APPENDIX G

Program Listing

```
function
                                                               'qui OutputFcn',
                   varargout
OBR (varargin)
                                           @OBR_OutputFcn, ...
                                                               'qui LayoutFcn',
% OBR M-file for OBR.fig
      OBR, by itself, creates a new
                                            [], ...
OBR or raises the existing
                                                               'qui Callback',
응
      singleton*.
                                            []);
양
                                           if nargin && ischar(varargin{1})
       H = OBR returns the handle to
용
                                               gui State.gui Callback
a new OBR or the handle to
                                           str2func(varargin{1});
      the existing singleton*.
                                           end
용
                                           if nargout
OBR('CALLBACK', hObject, eventData, han
                                               [varargout{1:nargout}]
dles,...) calls the local
                                           gui mainfcn(gui State, varargin{:});
        function named CALLBACK in
        with the given input
                                               qui mainfcn (qui State,
arguments.
                                           varargin(:));
                                           end
        OBR('Property','Value',...)
                                           % End initialization code - DO NOT
creates a new OBR or raises the
                                           EDIT
              existing singleton*.
Starting from the left, property
                                           % --- Executes just before OBR is
value pairs
      are
                                           made visible.
         applied to the GUI before
                                                        OBR OpeningFcn(hObject,
                                           function
OBR OpeningFunction gets called. An
                                           eventdata, handles, varargin)
      unrecognized property name or
                                           % This function has no output args,
invalid value makes property
                                           see OutputFcn.
application
                                           % hObject handle to figure
       stop. All inputs are passed
                                           % eventdata reserved - to be defined in a future version of
to OBR OpeningFcn via varargin.
                                           MATLAB
        *See GUI Options on GUIDE's
                                           % handles
                                                         structure with handles
Tools menu. Choose "GUI allows only
                                           and user data (see GUIDATA)
one
                                           % varargin command line arguments
                                           to OBR (see VARARGIN)
ે
      instance to run (singleton)".
                                           set (handles.BrowseImage, 'Enable', 'on
  See
         also: GUIDE,
                            GUIDATA,
                                           '):
GUIHANDLES
                                           set (handles.CleanImage, 'Enable', 'off
                                            ');
% Copyright 2002-2003 The MathWorks,
                                           set (handles.DeskewImage, 'Enable', 'of
Inc.
                                            f');
                                           set (handles.SegmentImage, 'Enable',
% Edit the above text to modify the
                                            'off');
response to help OBR
                                           set (handles.RecognizeImage, 'Enable',
                                            'off');
% Last Modified by GUIDE v2.5 23-
                                            set (handles.ImageDisplay, 'xcolor', 'w
Oct-2009 14:20:06
                                            ','ycolor','w','xtick',[],'ytick',[]
                                           );
% Begin initialization code - DO NOT
                                           set(handles.Deskew,'xcolor','w','yco
                                           lor','w','xtick',[],'ytick',[]);
qui Singleton = 1;
                                           set (handles.Backside, 'xcolor', 'w', 'y
gui State
                  struct('qui Name',
                                           color','w','xtick',[],'ytick',[]);
mfilename, ...
                                           set (handles.Oneclick, 'Enable', 'off')
                   'gui Singleton',
gui_Singleton, ...
                   'gui OpeningFcn',
@OBR OpeningFcn, ...
```

```
% Choose default command line output
                                          % handles
                                                       structure with handles
for OBR
                                          and user data (see GUIDATA)
handles.output = hObject;
                                          cla(handles.ImageDisplay, 'reset');
                                          cla(handles.Deskew, 'reset');
% Update handles structure
                                          cla(handles.Backside, 'reset');
quidata(hObject, handles);
                                          cla;
                                           set(handles.text1,'String','Browsed
                                           Image');
% UIWAIT makes OBR wait for user
response (see UIRESUME)
                                           set (handles.BrowseImage, 'Enable', 'on
% uiwait(handles.figure1);
                                           set (handles.DeskewImage, 'Enable', 'of
% --- Outputs from this function are
                                           set (handles.CleanImage, 'Enable', 'off
returned to the command line.
                                           ');
function varargout
                                           set(handles.SegmentImage, 'Enable', 'o
OBR OutputFcn(hObject, eventdata,
handles)
                                           set (handles.RecognizeImage, 'Enable',
% varargout cell array for
                                           'off');
returning output args (see
                                           set (handles.RecognizedImage, 'String'
VARARGOUT);
                                           ,'');
% hObject handle to figure
                                           set(handles.ImageDisplay,'xcolor','w
% eventdata reserved - to be defined in a future version of
                                           ','ycolor','w','xtick',[],'ytick',[]
                                           );
MATLAB
                                           set (handles.Deskew, 'xcolor', 'w', 'yco
% handles structure with handles
                                           lor','w','xtick',[],'ytick',[]);
                                           set (handles.Backside, 'xcolor', 'w', 'y
and user data (see GUIDATA)
                                           color','w','xtick',[],'ytick',[]);
                                           set(handles.text1,'visible','on');
% Get default command line output
from handles structure
                                           set(handles.Oneclick, 'Enable', 'off')
varargout{1} = handles.output;
                                           set(handles.Angle,'String',' ');
% --- Executes on button press in
BrailleContractions.
                                           % --- Executes on button press in
                                          BrailleAlphabet.
function
BrailleContractions Callback(hObject
                                           function
, eventdata, handles)
                                           BrailleAlphabet Callback(hObject,
  hObject
                                           eventdata, handles)
                        handle
BrailleContractions (see GCBO)
                                           % hObject
                                                                   handle
% eventdata reserved - to be defined in a future version of
                                          BrailleAlphabet (see GCBO)
                                           % eventdata reserved - to be
MATLAB
                                          defined in a future version
            structure with handles
                                          MATLAB
% handles
and user data (see GUIDATA)
                                           % handles
                                                       structure with handles
                                           and user data (see GUIDATA)
%set(handles.contractionsmenu,'visib
le','on');
                                          backgroundImage
                                           importdata('C:\MATLAB701\work\image\
backgroundImage
importdata('C:\MATLAB701\work\image\
                                          braille.jpg');
Braille Contractions (new).jpg');
                                           figure('Name','Braille
                                          Alphabet','NumberTitle','off'),imsho
figure('Name','Braille
Alphabet', 'NumberTitle', 'off'), imsho
                                          w(backgroundImage);
w(backgroundImage);
                                           % --- Executes on button press in
% --- Executes on button press in
                                           BrowseImage.
ResetAll.
                                           function
function ResetAll Callback (hObject,
                                          BrowseImage Callback(hObject,
eventdata, handles)
                                           eventdata, handles)
% hObject handle to ResetAll (see
                                          % hObject handle to BrowseImage
                                           (see GCBO)
% eventdata reserved - to be
                                           % eventdata reserved - to be
defined in a future version of
                                          defined in a future version of
MATLAB
                                          MATLAB
```

```
% handles
            structure with handles
                                            level=graythresh(gray);
and user data (see GUIDATA)
                                            bw=im2bw(gray,level);
set(handles.RecognizedImage,'String'
                                            imwrite(bw,'C:\MATLAB701\work\dump\b
,'');
                                            w.jpq');
                 pathname]
                                            img=imread('C:\MATLAB701\work\dump\b
[filename,
uigetfile({'*.jpg';'*.bmp'},'Pick an
                                            w.jpg');
Image(.jpg,.bmp) File');
                                            img=double(img);
%[filename,
                 pathname]
                                            inv(:,:,:) = 255 - img(:,:,:);
uigetfile({'*.jpg';'*.jpeg';'*.bmp'}
                                            inv=uint8(inv);
,'File
                                            inv=im2bw(inv);
%Selector');
                                            inv=bwareaopen(inv,30);
if ~ischar([pathname filename]) ||
                                            imwrite(inv,'C:\MATLAB701\work\dump\
isempty([pathname filename])
                                            inverse.jpg');
else
info= imfinfo([pathname filename]);
                                            %rotated initalization
I = imread([pathname, filename]);
                                            rotated = inv;
%Place the
             image file to the
                                            mediangray=gray;
variable I;
                                            save mediangray;
                                            totalangle=0;
save I;
set (handles.text1, 'String', info.File
                                            %Showing the skewed image with lines
name);
set(handles.Oneclick, 'Enable', 'on');
                                            L= bwlabel(rotated);
set (handles.BrowseImage, 'Enable', 'of
                                            s = regionprops(L,'centroid');
f');
                                            figure ('Name', 'Slopes: Yellow
set (handles.CleanImage, 'Enable', 'off
                                            Centers','NumberTitle','off'),imshow
');
set(handles.DeskewImage, 'Enable', 'on
                                            (rotated);
');
                                            hold on
                            'Enable',
                                            numObj = numel(s);
set(handles.SegmentImage,
'off');
set (handles.RecognizeImage, 'Enable',
                                            if numObj<3
                                            ee='Insufficient Braille Dots'
axes(handles.ImageDisplay)
                                            cla(handles.ImageDisplay, 'reset');
image(I);
                                            cla(handles.Deskew,'reset');
axis off
save info
                                            cla(handles.Backside, 'reset');
end
                                            cla:
                                            set (handles.text1, 'String', 'Browsed
                                            Image');
% --- Executes on button press in
                                            set (handles.BrowseImage, 'Enable', 'on
DeSkewImage.
