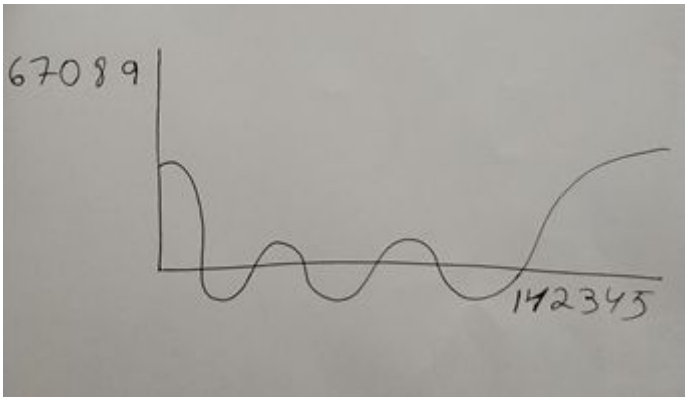
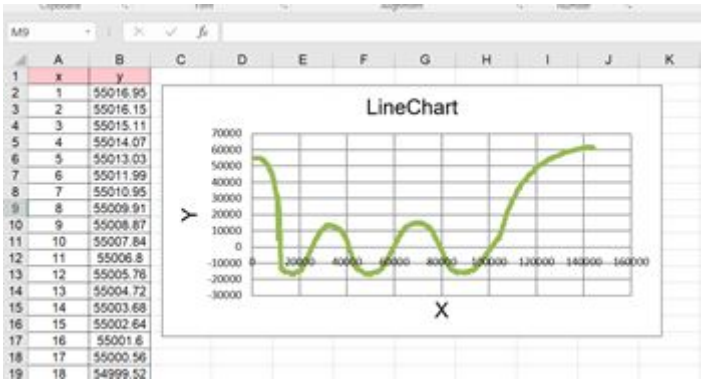
GTT

Graph To Table
User Manual





Mark Rachman and Nir Kravetzky

Table of Contents

1. What is it GTT ?

GTT is acronym for Graph To Table. The purpose of the program is to convert handwritten graph from an image to a set of coordinates. The program is written in Python. The input is a simple handwritten graph with numbers alongside the axes - **those numbers represents the maximum point in the graph**.
The output is an excel/csv file with the (X, Y) values of the graph.

2. Requirements

In order to run the GTT you must have **python 3.6+** installed.
In addition, you need to install the following libraries:

- Numpy - library that helps in calculating complex math problems.
- OpenCV - library that helps us to detect lines in the images and dots.
- PIL - library for image reading and processing.
- KERAS - library for deep learning in order to recognize digits.
- TensorFlow - library for deep learning in order to recognize digits.
- Openpyxl - library for the excel output.
- Matplotlib - library for creating graphs.
- (Not required) h5py - needed in order to disable certain log-warnings that appear in previous versions of TensorFlow.

For easy installation, we created a bat file that will install all your required libraries - install_libraries.bat

Please make sure that you have the following versions:

- Python 3.6+
  - In order to check your python version run the following command:
    - python --version
- OpenCV 3.4.1+
  - In order to check your openCv version run the following commands:
    - Python
    - import cv2
    - cv2.__version__
- You will be able to run GTT with lower versions, but your image detection quality will not be good enough.

3. Installation

In order to install GTT simply extract the source code and install all the required libraries. For smoother start, we created a simple batch file that will install all the required libraries (except for python): 'install_libraries.bat'.
In addition, please make sure that GTT.py has privileges to read and write inside his folder (GTT creates log files and folders).
Please pay attention - all the created log files will not be deleted by the software in order to allow the user to view and analyze those files.

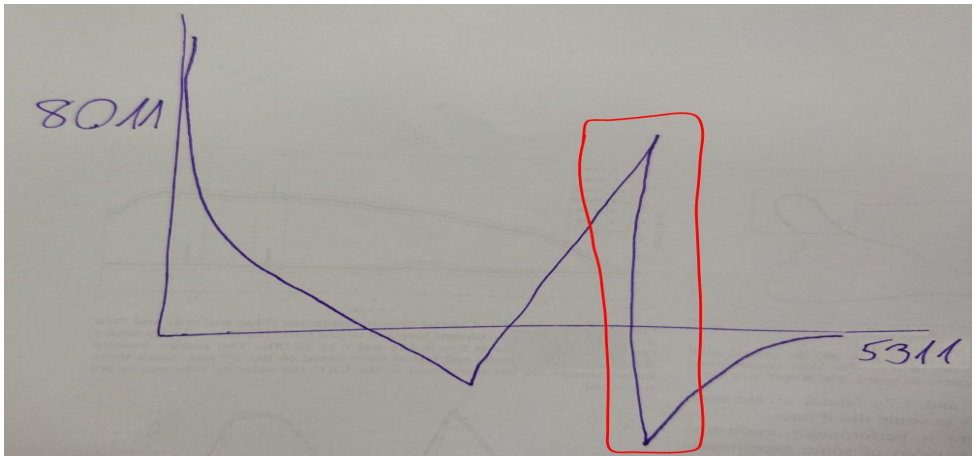
4. Directory Structure

The main directory contains:

