SearchPart_09.13

User manual

SearchPart - program to search for electronic componnets on circuit board

For ideas of improvement check
https://github.com/Berni1557/SearchPart
or send an Email at
SearchPart@yahoo.com
and help us to change the world!

Contents

1. Introduction to SearchPart

SearchPart is a program that searches for electronic parts in images of circuit boards. The user takes images of circuit boards with a conventional camera and loads the images in SearchPart. The program searches automatically for components and stores the part values or part names and positions of the parts in a data set. After that, the user can search for a certain part and SearchPart checks if the part exists on the circuit boards. SearchPart shows the user the image of the circuit board and the position of the part that the user can resold and reuse.

1.1 SearchPart can help us to reduce electronic waste!

Electronic equipment and gadgets are the fastest growing waste stream in many countries. For many, electronics are part of modern life – cell phones, laptops, TVs and a growing number of gadgets. Every year we buy new, updated equipment to support our needs -there are upwards of 300 million computers and one billion cell phones produced every year. All of these electronics become obsolete or unwanted, often within 2-3 years of purchase. This global mountain of waste is expected to continue growing 8% per year, indefinitely (BCC Research) [1].

But what happens to the e-waste?

Unfortunately, an incredibly small percentage of e-waste is recycled. Even when we take it to a recycling center it's usually not actually recycled – not in the way most of us think of that term.

A small percentage of e-waste is estimated to be sent to recyclers. In the U.S., as little as 11%-14%. The remainder is most often dumped or burned – either in formal landfills and incinerators, or informally dumped or burned. These inappropriate disposal methods for electronic waste fail to reclaim valuable materials or manage the toxic materials safely. In effect, our soil, water and air are easily contaminated.

One solution could be SearchPart!

1.2 SearchPart can help you to reuse electronic parts in private!

In private, people already unsolder electronic components like resistors, switches or electonic chips to reuse them. The problem is to find a certain part under a large amount of circuit boards without knowing to find it.

One solution could be SearchPart!

1.3 SearchPart can help you to improve your knowledge in image processing!

SearchPart is also for motivating to work in Image Processing. Everybody can test new algorithms and ideas for Pattern recognition. You can do research and improve your knowledge in the field of image processing. Therefore the improvement of a special module of SearchPart could be a student project or theses.

The program is modularized that everybody can improve the program by replacing existing algorithms with a better one or extend the program with the possibility to search for a new components.

One solution could be SearchPart!

To make SearchPart more powerful we need your help!

2. System requirements and installation of SearchPart

SearchPart is running under Windows and Ubuntu 12.04 LTS but it should also work under other linux distributions. We would be very happy to get information about possible other working environments. SearchPart is a MATLAB based program and was developed in MATLAB 7.12.0 (R2011a). Therefore at minimum this version is necessary. Additionally the MATLAB Image Processing Toolbox has to be installed and the open source software tesseract 3.01 for optical character recognition.

2.1 Installation of SearchPart under Ubuntu

SearchPart was developed under Ubuntu 12.04 LTS but should also run with other linux distributions. First you have to install MATLAB. How to install MATLAB is written under http://wiki.ubuntuusers.de/Matlab or additional websites.

Two ways for installing SearchPart. If you want to use SeachPart regularly, the second way should be prefered.

Install SearchPart first way

1. Copy the SearchPart_09.13_linux folder to the position you prefer.
2. Install tesseract:
   SearchPart needs tesseract 3.0.1 for optical character recognition. Tesseract is an open source software for optical character recognition and released under the Apache License 2.0. For more information read http://wiki.ubuntuusers.de/tesseract-ocr. To install tesseract simply open a terminal and install it with the command **sudo apt-get install tesseract-ocr**.
3. Open MATLAB browse to the SearchPart_9.13 folder and type „SearchPart“ in the matlab command line. SearchPart should start immediately.

Install SearchPart second way

1. Copy the SearchPart_09.13_linux folder to the position you prefer.
2. Set matlab path: Open the SearchPart_install.sh file in the SearchPart_09.13_linux folder and replace the matlab path with your matlab path. The path could look like this „/usr/local/MATLAB/R2011a/bin/matlab“. Therefore the example file looks like in Figure 1.

SearchPart_install.sh

```
#!/bin/bash
echo "Installing SearchPart"
var1=`pwd`
var2=$var1/SearchPart_09.13

echo "Creating file SearchPart_start.sh"
echo "  #!/bin/bash " > SearchPart_start.sh
echo "/usr/local/MATLAB/R2011a/bin/matlab -nodesktop -r \"userpath('$var2');SearchPart();pause(10^10);\"" >> SearchPart_start.sh

echo "Copy SearchPart_start1.sh to bin directory"
sudo cp SearchPart_start.sh /usr/bin/SearchPart_start.sh
sudo rm SearchPart_start.sh

echo "Creating startfile"
echo "[Desktop Entry]" > SearchPart.desktop
echo "Name=SearchPart" >> SearchPart.desktop
echo "Type=Application" >> SearchPart.desktop
echo "Comment=SearchPart" >> SearchPart.desktop
echo "Exec=SearchPart_start.sh" >> SearchPart.desktop
echo "Icon=label.png" >> SearchPart.desktop
echo "GenericName=SearchPart" >> SearchPart.desktop
echo "Name[de_DE]=SearchPart" >> SearchPart.desktop
chmod u+x SearchPart.desktop


echo "Installing tesseract"
sudo apt-get install tesseract-ocr
sudo cp SearchPart_09.13/label/label.png /usr/share/icons/label.png

echo "Creating Desktop icon"
cp SearchPart.desktop $HOME/Desktop
echo "Have fun!"
exit 0

```

Figure 1: Change matlab path in SearchPart_install.sh file

3. Set permissions to the SearchPart_install.sh. Open a terminal, browse to the SearchPart_09.13_linux folder and type **chmod +x SearchPart_install.sh**
4. Type **./SearchPart_install.sh** that installs tesseract, sets SearchPart settings and creates a desktop icon automatically. In between you have to enter your password. Tesseract is an open source software for optical character recognition and released under the Apache License 2.0. For more information read http://wiki.ubuntuusers.de/tesseract-ocr.

   That's it! Just click at the SearchPart icon on your desktop and the program starts.

2.2 Installation under Windows

SearchPart was developed under Windows 7 but should also run with other windows distributions. For installation under windows, you have to instal MATLAB. Further information for installing MATLAB under http://www.mathworks.com/help/install/ or additional websites.

Two ways for installing SearchPart.

Install SearchPart first way

1. Copy the SearchPart_09.13_windows folder to the position you prefer.
2. Install tesseract-ocr:
   SearchPart needs Tesseract_OCR 3.02.02 for optical character recognition. Tesseract is an open source software for optical character recognition and released under the Apache

License 2.0. For more information read http://wiki.ubuntuusers.de/tesseract-ocr. To install tesseract simply double click the Tesseract_OCR 3.02.02.exe in the SearchPart_09.13_windows folder.

3. Open MATLAB and browse to the SearchPart_9.13 folder in the SearchPart_9.1_windows folder and enter „SearchPart“ in the matlab command line. Press enter and SearchPart should start immediately.

Install SearchPart second way

1. Copy the SearchPart_09.13_windows folder to the position you prefer.
2. To install SearchPart under Windows, browse to the SearchPart_09.13_windows folder and double click the SearchParts_install symbol. The installation starts and the program creates a Desktop icon. Additional the program installs Tesseract_OCR 3.02, an open source software for character recognition which is necessary for SearchPart and Java run-time (if nor installed already).

Now you just have to double click the SearchPart_start desktop icon and SearchPart starts automatically.

3. SearchPart Environment

How to start SearchPart depends on your installation way. If you did the second way, you just have to double click on the desktop icon.

If you did the second way, start MATLAB and set the current folder to the SearchPart_09.13 folder. Type „SearchPart“ in the Command Window and the program starts.

The information window Fig.: 2 gives you some information about the program version and support. Press ok.