                                            ');
                                            set(handles.DeskewImage, 'Enable', 'of
function
DeSkewImage Callback(hObject,
eventdata, handles)
                                            set (handles.CleanImage, 'Enable', 'off
% hObject
               handle to DeSkewImage
                                            ');
(see GCBO)
                                            set (handles.SegmentImage, 'Enable', 'o
% eventdata
                reserved - to be
                                            set(handles.RecognizeImage, 'Enable',
defined in a future version of
MATLAB
                                            'off');
% handles
              structure with handles
                                            set(handles.RecognizedImage,'String'
and user data (see GUIDATA)
                                            set (handles.ImageDisplay, 'xcolor', 'w
load I;
                                            ','ycolor','w','xtick',[],'ytick',[]
cropedimage = I; %copies I to
                                            );
cropedimage
                                            set(handles.Deskew,'xcolor','w','yco
                                            lor','w','xtick',[],'ytick',[]);
save cropedimage;
gray=imadjust(rgb2gray(cropedimage))
                                            set(handles.Backside,'xcolor','w','y
; %Grayscales cropedimage and then
                                            color','w','xtick',[],'ytick',[]);
                                            set(handles.text1,'visible','on');
increase its contrast
imwrite(gray, 'C:\MATLAB701\work\dump
                                            set (handles.Oneclick, 'Enable', 'off')
\gray.jpg'); %saves the image bw
imwrite(gray, 'C:\MATLAB701\work\dump
                                            set(LOADING,'visible','off');
\gray.jpg'); %saves the image bw
                                            clear
```

```
ctrr=ctrr+1;
else
                                                          end
                                                        else
for k = 1: numObj
                                                          if ((slope < 0.05) &&
    plot(s(k).Centroid(1),
                                            (slope > -0.05)) \&\& distance < 150
                                                             ctrr=ctrr+1;
s(k).Centroid(2), 'y*');
   plot(s(k).Centroid(1),
                                                          end
s(k).Centroid(2), 'yo');
                                                        end
    for i=1 : numObj
                                                     end
        if
                  (s(i).Centroid(1) -
                                                end
s(k).Centroid(1)) >0
                                            end
    slope=double((s(i).Centroid(2)-
s(k).Centroid(2))/(s(i).Centroid(1)-
                                            %Declaration of array and pointer
s(k).Centroid(1));
                                            array =1:ctrr;
    distance
                                            point=0;
sgrt((s(i).Centroid(1)-
s(k).Centroid(1))^2+(s(i).Centroid(2
                                            %Extraction of the useful slopes
)-s(k).Centroid(2))^2;
                                            hold on
                                            numObj = numel(s);
    format long
                                            for k = 1 : numObj
    slope;
             if ((slope < 0.1) &&
                                                for i=1 : numObj
(slope > -0.1)) && distance < 150
                                                     if
                                                                (s(i).Centroid(1) -
                                            s(k).Centroid(1)) >0
plot([s(k).Centroid(1)
                                                 slope=double((s(i).Centroid(2)-
s(i).Centroid(1)], [s(k).Centroid(2)
                                            s(k).Centroid(2))/(s(i).Centroid(1)-
s(i).Centroid(2)],'r');
                                            s(k).Centroid(1));
                                                 distance
             end
        end
                                            sqrt((s(i).Centroid(1) -
                                            s(k).Centroid(1))^2+(s(i).Centroid(2
    end
end
                                            )-s(k).Centroid(2))^2;
                                                 format long
hold off;
                                                 slope;
                                                       if loop==1
                                                          if ((slope < 0.1)
                                            (slope > -0.1)) \&\& distance < 150
%Loop start and angle extraction
                                                             point=point+1;
loop=1;
angle=1;
z=0;
                                            array(:,point) = slope;
while z \le 5
                                            %plot([s(k).Centroid(1)
                                            s(i).Centroid(1)], [s(k).Centroid(2)]
L= bwlabel(rotated);
                                            s(i).Centroid(2)],'r');
s = regionprops(L,'centroid');
                                                          end
%get total number of accepted slopes
                                                        else
ctrr=0;
                                                          if ((slope < 0.05) &&
numObj = numel(s);
                                             (slope > -0.05)) \&\& distance < 150
for ctr = 1 : numObj
                                                             point=point+1;
    for ctr2=1 : numObj
               (s(ctr2).Centroid(1)-
        i f
                                            array(:,point)=slope;
s(ctr).Centroid(1))>0
                                                          end
                                                        end
slope=double((s(ctr2).Centroid(2)-
                                                     end
s(ctr).Centroid(2))/(s(ctr2).Centroi
                                                end
d(1)-s(ctr).Centroid(1));
                                            end
    distance
                                            hold off;
sqrt((s(ctr2).Centroid(1)-
s(ctr).Centroid(1))^2+(s(ctr2).Centr
                                            arrav;
oid(2) - s(ctr) \cdot Centroid(2))^2;
                                            medianave=median(array);
    format long
    slope;
                                            %Deskewing
           if loop==1
                                            angle=atan(medianave);
             if ((slope < 0.1)
                                            angle=(angle*180)/pi;
(slope > -0.1)) \&\& distance < 150
                                            rotated= imrotate(rotated, angle);
```

```
mediangray=imrotate (mediangray,
                                                    end
angle);
                                                end
save mediangray;
                                            end
imwrite(mediangray,'C:\MATLAB701\wor
                                            hold off;
k\dump\mediangray.jpg');
imwrite(rotated,'C:\MATLAB701\work\d
ump\Rotated.jpg');
                                            %dis1=mediangray;
                                            %dis1=im2bw(dis1);
clear array;
loop=loop+1;
                                            dis1=imread('C:\MATLAB701\work\dump\
totalangle=totalangle+angle;
                                            mediangray.jpg');
                                            dis1 =cat (3, dis1, dis1, dis1);
z=z+1;
end
                                            axes(handles.Deskew);
                                            image(dis1);
%final display label
                                            axis off;
figure('Name','Deskew
run','NumberTitle','off'),imshow(rot
                                            totalangle
ated);
                                            set (handles.Angle, 'String', totalangl
L = bwlabel(rotated);
s = regionprops(L,'centroid');
hold on
                                            set (handles.BrowseImage, 'Enable', 'of
numObj = numel(s);
                                            f');
for k = 1: numObj
                                            set (handles.CleanImage, 'Enable', 'on'
    plot(s(k).Centroid(1),
                                            );
s(k).Centroid(2), 'y*');
                                            set (handles.DeskewImage, 'Enable', 'of
    plot(s(k).Centroid(1),
                                            f');
s(k).Centroid(2), 'yo');
                                            set (handles.SegmentImage,
                                                                         'Enable',
                                            'off');
    %backside filtering
                                            set(handles.RecognizeImage, 'Enable',
    x=s(k).Centroid(1);
                                            'off');
    y=s(k).Centroid(2)-7.5;
                                            end
    plot(x,y, 'b*');
                                            % --- Executes on button press in
    p=impixel (mediangray, x, y);
                                            RecognizeImage.
    for i=1 : numObj
                                            function
                                            RecognizeImage_Callback(hObject,
       i f
             (s(i).Centroid(1)-
s(k).Centroid(1))>0
                                            eventdata, handles)
                                               hObject
    slope=double((s(i).Centroid(2)-
                                                                      handle
                                                                                t o
s(k).Centroid(2))/(s(i).Centroid(1)-
                                            RecognizeImage (see GCBO)
s(k).Centroid(1));
                                            % eventdata
                                                             reserved - to be
                                            defined in a future version
    distance
sqrt((s(i).Centroid(1) -
                                            MATT.AR
s(k).Centroid(1))^2+(s(i).Centroid(2
                                            % handles
                                                          structure with handles
)-s(k).Centroid(2))^2);
                                            and user data (see GUIDATA)
    format long
                                            set(LOADING,'Visible','on')
            if loop==1
                if ((slope < 0.1) &&
(slope > -0.1)) && distance <150
                                            load charcount;
                                            charcount
plot([s(k).Centroid(1)
                                            if charcount==0
s(i).Centroid(1)], [s(k).Centroid(2)
s(i).Centroid(2)],'r');
                                            ee='No Character Detected'
                end
                                            set(LOADING,'Visible','off')
                                            set(handles.RecognizedImage,'String'
                if ((slope < 0.05)
                                            ,ee);
&& (slope > -0.05)) && distance <150
                                            set(handles.BrowseImage, 'Enable', 'on
                                            ');
plot([s(k).Centroid(1)
                                            set (handles.CleanImage, 'Enable', 'off
s(i).Centroid(1)], [s(k).Centroid(2)
s(i).Centroid(2)],'r');
                                            set (handles.DeskewImage, 'Enable', 'of
                end
                                            f');
            end
```

```
db2=[db1 db(epox+1,:) '%']
set(handles.SegmentImage, 'Enable',
                                               elseif charcount==1
set (handles.RecognizeImage, 'Enable',
                                               db2='12345'
'off');
                                               end
                                               conn = database('Braille', '',
                                            '');
elseif charcount>=1
load binary;
                                           setdbprefs('DataReturnFormat','cella
binary
                                           rray')
                                               curs = exec(conn, ['select Words
                                           from Grade2 where ID like ' '''' db2
%for u = 1 : charcount;
                                            '''')
11=1
                                               curs = fetch(curs)
while(u<=charcount)</pre>
                                               aa = curs.data
    z=num2str(binary(1,u));
   b=[z];
    for 0 = 2 : 6
                                               %Empty query checker
    a=binary(o,u);
                                               n = numel(curs.data)
    str1=num2str(a);
                                               b=0;
   b=[b str1];
                                               if n==1
                                               b = strcmp(bb,aa)
   end
    db(u,:) = strvcat(b);
                                               end
    u=u+1;
                                               %left & right checker
end
                                               if epox+1<charcount
db
                                           rgt=strcmp(db(epox+1,:),'000000')
                                               end
                                               if epox-met>=1%error catcher if
    db1='123456';
   conn = database('Braille', '',
                                           ctr-met will equal to 0
'');
                                               lft=strcmp(db(epox-
                                           met,:),'000000')
setdbprefs('DataReturnFormat','cella
                                               end
         =
                                               %query for standalone grade 2
   curs
             exec(conn, ['select
Letter from Gradel where ID= ' ''''
                                           braille characters(group) for the
db1 '''']);
                                           first
   curs = fetch(curs);
                                               %including the first char
                                               if (b == 1 && n == 1) && rgt==1
   bb = curs.data;
                                           && met>1 && epox-met<1
   db1='000000';
                                                   curs = exec(conn, ['select
   conn = database('Braille', '',
                                           Words from Grade2 where ID= ' ''''
'');
                                           db1 ''''])
                                                   curs = fetch(curs)
setdbprefs('DataReturnFormat','cella
                                                   aa = curs.data
rray');
                                                   n = numel(curs.data)
   curs
             exec(conn, ['select
                                                   b=strcmp(bb,aa)
Letter from Gradel where ID= ' ''''
                                                   arrayy(epox) = aa
db1 '''']);
   curs = fetch(curs);
                                                   spaceb=met-1
   space = curs.data;
                                                   while spaceb>=1
                                                   arrayy(epox-spaceb) = space;
end
                                                   spaceb=spaceb-1;
                                                   end
if charcount>=1
set (handles.RecognizedImage, 'String'
                                                   db1=db(epox+1,:)
,'Loading...');
                                                   met=1
db1 = [db(1,:)]
                                                   epox=epox+1;
met=1;
epox=1;
                                                %Sure standalone
while(epox<=charcount)</pre>
                                                elseif (b == 1 && n == 1) &&
                                           lft==1 && rgt==1 && met>1
    epox
    lft=0;
                                                   curs = exec(conn, ['select
                                           Words from Grade2 where ID= ' ''''
    rgt=0;
                                           db1 ''''))
    if epox<charcount && charcount>1
```

```
curs = fetch(curs)
                                                  db1=db(epox+1,:)
       aa = curs.data
                                                  met=1
       n = numel(curs.data)
                                                  epox=epox+1;
       b=strcmp(bb,aa)
                                                  end
       arrayy(epox) = aa
                                                 %Suffix
       spaceb=met-1
                                                 elseif (b == 1 && n == 1) &&
       while spaceb>=1
                                           lft==0 && rgt==1 && met>1
       arrayy(epox-spaceb) = space;
        spaceb=spaceb-1;
                                                  curs = exec(conn, ['select
                                           Words from Suffix where ID= ' ''''
       end
                                           db1 '''')
