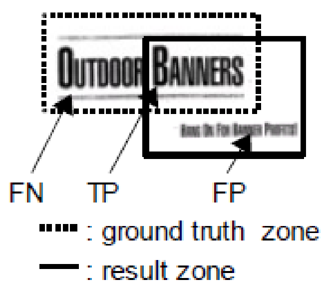


Figure 4: Pixel level definition

3.2.2 Zone matching

Once we construct the MST we define four types of zone matching by type of overlap. They include 'one-to-one', 'one-to-many', 'many-to-one' and 'many-to-many'. To find only one ground truth zone which is matched to a result zone, we use MST to find the best matching case. We define 4 types of zone matching when we consider the label correspondence for zone segment matching.

- MATCHED : A result zone matching a ground truth zone with same label
- DETECTED : A result zone matching a ground truth zone with different label
- FALSEALARM : A result zone matching no ground truth zone
- MISSED : A ground truth zone which is not matched by any result zone

3.2.2.1 One-to-one

This case is simplest case of overlap. Only one result zone overlaps with one ground truth zone. Figure 3 shows an example of a one-to-one overlap and the matching score table. In this case, *R1* is defined as 'MATCHED' if *L(R1)=L(G1)* and the matching score is greater than a threshold, otherwise *R1* is defined as 'DETECTED' if *L(R1)≠ L(G1)*. The default threshold is *80%*.

(a) example

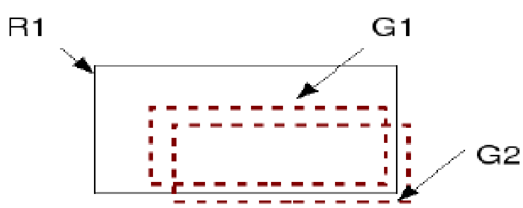
result\GT	G1
R1	85.00%

(b) matching score

Figure 5: Example of One to One Overlap

3.2.2.2 One-to-many

If one result overlaps multiple ground truth zones, we need to define which ground truth zones are matched to the result zone. Figure 4 shows an example of a one-to-many overlap. In this case, if *L(R1)=L(G1)=L(G2)*, we define *R1* as 'MATCHED' to *G1* and *G2* as 'MISSED'. IF *L(R1)=L(G2)≠ L(G1)*, *R1* is 'MATCHED' to *G2* and *G1* is 'MISSED'. IF *L(R1)≠ L(G1)≠ L(G2)*, *R1* is 'DETECTED' by *G1* and *G2* is 'MISSED'.



(a) example

result\GT	G1	G
R1	90.00%	85.(

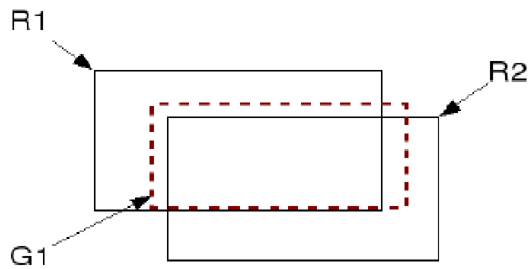
(b) matching score

Figure 6: example of one-many overlap

3.2.2.3 Many-to-one

If multiple result zones overlap one ground truth zone, we need to define which result zones are matched to the ground truth zone. Figure 5 shows an example of a many-to-one overlapping case. In this case, if *L(R1)=L(R2)=L(G1)*, we define *R1* as 'MATCHED' to *G1* and *R2* as 'FALSE ALARM'. If *L(R1)≠ L(R2)=L(G1)*, *R2* is

‘MATCHED’ by *G*1 and *R1* is ‘FALSE ALARM’. If *L(R1)≠ L(R2)≠ L(G1)*, *R1* is ‘DETECTED’ to *G1* and *R2* is a ‘FALSE ALARM’.



(a) example

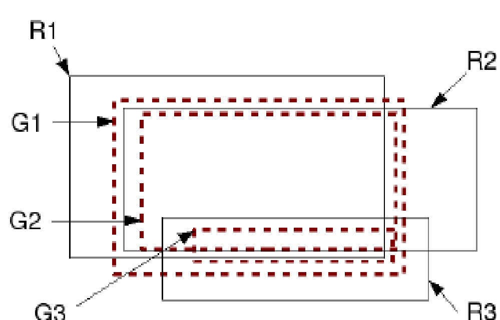
result\GT	G1
R1	95.00%
R2	90.00%

(b) matching score

Figure 7: Example of Many-to-One overlap

3.2.2.4 Many-to-many

This case is the most complicated type of overlap. Multiple result zones overlap multiple ground truth zones. Figure 6 shows an example of a many-to-many overlap. In this case, we find the maximum number of ‘MATCHED’ cases in the first step, and then find the maximum number of ‘DETECTED’ cases among the remainder from the first step.



(a) example

result\GT	G1	G2	G3
R1	90.00%	85.00%	50.00%
R2	80.00%	82.00%	45.00%
R3	30.00%	0.00%	85.00%

(b) matching score

Figure 8: Example of Many-To-Many overlap

3.2.3 System Architecture

3.2.3.1 Input

PETS uses XML which follows the GEDI data format specification for input of results and ground truth. GEDI is the Ground truthing Editor and Document Interface which can represent the document image structure well.

There are three xml tags which are essential to represent the document element. All of these are generated by default with GEDI.

<DL_DOCUMENT>

This tag represents the document image itself and this is the parent tag of the DL_PAGE tag. This tag should have several attributes including 'NrOfPages', 'docTag' and 'src'.