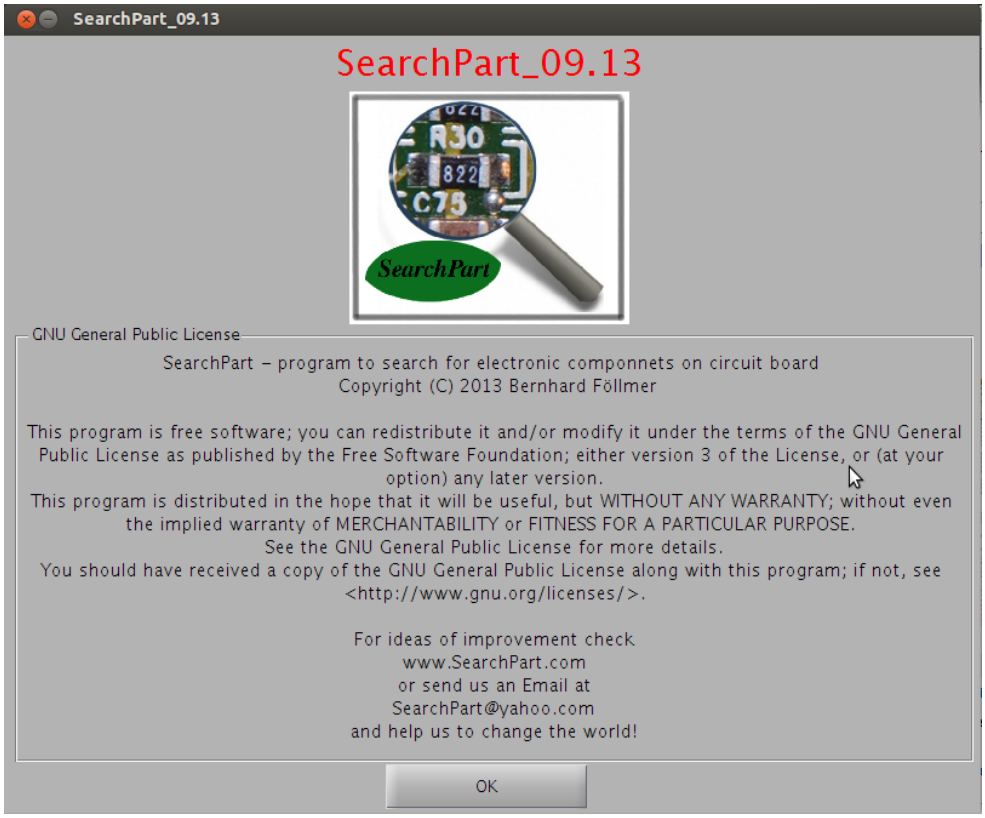


Figure 2: information window

In a next window you are asked if you want to create a new session or if you want to load an existing session (Fig.:3).

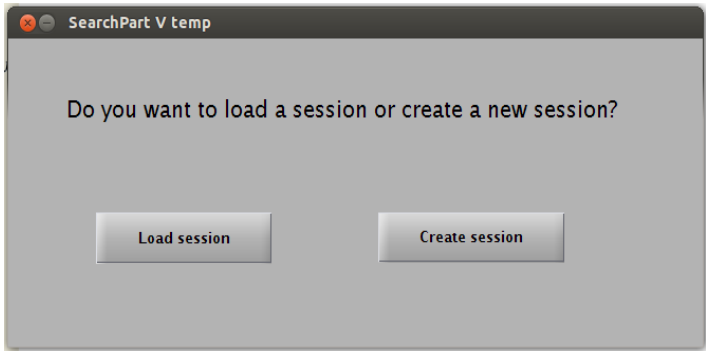


Figure 3: Load or create session

3.1 Load session

If you press „Load session“, you can browse to the session folder and pick the .mat file with the name of a session (Fig.: 4). An example session is in the example folder.

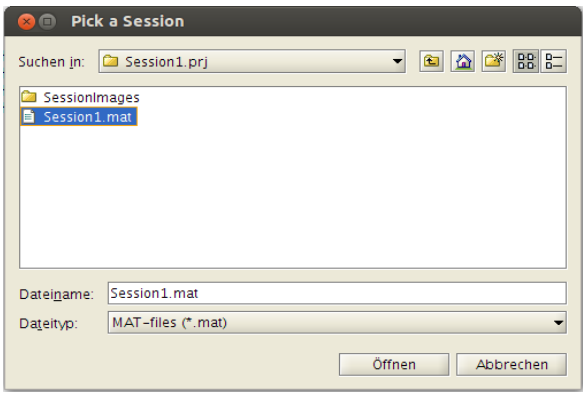


Figure 4: Browse to session

3.2 Create session

If you press „Create session“, a window opens where you can press „Select folder“ and browse to the folder of the images you want to search through (Fig.:5).

Figure 5: Select Images

After selecting images you want to process, you can chose the location of your session. Therefore press „Select project location“ browse to the place you want to save your session and chose the name of your session (Fig.:6). The program creates the session and saves it in a MAT-file(e.g. sample_session.mat). The selected images are copied in the SessionImages folder in your Session folder. If the program can not find the scale symbol, the program skips the image and displays „Image could not be loaded“ in the info box.

Figure 6: Select Project location

After choosing a location, the program opens the window where you can choose images and components you are searching for automatically(Fig.: 7).

3.3 Execute GUI

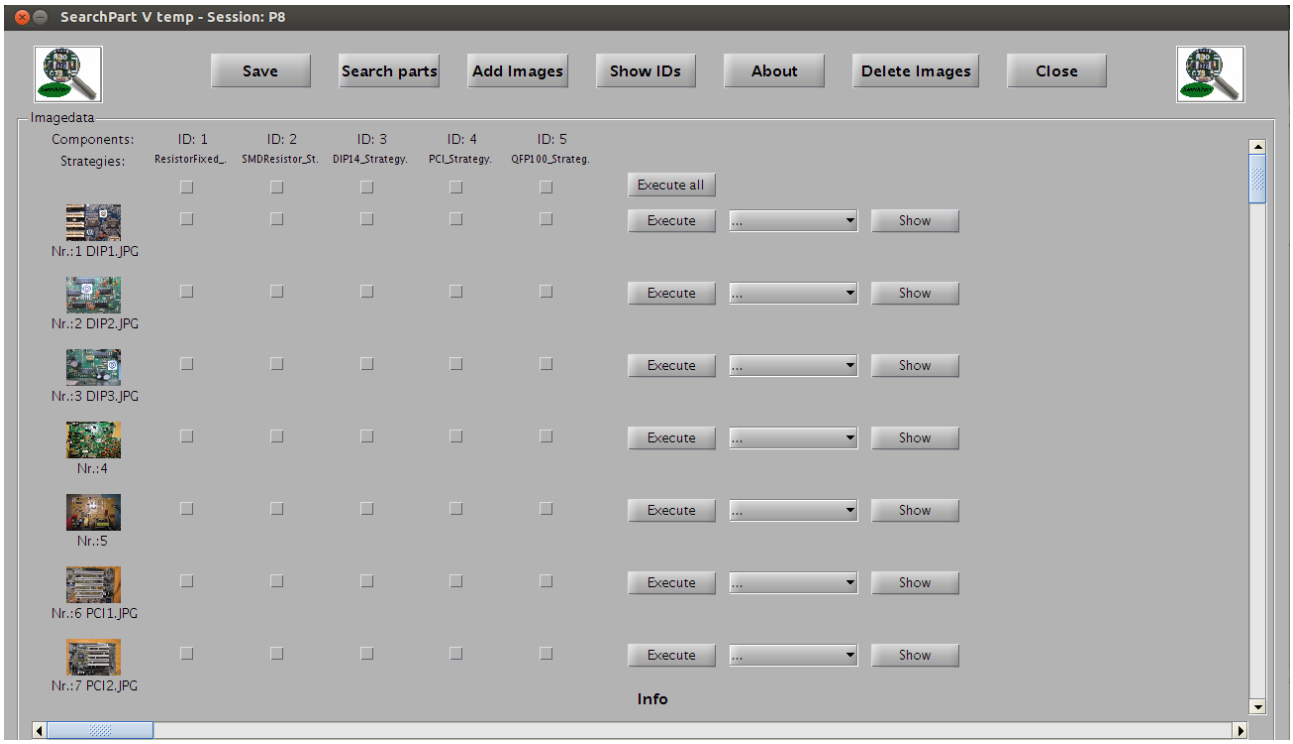


Figure 7: Search for components in images

If you press the „Save“ button on the left side, the program saves the session with all search results.

If you press the „Close“ button on the right side, you can close the program. In the left column you can see all pictures of the session with image name and image number. Every component you are searching for has a special ID. Which component corresponds to which ID is defined in the ID-LIST.pdf in the SearchPart folder. You can see the list if you press the „Show IDs“ button. Select the component your are searching for and the images by pressing the checkboxes. If you press the upper checkbox, you can select a component for all images of the session.