                                                  curs = fetch(curs)
       db1=db(epox+1,:)
                                                  aa = curs.data
       met=1
       epox=epox+1;
                                                  n = numel(curs.data)
                                                  b=strcmp(bb,aa)
                                                  if (b == 1 && n == 1)
     %Prefix
     elseif (b == 1 && n == 1) &&
                                                  ppp=met-1
lft==1 && rgt==0 && met>1
                                                  while ppp>=1
       curs = exec(conn, ['select
                                                  epox=epox-ppp
Words from Pre_fix where ID= ' ''''
                                                  db1=db(epox,:)
db1 ''''])
                                                  curs = exec(conn, ['select
       curs = fetch(curs)
                                          Letter from Gradel where ID= ' ''''
                                          db1 '''')
       aa = curs.data
       n = numel(curs.data)
                                                  curs = fetch(curs)
       b=strcmp(bb,aa)
                                                  aa = curs.data
                                                  n = numel(curs.data)
       if (b == 1 \&\& n == 1)
                                                  b=strcmp(bb,aa)
       ppp=met-1
                                                  arrayy(epox) = aa
       while ppp>=1
                                                  db1=db(epox+1,:)
       epox=epox-ppp
                                                  met=1
       db1=db(epox,:)
                                                  epox=epox+ppp;
       curs = exec(conn, ['select
                                                  ppp=ppp-1;
Letter from Gradel where ID= ' ''''
                                                  end
db1 ''''])
       curs = fetch(curs)
                                                  else
                                                  curs = exec(conn, ['select
       aa = curs.data
       n = numel(curs.data)
                                          Words from Suffix where ID= ' ''''
                                          db1 ''''])
       b=strcmp(bb,aa)
                                                  curs = fetch(curs)
       arrayy(epox) = aa
       db1=db(epox+1,:)
                                                  aa = curs.data
                                                  n = numel(curs.data)
       met=1
       epox=epox+ppp;
                                                  b=strcmp(bb,aa)
       ppp=ppp-1;
                                                  arrayy(epox) = aa
       end
                                                  spaceb=met-1
                                                  while spaceb>=1
       else
       curs = exec(conn, ['select
                                                  arrayy(epox-spaceb)=space;
Words from Pre fix where ID= ' ''''
                                                  spaceb=spaceb-1;
db1 ''''])
                                                  end
       curs = fetch(curs)
       aa = curs.data
                                                  db1=db(epox+1,:)
       n = numel(curs.data)
                                                  met=1
       b=strcmp(bb,aa)
                                                  epox=epox+1;
       arrayy(epox) = aa
                                                  end
       spaceb=met-1
                                                  %Midfix
       while spaceb>=1
                                                  elseif (b == 1 && n == 1) &&
       arrayy(epox-spaceb) = space;
        spaceb=spaceb-1;
                                          lft==0 && rgt==0 && met>1
       end
```

```
curs = exec(conn, ['select
Words from Mid fix where ID= ' ''''
                                                  epox=epox+1;
db1 '''')
       curs = fetch(curs)
                                              elseif (b==1 && n==1)&& met==1
                                          && lft==1 && epox+1>charcount
       aa = curs.data
       n = numel(curs.data)
                                                  curs = exec(conn, ['select
       b=strcmp(bb,aa)
                                          Words from Stand Alone where ID= '
                                          '''' db1 ''''])
       if (b == 1 \&\& n == 1)
                                                 curs = fetch(curs)
       ppp=met-1
                                                  aa = curs.data
       while ppp>=1
                                                  n = numel(curs.data)
                                                  b=strcmp(bb,aa)
       epox=epox-ppp
       db1=db(epox,:)
                                                  arrayy(epox) = aa
       curs = exec(conn, ['select
                                                  if epox<charcount
Letter from Gradel where ID= ' ''''
                                                 db1=db(epox+1,:)
db1 ''''])
                                                  end
       curs = fetch(curs)
                                                  met=1
       aa = curs.data
                                                  epox=epox+1;
       n = numel(curs.data)
                                               elseif (b==1 && n==1)&& met==1
       b=strcmp(bb,aa)
       arrayy(epox) = aa
                                          && rgt==1 && epox-met<1
                                                  curs = exec(conn, ['select
       db1=db(epox+1,:)
       met=1
                                          Words from Stand Alone where ID= '
                                          '''' db1 ''''])
       epox=epox+ppp;
       ppp=ppp-1;
                                                  curs = fetch(curs)
                                                  aa = curs.data
       end
                                                  n = numel(curs.data)
       else
                                                  b=strcmp(bb,aa)
       curs = exec(conn, ['select
                                                  arrayy(epox) = aa
Words from Mid fix where ID= ' ''''
                                                  if epox<charcount
db1 ''''])
                                                  db1=db(epox+1,:)
       curs = fetch(curs)
                                                  end
       aa = curs.data
                                                  met=1
       n = numel(curs.data)
                                                  epox=epox+1;
       b=strcmp(bb,aa)
       arrayy(epox) = aa
                                                  elseif (b==1 && n==1)&&
                                          met==1 && epox+1>charcount && epox-
       spaceb=met-1
                                          met<1
       while spaceb>=1
                                                  curs = exec(conn, ['select
                                          Words from Stand Alone where ID= '
       arrayy(epox-spaceb)=space;
                                          '''' db1 ''''])
       spaceb=spaceb-1;
                                                  curs = fetch(curs)
       end
                                                  aa = curs.data
       db1=db(epox+1,:)
                                                  n = numel(curs.data)
       met=1
                                                  b=strcmp(bb,aa)
       epox=epox+1;
                                                  arrayy(epox) = aa
       end
                                                  if epox<charcount
                                                  db1=db(epox+1,:)
                                                  end
       %try
                                                  met=1
   elseif (b==1 && n==1)&& met==1
                                                  epox=epox+1;
&& lft==1 && rgt==1
                                          %not sure
       curs = exec(conn, ['select
Words from Stand Alone where ID= '
'''' db1 ''''])
       curs = fetch(curs)
                                              %query for grade 1 braille(since
       aa = curs.data
                                          grade 2 1 braille char standalone
       n = numel(curs.data)
                                          are unique)
                                                      (b==1 \&\& n == 1) \&\&
       b=strcmp(bb,aa)
                                             elseif
       arrayy(epox) = aa
       if epox<charcount
                                                 curs = exec(conn, ['select
                                          Letter from Gradel where ID= ' ''''
       db1=db(epox+1,:)
                                          db1 ''''])
       end
```

```
curs = fetch(curs)
                                                        curs = exec(conn,
       aa = curs.data
                                         ['select Words from Pre Fix where
                                         ID= ' '''' db1 ''''])
       n = numel(curs.data)
       b=strcmp(bb,aa)
                                                        curs = fetch(curs)
                                                        aa = curs.data
       %query for grade 2 single
                                                        n = numel(curs.data)
braille char standalone (condition is
                                                        b=strcmp(bb,aa)
       %there's no match in the
                                                     elseif n>1 && lft==1 &&
grade 1 DB)
                                         rgt==1
       if (b == 1 && n == 1)
                                                        curs = exec(conn,
       curs = exec(conn, ['select
                                                    Words from Stand Alone
                                         ['select
                                         where ID= ' '''' db1 ''''])
Words from Grade2 where ID= ' ''''
db1 '''')
                                                       curs = fetch(curs)
       curs = fetch(curs)
                                                        aa = curs.data
       aa = curs.data
                                                        n = numel(curs.data)
       n = numel(curs.data)
                                                        b=strcmp(bb,aa)
       b=strcmp(bb,aa)
                                                     end
           if n>1 && epox-met<1 &&
                                                 end
rgt==0
              curs = exec(conn,
                                                 %proceed here if match is
['select Words from Pre Fix where
                                         already found
ID= ' '''' db1 ''''])
                                                 arrayy(epox) = aa
              curs = fetch(curs)
                                                 if epox<charcount
              aa = curs.data
                                                 db1=db(epox+1,:)
              n = numel(curs.data)
                                                 end
              b=strcmp(bb,aa)
                                                met=1
                                                 epox=epox+1;
           elseif n>1 && lft==0 &&
                                             else
rgt==1 && epox==1
              curs = exec(conn,
                                             %db1 incrementation if there's
['select Words from Stand Alone
                                        no match in any of the database
where ID= ' '''' db1 ''''])
                                            if epox+1<=charcount
              curs = fetch(curs)
                                             db1=[db1 db(epox+1,:)]
              aa = curs.data
                                             met=met+1
              n = numel(curs.data)
                                             epox=epox+1;
              b=strcmp(bb,aa)
                                             %trial
           elseif n>1 && lft==0 &&
                                             else
                                                 curs = exec(conn, ['select
rgt==0
              curs = exec(conn,
                                         Words from Grade2 where ID= ' ''''
['select Words from Mid Fix where
                                         db1 ''''])
ID= ' '''' db1 '''')
                                                curs = fetch(curs)
              curs = fetch(curs)
                                                 aa = curs.data
              aa = curs.data
                                                 n = numel(curs.data)
              n = numel(curs.data)
                                                 b=strcmp(bb,aa)
              b=strcmp(bb,aa)
                                                 arrayy(epox) = aa
           elseif n>1 && lft==0 &&
                                                 spaceb=met-1
rat==1
                                                 while spaceb>=1
              curs = exec(conn,
                                                 arrayy(epox-spaceb)=space;
['select Words from Suffix where ID=
                                                 spaceb=spaceb-1;
' '''' db1 ''''])
                                                 end
              curs = fetch(curs)
              aa = curs.data
                                                epox=epox+1
              n = numel(curs.data)
                                             end
              b=strcmp(bb,aa)
                                             %trial
                                         end
           elseif n>1 && lft==1 &&
                                             close(conn)
rqt==0
                                             close(curs)
```

```
end
                                           set (hObject, 'BackgroundColor', 'white
result = [arrayy(:)]
                                           %end
ee=strvcat(arrayy(1));
                                           % --- Executes when figure1 is
for cc = 2: charcount
ff=strvcat(arrayy(cc));
                                          resized.
                                           function figure1_ResizeFcn(hObject,
eventdata, handles)
ee=[ee ff];
end
                                           % hObject
                                                       handle to figure1 (see
                                           GCBO)
set(LOADING,'Visible','off')
                                           % eventdata
                                                          reserved - to be
                                           defined in a future version of
set (handles.RecognizedImage, 'String'
,ee);
                                           MATLAB
                                           % handles
                                                         structure with handles
                                           and user data (see GUIDATA)
set (handles.BrowseImage, 'Enable', 'on
set (handles.CleanImage, 'Enable', 'off
                                           % --- Executes on button press in
');
                                           pushbutton16.
set(handles.DeskewImage, 'Enable', 'of
                                           function
                                           pushbutton16_Callback(hObject,
f');
set (handles.SegmentImage,
                           'Enable',
                                           eventdata, handles)
                                           % hObject
                                                        handle to pushbutton16
set (handles.RecognizeImage, 'Enable',
                                           (see GCBO)
'off');
                                                           reserved - to be
                                           % eventdata
                                           defined in a future version of
end
                                           MATTAR
% --- Executes on button press in
                                                        structure with handles
                                           % handles
AnalyzeImage.
                                           and user data (see GUIDATA)
function
AnalyzeImage Callback(hObject,
                                           %erasing
eventdata, handles)
                                           load mediangray;
% hObject handle to AnalyzeImage
                                           for po =1:3
(see GCBO)
                                           rotated=imread('C:\MATLAB701\work\du
% eventdata reserved - to be
                                           mp\rotated.jpg');
defined in a future version of
                                           rotated=uint8(rotated);
MATLAB
                                           rotated=im2bw(rotated);
% handles
            structure with handles
and user data (see GUIDATA)
                                           %figure('Name','Erasing','NumberTitl
                                           e','off'),imshow(mediangray);
                                           hold on
                                           L = bwlabel(rotated);
% --- Executes on button press in
                                           s = regionprops(L,'centroid');
Exit.