- NrOfPages : number of pages which this image document consists of.
- docTag : usually 'xml'
- src : name of document image

<DL_PAGE>

This tag represents the page of document image and this is the parent tag of the DL_ZONE tag. It is possible that there are several DL_PAGE tags, if the document image is multiple page image such as 'TIFF' or 'GIF'. This tag should have several attribute such as 'gedi_type', 'pageID', 'src', 'height' and 'width'.

- gedi_type : usually DL_PAGE
- pageID : identity of the page
- src : name of document image
- width : horizontal length of the page the image
- height : vertical length of the page of the image

<DL_ZONE>

This tag represents the zone in the page of image. This tag should have 5 essential attributes and can have user defined attribute. The essential attributes are 'gedi_type', 'id', 'col', 'row', 'width' and 'height'.

- gedi_type : label of the zone such as Table, Text, Stamp and so on.
- id : identity of the zone
- col : upper right column point of the zone
- row : upper right row point of the zone
- width : horizontal length of the zone
- height : vertical length of the zone
- orientationD : rotation degree of the oriented box. It uses degrees
- polygon : points that consist of (col,row) for boundary point
- lineID : identity of the one line of text
- RLEIMAGE : Run-length code of the zone


```
MENT src="aaa3aa00.tif" docTag="xml" NrOfPages="2">
PAGE gedi_type="DL_PAGE" src="aaa3aa00.tif" pageID="1" width="2544" height="3296">
<DL_ZONE gedi_type="HANDPRINT" id="1" polygon="(153,282);(1155,145);(1163,390);(182,527);(149,274)" contents
<DL_ZONE gedi_type="MACHINEPRINT" id="3" col="1712" row="2747" width="137" height="75" contents=""> </DL_ZON
<DL_ZONE gedi_type="MARKUP" id="4" col="1579" row="1525" width="283" height="108"> </DL_ZONE>
<DL_ZONE gedi_type="MARKUP" id="5" col="842" row="1566" width="370" height="110" contents=""> </DL_ZONE>
<DL_ZONE gedi_type="MACHINEPRINT" id="6" col="1868" row="2700" width="172" height="92" orientationD="0.0" cc
<DL_ZONE gedi_type="HANDPRINT" id="10" col="463" row="1178" width="738" height="100" orientationD="28.196" c
<DL_ZONE gedi_type="MARKUP" id="11" polygon="(1317,1338);(1496,1129);(1702,1259);(1605,1499);(1371,1453);(13
_PAGE>
PAGE gedi_type="DL_PAGE" src="aaa3aa00.tif" pageID="2" width="2544" height="3296">
<DL_ZONE gedi_type="HANDPRINT" id="7" col="108" row="486" width="1878" height="536" contents=""> </DL_ZONE>
<DL_ZONE gedi_type="MACHINEPRINT" id="9" col="1652" row="2724" width="416" height="99" orientationD="1.102"
<DL_ZONE gedi_type="MARKUP" id="12" col="448" row="1496" width="537" height="96" orientationD="18.772" conte
_PAGE>
JMENT>
```


Figure 9: Example of GEDI XML

3.2.4 Process

3.2.4.1 Selective merging

In some cases, it is necessary that multiple result zones are merged to one larger result zone and get the matching score for a ground truth zone. For example, when the zone segmentation algorithm is trained to segment the text region by word, but the ground truth zone is a large text area, we need merging mechanism to evaluate the algorithm. PETS check that a result zone fulfill the condition of merging using coverage score(CS). CS is defined as follow.

$$CS_{mn} = \frac{n(TP_{mn})}{n(TP_{mn}) + n(FP_n)} \qquad (4)$$

when the subset of R , $S(R) \subset R$, and $S(R) = \{r_n | CS_{mn} > threshold, r_m \in R\}$ then, $S(R)$ will be merged into one zone and get matching score to g_n . for the selective merging, PETS uses list of entity which is allowed to merge.

3.2.5 Output

There are two types of output from the program. One is the text file which has the detail result of evaluation process for every zone and the summarized results for all zones. The other is the output for visualization. It is useful when user check the matching result directly using GEDI.

For the individual result, symbols, 'O', '-' and 'X', are used to represent the result of each zone. 'O' indicates that the result zone detected one ground truth zone with matching score over the threshold and the type of two zones matched. '-' is same as 'O' in terms of detection, but the type of two zones did not matched. 'X' means that the result zone is a false alarm. The matching score is displayed if the result zone detects the one of ground truth zones with matching score over threshold. There are also overall result for each page at the end of the individual result.

```
=========================
Result of Individual File
=========================

[O] : Matched, [-] : Detected, [X] : False Alarm

AAW_ARB_20070101.0003_1_LDC0002.tif
==================================

Page ID : 1
-------------
[O]    M0,     zone,    z0,     zone, 99.79%
[O]    M1,     zone,    z1,     zone, 99.89%
[O]    M2,     zone,    z2,     zone, 99.60%
[O]    M3,     zone,    z3,     zone, 99.68%
[X]    M4,     zone
[X]    M5,     zone
[O]    M6,     zone,    z9,     zone, 81.76%
[O]    M7,     zone,   z10,     zone, 99.93%
[O]    M8,     zone,   z11,     zone, 99.92%
[O]    M9,     zone,   z12,     zone, 99.90%
[O]   M10,     zone,   z13,     zone, 99.87%
[O]   M11,     zone,   z14,     zone, 99.71%
[O]   M12,     zone,   z15,     zone, 99.88%
[X]     5,     zone
[X]     6,     zone
[X]    13,     zone
[OVERALL] 11/0/5/16, 68.75%
```