After selcting components and images you can run the search by pressing the „Execute“ button to search for a special image or pressing „execute all“ for searching all selected components in the specified images. When the search is running, the progressbar approximates the remaining time until the search process will be finished. After the search process has been finished, you can use the popup menu to choose a component for an image and press „show“ to see the image. Markers show you the position of the parts.

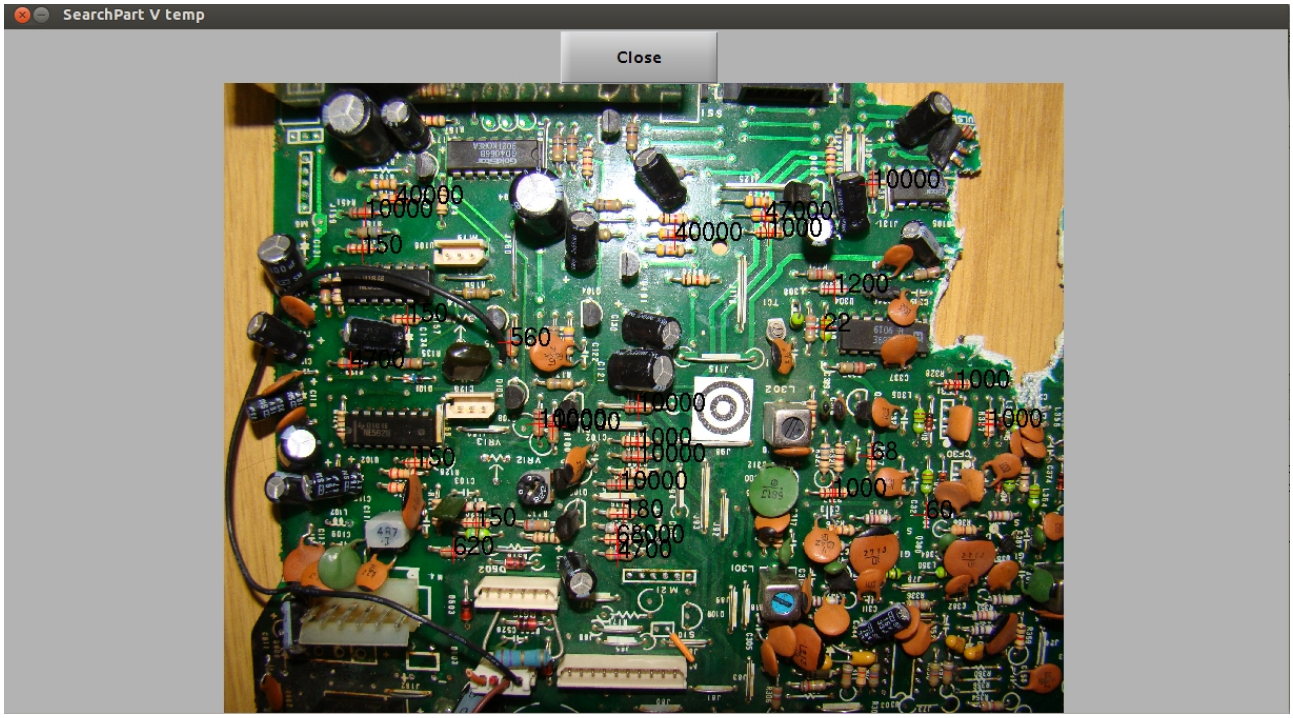


Figure 8: Marked parts

If you are searching for a special part, press the „Search parts“ button and the program opens the Search GUI.

When executing a lot of images at once you have to extend your Java Heap memory size in your matlab settings. To do that open MATLAB click File->Preferences->General->Java Heap Memory and increase your Java Heap memory size.

3.4 Search GUI

If you are in the execute GUI and press „Search Parts“, you come to the Search GUI (Fig.: 9). Here

you can search for certain parts. First pick a component you are searching for from the popup menu, after that you can pick an image you want to search through from the popup menu or select all to search in all images of the session. If you are looking for a certain value or name of a part, you can enter the value (SI base units) or the the name of the component. Press the „Search“ button and the program searches for the part in the session. The program can only find parts which were already found in the execute GUI. After pressing „Search“, you can choose one part from the lower popup menu. The number behind the part name indicates how many times that sort of part has been found in the session.

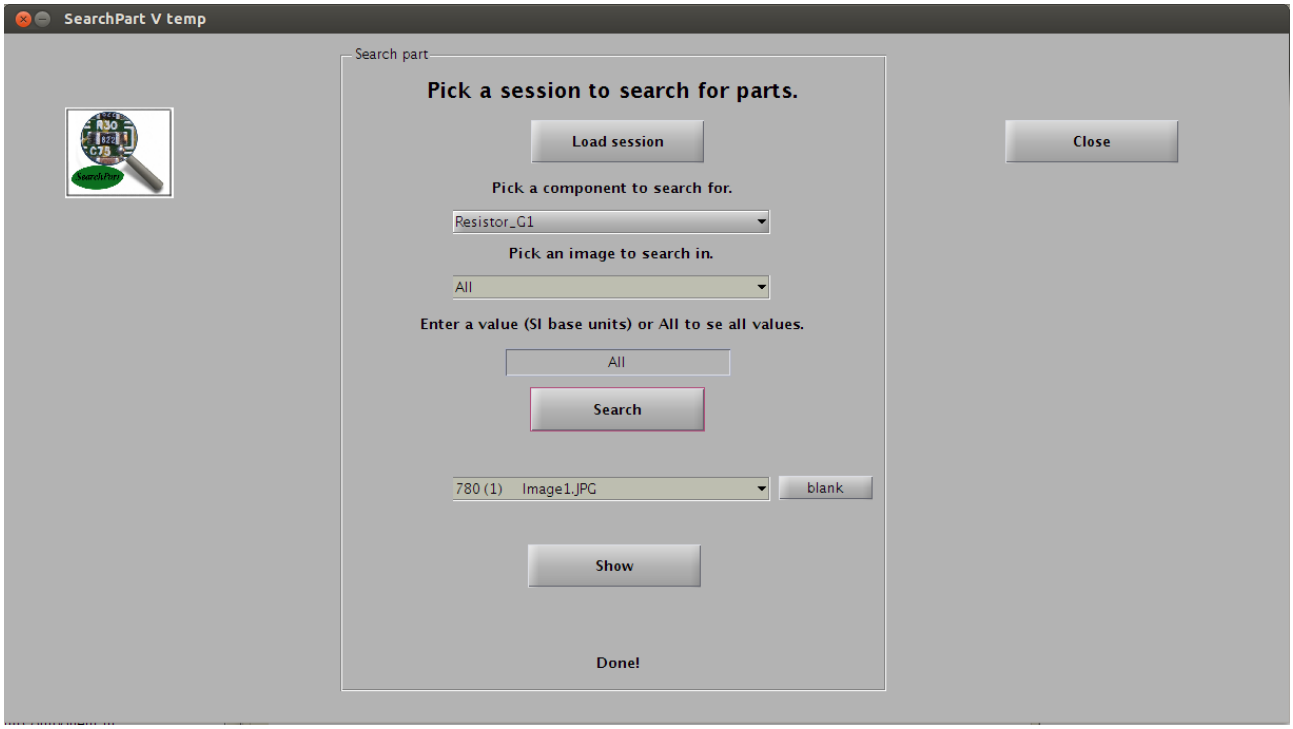


Figure 9: Search GUI

By pressing the „Show“ button SearchPart shows you the image and marks the position of the part. If the part still exists, it is marked by a red cross. If the part has already been resoldered or has been marked at the wrong place, it is marked with a blue cross.