                                           numObi = numel(s);
              Exit Callback (hObject,
                                           for k = 1 : numObj
function
eventdata, handles)
             handle to Exit (see
                                               %backside filtering
% hObject
GCBO)
                                              x=s(k).Centroid(1);
% eventdata reserved - to be
                                              y=s(k).Centroid(2);
                                              x2=s(k).Centroid(1)-4;
defined in a future version of
MATLAB
                                              y2=s(k).Centroid(2)-7;
% handles
            structure with handles
                                              plot(x2,y2, 'b.');
and user data (see GUIDATA)
                                              p=impixel (mediangray, x2, y2);
close
                                               if p(1,1) \le 210
                                                  plot(s(k).Centroid(1),
%handles.output = hObject;
                                           s(k).Centroid(2),'g*');
                ispc
                                  23
                                                 plot(s(k).Centroid(1),
isequal(get(hObject, 'BackgroundColor
                                           s(k).Centroid(2),'go');
get(0,'defaultUicontrolBackgroundCol
                                           imread('C:\MATLAB701\work\dump\media
or'))
                                           ngray.jpg');
```

```
rotated=imread('C:\MATLAB701\work\du
imread('C:\MATLAB701\work\dump\rotat
                                            mp\rotated.jpg');
                                            rotated=uint8(rotated);
ed.jpg');
        col = [(s(k).Centroid(1)-7)]
                                            rotated=im2bw(rotated);
                                            figure('Name','Segments','NumberTitl
(s(k).Centroid(1)+7)
(s(k).Centroid(1)+7)
                                            e','off'),imshow(rotated);
(s(k).Centroid(1)-7)];
        row = [(s(k).Centroid(2)-7)]
                                            L = bwlabel(rotated);
(s(k).Centroid(2)-7)
                                            s = regionprops(L,'centroid');
(s(k).Centroid(2)+7)
                                            hold on
(s(k).Centroid(2)+7)];
                                            numObj = numel(s);
        J = roifill(I2, col, row);
                                            if numObj>0
        M = roifill(K,col,row);
                                            %xholder
imwrite(J,'C:\MATLAB701\work\dump\me
                                            xholder=1:numObj;
                                            for iz = 1 : numObj
diangray.jpg');
                                            xholder(:,iz) = s(iz).Centroid(1);
imwrite(M,'C:\MATLAB701\work\dump\ro
                                            format short;
tated.jpg');
                                            end
        rotated=M;
                                            xholder
        mediangray=J;
        save mediangray;
                                            %vholder
    end
                                            yholder=1:numObj;
                                             for iz = 1 : numObj
end
                                            yholder(:,iz) = s(iz).Centroid(2);
hold off;
                                            format short;
end
                                            end
                                            yholder
dis2=imread('C:\MATLAB701\work\dump\
mediangray.jpg');
                                            %xvholder
dis2 =cat (3, dis2, dis2, dis2);
                                            xyholder=[1:numObj;1:numObj];
axes(handles.Backside);
                                            for iz = 1 : numObj
                                            xyholder(1,iz) = s(iz).Centroid(1);
image (dis2);
axis off;
                                            xyholder(2,iz)=s(iz).Centroid(2);
                                            format short;
                                            end
                                            xyholder
set (handles.BrowseImage, 'Enable', 'of
set (handles.CleanImage, 'Enable', 'off
                                            v1 = -999;
                                            y2 = -999;
');
set(handles.DeskewImage, 'Enable', 'of
                                            y3=-999;
set(handles.SegmentImage, 'Enable',
                                            for pt=1 : numObj
                                                y2=(yholder(:,pt));
set (handles.RecognizeImage, 'Enable',
'off');
                                                 for pt2=1 : numObj
                                                     if yholder(:,pt2)>=y2-22 &&
                                            yholder(:,pt2)<y2-9
% --- Executes on button press in
                                                     y1=yholder(:,pt2);
SegmentImage.
                                                     break;
function
                                                     end
SegmentImage Callback(hObject,
                                                 end
eventdata, handles)
% hObject
            handle to SegmentImage
                                                 for pt2=1 : numObj
(see GCBO)
                                                     if yholder(:,pt2)>=y2+9 &&
% eventdata
                 reserved - to be
                                            yholder(:,pt2)<y2+22</pre>
defined in a future version of
                                                     y3=yholder(:,pt2);
                                                     break;
             structure with handles
% handles
                                                     end
and user data (see GUIDATA)
                                                 end
%segmentation
```

```
if y3>=y2+9 && y3<y2+22 && y1>=y2-22
                                                end
&& y1<y2-9
break;
                                                plot(firstx,80,'g*');
                                                plot(secondx,80,'g*');
end
                                                plot([firstx secondx], [80 80],'b');
end
                                                 for i = 1 : 50
%charcount
                                                     for pt=1:numObj
charcount=1;
pt=0;
                                                (xholder(:,pt)>=secondx+21
                                                                                       & &
firstx=xholder(:,1);
                                                xholder(:,pt) < secondx+34)</pre>
                                                                                       83
secondx=0;
                                                (yholder(:,pt)>=y1-4
                                                                                       83
                                                yholder(:,pt) \le y\bar{3}+4)
for pt=1 : numObj
    save charcount;
                                                             firstx=xholder(:,pt);
                                                             for pt2=1:numObj
        if (xholder(:,pt)>=firstx+16
      xholder(:,pt)<firstx+21)</pre>
                                      & &
                                                                  if
(yholder(:,pt)>=y1-4
                                      & &
                                                (xholder(:,pt2)>=firstx+16
                                                                                       & &
yholder(:,pt) \le y3+4)
                                                xholder(:,pt2)<firstx+21)</pre>
                                                                                       23
             secondx=xholder(:,pt);
                                                (yholder(:,pt)>=y1-4
                                                                                       8 8
             break:
                                                yholder(:,pt) \le y3+4)
        elseif
(xholder(:,pt)>=firstx+21
                                      ኤ ኤ
                                                secondx=xholder(:,pt2);
xholder(:,pt) <= firstx+32)</pre>
                                                                      break;
(yholder(:,pt)>=y1-4
                                                                  else
                                      83
yholder(:,pt) \le y3+14)
             secondx=firstx;
                                                secondx=firstx+20;
             firstx=secondx-20;
                                                                      break:
             break:
                                                                  end
         elseif
                                                             end
(xholder(:,pt)>=firstx+47
                                                             charcount=charcount+1;
                                      23
xholder(:,pt) <= firstx+51)</pre>
                                                             break;
                                      & &
(yholder(:,pt)>=y1-4
                                      & &
                                                         elseif
yholder(:,pt) \le y3+4)
                                                (xholder(:,pt)>=secondx+44
                                                                                       23
                                                xholder(:,pt) <secondx+53)</pre>
             secondx=firstx+20;
                                                                                       83
                                                (yholder(:,pt)>=y1-4
             break;
                                                                                       አ አ
                                                yholder(:,pt) \le y3+4)
          %subject to change
         elseif
                                                             secondx=xholder(:,pt);
(xholder(:,pt)>=firstx+74
                                                             firstx=secondx-20;
xholder(:,pt) <= firstx+83)</pre>
                                      & &
                                                             charcount=charcount+1;
(yholder(:,pt)>=y1-4
                                      23
                                                             break;
                                                         elseif
yholder(:,pt) \le y3+4)
             secondx=firstx;
                                                (xholder(:,pt)>=secondx+74
                                                                                       አ አ
             firstx=secondx-20;
                                                xholder(:,pt) < secondx+83)</pre>
                                                                                       & &
             break;
                                                (yholder(:,pt)>=y1-4
                                                                                       & &
         elseif
                                                yholder(:,pt) \le y3+4)
(xholder(:,pt)>=firstx+96
                                      & &
                                                             firstx=xholder(:,pt);
xholder(:,pt)<=firstx+102)</pre>
                                      & &
                                                             for pt2=1:numObj
(yholder(:,pt)>=y1-4
                                                                  if
                                      83
yholder(:,pt) \le y3+4)
                                                (xholder(:,pt2)>=firstx+16
                                                                                       83
             secondx=firstx+20;
                                                xholder(:,pt2)<firstx+21)</pre>
                                                                                       23
             break:
                                                (yholder(:,pt)>=y1-4
                                                                                       & &
           elseif
                                                yholder(:,pt) \le y3+4)
(xholder(:,pt)>=firstx+113
                                      & &
xholder(:,pt)<=firstx+124)</pre>
                                      & &
                                                secondx=xholder(:,pt2);
(yholder(:,pt)>=y1-4
                                      23
                                                                      break;
yholder(:,pt) \le y3+4)
                                                                  else
             secondx=firstx+20;
             break;
                                                secondx=firstx+20;
          %end subject to change
                                                                      break;
                                                                  end
             firstx=xholder(:,pt);
                                                             end
                                                             charcount=charcount+2;
        end
```

```
break;
                                                (yholder(:,pt)>=y1-4
                                                                                      & &
        elseif
                                               yholder(:,pt) \le y3+4)
(xholder(:,pt)>=secondx+96
                                      & &
                                                             secondx=firstx+20;
xholder(:,pt) < secondx+102)</pre>
                                      23
                                                            break:
(yholder(:,pt)>=y1-4
                                      83
yholder(:,pt) \le y3+4)
                                                        else
                                                             firstx=xholder(:,pt);
             secondx=xholder(:,pt);
                                                        end
             firstx=secondx-20;
                                               end
             charcount=charcount+2;
             break;
                                                                      for pt2 = 1:
                                               numObj
                                                                     if
        end
    end
                                               xyholder(1,pt2)>=firstx-10
                                                                                      & &
plot(firstx,80,'g*');
                                               xyholder(1,pt2)<=firstx+10</pre>
plot(secondx, 80, 'q*');
plot([firstx secondx], [80 80],'b');
                                               xyholder(2,pt2) >= y1-8
                                                                                      ያ ያ
end
                                               xyholder(2,pt2) \le y1+8
charcount
                                               binary(1, marker)=1;
save charcount;
                                                                     elseif
                                               xyholder(2,pt2) >= y2-8
                                                                                      & &
                                               xyholder(2,pt2) \le y2+8
y1
                                               binary(2, marker) = 1;
y2
                                                                     elseif
уЗ
                                               xyholder(2,pt2) >= y3-8
                                                                                      83
                                               xyholder(2,pt2) \le y3+8
%test
                                               binary(3, marker)=1;
binary=[1:charcount ; 1:charcount ;
                                                                      end
1:charcount ; 1:charcount
                                                                     end
1:charcount; 1:charcount];
                                                                     end
for cl=1:6
    for cl2=1:charcount
                                                                      for pt2 = 1:
        binary(cl,cl2)=0;
                                               numObj
    end
                                                                     if
end
                                               xyholder(1,pt2) >= secondx-10
                                                                                      & &
                                               xyholder(1,pt2) <= secondx+10
marker=1;
firstx=xholder(:,1);
                                               xyholder(2,pt2) >= y1-8
                                                                                      23
secondx=0;
                                               xyholder(2,pt2) \le y1+4
for pt=1 : numObj
        if (xholder(:,pt)>=firstx+16
                                               binary(4, marker)=1;
      xholder(:,pt)<firstx+21)</pre>
                                      & &
                                                                     elseif
(yholder(:,pt)>=y1-4
                                               xyholder(2,pt2) >= y2-8
                                                                                      & &
                                      & &
yholder(:,pt) \le y3+4)
                                               xyholder(2,pt2) \le y2+8
            secondx=xholder(:,pt);
           break;
                                               binary(5, marker)=1;
                                                                     elseif
        elseif
                                               xyholder(2,pt2)>=y3-8
                                                                                      & &
(xholder(:,pt)>=firstx+21
                                      & &
                                               xyholder(2,pt2) \le y3+8
xholder(:,pt) <= firstx+32)</pre>
                                      & &
(yholder(:,pt)>=y1-4
                                      & &
                                               binary(6, marker)=1;
yholder(:,pt) \le y3+4)
                                                                      end
             secondx=firstx;
                                                                     end
             firstx=secondx-20;
                                                                     end
             break;
        elseif
(xholder(:,pt)>=firstx+47
                                      & &
                                               for i = 1 : 50
xholder(:,pt)<=firstx+51)</pre>
                                      & &
                                                    for pt=1:numObj
```

```
i f
                                                 (yholder(:,pt)>=y1-4
                                                                                       & &
(xholder(:,pt)>=secondx+21
                                                yholder(:,pt) \le y3+4)
                                       & &
xholder(:,pt) < secondx+32)</pre>
                                       & &
(yholder(:,pt)>=y1-4
                                       23
                                                secondx=xholder(:,pt);
yholder(:,pt) \le y3+4)
                                                                  firstx=secondx-20;
                                                                  marker=marker+2;
             firstx=xholder(:,pt);
                                                              break;
             for pt2=1:numObj
                 if
                                                                       end
(xholder(:,pt2)>=firstx+16
                                       ኤ ኤ
xholder(:,pt2)<firstx+21)</pre>
                                       & &
                                                     end
(yholder(:,pt)>=y1-4
                                       83
yholder(:,pt) \le y3+4)
                                                                       for pt2 = 1:
secondx=xholder(:,pt2);
                                                numObj
                      break;
                 else
                                                xyholder(1,pt2) >= firstx-10
                                                                                       ያ ያ
                                                xyholder(1,pt2)<=firstx+10
secondx=firstx+20;
                                                xyholder(2,pt2) >= y1-8
                      break;
                                                                                       & &
                 end
                                                xyholder(2,pt2) \le y1+8
             end
             marker=marker+1;
                                                binary(1, marker)=1;
             break;
                                                                       elseif
                                                xyholder(2,pt2) >= y2-8
                                                                                       & &
        elseif
                                                xyholder(2,pt2) \le y2+8
(xholder(:,pt)>=secondx+47
                                       83
xholder(:,pt) < secondx+51)</pre>
                                       ኤ ኤ
                                                binary(2,marker)=1;
(yholder(:,pt)>=y1-4
                                       33
                                                                       elseif
yholder(:,pt) \le y3+4)
                                                xyholder(2,pt2) >= y3-8
                                                                                       & &
             secondx=xholder(:,pt);
                                                xyholder(2,pt2) \le y3+8
             firstx=secondx-20;
             marker=marker+1;
                                                binary(3, marker) = 1;
             break:
                                                                       end
                                                                       end
        elseif
                                                                       end
(xholder(:,pt)>=secondx+74
                                       8.8
xholder(:,pt)<secondx+83)</pre>
                                                                       for pt2 = 1:
(yholder(:,pt)>=y1-4
                                                numObj
yholder(:,pt) \le y3+4)
             firstx=xholder(:,pt);
                                                xyholder(1,pt2)>=secondx-10
                                                                                       & &
                                                xyholder(1,pt2) <= secondx+10
             for pt2=1:numObj
                                                                       i f
                 if
                                                xyholder(2,pt2) >= y1-8
                                                                                       & &
(xholder(:,pt2)>=firstx+16
                                       & &
                                                xyholder(2,pt2) \le y1+8
xholder(:,pt2)<firstx+21)</pre>
                                       & &
(yholder(:,pt)>=y1-4
                                       & &
                                                binary(4, marker)=1;
                                                                       elseif
yholder(:,pt) \le y3+4)
                                                xyholder(2,pt2) >= y2-8
                                                                                       83
secondx=xholder(:,pt2);
                                                xyholder(2,pt2) \le y2+8
                 break;
                                                binary(5, marker) = 1;
                                                                       elseif
                 secondx=firstx+20;
                                                xyholder(2,pt2) >= y3-8
                                                                                       & &
                 break;
                                                xyholder(2,pt2) \le y3+8
                 end
                                                binary(6, marker)=1;
             marker=marker+2;
                                                                       end
             break;
                                                                       end
                                                                       end
        elseif
(xholder(:,pt)>=secondx+96
                                       & &
                                                save binary;
xholder(:,pt) < secondx+102)</pre>
                                      & &
                                                end
```

```
elseif numObj==0
                                             totalangle=0;
charcount=0
save charcount;
                                             %Showing the skewed image with lines
end
                                             L= bwlabel(rotated);
                                             s = regionprops(L,'centroid');
                                             %figure('Name','Slopes','NumberTitle
set (handles.BrowseImage, 'Enable', 'of
                                             ','off'),imshow(rotated);
set (handles.CleanImage, 'Enable', 'off
                                             %title('Yellow are the centers');
                                             hold on
');
set (handles.DeskewImage, 'Enable', 'of
                                             numObj = numel(s);
                             'Enable',
set (handles.SegmentImage,
                                             if numObj<3
'off');
set(handles.RecognizeImage, 'Enable',
                                             ee='Insufficient Braille Dots'
'on');
                                             cla(handles.ImageDisplay,'reset');
                                             cla(handles.Deskew, 'reset');
                                             cla(handles.Backside, 'reset');
                                             cla;
% --- Executes on button press in
                                             set (handles.text1, 'String', 'Browsed
                                             Image');
Oneclick.