Figure 10: Example of Individual Results Page

In the summary of section, there is information on individual zones and a confusion matrix for the evaluation results. These are very commonly used tools for the analysis of classification results. Finally, summarized results for each type of zone are shown. The precision, recall, F-score, missing rate and false alarm rate represents the result of the segmentation algorithm and the classification algorithm.

```
==================
Summary of Results
==================

- Number of Files : 7
- Accuracy of Zone Detecting : 34.76%

01. Information on Zones
========================

Label   Class of Zone             Num. in Result     Num. in G-T
----------------------------------------------------------------
1       zone                                 210

02. Confusion Matrix
====================

 Result\GT    unmatch           1
---------------------------------
   unmatch   0( 0.0%)*  24(24.7%)
         1 137(65.2%)   73(34.8%)*

03. Result Table
================

Label      GT    Result   Correct  Precsion    Recall   F-Score
----------------------------------------------------------------
    1      97       210        73    34.76%    75.26%    47.56%
```

Figure 11: Example of Summary Results

The other output of the program is a GEDI file which is very useful when user wants to check the results of algorithm visually. There are four types of zones in the visual output, 'MATCHED', 'DETECTED', 'FALSEALARM' and 'MISSED'. Each zone has more information which is useful for understanding the result of algorithm include 'GTID', 'GTClass', 'RESID', 'RESClass', 'MZID' and 'Score'.

- GTID : ID of the ground truth zone
- GTClass : Type of the ground truth zone
- RESID : ID of the result zone
- RESClass : Type of the result zone
- MZID : ID of the merged zone, merged zones have same MZID
- Score : Matching score of the result zone

3.2.6 PETS Software Usage

Additional details of the software are available with the software distribution.

```
Name :
     PETS Performance Evaluation ToolS for zone segmentation
     and classification

Synopsis :
     Unix/Linux platform command : PETS
     Window platform command : PETS.exe
     command r {<FILE>|<DIR>} g{<FILE>|<DIR>} i {<FILE>|<DIR>}
               [o <FILE>] [v <DIR>] [m <FILE>] [t <NUM>]
               [detail] [lid] [rle] [segonly|zoneclass]
               [az <FILE>|naz <FILE>]
Options :
     r {<FILE>|<DIR>}
          : Location of Results File(s)

     g {<FILE>|<DIR>}
          : Location of Ground Truth File(s)

     i {<FILE>|<DIR>}
          : Location of Image File(s). Default location is the
          location of ground truth

     o <FILE>
          : Name of File for Evaluation Results. Default is
          'PETS Eval.txt'.

     v {<FILE>|<DIR>}
          : directory where Xml output of GEDI format will be
          saved

     lid
          : Zones which have same 'lineID' attribute in Ground
          truth will be merged to one zone

     rle
          : runlength code will be add to visualization output

     detail
          : enable detailed output for each zone

     t <NUM>
          : set the threshold by user for determining a zone
          match based on pixel counts. Default is 80(%).
```

```
m <FILE>
     : result zones which are in a ground truth zone will
     be merged if it’s types are in the <FILE>. First
     line of the FILE should have numeric data which is
     used as threshold for zone merging.

segonly
     : Evaluation will perform detection by not consider
     zone labels for matching.

zoneclass
     : Evaluation will rely on ZoneIDs for
     correspondence, considering only zone labels for
     results

az <FILE>
     : Zones which its types are in the <FILE> will be
     treated in the program, otherwise deleted from the
     result.

naz <FILE>
     : Zones which its types are in the <FILE> will be
     deleted from the result.
```

3.3 ImDiff

There is a class of document analysis problems which require analysis, and thus evaluation, at the pixel level and we refer to these problems as detection problems. One example of this includes line detection and removal and second example is general noise removal. Pixels of these classes are often interspersed with content so the “detection” must be done at a pixel level.

For this class of problems, we assume that we have a set of pixels we want to detect, which we refer to as the template. From this template, we can look at the number of pixels that are missed and falsely detected, as well as a percentage of missed as a function of the number of pixels in the template, and the percentage of false, as a function of the number of pixels in the original or result image.

Accurate evaluation at pixel level requires precise knowledge of the set of pixels which belong to a particular class which we are interested in detecting. This requires ground truthing to be done at pixel level, which is almost impractical. Hence we resort to an approach which provides us with ground truth data which is very close to the pixel level ground truthing. For example, in case of line detection and removal evaluation, initially we have scanned images of ruled lines. Noise gets introduced when these images are binarized, and hence all the foreground pixels do not correspond to lines now. So, to get rid of these noises and obtain a clean set of line pixels, we do a line based ground truthing of these images using GEDI. We use this final filtered template for evaluation.

Currently, this capability is not included in PETS or any other type evaluation software. We have developed this software independently and will be

3.3.1 Background

We define five different type of images used in our evaluation as follows:

a) **ContentImage:** Images with only content in it. (For example - text, logo)

b) **TemplateImage** - Images with only the "pixels" which we are interested in detecting or removing (Ex- Line pixels). In case of line detection evaluation, the template is the image with only line pixels in it and for line removal evaluation, the template is defined as the line pixels that we are interested in removing, after this template is added to ContentImage. For generating these templates, we have scanned ruled line pages at 300 dpi.

c) **OriginalImage/InputImage** - Images which are generated by the addition of contentImage and templateImage. This can be done by simply performing the logical 'OR' operation on the corresponding pixels of two binary images. This is the input to the detection or removal algorithm which we want to evaluate.

d) **DegradedTemplates:** The templates obtained by just scanning the ruled documents may not be a good match to what we observe in real documents. For example the line may be broken at several places or the thickness of a line may not be uniform due to some processing that degraded the image. We try to obtain such realistic InputImages for evaluation by degrading the templates manually using some image manipulation software. (Ex- GIMP)

e) **OutputImages:** The output of the method we are evaluating.