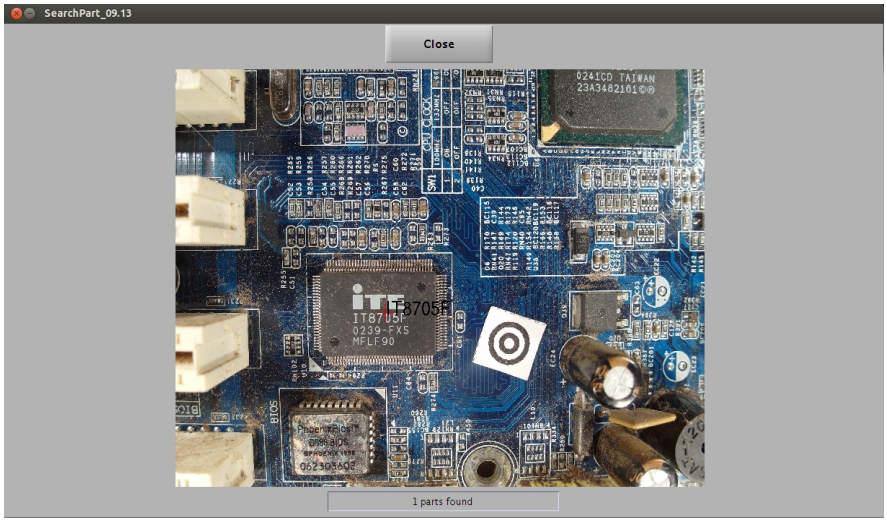


Figure 10: Marked part

On the right side of the popup menu is a „blank“ button. This button gives you the possibility to mark already resoldered or incorrect parts. If you resold one of the parts, you can press the „blank“ button and if you search the part next time, „(b)“ will appear behind the part name. It shows you that the part doesn't exist anymore or is incorrect. If you press „Show“, all blank parts are marked blue.

3.5 Add Images

To add new images to the session, press the „Add Images“ button in the „Execute GUI“, and select the images you want to add to your session.

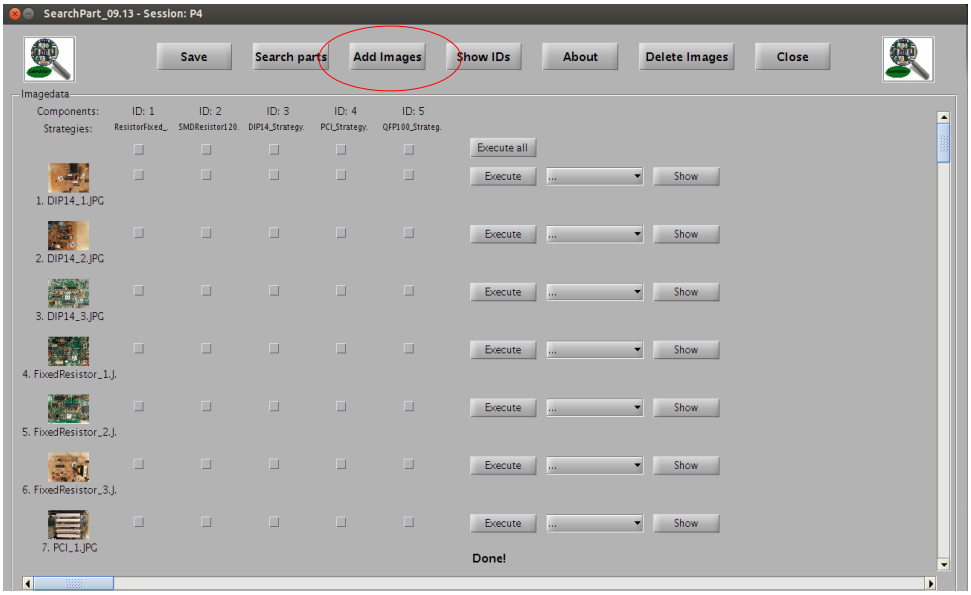


Figure 11: Add images button

3.6 Delete Images

To delete images of a session, press the „Delete Images“ button in the „Execute GUI“ (Figure 12).

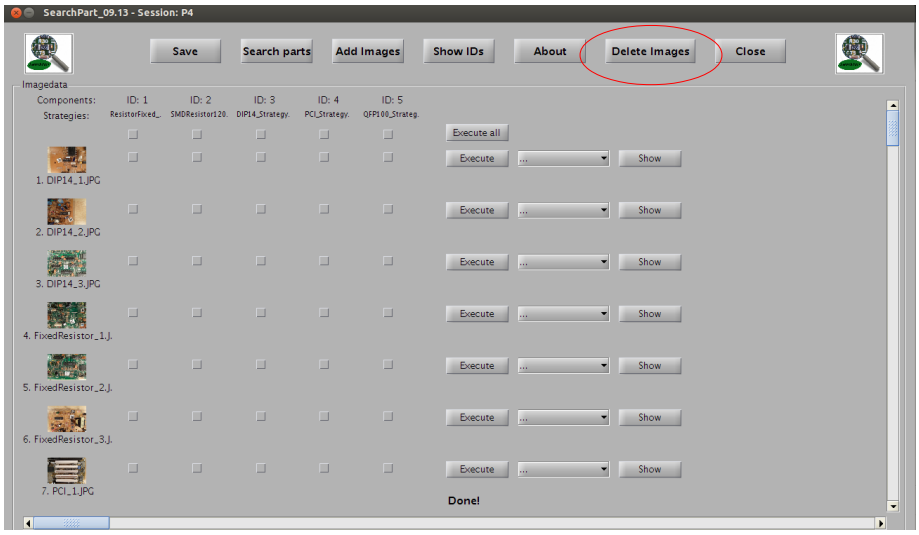


Figure 12: Delete images button

Now you can press the „Delete“ button on the right of the image you want to delete from the session. Press the „Close“ button to come back to the „Execute GUI“.

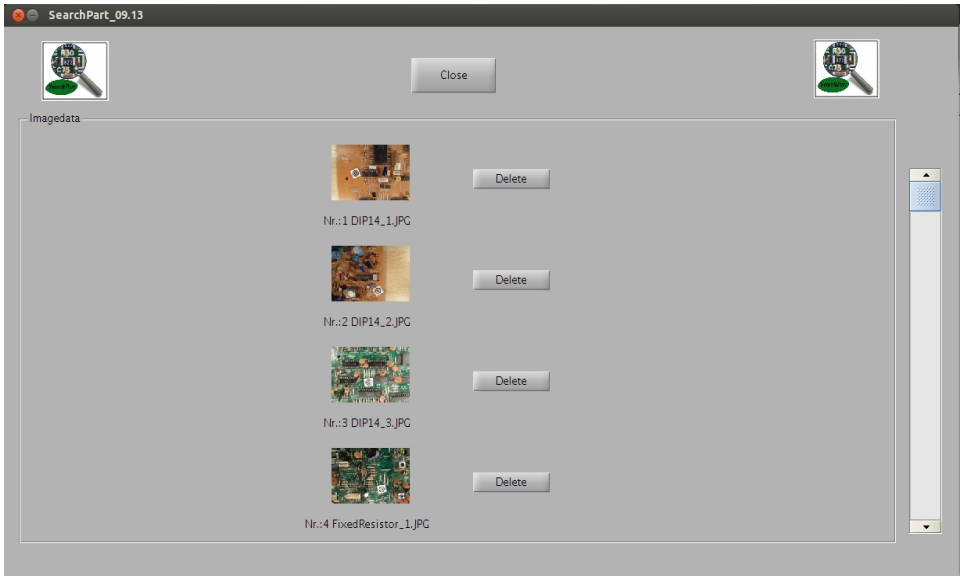


Figure 13: Delete images

4. How to take the images of circuit boards

One of the most important question is how to capture images. The idea of SearchPart is that you do not need a special high developed image acquisition system. You can capture images by a commercial camera. The images in the data folder where made for testing new algorithms with different cameras (Samsung EX2F, Canon PowerShot D10, Medion 10 MP). The construction for image acquisition is shown in Fig.:15 and was done by four simple table lamps prepared with a diffusion film.

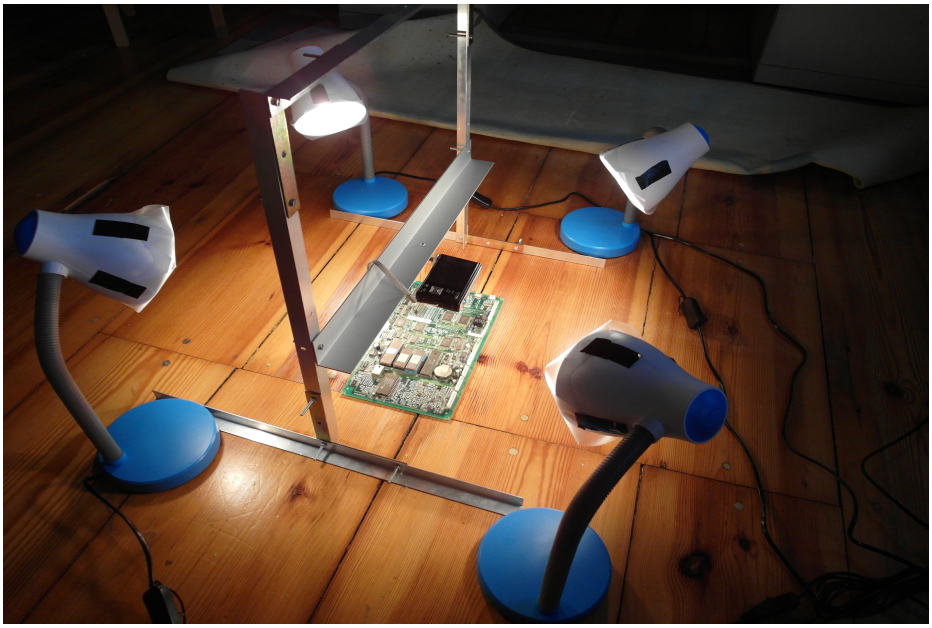


Abbildung 14: Image acquisition

You do not need this or any other equipment for capturing images, you should get reasonable images without.