function Oneclick Callback (hObject,
                                             set(handles.BrowseImage, 'Enable', 'on
eventdata, handles)
                                             ');
% hObject
             handle to Oneclick (see
                                             set (handles.DeskewImage, 'Enable', 'of
GCBO)
   eventdata
                 reserved - to be
                                             set (handles.CleanImage, 'Enable', 'off
defined in a future version of
                                             ');
MATLAB
                                             set(handles.SegmentImage, 'Enable', 'o
% handles
              structure with handles
and user data (see GUIDATA)
                                             set(handles.RecognizeImage, 'Enable',
                                             'off');
                                             set(handles.RecognizedImage,'String'
set(LOADING,'Visible','on')
                                             set (handles. ImageDisplay, 'xcolor', 'w
                                             ','ycolor','w','xtick',[],'ytick',[]
load I;
                                             );
cropedimage = I;
                                             set(handles.Deskew,'xcolor','w','yco
                      %copies
                               T to
cropedimage
                                             lor','w','xtick',[],'ytick',[]);
save cropedimage;
                                             set(handles.Backside,'xcolor','w','y
                                             color','w','xtick',[],'ytick',[]);
set(handles.text1,'visible','on');
gray=imadjust(rgb2gray(cropedimage))
; %Grayscales cropedimage and then
increase its contrast
                                             set(handles.Oneclick, 'Enable', 'off')
imwrite(gray,'C:\MATLAB701\work\dump
\gray.jpg'); %saves the image bw
                                             set(LOADING,'visible','off');
imwrite(gray, 'C:\MATLAB701\work\dump
                                             clear
\gray.jpg'); %saves the image bw
level=graythresh(gray);
bw=im2bw(gray,level);
                                             else
imwrite(bw,'C:\MATLAB701\work\dump\b
                                             for k = 1: numObj
                                                 for i=1 : numObj
w.jpg');
img=imread('C:\MATLAB701\work\dump\b
                                                     if
                                                                 (s(i).Centroid(1)-
w.jpg');
                                             s(k).Centroid(1)) >0
img=double(img);
                                                 slope=double((s(i).Centroid(2)-
inv(:,:,:)=255-img(:,:,:);
                                             s(k).Centroid(2))/(s(i).Centroid(1)-
inv=uint8(inv);
                                             s(k).Centroid(1));
inv=im2bw(inv);
                                                 distance
inv=bwareaopen(inv,30);
                                             sqrt((s(i).Centroid(1)-
imwrite(inv,'C:\MATLAB701\work\dump\
                                             s(k).Centroid(1))^2+(s(i).Centroid(2
inverse.jpg');
                                             )-s(k).Centroid(2))^2;
                                                 format long
%rotated initalization
                                                 slope;
rotated = inv;
                                                           if ((slope < 0.1)
mediangray=gray;
                                             (slope > -0.1)) \&\& distance < 150
save mediangray;
                                                           end
```

```
end
                                                 distance
    end
                                             sqrt((s(i).Centroid(1) -
end
                                             s(k).Centroid(1))^2+(s(i).Centroid(2
                                             )-s(k).Centroid(2))^2;
hold off;
                                                  format long
                                                  slope;
                                                        if loop==1
\mbox{\ensuremath{\mbox{\$}Loop}} start and angle extraction
                                                           if ((slope < 0.1)
                                              (slope > -0.1)) && distance < 150
loop=1;
angle=1;
                                                              point=point+1;
z=0;
while z \le 5
                                             array(:,point)=slope;
L= bwlabel(rotated);
                                             %plot([s(k).Centroid(1)
s = regionprops(L,'centroid');
                                             s(i).Centroid(1)], [s(k).Centroid(2)
                                             s(i).Centroid(2)],'r');
%get total number of accepted slopes
ctrr=0;
                                                         else
numObj = numel(s);
                                                           if ((slope < 0.05) &&
                                              (slope > -0.05)) \&\& distance < 150
for ctr = 1 : numObj
    for ctr2=1 : numObj
                                                              point=point+1;
        if
               (s(ctr2).Centroid(1) -
s(ctr).Centroid(1))>0
                                             array(:,point)=slope;
slope=double((s(ctr2).Centroid(2)-
                                                         end
s(ctr).Centroid(2))/(s(ctr2).Centroi
                                                      end
d(1)-s(ctr).Centroid(1)));
                                                 end
    distance
                                             end
                                             hold off;
sqrt((s(ctr2).Centroid(1)-
s(ctr).Centroid(1))^2+(s(ctr2).Centr
oid(2) - s(ctr) \cdot Centroid(2))^2;
                                             array;
    format long
                                             medianave=median(array);
    slope;
           if loop==1
                                             %Deskewing
             if ((slope < 0.1)
                                             angle=atan(medianave);
(slope > -0.1)) \&\& distance < 150
                                             angle=(angle*180)/pi;
                ctrr=ctrr+1;
                                             rotated= imrotate(rotated, angle);
                                             mediangray=imrotate(mediangray,
             end
           else
                                             angle);
             if
                 ((slope < 0.05)
                                             save mediangray;
                                             imwrite(mediangray,'C:\MATLAB701\wor
(slope > -0.05)) \&\& distance < 150
                                             k\dump\mediangray.jpg');
                ctrr=ctrr+1;
                                             imwrite(rotated,'C:\MATLAB701\work\d
             end
           end
                                             ump\Rotated.jpg');
        end
    end
                                             clear array;
                                             loop=loop+1;
                                             totalangle=totalangle+angle;
%Declaration of array and pointer
                                             z=z+1;
array =1:ctrr;
                                             end
point=0;
                                             %final display label
%Extraction of the useful slopes
                                             L = bwlabel(rotated);
hold on
                                             s = regionprops(L,'centroid');
numObj = numel(s);
                                             hold on
for k = 1: numObj
                                             numObj = numel(s);
                                             for k = 1: numObj
    for i=1 : numObj
        if
                   (s(i).Centroid(1)-
                                                 %backside filtering
s(k).Centroid(1)) >0
                                                 x=s(k).Centroid(1);
    slope=double((s(i).Centroid(2)-
                                                 v=s(k).Centroid(2)-7.5;
s(k).Centroid(2))/(s(i).Centroid(1)-
                                                 p=impixel (mediangray, x, y);
s(k).Centroid(1));
                                                  for i=1 : numObj
```

```
(s(i).Centroid(1)-
s(k).Centroid(1))>0
                                            imread('C:\MATLAB701\work\dump\rotat
    slope=double((s(i).Centroid(2)-
                                            ed.jpg');
s(k).Centroid(2))/(s(i).Centroid(1)-
                                                     col = [(s(k).Centroid(1)-7)]
                                             (s(k).Centroid(1)+7)
s(k).Centroid(1));
    distance
                                             (s(k).Centroid(1)+7)
sqrt((s(i).Centroid(1) -
                                             (s(k).Centroid(1)-7)];
s(k).Centroid(1))^2+(s(i).Centroid(2
                                                    row = [(s(k).Centroid(2)-7)]
)-s(k).Centroid(2))^2;
                                             (s(k).Centroid(2)-7)
    format long
                                             (s(k).Centroid(2)+7)
                                             (s(k).Centroid(2)+7)];
            if loop==1
                                                    J = roifill(I2, col, row);
                if ((slope < 0.1) &&
                                                    M = roifill(K,col,row);
(slope > -0.1)) && distance <150
                end
                                            imwrite(J,'C:\MATLAB701\work\dump\me
            else
                                            diangray.jpg');
                if ((slope < 0.05)
&& (slope > -0.05)) && distance <150
                                            imwrite(M,'C:\MATLAB701\work\dump\ro
                                            tated.jpg');
                end
            end
                                                     rotated=M;
        end
                                                    mediangray=J;
    end
                                                     save mediangray;
end
                                                end
hold off;
                                            end
                                            hold off;
dis1=imread('C:\MATLAB701\work\dump\
                                            end
mediangray.jpg');
dis1 =cat (3, dis1 ,dis1 ,dis1);
axes(handles.Deskew);
image (dis1);
                                            dis2=imread('C:\MATLAB701\work\dump\
axis off;
                                            mediangray.jpg');
                                            dis2 =cat (3, dis2 ,dis2 ,dis2);
totalangle
                                            axes(handles.Backside);
                                            image(dis2);
                                            axis off;
%erasing
load mediangray;
for po =1:3
rotated=imread('C:\MATLAB701\work\du
                                            %segmentation
                                            rotated=imread('C:\MATLAB701\work\du
mp\rotated.jpg');
rotated=uint8(rotated);
                                            mp\rotated.jpg');
rotated=im2bw(rotated);
                                            rotated=uint8(rotated);
                                            rotated=im2bw(rotated);
%figure('Name','Erasing','NumberTitl
e','off'),imshow(mediangray);
                                            L = bwlabel(rotated);
hold on
                                            s = regionprops(L,'centroid');
L = bwlabel(rotated);
                                            hold on
s = regionprops(L,'centroid');
                                            numObj = numel(s);
numObj = numel(s);
for k = 1 : numObj
                                            if numObj>0
                                            %xholder
    %backside filtering
                                            xholder=1:numObj;
    x=s(k).Centroid(1);
                                            for iz = 1 : numObj
    y=s(k).Centroid(2);
                                            xholder(:,iz) = s(iz).Centroid(1);
    x2=s(k).Centroid(1)-4;
                                            format short;
    y2=s(k).Centroid(2)-7;
                                            end
    p=impixel(mediangray, x2, y2);
                                            xholder
    if p(1,1) \le 210
        Ι2
                                            %vholder
imread('C:\MATLAB701\work\dump\media
                                            vholder=1:numObj;
ngray.jpg');
                                            for iz = 1 : numObj
                                            yholder(:,iz) = s(iz).Centroid(2);
                                            format short;
```

and		h ma a la .	