As an example, have created 5 TemplateImage, 2 DegradedTemplates and 10 ContentImages. We obtained 50 clean Input Images using 5 TemplateImages and 10 ContentImages and 10 degraded Input Images using the 2 degraded templates. Then our line detection algorithm is run on these 60 documents. In case of Line detection the output of the method is the "pixels" that are detected as line pixels, while for Line removal the output is the image with all the "pixels" detected as line, removed from the input image.

3.3.2 Evaluation Methodology

Evaluation is based on the purpose or goal of the method we want to evaluate. Since we have both the original document and the ground truth templates, we can obtain the pixel based performance of a given algorithm by finding the total number of missed pixels and false pixels in the output image which is defined as follows:

<u>CASE 1: If the goal is detection then we define the two measures as:</u>

Missed Detection: Foreground pixels that are present in the template image but are missing in the output image.

False Detection: Foreground pixels that are not present in template but are present in the output image.

The implementation of above measures can be done by following these equations:

DetectTemplate = LineTemplate ; **(1)**

MissedDetection = DetectTemplate - (DetectTemplate & OutImage); **(2)**

FalseDetection = OutImage - (DetectTemplate & OutImage); **(3)**

Where '&' and '|' represents logical AND and OR operations on the two binary images. Line Template is the image with only ruled lines it. In case of line detection evaluation, the detection template used to evaluate the method is same as line template.
Also, in all the above and following equations we have assumed that the foreground is represented as 1 and background as 0.

We can express the above measure relative to the total number of pixels as follow:

Missed %= (Number of Missed Pixels / Pixels in template)*100 **(4)**

False % = (Number of False Pixels / Pixels in output image)*100 **(5)**

<u>CASE 2 : If the goal is removal then we have the same measures defined as :</u>

Missed Detections: Foreground pixels that are present in the template and remain in the output image. i.e. The method failed to remove to these pixels.

False Detections: Foreground pixels that are not present in the template and the output image, but are present in the original image. i.e. The method wrongly removed these pixels.

RemovalTemplate = LineTemplate – (LineTemplates & ContentImage); **(6)**

OriginalImage = RemovalTemplate | ContentImage; **(7)**

MissedDetection = OutputImage & RemovalTemplate; **(8)**

FalseDetection = OriginalImage - (RemovalTemplate | OutputImage); **(9)**

Where '&' and '|' represents logical AND and OR operations on the two binary images. Substituting the OriginalImage from 6 in 9, we get

FalseDetection = (RemovalTemplate | ContentImage) –
(RemovalTemplate | OutputImage) **(10)**

So effectively the false pixels are the content pixels which were removed from the input image, as template pixels. Further insight into false removal can be obtained if we classify them as:

a) **FalseLine:** Falsely removed pixels which belong to a line.
b) **FalseRandom:** Falsely removed pixels which do not belong to a line.

FalseLine = FalseDetection & LineTemplate **(11)**

FalseRandom = FalseDetection – FalseLine **(12)**

In this way, we can get a deeper insight into what is happening with the line and content pixels. For the same value of FalseDetection, if the number of FalseRandom is high for a given method then it may be assigned a bigger cost for overall evaluation of the method.

We can express the above measure relative to the total number of pixels as follow:

Missed %= (Number of Missed Pixels / Pixels in template)*100 **(13)**

False % = (Number of False Pixels / Pixels in original image)*100 **(14)**

To combine the above two measures use the F-β measures, for which we first find the total number of true positives (or true detection or removal), precision and recall as follows:

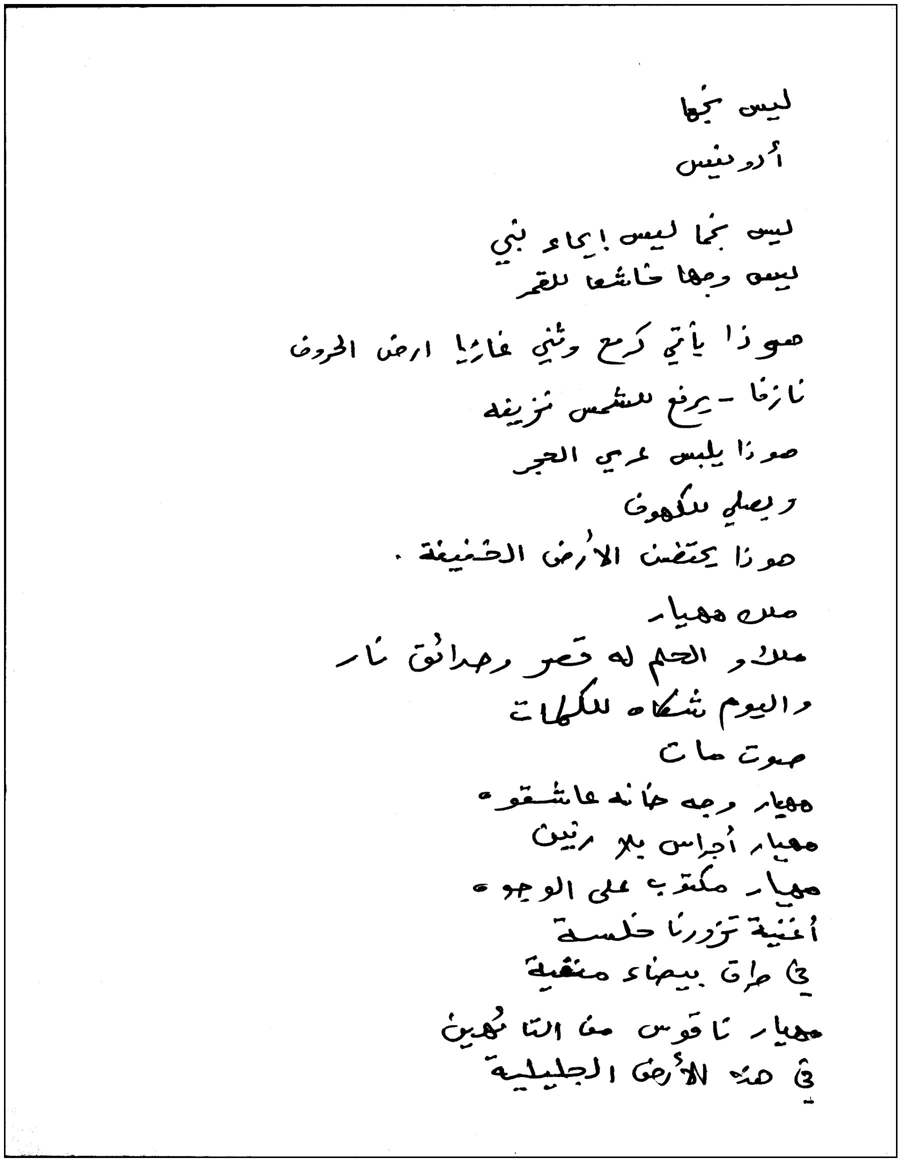
True detections (tp): Foreground pixels that were removed from input image (as line pixels) and are present in the template.