4.1 Take pictures of circuit boards with scale symbol

To search for parts on circuit boards, you have to take an image of the circuit board. To scale the image, it is necessary to put the scale symbol on the board. The scale symbol is in the scalesymbol.pdf in the SearchPart folder. When you print out the symbol, be careful that the printer prints at the scale of 100%. For correct scaling put the symbol directly on the printed circuit boards as you can seen in Fig.:15 and not on top of a part.

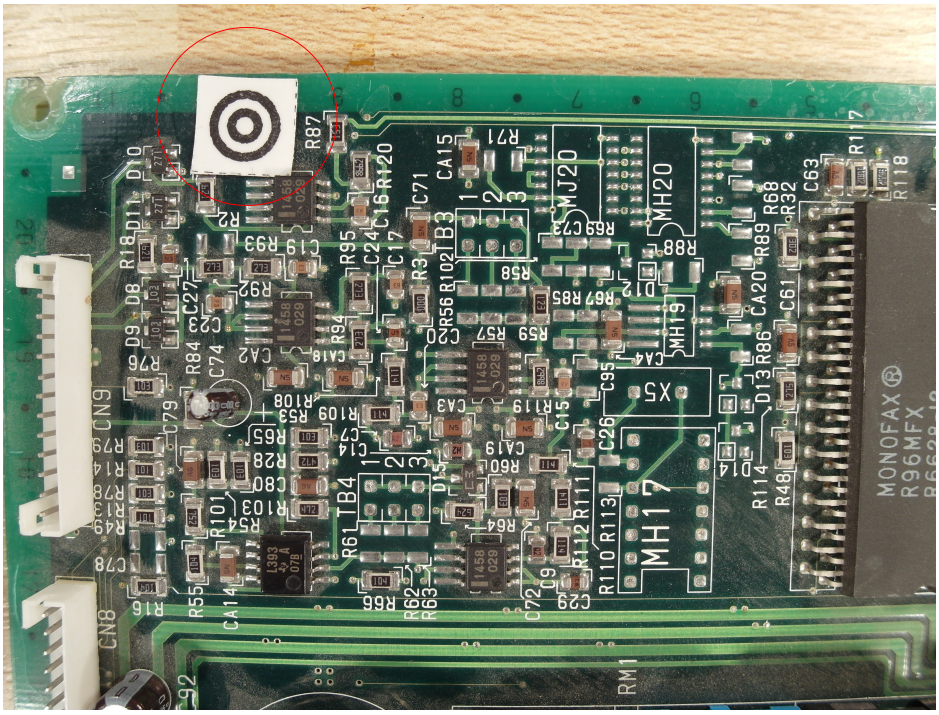


Figure 15: Scale symbol on circuit board

You can take images with the digital camera you prefer, but you should take care that the quality of the image is sufficient for the search task. Use the macro mode of the camera to take optimal pictures. The quality of the images decides about the accuracy rate of SearchPart.

Some important facts you should take care of:

Even illumination

You do not need a special illumination to take images for SearchPart, you can use every lamp or light source you want. But a even illumination is an important condition to detect parts in all areas of the image. Therefore the position of the light is important. Daylight is very diffuse and supports a good color rendering. Figure 16 shows an image with bad, uneven illumination.

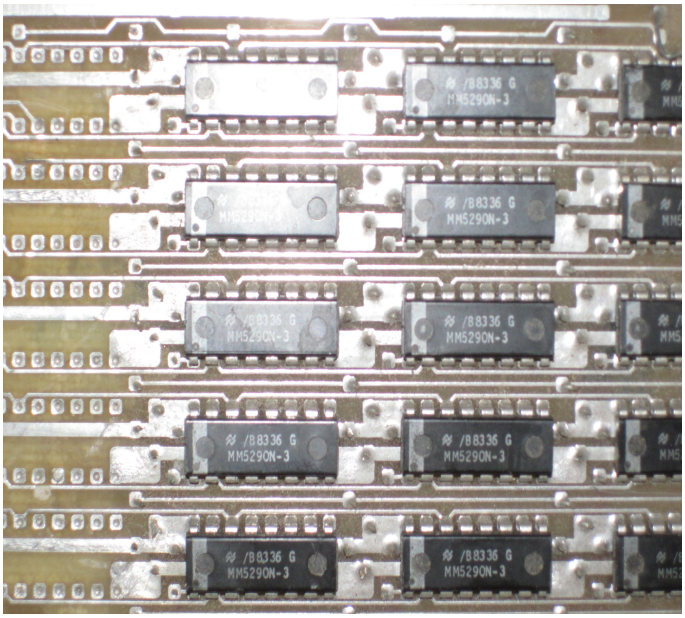


Figure 16: Uneven Illumination

Resolution of the Images

A high resolution is important for the detection, but most of the conventional cameras have high resolution and are sufficient for taking images. A higher resolution can not always increase the accuracy rate. Figure 16 shows an image with low resolution.

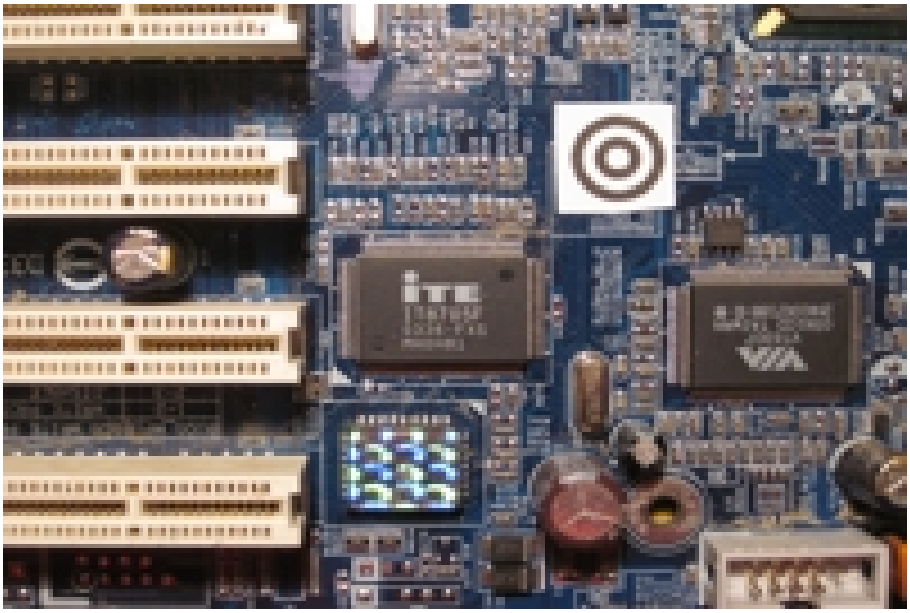


Figure 17: Image with to low resolution

Blurry Images

Try to avoid blurry images because of moving camera or too little illumination. A bad example for a blurry image is shown in 17.

Figure 18: Too blurred image

5. Components in SearchPart_09.13

In SearchPart_09.13 there are 5 different components to search for. These are:

- Axial-lead fixed resistors
- DIP 14 IC-package
- SMD-Resistor, 1206
- PCI Local Bus
- QFP100 package
- SMD resistor 0805

A detailed description is in the appendix and in the ID-List folder.

New components can be inserted in the initComponent function in the SearchPart_09.13 folder. For more details read chapter 6. Link new strategies.

6. Link new strategies

SearchPart is an OpenSource project which is easy to extend. The program is modularized so that everybody can simply remove, add or change searching strategies (searching algorithms).

6.1 Adding new search algorithms

You can add your new search algorithm in 4 steps.