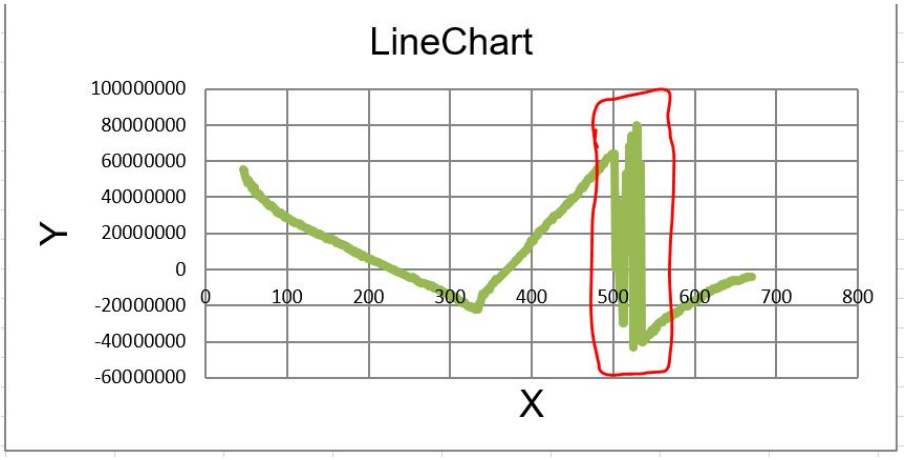
- GTT.py - User entrance point to the program. The main Python file.
- Install_libraries.bat - Batch file for easy installation of required libraries.

Other directories are:

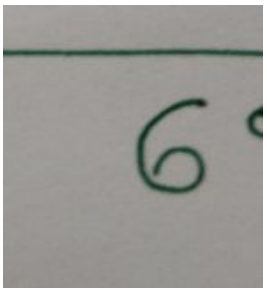
- “/docs” - contains the final documents (including this one).
- “/graphs” - contains graph examples you can run the program on.
    - We collected some ‘bad’ images in order to demonstrate common problems you might encounter:
        - ‘bad_incline.jpg’ / ‘bad_inclince2.jpg’ - those images contains at least two points with the same x values but different y values:



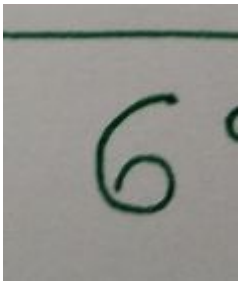
The final output will try to mitigate those differences and the result will look something like:



- ‘green_no_grpah.jpg’ - be aware of the image quality. For example we attached two images (green_no_graph.jpg and green_graph.jpg) with the same X and Y axis except that the green_no_graph quality is insufficient:



green_no_graph.jpg



green_graph.jpg

- “/modules” - contains all the logic of the program.
- “/website” - contains the files for the presentation website.

5. How To Use

GTT is very simple to use - in order to run it open a command line at the extracted folder location (where you have downloaded all GTT files), and type the following command:

```
GTT.py --source=img_path [--log_level=logLevel] [--scale=scaleLevel] [--x_y_max X_max Y_MAX]
```

[] - is an optional parameter.
In addition to this paper, you may ask for help with the following command:
GTT.py --help

Argument list:

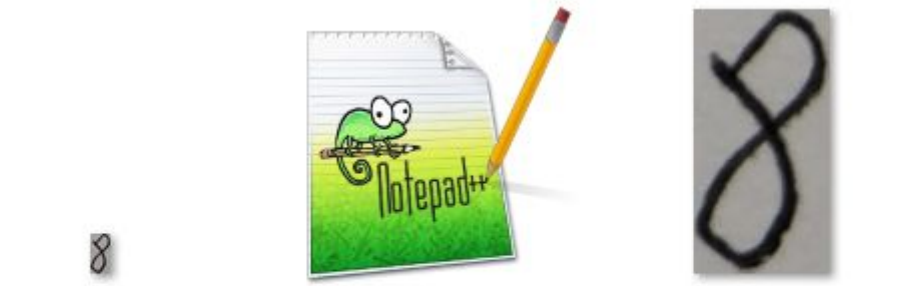
- source - this is the image path. You can use any format you want (png, jpg, jpeg and so on).
- log_level - we have 5 log levels in GTT. The default log level is 0. Please visit chapter 5. Log level information to understand it better.
- scale - scale is the frequency rate of the X axis. The default value is 1. The scale value must be a natural number bigger than 1.
  For example, let's look at all the natural numbers from 0 to 10.
  - Scale rate 1 will produce an output that has all the numbers of 0 to 10.
  - Scale rate 2, will produce all the even numbers from 0 to 10 (0, 2, 4, 6, 8, 10)
  - Scale rate 10, will produce only 0 and 10.
- x_y_max - sometimes, the digit recognizer might recognize different numbers from the user's intention. To help with this case we added the following configuration. With this configuration you can enter manually your numbers. The given point will be the maximum point in the graph line - the format shall be (x,y). If this parameter is provided, the numbers detection library will not be called.