end yholder		break; elseif	
ynorder	(xholder	(:,pt)>=firstx+47	& &
%xyholder		(:,pt) <=firstx+51)	& &
<pre>xyholder=[1:numObj;1:numObj];</pre>		(:,pt)>=y1-4	& &
for iz = 1 : numObj	_	(:,pt) <=y3+4);	
xyholder(1,iz)=s(iz).Centroid(1);	_	secondx=firstx+20;	
xyholder(2,iz)=s(iz).Centroid(2);		break;	
format short;		%subject to change	
end		elseif	
xyholder	·	(:,pt)>=firstx+74	& &
		(:,pt)<=firstx+83)	& &
1 000		(:,pt) >= y1-4	& &
y1=-999;	Auoraer ((:,pt) <=y3+4);	
y2=-999;		<pre>secondx=firstx; firstx=secondx-20;</pre>	
y3=-999;		break;	
for pt=1 : numObj		elseif	
y2=(yholder(:,pt));	(vholder	c(:,pt)>=firstx+96	& &
y2-(y1101de1(.,pc)),		(:,pt)<=firstx+102)	& &
for pt2=1 : numObj		(:,pt) >= y1-4	& &
if yholder(:,pt2)>=y2-22		(:,pt)<=y3+4);	u u
yholder(:,pt2) <y2-9< td=""><td>7</td><td>secondx=firstx+20;</td><td></td></y2-9<>	7	secondx=firstx+20;	
y1=yholder(:,pt2);		break;	
break;		elseif	
end	(xholder	c(:,pt)>=firstx+113	& &
end	xholder((:,pt)<=firstx+124)	& &
	(yholder	c(:,pt)>=y1-4	& &
for pt2=1 : numObj	yholder((:,pt)<=y3+4);	
if $yholder(:,pt2) >= y2+9$	ά &	<pre>secondx=firstx+20;</pre>	
yholder(:,pt2) <y2+22< td=""><td></td><td>break;</td><td></td></y2+22<>		break;	
y3=yholder(:,pt2);		%end subject to change	
break;		else	
end		<pre>firstx=xholder(:,pt);</pre>	
end	1	end	
if2>2+0 cc2<2+02 cc1>0	end		
if y3>=y2+9 && y3 <y2+22 &&="" y1="">=y2- && y1<y2-9< td=""><td></td><td>caty 00 la*!).</td><td></td></y2-9<></y2+22>		caty 00 la*!).	
break;		cstx,80,'g*'); condx,80,'g*');	
end	_	rstx secondx], [80 80],'b	').
end	Pioc ([ii	ista secondaj, [00 00], b	<i>,</i> ,
end	for i =	= 1 : 50	
%charcount	for	pt=1:numObj	
<pre>charcount=1;</pre>		if	
pt=0;	(xholder	(:,pt)>=secondx+21	& &
<pre>firstx=xholder(:,1);</pre>		(:,pt) <secondx+34)< td=""><td>& &</td></secondx+34)<>	& &
secondx=0;		c(:,pt)>=y1-4	& &
for pt=1 : numObj	yholder((:,pt)<=y3+4);	
save charcount;	_	<pre>firstx=xholder(:,pt);</pre>	
<pre>if (xholder(:,pt)>=firstx+</pre>		for pt2=1:numObj	
&& xholder(:,pt) <firstx+21)< td=""><td>ίδ. </td><td>if</td><td></td></firstx+21)<>	ίδ. 	if	
(yholder(:,pt)>=y1-4	·	c(:,pt2)>=firstx+16 (:,pt2) <firstx+21)< td=""><td>& &</td></firstx+21)<>	& &
<pre>yholder(:,pt) <= y3+4); secondx=xholder(:,pt);</pre>		:,pt2)<:1rstx+21) :(:,pt)>=y1-4	& & & &
break;		(:,pt) <=y1-4 (:,pt) <=y3+4);	αα
elseif	Augraei (·, Pc/ /- 32/4/,	
(xholder(:,pt)>=firstx+21	& secondx=	<pre>exholder(:,pt2);</pre>	
xholder(:,pt)<=firstx+32)	ik seconda-	break;	
(yholder(:,pt)>=y1-4	i &	else	
yholder(:,pt)<=y3+14);			
secondx=firstx;	secondx=	firstx+20;	
firstx=secondx-20;		break;	

```
end
                                                %test
             charcount=charcount+1;
             break;
                                               binary=[1:charcount ; 1:charcount ;
        elseif
                                               1:charcount ; 1:charcount
(xholder(:,pt)>=secondx+44
                                               1:charcount; 1:charcount];
                                      23
xholder(:,pt)<secondx+53)</pre>
                                      & &
                                                for cl=1:6
(yholder(:,pt)>=y1-4
                                                    for cl2=1:charcount
                                      ያ ያ
yholder(:,pt) \le y3+4);
                                                        binary(cl,cl2)=0;
             secondx=xholder(:,pt);
             firstx=secondx-20;
                                                end
             charcount=charcount+1;
             break:
                                               marker=1;
                                               firstx=xholder(:,1);
        elseif
(xholder(:,pt)>=secondx+74
                                                secondx=0;
                                      ኤ ኤ
xholder(:,pt) < secondx + 83)</pre>
                                      & &
                                                for pt=1 : numObj
(yholder(:,pt)>=y1-4
                                      ያ ያ
                                                        if (xholder(:,pt)>=firstx+16
yholder(:,pt) <=y3+4);;</pre>
                                                      xholder(:,pt)<firstx+21)</pre>
             firstx=xholder(:,pt);
                                                (yholder(:,pt)>=y1-4
                                                                                      23
             for pt2=1:numObj
                                               yholder(:,pt) \le y3+4);
                 if
                                                            secondx=xholder(:,pt);
(xholder(:,pt2)>=firstx+16
                                                           break;
                                      83
xholder(:,pt2)<firstx+21)</pre>
                                      ኤ ኤ
(yholder(:,pt)>=y1-4
                                                        elseif
yholder(:,pt) \le y3+4);
                                                (xholder(:,pt)>=firstx+21
                                                                                      83
                                               xholder(:,pt) <=firstx+32)</pre>
                                                                                      & &
secondx=xholder(:,pt2);
                                                (yholder(:,pt)>=y1-4
                                                                                      83
                                               yholder(:,pt) \le y3+4);
                     break;
                                                             secondx=firstx;
                 else
                                                             firstx=secondx-20;
secondx=firstx+20;
                                                             break:
                     break;
                 end
                                                        elseif
                                                (xholder(:,pt)>=firstx+47
             end
                                                                                      23
             charcount=charcount+2;
                                                xholder(:,pt) <= firstx+51)</pre>
                                                                                      & &
             break:
                                                (yholder(:,pt)>=y1-4
                                                                                      አ አ
        elseif
                                               yholder(:,pt) \le y3+4);
(xholder(:,pt)>=secondx+96
                                                             secondx=firstx+20;
                                      23
xholder(:,pt) < secondx+102)</pre>
                                                             break;
(yholder(:,pt)>=y1-4
yholder(:,pt) \le y3+4);
                                                        else
                                                             firstx=xholder(:,pt);
             secondx=xholder(:,pt);
                                                        end
             firstx=secondx-20;
                                               end
             charcount=charcount+2;
             break;
                                                                      for pt2 = 1:
                                               numObj
        end
                                                                      i f
                                               xyholder(1,pt2)>=firstx-10
    end
                                                                                      & &
plot(firstx,80,'g*');
                                               xyholder(1,pt2)<=firstx+10;</pre>
plot(secondx,80,'g*');
                                                                      i f
plot([firstx secondx], [80 80],'b');
                                               xyholder(2,pt2) >= y1-8
                                                                                      & &
                                               xyholder(2,pt2) \le y1+8;
end
charcount
                                               binary(1, marker)=1;
save charcount;
                                                                      elseif
                                               xyholder(2,pt2) >= y2-8
                                                                                      83
                                               xyholder(2,pt2) \le y2+8;
                                               binary(2, marker) = 1;
у1
                                                                      elseif
y2
                                               xyholder(2,pt2) >= y3-8
                                                                                      & &
                                               xyholder(2,pt2) \le y3+8;
уЗ
```

```
(yholder(:,pt)>=y1-4
                                                                                       & &
binary(3, marker)=1;
                                                yholder(:,pt) \le y3+4);
                                                              secondx=xholder(:,pt);
                      end
                      end
                                                              firstx=secondx-20;
                                                             marker=marker+1;
                      end
                                                             break;
                      for pt2 = 1:
                                                         elseif
numObj
                                                (xholder(:,pt)>=secondx+74
                                                                                       & &
xyholder(1,pt2) >= secondx-10
                                       & &
                                                xholder(:,pt) < secondx+83)</pre>
                                                                                       ያ ያ
xyholder(1,pt2) <= secondx+10;
                                                (yholder(:,pt)>=y1-4
                                                                                       & &
                                                yholder(:,pt) \le y3+4);
                      if
                                                              firstx=xholder(:,pt);
xyholder(2,pt2) >= y1-8
                                       & &
xyholder(2,pt2) \le y1+4
                                                              for pt2=1:numObj;
binary(4, marker)=1;
                                                                  if
                      elseif
                                                (xholder(:,pt2)>=firstx+16
                                                                                       & &
xyholder(2,pt2) >= y2-8
                                       & &
                                                xholder(:,pt2)<firstx+21)</pre>
                                                                                       & &
xyholder(2,pt2) \le y2+8;
                                                (yholder(:,pt)>=y1-4
                                                                                       23
                                                yholder(:,pt) \le y3+4);
binary(5,marker)=1;
                                                secondx=xholder(:,pt2);
                      elseif
xyholder(2,pt2) >= y3-8
                                       & &
                                                                  break;
xyholder(2,pt2)<=y3+8;
                                                                  else
                                                                  secondx=firstx+20;
binary(6, marker)=1;
                                                                  break:
                      end
                                                                  end
                      end
                      end
                                                              end
                                                              marker=marker+2;
                                                              break;
for i = 1 : 50;
                                                         elseif
    for pt=1:numObj;
                                                (xholder(:,pt)>=secondx+96
                                                                                       23
                                                xholder(:,pt) < secondx+102)</pre>
        if
                                                                                       & &
(xholder(:,pt)>=secondx+21
                                       ኤ ኤ
                                                (yholder(:,pt)>=y1-4
                                                                                       አ አ
xholder(:,pt) < secondx + 32)</pre>
                                       & &
                                                yholder(:,pt) \le y3+4);
(yholder(:,pt)>=y1-4
                                       23
yholder(:,pt) \le y3+4);
                                                secondx=xholder(:,pt);
             firstx=xholder(:,pt);
                                                                  firstx=secondx-20;
                                                                  marker=marker+2;
             for pt2=1:numObj;
                                                             break:
                 if
(xholder(:,pt2)>=firstx+16
                                       & &
                                                                       end
xholder(:,pt2)<firstx+21)</pre>
                                      & &
(yholder(:,pt)>=y1-4
                                                     end
yholder(:,pt) \le y3+4);
secondx=xholder(:,pt2);
                                                                       for pt2 = 1:
                      break;
                                                numObj;
                 else
                                                xyholder(1,pt2) >= firstx-10
                                                                                       83
secondx=firstx+20;
                                                xyholder(1,pt2)<=firstx+10;</pre>
                      break;
                 end
                                                xyholder(2,pt2) >= y1-8
                                                                                       & &
             end
                                                xyholder(2,pt2) \le y1+8;
             marker=marker+1;
             break;
                                                binary(1, marker)=1;
                                                                      elseif
        elseif
                                                xyholder(2,pt2) >= y2-8
                                                                                       & &
(xholder(:,pt)>=secondx+47
                                                xyholder(2,pt2) \le y2+8;
xholder(:,pt) < secondx+51)</pre>
                                       & &
                                                binary(2, marker)=1;
```

```
load binary;
xyholder(2,pt2) >= y3-8
                                     & &
                                              binary
xyholder(2,pt2) \le y3+8;
                                              11=1
binary(3, marker) = 1;
                                              while(u<=charcount)</pre>
                                                  z=num2str(binary(1,u));
                     end
                                                  b=[z];
                     end
                     end
                                                  for 0 = 2 : 6