Step 1

To add a new algorithm just make a folder named after your component and place all your functions and data which are necessary for your code. Write a matlab function which gets as inputs the image, scalefactor and as optional inputs a handle of the progress bar object and a structure named searchPartLibrary of possibly useful functions for your code.

A dummy of that function is shown on next page. You can take this code and change the name „dummyAlgorithm“ to the name of your algorithm. You can choose the name by yourself. The same code exists already in the dummy folder under SearchPart_09.13->strategies->dummy.

```
%dummy of your algorith
function [position,name]=dummyAlgorithm(image,scaleFactor,bar,searchPartLibrary)

%Inputs:
%
% image -               image of the circuid board
%
% scaleFactor -         is the scale factor which gives you information about
%                       the size of the image (100 pixel~10mm)
%
% bar -                 (optional) is an processbar object. Just place
bar.update(); between your code to update the progress bar
%
% searchPartLibrary -   (optional) is a structure of maybe usefull
%                       algorithms for your code
%
%Outputs:
%
% position -            is an nx2 array of the positions of the parts where n is
the numbers of parts
%
% name -                is a cell array of n strings with the name of the
%                       part or the value of the part (SI base units)

%%%%%%%%%%%%%%%%%%%%%%%% paste your algorithm here! %%%%%%%%%%%%%%%%%%%%%%%%
```

```
[Ir,~]=SearchPartLibrary.rotCorrection(I);                          % rotation
correction is not necessary but probably usefull
% do some stuff
bar.update();                                                       % update
progress bar
pause(5);                                                           % pause command
is just to see what happends
position=[232, 914;830, 1710;1060, 320];                            % 3x2 array of
part positions
name={'dummy_NE592N','dummy_SN74HC238N','dummy_74HC4086N'};         % 1x3 array of
part names

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end
```

Step 2

Now create a Strategy object. To do so, you can open the dummyStrategy and replace the „dummyStrategy“ by the name of your strategy and replace the „dummyAlgorithm“ by the name of your algorithm in Step 1.

```
classdef dummy_Strategy < Strategy
% dummyStrategy
    methods
        %here paste your strategy name
        function obj = dummy_Strategy(varargin)
            obj = obj@Strategy(varargin{:});
        end
        function [position,
value]=executeAlgo(obj,image,scaleFactor,bar,SearchPartLibrary)
            [pathstr, ~, ~] = fileparts(mfilename('fullpath'));
            addpath(genpath(pathstr));
            %here paste your function

[position,value]=dummyAlgorithm(image,scaleFactor,bar,SearchPartLibrary);
        end
    end
end
```

Step 3

In step 3 you have to add your component to the component initialization list. To do so open the „initComponent“ function in the SearchPart folder and add your component. The file could look like this:

```
% Initialize component list used in the session
% Output: componentListTemp is a list of all components with the strategy used
```

```
in the session
function [componentListTemp]=initComponent()

componentListTemp=[];

% Add you strategy
% componentListTemp{num}=Component('ComponentID',<ID>,'Strategy',<name of
strategy
object(<ID>,<duration>)>,'Componentname',<Componentname>,'HasDatasheet',<true/fa
lse>);
componentListTemp{end+1}=Component('ComponentID',1,'Strategy',ResistorFixed_Stra
tegy('duratione',60),'Componentname','ResistorFixed','HasDatasheet',false);
componentListTemp{end+1}=Component('ComponentID',2,'Strategy',SMDResistor1206_St
rategy('duratione',70),'Componentname','ResistorSMD','HasDatasheet',false);
componentListTemp{end+1}=Component('ComponentID',3,'Strategy',DIP14_Strategy('du
ratione',80),'Componentname','DIP-14','HasDatasheet',true);
componentListTemp{end+1}=Component('ComponentID',4,'Strategy',PCI_Strategy('dura
tione',15),'Componentname','PCI','HasDatasheet',false);
componentListTemp{end+1}=Component('ComponentID',5,'Strategy',QFP100_Strategy('d
uratione',90),'Componentname','QFP100','HasDatasheet',true);

% dummy strategy
componentListTemp{end+1}=Component('ComponentID',6,'Strategy',dummy_Strategy('du
ratione',5),'Componentname','DummyComponent','HasDatasheet',false);
```

componentListTemp(num) -	num is the next entry in the componentList.
ComponentID -	is the ID of the component the algorithm is searching for. Check the ID-List for more information or use your own ID for new components
Strategy -	name of the strategy-function with the duration parameter which is the approximated duration for searching through one image in seconds
Componentname -	name of the component searching for (can be selected by yourself)
HasDatasheet -	set true if a datasheet exists on http://www.datasheetarchive.com/ for this component; otherwise it's false. If it's set to true, SearchPart tries to download the datasheet in the 'Execute GUI' by the name. A list of all parts on datasheetarchive.com is in the datasheetarchive.txt file in the SearchpartLibrary folder and in the SearchPart library.

That's it! Now you can start SearchPart and your ID and strategy name appears in the „Execute GUI“ (Figure: 19).

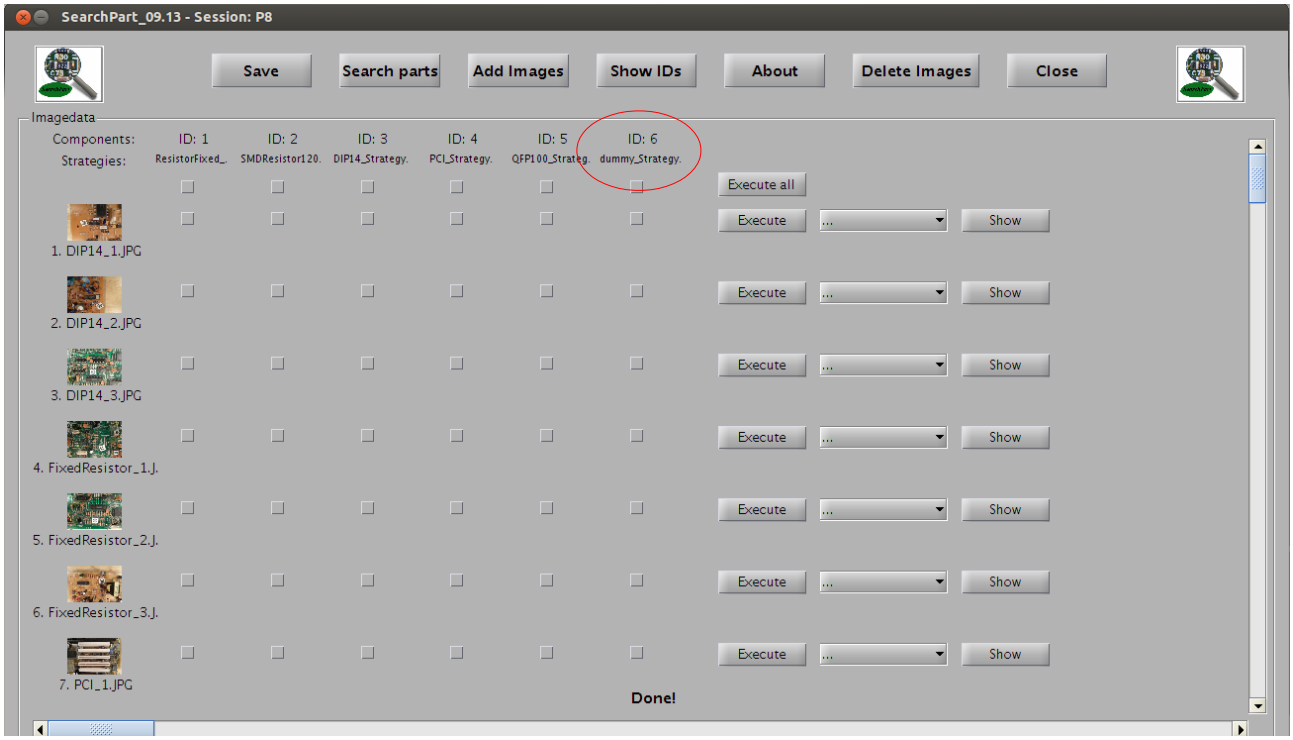


Figure 19: Execute GUI after adding dummy Strategy

6.2 SearchPartLibrary

To simplify programming of new strategies, SearchPart has a library **SearchPartLibrary** with hopefully useful functions that can be used in your code. In this version the library is still small but will be extended in further versions. You should be careful with the library because the functions and datasets are not necessarily better or more useful than others. You are the developer and decide if the functions are useful and sufficient for your searching tasks.