Precision (P) = tp/(tp + FalseDetection) **(15)**

Recall (R) = tp/(tp + MissedDetection) **(16)**

Fβ-Measure = (1+ β^2) P*R/ ((β^2)*P + R)

We find the Fβ-Measure for β = 1,2,3 to give different weights to recall and precision.


ليس نجماً

أدونيس

ليس نجماً ليس إيحاء نبي

ليس وجهاً خاشعاً للقمر

هوذا يأتي كرمح وثني غازياً أرض الحروف

نازفاً - يرفع للشمس نزيفه

هوذا يلبس عري الحجر

ويصلي للكهوف

هوذا يحتضن الأرض الخفيفة .

ملك مهيار

ملك، والحلم له قصر وحدائق نار

واليوم شكاه للكلمات

صوت مات

مهيار وجه خانه عاشقوه

مهيار أجراس بلا رنين

مهيار مكتوب على الوجوه

أغنية تزورنا خلسة

في طرق بيضاء منفية

مهيار ناقوس من التائهين

في هذه الأرض الجليلية

Figure 12: Example of a ground truth Text image.

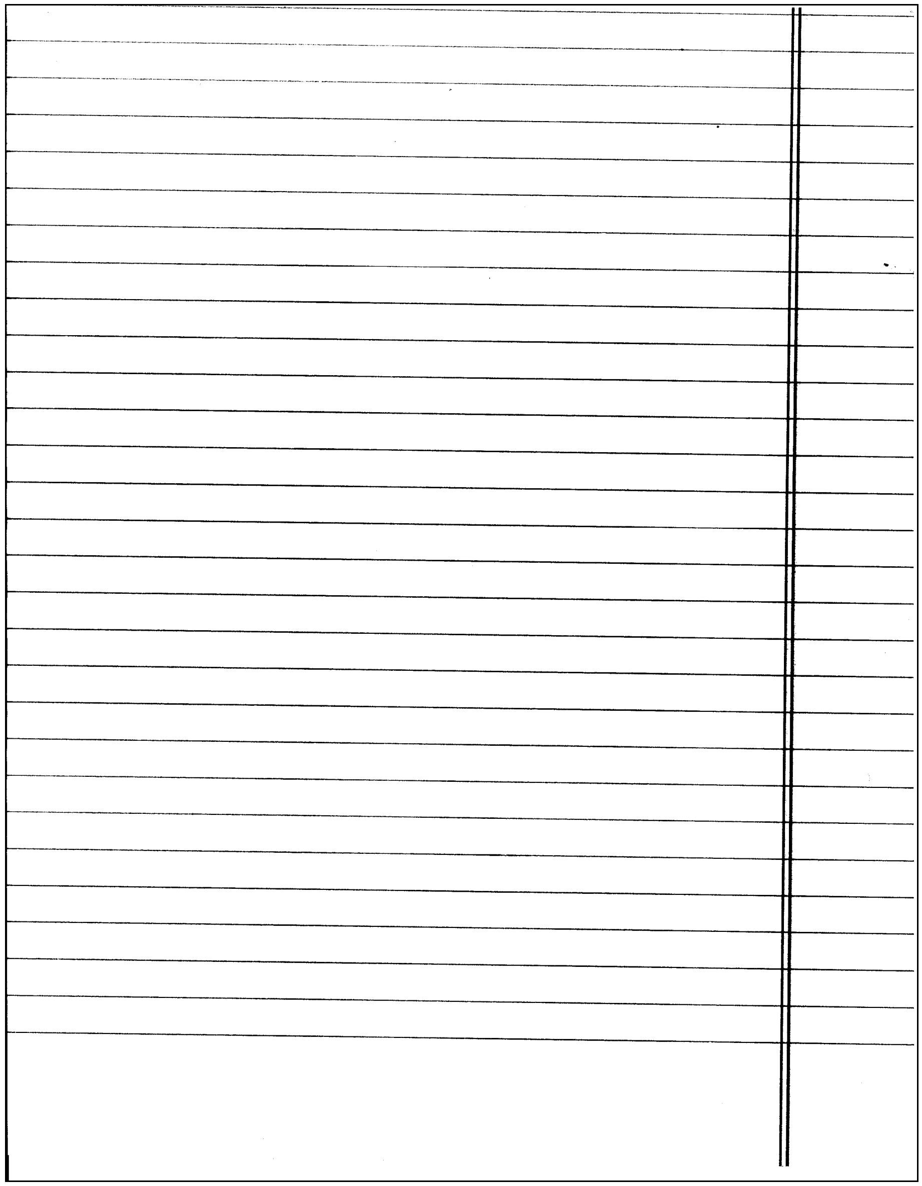
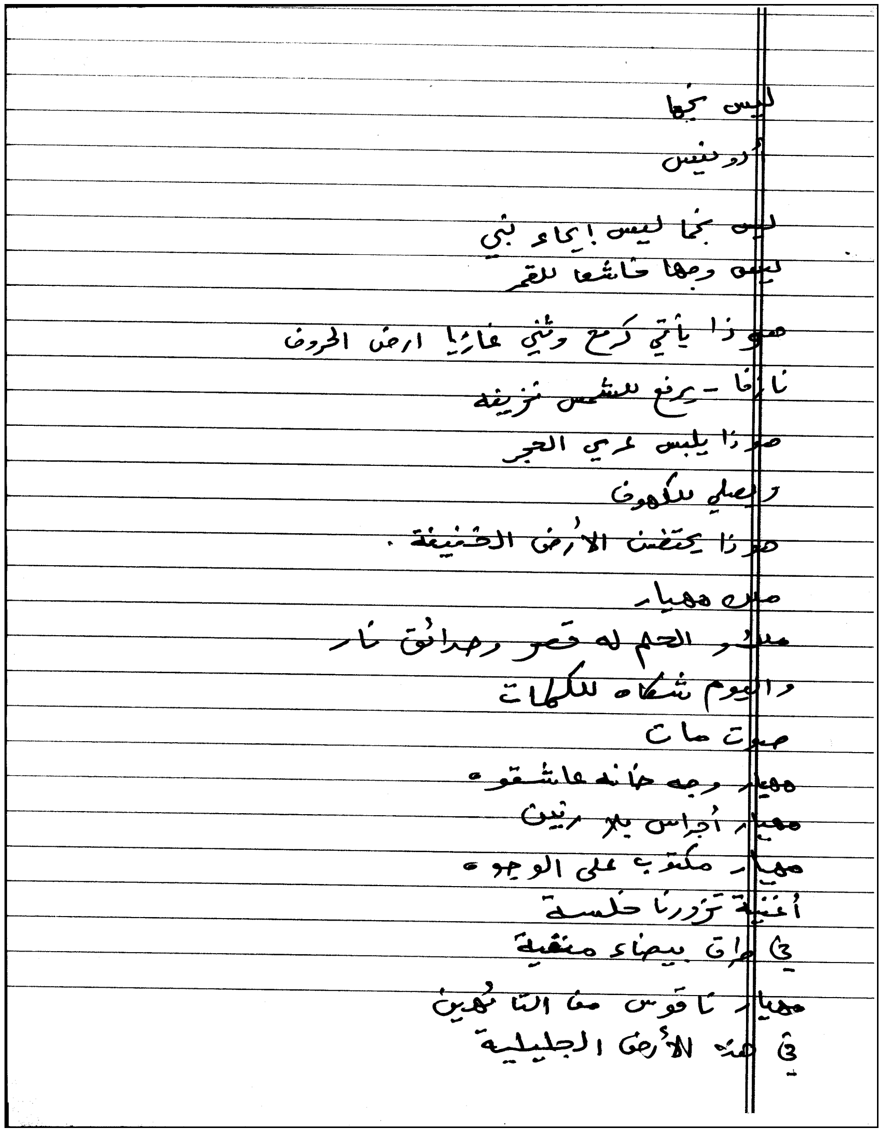


Figure 13: Example of a ground truth Text Line Image



ليس نجما
أو نبيا
ليس نجما ليس إيحاء نبي
ليس وجها خاشعا للقمر
هو ذا يأتي كرمح وثني غازيا أرض الحروف
نازفا - يرفع للشمس نزيفه
هو ذا يلبس عري الحجر
ويصلي للكهوف
هو ذا يحتضن الأرض الخفيفة .
ملك مهيار
ملك والحلم له قصر وحدائق نار
واليوم شكاه للكلمات
صوت مات
مهيار وجه خانه عاشقوه
مهيار أجراس بلا رنين
مهيار مكتوب على الوجوه
أغنية تزورنا خلسة
في طرق بيضاء منفية
مهيار ناقوس من التائهين
في هذه الأرض الجليلية

Figure 14: Image created by superposition (Previous 2 Figures)