                                                  a=binary(o,u);
                     for pt2 = 1:
                                                  str1=num2str(a);
numObj;
                                                  b=[b str1];
                                                  end
xyholder(1,pt2)>=secondx-10
                                                  db(u,:) = strvcat(b);
                                     & &
                                                  u=u+1;
xyholder(1,pt2) <=secondx+10;</pre>
                     i f
                                              end
xyholder(2,pt2) >= y1-8
                                     & &
xyholder(2,pt2) \le y1+8;
                                              db
                                                  db1='123456';
binary(4, marker)=1;
                     elseif
                                                  conn = database('Braille', '',
xyholder(2,pt2) >= y2-8
                                     33
xyholder(2,pt2) \le y2+8;
                                              setdbprefs('DataReturnFormat','cella
binary(5, marker)=1;
                                              rray');
                     elseif
                                                         = exec(conn,
                                                  curs
                                                                           ['select
                                              Letter from Gradel where ID= ' ''''
xyholder(2,pt2) >= y3-8
                                     & &
                                              db1 '''']);
xyholder(2,pt2) \le y3+8;
                                                  curs = fetch(curs);
binary(6, marker) = 1;
                                                  bb = curs.data;
                     end
                                                  db1='000000';
                     end
                     end
                                                  conn = database('Braille', '',
                                              '');
save binary;
end
elseif numObj==0
                                              setdbprefs('DataReturnFormat','cella
charcount=0
                                              rray');
save charcount;
                                                                           ['select
                                                  curs
                                                             exec(conn,
                                              Letter from Gradel where ID= ' ''''
end
                                              db1 '''']);
                                                  curs = fetch(curs);
load charcount;
                                                  space = curs.data;
charcount
if charcount==0
                                              end
ee='No Character Detected'
set(LOADING,'Visible','off')
                                              if charcount>=1
set (handles.RecognizedImage, 'String'
                                              db1 = [db(1,:)]
,ee);
                                              met=1;
set(handles.BrowseImage, 'Enable', 'on
                                              epox=1;
                                              while(epox<=charcount)</pre>
');
set (handles.CleanImage, 'Enable', 'off
                                                  epox
                                                  lft=0;
');
set (handles.DeskewImage, 'Enable', 'of
                                                  rgt=0;
f');
                                                  if epox<charcount && charcount>1
set(handles.SegmentImage,
                             'Enable',
                                                  db2=[db1 db(epox+1,:) '%']
'off');
                                                  elseif charcount==1
                                                  db2='12345'
set(handles.RecognizeImage, 'Enable',
                                                  end
set(handles.Oneclick, 'Enable', 'off')
                                                  conn = database('Braille', '',
                                              '');
```

elseif charcount>=1

```
setdbprefs('DataReturnFormat','cella
                                                  spaceb=met-1
                                                  while spaceb>=1
   curs = exec(conn, ['select Words
                                                  arrayy(epox-spaceb)=space;
from Grade2 where ID like ' '''' db2
                                                  spaceb=spaceb-1;
''''])
                                                  end
    curs = fetch(curs)
    aa = curs.data
                                                  db1=db(epox+1,:)
                                                  met=1
    %Empty query checker
                                                  epox=epox+1;
    n = numel(curs.data)
                                                 %Prefix
    b=0:
    if n==1
                                                 elseif (b == 1 && n == 1) &&
                                           lft==1 && rgt==0 && met>1
    b = strcmp(bb, aa)
                                                   curs = exec(conn, ['select
    end
    %left & right checker
                                           Words from Pre_fix where ID= ' ''''
                                           db1 '''')
    if epox+1<charcount
                                                  curs = fetch(curs)
rgt=strcmp(db(epox+1,:),'000000')
                                                  aa = curs.data
                                                  n = numel(curs.data)
    if epox-met>=1%error catcher if
                                                  b=strcmp(bb,aa)
ctr-met will equal to 0
    lft=strcmp(db(epox-
                                                   if (b == 1 \&\& n == 1)
met,:),'000000')
                                                   ppp=met-1
    end
                                                   while ppp>=1
                                                   epox=epox-ppp
    %query for standalone grade 2
                                                  db1=db(epox,:)
braille characters(group) for the
                                                  curs = exec(conn, ['select
first
                                          Letter from Gradel where ID= ' ''''
    %including the first char
                                           db1 '''')
    if (b == 1 && n == 1) && rgt==1
                                                  curs = fetch(curs)
&& met>1 && epox-met<1
                                                  aa = curs.data
       curs = exec(conn, ['select
                                                  n = numel(curs.data)
Words from Grade2 where ID= ' ''''
                                                  b=strcmp(bb,aa)
db1 ''''])
                                                  arrayy(epox) = aa
       curs = fetch(curs)
                                                  db1=db(epox+1,:)
        aa = curs.data
                                                  met=1
        n = numel(curs.data)
                                                  epox=epox+ppp;
       b=strcmp(bb,aa)
                                                  ppp=ppp-1;
       arrayy(epox) = aa
                                                   end
        spaceb=met-1
                                                  else
                                                  curs = exec(conn, ['select
        while spaceb>=1
                                         Words from Pre_fix where ID= ' ''''
        arrayy(epox-spaceb) = space;
                                          db1 '''')
        spaceb=spaceb-1;
        end
                                                  curs = fetch(curs)
                                                  aa = curs.data
        db1=db(epox+1,:)
                                                  n = numel(curs.data)
       met=1
                                                  b=strcmp(bb,aa)
        epox=epox+1;
                                                  arrayy(epox) = aa
                                                  spaceb=met-1
     %Sure standalone
                                                  while spaceb>=1
     elseif (b == 1 && n == 1) &&
                                                  arrayy(epox-spaceb)=space;
lft==1 && rgt==1 && met>1
                                                  spaceb=spaceb-1;
       curs = exec(conn, ['select
                                                  end
Words from Grade2 where ID= ' ''''
db1 '''')
                                                  db1=db(epox+1,:)
       curs = fetch(curs)
                                                  met=1
       aa = curs.data
                                                  epox=epox+1;
       n = numel(curs.data)
                                                   end
       b=strcmp(bb,aa)
                                                 %Suffix
        arrayy(epox) = aa
```

```
elseif (b == 1 && n == 1) &&
lft==0 && rgt==1 && met>1
                                                  if (b == 1 \&\& n == 1)
                                                  ppp=met-1
       curs = exec(conn, ['select
                                                  while ppp>=1
Words from Suffix where ID= ' ''''
                                                  epox=epox-ppp
db1 ''''])
                                                  db1=db(epox,:)
       curs = fetch(curs)
                                                  curs = exec(conn, ['select
                                          Letter from Gradel where ID= ' ''''
       aa = curs.data
                                          db1 ''''])
       n = numel(curs.data)
       b=strcmp(bb,aa)
                                                  curs = fetch(curs)
                                                  aa = curs.data
       if (b == 1 \&\& n == 1)
                                                  n = numel(curs.data)
       ppp=met-1
                                                  b=strcmp(bb,aa)
       while ppp>=1
                                                  arrayy(epox) = aa
       epox=epox-ppp
                                                  db1=db(epox+1,:)
       db1=db(epox,:)
                                                  met=1
       curs = exec(conn, ['select
                                                  epox=epox+ppp;
Letter from Gradel where ID= ' ''''
                                                  ppp=ppp-1;
db1 ''''])
                                                  end
       curs = fetch(curs)
       aa = curs.data
                                                   else
                                                  curs = exec(conn, ['select
       n = numel(curs.data)
       b=strcmp(bb,aa)
                                           Words from Mid fix where ID= ' ''''
                                           db1 ''''])
       arrayy(epox) = aa
       db1=db(epox+1,:)
                                                  curs = fetch(curs)
                                                  aa = curs.data
       met=1
                                                  n = numel(curs.data)
       epox=epox+ppp;
                                                  b=strcmp(bb,aa)
       ppp=ppp-1;
       end
                                                  arrayy(epox) = aa
                                                  spaceb=met-1
       curs = exec(conn, ['select
                                                  while spaceb>=1
Words from Suffix where ID= ' ''''
                                                  arrayy(epox-spaceb)=space;
db1 ''''])
                                                  spaceb=spaceb-1;
       curs = fetch(curs)
                                                  end
       aa = curs.data
       n = numel(curs.data)
                                                  db1=db(epox+1,:)
       b=strcmp(bb,aa)
                                                  met=1
       arrayy(epox) = aa
                                                  epox=epox+1;
                                                  end
       spaceb=met-1
       while spaceb>=1
       arrayy(epox-spaceb) = space;
                                          %try
                                               elseif (b==1 && n==1)&& met==1
       spaceb=spaceb-1;
       end
                                           && lft==1 && rgt==1
                                                  curs = exec(conn, ['select
       db1=db(epox+1,:)
                                           Words from Stand Alone where ID= '
                                           '''' db1 ''''])
       met=1
                                                  curs = fetch(curs)
       epox=epox+1;
       end
                                                  aa = curs.data
                                                  n = numel(curs.data)
                                                  b=strcmp(bb,aa)
      %Midfix
                                                  arrayy(epox) = aa
      elseif (b == 1 \&\& n == 1) &&
                                                  if epox<charcount
lft==0 && rgt==0 && met>1
                                                  db1=db(epox+1,:)
                                                  end
       curs = exec(conn, ['select
                                                  met=1
Words from Mid fix where ID= ' ''''
                                                  epox=epox+1;
db1 ''''])
                                               elseif (b==1 && n==1)&& met==1
       curs = fetch(curs)
       aa = curs.data
                                          && lft==1 && epox+1>charcount
       n = numel(curs.data)
       b=strcmp(bb,aa)
```

```
%query for grade 2 single
       curs = exec(conn, ['select
Words from Stand Alone where ID= '
                                          braille char standalone (condition is
'''' db1 ''''])
       curs = fetch(curs)
                                                   %there's no match in the
                                           grade 1 DB)
       aa = curs.data
       n = numel(curs.data)
                                                   if (b == 1 && n == 1)
       b=strcmp(bb,aa)
                                                   curs = exec(conn, ['select
                                           Words from Grade2 where ID= ' ''''
       arrayy(epox) = aa
       if epox<charcount
                                           db1 ''''])
       db1=db(epox+1,:)
                                                  curs = fetch(curs)
       end
                                                   aa = curs.data
       met=1
                                                  n = numel(curs.data)
                                                  b=strcmp(bb,aa)
       epox=epox+1;
    elseif (b==1 && n==1)&& met==1
                                                      if n>1 && epox-met<1 &&
&& rgt==1 && epox-met<1
                                           rqt==0
       curs = exec(conn, ['select
                                                         curs = exec(conn,
Words from Stand Alone where ID= '
                                           ['select Words from Pre Fix where
'''' db1 ''''])
                                           ID= ' '''' db1 ''''])
                                                         curs = fetch(curs)
       curs = fetch(curs)
       aa = curs.data
                                                         aa = curs.data
       n = numel(curs.data)
                                                         n = numel(curs.data)
       b=strcmp(bb,aa)