If you create your strategy (refer to 6.1), you get the SearchPartLibrary structure and can access the functions and datasets by the point operator.

```
function [position,name]=dummyAlgorithm(image,scaleFactor,bar,searchPartLibrary)
```

SearchPartLibrary.rotCorrection()

The command **[I1, phi]=SearchPartLibrary.rotCorrection(I);** is used to correct image rotation of the image I and outputs the rotated image I1 and the rotated angle. The function uses an algorithm in the frequency domain and fills the resulting border caused by rotation correction with black pixel. An example is shown in Fig.: 22.

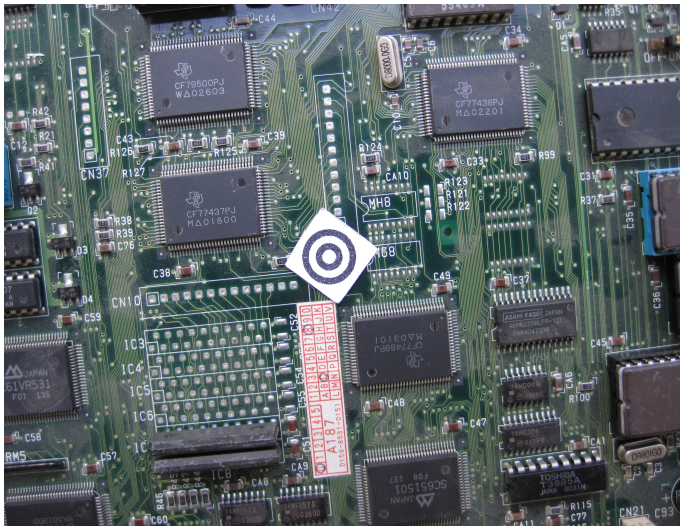


Figure 21: Original rotated image I

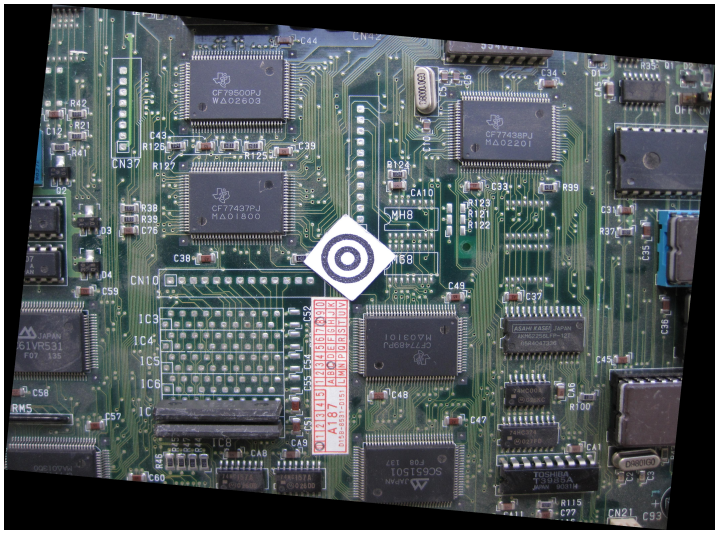


Figure 20: Rotation corrected image I1 (phi=5.5 deg)

An example of how to use the function SearchPartLibrary.rotCorrection() is shown in the sampleSearchPartLibrary folder which is in the SearchPartLibrary folder.

SearchPartLibrary.OCRtesseractLetter()

The command **[letter]=SearchPartLibrary.OCRtesseractLetter(I);** takes the image of a character and does character recognition. The input image has to be a binary or grayscaled image with a white character on black backround. The output is a character recognized by the open source character recognition tool “tesseract 3.0.2”. An example is shown in Fig.:22, 23 and 24.

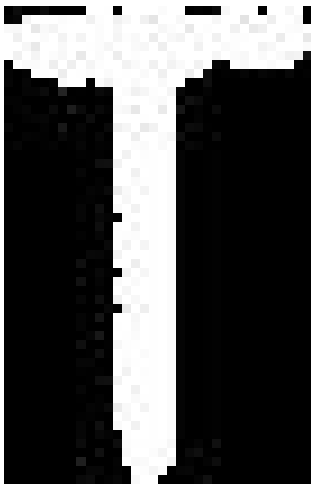


Figure 24: letter: T

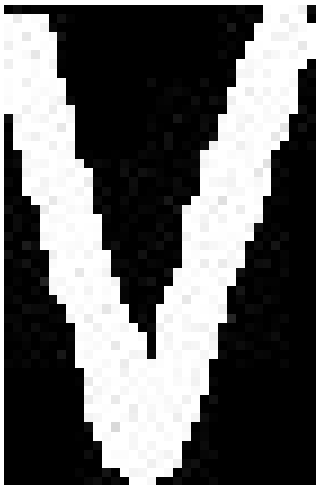


Figure 22: letter: V

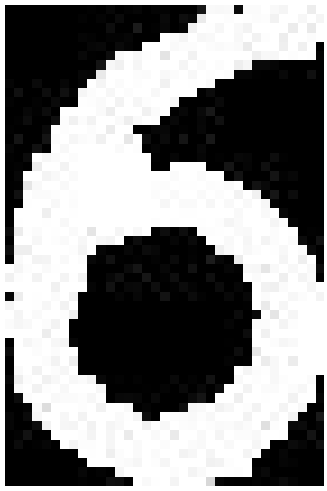


Figure 23: letter: 6

An example of how to use the function SearchPartLibrary.OCRtesseractLetter() is shown in the

sampleSearchPartLibrary folder which is in the example folder.

SearchPartLibrary.loadDatasheetNamesAlphabet()

The function **SearchPartLibrary.loadDatasheetNamesAlphabet()** loads and puts out the dataset DatasheetNamesAlphabet. DatasheetNamesAlphabet is a cell array of part names of electronic components. The list is created from the datasheet database http://www.datasheetarchive.com/ and a list is placed in the datasheetarchive.txt file in the SearchPartLibrary folder. The part names are ordered alphabetically where each cell array entry consists of the names starting with the same character. The array starts with the initial character 0,1, …, 9, A,B,...,Z therefore the cell array DatasheetNamesAlphabet consists of 36 entries.

SearchPartLibrary.loadDatasheetNamesSize()

The function **SearchPartLibrary.loadDatasheetNamesSize()** loads and outputs dataset DatasheetNamesSize. DatasheetNamesSize is a cell array of the same part names in DatasheetNamesAlphabet. The difference is that the names are ordered by the length. That means that the first entry consists of all part names with the length of one character, the second entry contains part names with a length of 2 characters and so on. The array comprises part names with a maximum length of 19 characters.

7. Examples

To see how SearchPart works, there is an example Session with some images in the example folder. Additional data are on the SearchPart website https://github.com/Berni1557/SearchPart . The results show you that SearchPart doesn't work perfectly yet and it's your chance to change that. Possible recognition errors are shown in the following pictures. They should motivate you to improve SearchPart.