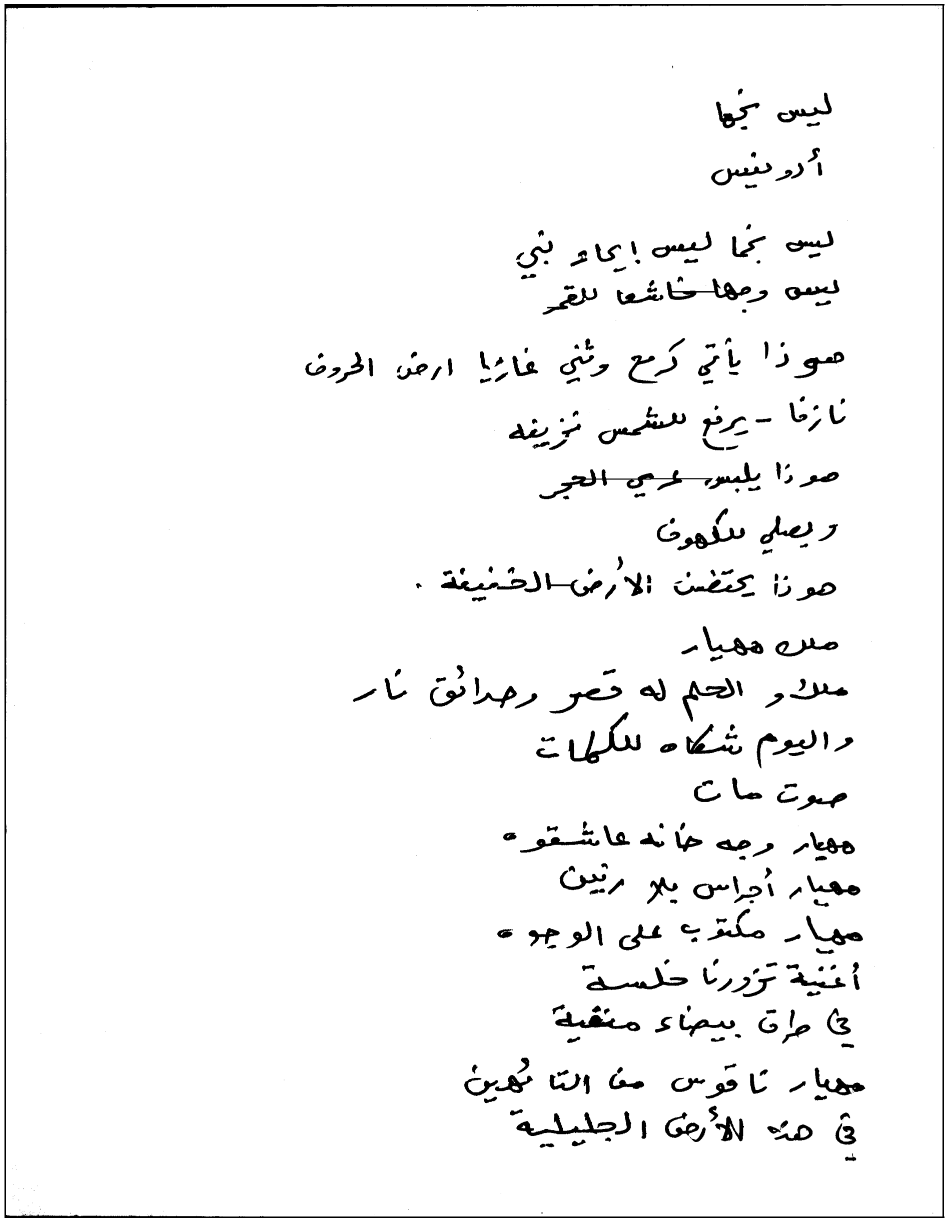


Figure 15: A sample output of Line Removal Output.

We can see some of the parts of Line still remaining over the text and some of the text pixels missing.