                                                         b=strcmp(bb,aa)
       arrayy(epox) = aa
       if epox<charcount
                                                      elseif n>1 && lft==0 &&
       db1=db(epox+1,:)
                                           rgt==1 && epox==1
       end
                                                               =
                                                                   exec(conn,
                                                         curs
                                                    Words from Stand Alone
       met=1
                                           ['select
                                           where ID= ' '''' db1 '''')
       epox=epox+1;
                                                         curs = fetch(curs)
    elseif (b==1 && n==1)&& met==1
                                                         aa = curs.data
&& epox+1>charcount && epox-met<1
                                                         n = numel(curs.data)
       curs = exec(conn, ['select
                                                         b=strcmp(bb,aa)
Words from Stand Alone where ID= '
'''' db1 ''''])
                                                      elseif n>1 && lft==0 &&
       curs = fetch(curs)
                                           rqt==0
                                           curs = exec(conn,
['select Words from Mid_Fix where
       aa = curs.data
       n = numel(curs.data)
                                           ID= ' '''' db1 ''''])
       b=strcmp(bb,aa)
       arrayy(epox) = aa
                                                         curs = fetch(curs)
                                                         aa = curs.data
       if epox<charcount
       db1=db(epox+1,:)
                                                         n = numel(curs.data)
       end
                                                         b=strcmp(bb,aa)
       met=1
                                                      elseif n>1 && lft==0 &&
       epox=epox+1;
                                           rqt==1
%not sure
                                                         curs = exec(conn,
                                           ['select Words from Suffix where ID=
                                            '''' db1 ''''])
    %query for grade 1 braille(since
grade 2 1 braille char standalone
                                                         curs = fetch(curs)
are unique)
                                                         aa = curs.data
             (b==1 \&\& n == 1) \&\&
                                                         n = numel(curs.data)
   elseif
                                                         b=strcmp(bb,aa)
       curs = exec(conn, ['select
Letter from Gradel where ID= ' ''''
                                                      elseif n>1 && lft==1 &&
db1 ''''])
                                           rgt==0
       curs = fetch(curs)
                                                         curs = exec(conn,
       aa = curs.data
                                           ['select Words from Pre Fix where
                                           ID= ' '''' db1 ''''])
       n = numel(curs.data)
                                                         curs = fetch(curs)
       b=strcmp(bb,aa)
                                                         aa = curs.data
                                                         n = numel(curs.data)
                                                         b=strcmp(bb,aa)
```

```
ee=strvcat(arrayy(1));
            elseif n>1 && lft==1 &&
                                           for cc = 2: charcount
rgt==1
                                           ff=strvcat(arrayy(cc));
               curs
                    =
                         exec(conn,
                                           ee=[ee ff];
['select Words from Stand Alone
                                           end
where ID= ' '''' db1 ''''])
                                            66
              curs = fetch(curs)
               aa = curs.data
                                            set(LOADING,'Visible','off')
               n = numel(curs.data)
               b=strcmp(bb,aa)
                                            set (handles.RecognizedImage, 'String'
                                            ,ee);
                                            set (handles.BrowseImage, 'Enable', 'on
            end
                                            ');
        end
                                            set(handles.CleanImage, 'Enable', 'off
        %proceed here if match is
                                            ');
already found
                                            set (handles.DeskewImage, 'Enable', 'of
       arrayy(epox) = aa
                                            f');
        if epox<charcount
                                            set (handles.SegmentImage,
                                                                        'Enable',
        db1=db(epox+1,:)
                                            'off');
        end
                                            set (handles.RecognizeImage, 'Enable',
       met=1
                                            'off');
       epox=epox+1;
                                            set (handles.Oneclick, 'Enable', 'off')
    %db1 incrementation if there's
no match in any of the database
                                           end
   if epox+1<=charcount
   db1=[db1 db(epox+1,:)]
                                            % --- Executes on button press in
   met=met+1
                                           Showall.
   epox=epox+1;
                                            function
                                                       Showall Callback(hObject,
                                            eventdata, handles)
   %trial
                                            % hObject
                                                         handle to Showall (see
   else
                                           GCBO)
        curs = exec(conn, ['select
                                           % eventdata
                                                           reserved - to be
Words from Grade2 where ID= ' ''''
                                           defined in a future version of
db1 ''''])
                                           MATLAB
       curs = fetch(curs)
                                            % handles
                                                         structure with handles
        aa = curs.data
                                            and user data (see GUIDATA)
        n = numel(curs.data)
        b=strcmp(bb,aa)
                                            set (handles.BrowseImage, 'Visible', 'o
        arrayy(epox) = aa
                                           n');
                                            set(handles.CleanImage,'Visible','on
        spaceb=met-1
                                            ');
                                           set(handles.DeskewImage,'Visible','o
        while spaceb>=1
        arrayy(epox-spaceb)=space;
                                           n');
        spaceb=spaceb-1;
                                            set (handles.SegmentImage, 'Visible',
        end
                                            set (handles.RecognizeImage, 'Visible'
                                            ,'on');
        epox=epox+1
   end
                                           set(handles.Oneclick,'Visible','Off'
    %trial
                                           );
                                            set (handles.BrailleAlphabet, 'Visible
                                            ','on');
   end
                                            set (handles.BrailleContractions, 'Vis
                                            ible','on');
   close(conn)
                                            set(handles.ResetAll,'Visible','on')
   close(curs)
                                            set(handles.Exit,'Visible','on');
                                           set(handles.uipanel2,'Visible','on')
end
                                            set (handles.uipanel4, 'Visible', 'on')
result = [arrayy(:)]
```

```
set (handles.RecognizedImage, 'Visible
                                             set(handles.uipanel7,'Visible','off'
set(handles.uipanel7,'Visible','on')
                                             set(handles.uipanel8,'Visible','off'
                                             set (handles.text1, 'Visible', 'off');
set (handles.uipanel8, 'Visible', 'on')
                                             set(handles.ImageDisplay,'Visible','
                                             off');
set(handles.text1,'Visible','on');
set(handles.ImageDisplay,'Visible','
                                             set(handles.uipanel6,'Visible','off'
on');
set (handles.uipanel6, 'Visible', 'on')
                                             set (handles.text3, 'Visible', 'off');
                                             set(handles.Deskew,'Visible','off');
set(handles.text3,'Visible','on');
                                             set(handles.uipanel10,'Visible','off
set(handles.Deskew,'Visible','on');
                                             ');
set(handles.uipanel10,'Visible','on'
                                             set(handles.text12,'Visible','off');
                                             set (handles.Backside, 'Visible', 'off'
);
set(handles.text12,'Visible','on');
set (handles.Backside, 'Visible', 'on')
                                             set (handles.Showall, 'Visible', 'on');
                                             set(handles.Back,'Visible','off');
set(handles.Showall,'Visible','off')
                                             set (handles.Oneprocess, 'Visible', 'on
                                             ');
set(handles.Back,'Visible','on');
                                             set(handles.Banner,'Visible','on');
set (handles.Oneprocess, 'Visible', 'of
                                             set(handles.uipanel12,'Visible','off
f');
                                             ');
set(handles.Banner,'Visible','off');
set (handles.uipanel12, 'Visible', 'on'
                                             cla(handles.ImageDisplay, 'reset');
                                             cla(handles.Deskew, 'reset');
                                             cla(handles.Backside, 'reset');
% --- Executes on button press in
                                             cla:
                                             set (handles.text1, 'String', 'Browsed
Back.
               Back Callback (hObject,
                                             Image');
function
eventdata, handles)
                                             set (handles.BrowseImage, 'Enable', 'on
                handle to Back (see
GCBO)
                                             set (handles.DeskewImage, 'Enable', 'of
% eventdata
                reserved - to be
defined in a future version of
                                             set (handles.CleanImage, 'Enable', 'off
MATTAB
                                             ');
% handles
              structure with handles
                                             set (handles.SegmentImage, 'Enable', 'o
and user data (see GUIDATA)
                                             ff');
set (handles.BrowseImage, 'Visible', 'o
                                             set (handles.RecognizeImage, 'Enable',
                                             'off');
set (handles.CleanImage, 'Visible', 'of
                                             set(handles.RecognizedImage,'String'
f');
                                             ,'');
set(handles.DeskewImage,'Visible','o
                                             set(handles.ImageDisplay,'xcolor','w
                                             ','ycolor','w','xtick',[],'ytick',[]
set(handles.SegmentImage, 'Visible',
                                             );
                                             set (handles.Deskew, 'xcolor', 'w', 'yco
'off');
set (handles.RecognizeImage, 'Visible'
                                             lor','w','xtick',[],'ytick',[]);
                                             set (handles.Backside, 'xcolor', 'w', 'y
,'off');
set(handles.Oneclick,'Visible','Off'
                                             color','w','xtick',[],'ytick',[]);
                                             set(handles.text1,'visible','on');
);
set (handles.BrailleAlphabet, 'Visible
                                             set(handles.Oneclick, 'Enable', 'off')
','off');
set (handles.BrailleContractions, 'Vis
                                             set(handles.Angle,'String',' ');
ible','off');
                                             clear
set(handles.ResetAll,'Visible','off'
                                             % --- Executes on button press in
set(handles.Exit,'Visible','off');
                                             Oneprocess.
set (handles.uipanel2, 'Visible', 'off'
                                             function
                                             Oneprocess Callback (hObject,
) ;
set(handles.uipanel4,'Visible','off'
                                             eventdata, handles)
                                             % hObject
                                                             handle to Oneprocess
set (handles.RecognizedImage, 'Visible
                                             (see GCBO)
','off');
```

```
% eventdata reserved - to be
                                            set(handles.uipanel4,'Visible','on')
defined in a future version of
MATLAB
                                            set (handles.RecognizedImage, 'Visible
% handles
             structure with handles
                                            ','on');
                                            set (handles.uipanel7,'Visible','on')
and user data (see GUIDATA)
set (handles.BrowseImage, 'Visible', 'o
                                            set (handles.uipanel8, 'Visible', 'on')
set(handles.CleanImage,'Visible','of
                                            set(handles.text1,'Visible','on');
                                            set (handles.ImageDisplay, 'Visible','
set(handles.DeskewImage,'Visible','o
                                            on');
                                            set(handles.uipanel6,'Visible','off'
ff');
set(handles.SegmentImage, 'Visible',
                                            set(handles.text3,'Visible','off');
'off');
                                            set(handles.Deskew,'Visible','off');
set(handles.RecognizeImage,'Visible'
,'off');
                                            set(handles.uipanel10,'Visible','off
set(handles.Oneclick,'Visible','on')
                                            ');
                                            set(handles.text12,'Visible','off');
                                            set (handles.Backside, 'Visible', 'off'
set (handles.BrailleAlphabet, 'Visible
','on');
set (handles.BrailleContractions, 'Vis
                                            set(handles.Showall,'Visible','off')
ible','on');
set(handles.ResetAll,'Visible','on')
                                            set(handles.Back,'Visible','on');
                                            set (handles.Oneprocess, 'Visible', 'of
set(handles.Exit,'Visible','on');
                                            f');
set(handles.uipanel2,'Visible','on')
                                            set(handles.Banner,'Visible','off');
```