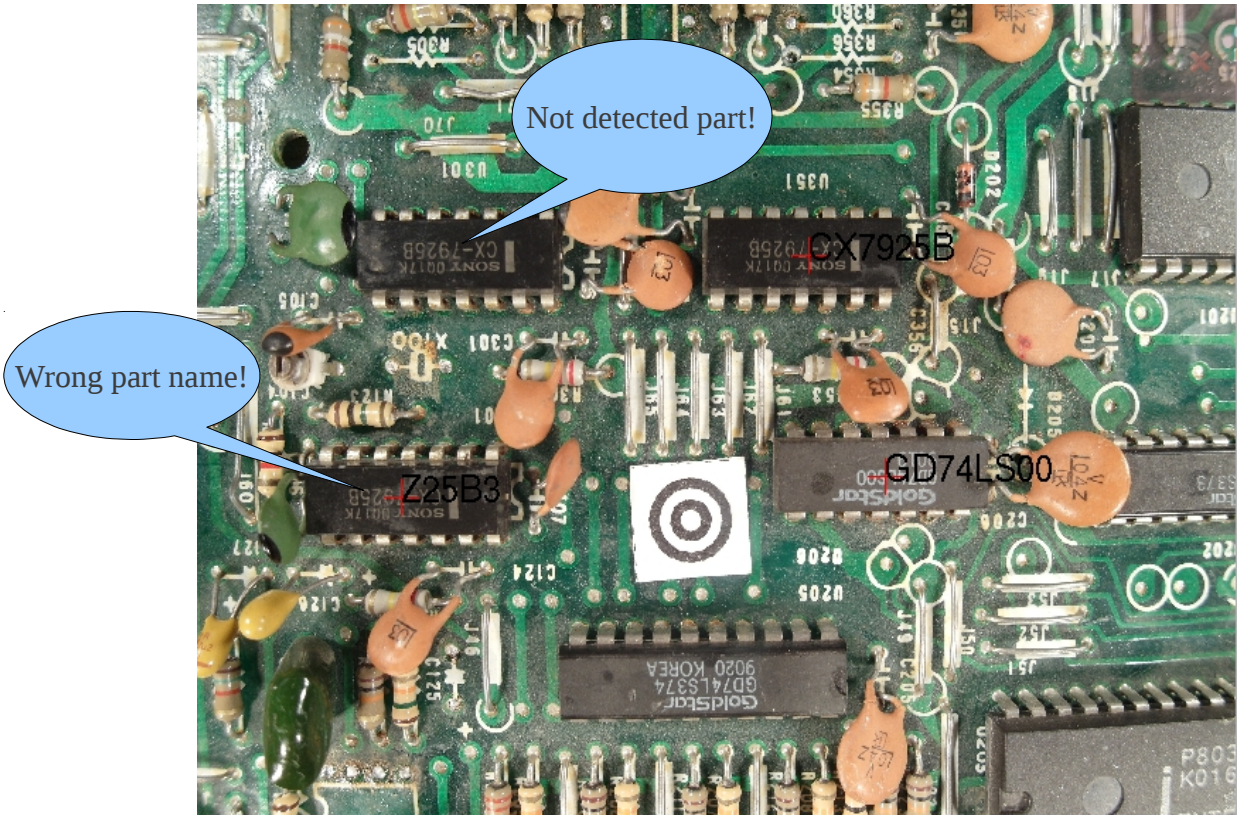


Figure 25: Recognition errors for DIP14

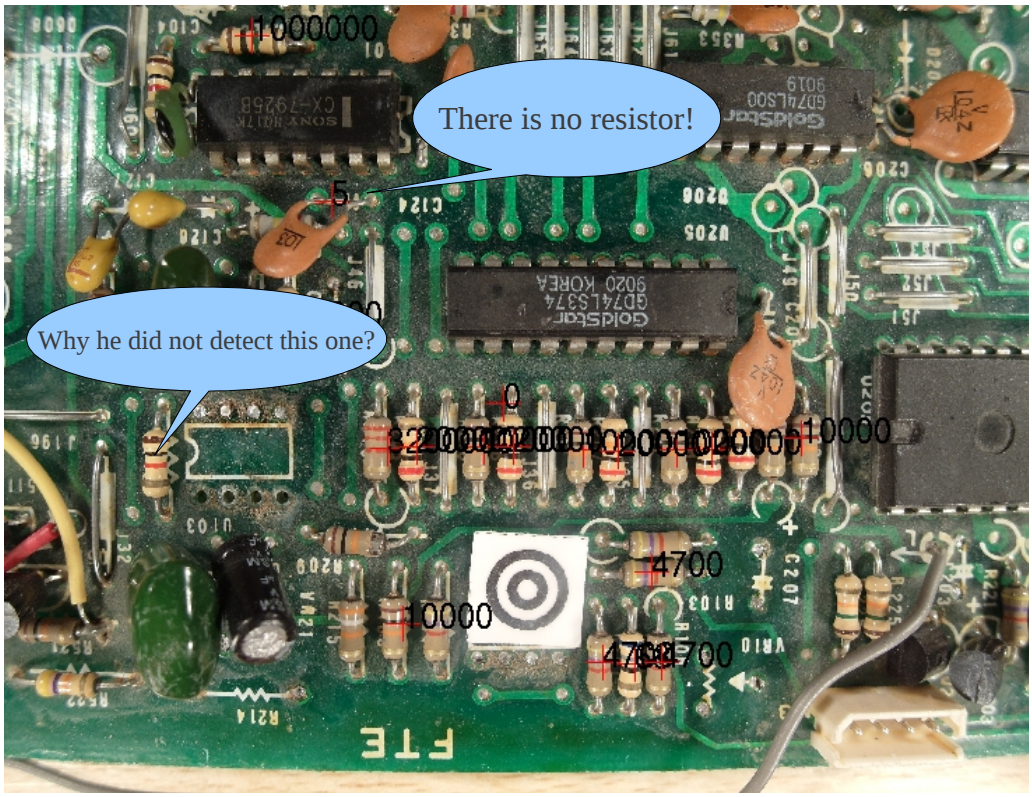


Figure 26: Recognition errors for axial-lead fixed resistors

When you think you know how to improve SearchPart, then try and help us to create a better SeachPart. Try new ideas and algorithms and send them to us. We will include your algorithm in the next version.

8. Dataset for testing and training

For testing and training new algorithms there is a large number of images from different cameras on the SearchPart webpage https://github.com/Berni1557/SearchPart in the datasets folder. This datasets can be used for searching new components. If you create your own image dataset please publish on the SearchPart website.

9. Glossary

Session	A session is a set of images where some components were partially searched. You can save and load sessions or add new images to continue working.
Component	A component is a kind of part for example SMD Resitor 1206, parts in an DIP 14 package, PCI Local Bus, parts in a QFP100 package and so on. All components have an ID to separate them more easily from each other.
Part	A part is one device like a resistor, an IC in a DIP14 package, an IC in a QFP100 package.
ID	Every Component has an ID to seperate them easyer from each other. The ID and the corresponding component is shown in the ID-List.pdf.
Session folder	The session folder is the folder where all images of a session are stored in the „SessionImages“ folder and the data of the Session are the „Sessionname.mat“ file.
strategies folder	The strategies folder contains all strategies. New strategies have to be copied in that folder.
SearchPartLibrary folder	The SearchPartLibrary contains all information about the SearchPart library. The folder is in the SearchPart folder.
SearchPart folder	The SearchPart folder contains all gui functions and objects for Searchpart.
Execute GUI	The „execute GUI“ is the GUI where all images and components are shown and can be executed to search for. The execution of the search takes some time, therefore it's possible to save the session with the searched data. An example for the „execution GUI“ is shown in Fig.: 15.
Search GUI	The „search GUI“ is the GUI where it's possible to search for a certain part. Select a component and type the name or value of the certain part. Searchpart can just find components which have been executed on the images in the „execute GUI“. An example for the „execution GUI“ is showen in Fig.: 11.
blank part	If a part was removed from a board, it has to be marked in the software. Therefore you can mark the part as blank in the Search GUI by clicking the blank button after searching for the part. If you search for a certain part, and behind the part is written „(b)“ the part doesn't exist anymore on the circuit board. In the images, the parts are marked with a blue cross instead of a red cross for existing parts.
strategy	To search for a component, you can change the strategy for searching. The strategy is the algorithm used for the search of the component. The used strategyname is shown in the Execute GUI under the ID number (Fig.: 15). The used strategy for each component is set in the „initComponent“ file in the SearchPart_temp folder.
scaleFactor	Is the scaling factor of the image. The scaling factor was calculated with the

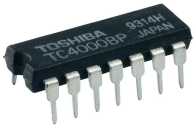
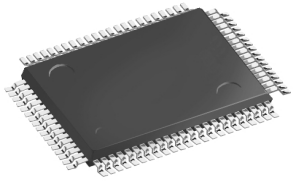
	scale symbol by loading images in the session. The scale unit is [pixel/mm].

10. References

[1] e-Stewards Initiative, http://e-stewards.org/the-e-waste-crisis/, 10.09.2013

11. Appendix

ID-List

ID	Description	Imageexample
1	**Axial-lead fixed resistor** Properties: linear resistance, through-hole resistor, color banded code, four and five bands	
2	**SMD resistor 1206** Properties: 1206 package, fixed resistor, linear resistance, resistance value written with 3 numbers	
3	**DIP 14 IC-packages** Properties: Dual in-line package (DIP or DIL), rectangular housing and two parallel rows of electrical connecting pins, through-hole soldering, inter-lead spacing (lead pitch) of 2.54 mm	
4	**PCI** Properties: local computer bus for attaching hardware devices in a computer	
5	**QFP100** Properties: IC chip in QFP100 package	

6	**SMD resistor 0805** Properties: 0805 package, fixed resistor, linear resistance, resistance value written with 3 numbers