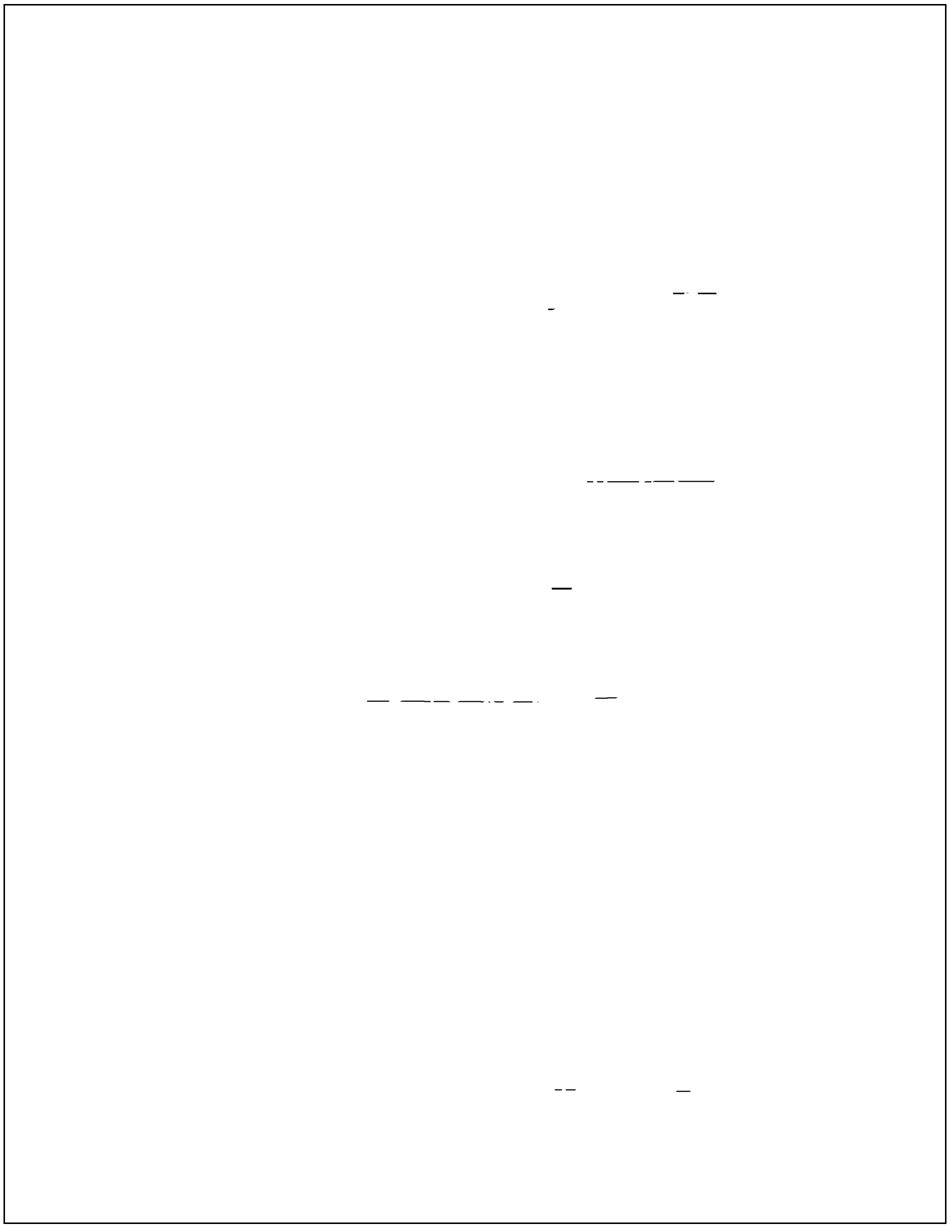


Figure 16: Missed Detections in the output image

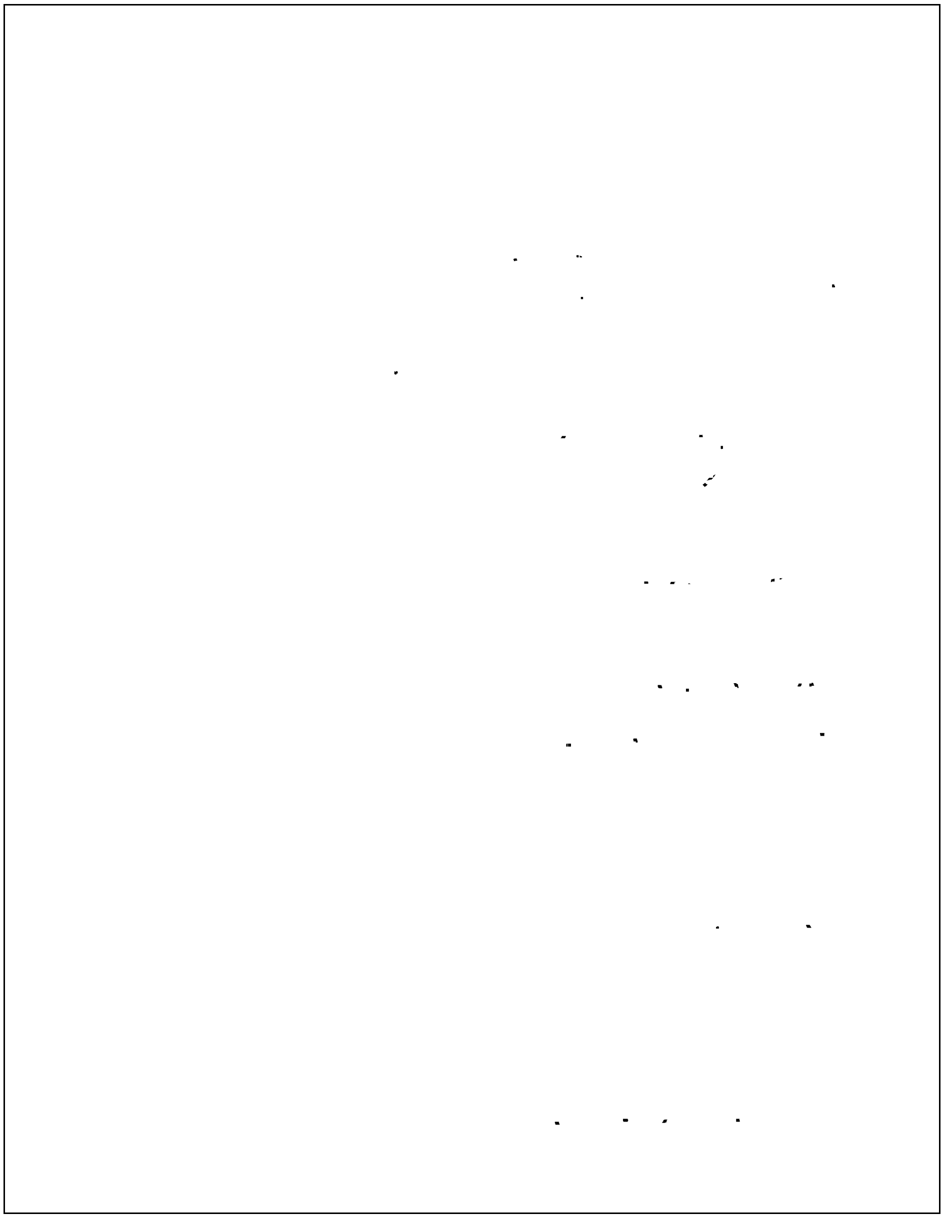


Figure 17: False detections in the output image

For the above example we have following:

Total number of text pixels: 265911
Total number of Line Only Pixels (which do not overlap with text): 307527
Missed Detections (line pixels that are not removed): 2028
False Detections (text pixels that are removed): 2971
Missed Percentage = 0.9661 %
False Percentage = 0.7627 %

4 Summary and Conclusions

The PETS project has produced a set of tools that can be integrated into a full framework for evaluation of document image analysis research. Combined with the existing OCR evaluation framework from UNLV the tools provide a powerful tool that can lead to wide dissemination of results in the community. An internet based portal which provides data, algorithms and evaluation capabilities as well as the ability to store and retrieve results will go a long way to achieving this vision.