6. Log Level Information

GTT might detect or recognize different digits than what we intended (after all don't forget - all the images are hand-written).
So, we support a log level mechanism that can help the user understand the process behind the scenes.
Log level can be any number between 0 to 4. The products of each log level contains the previous log level (for example, all the features of log level 3 are also available when using log level 4).

- Log level 0 - This is the default log level and the most basic one. It only prints the progress of the program without any additional information.
    - GTT progress may take several minutes. We print the progress in the following format:
        - "Starting data extraction from the image" – this process will convert the given image into a matrix and run the visual recognition algorithm.
        - "Editing final dot list" – this phase responsible to merge all the extracted data from the image, into a single data structure.
        - "Creating the excel file" - this process will generate the excel file and the graph inside it.
        - "Done" – GTT finished and the product is ready at the '/out' directory.
- Log level 1- This log level creates a 'log.txt' file that contains all the recognized digits in the image and their classification. The log file will be generated at folder '/log/log<X>/', where X is an internal identifier of the current running GTT process (each time you run GTT, it selects unique id that is not in use).
- Log level 2- This log level saves in addition to the text file, all the recognized images of the digits inside the image. It saves two types of images:
    - 'Non_filtered<X>.png' - this digit is suspected to be a valid digit.
    - 'filtered<X>.png' - this image is actually a digit that was recognized by the neural network.

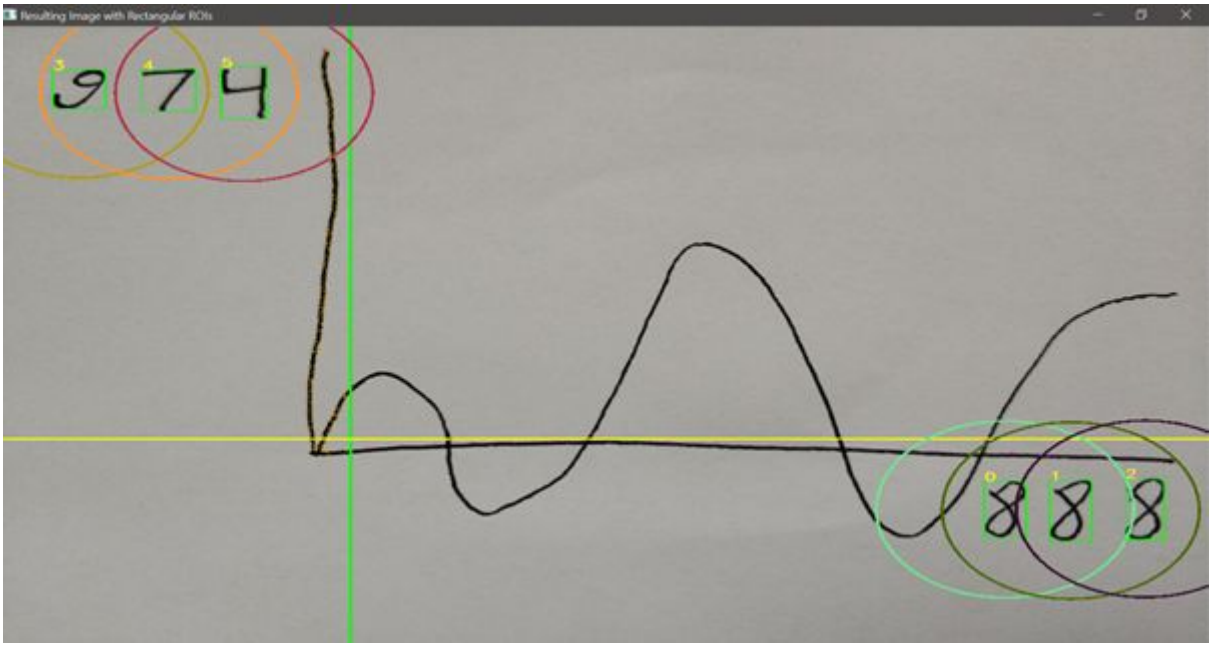
  In both cases, X number is an internal identifier number. In addition, filtered<X> number with index X fits to non_filtered<X> image file (Therefore digits without filtered image, were classified as non-digit).
  In order to see the classifier output regarding a specific digit, look at the 'log.txt' file, it contains all the conclusions regarding the digits images.
  For example, if we have the following log folder that contains the following files:



We can see, that GTT thought that “non_filtered0.png” may be a valid digit. Actually, it classified this number as a valid digit (we have the “filtered0.png” file). In order to see the classifier output open the “log.txt” file, and search for “non_filtered0.png”:

Now, you can see the image in binary format and the classification results. In this case, the classifier thinks that this digit is 8 in 100% (Wow!!).

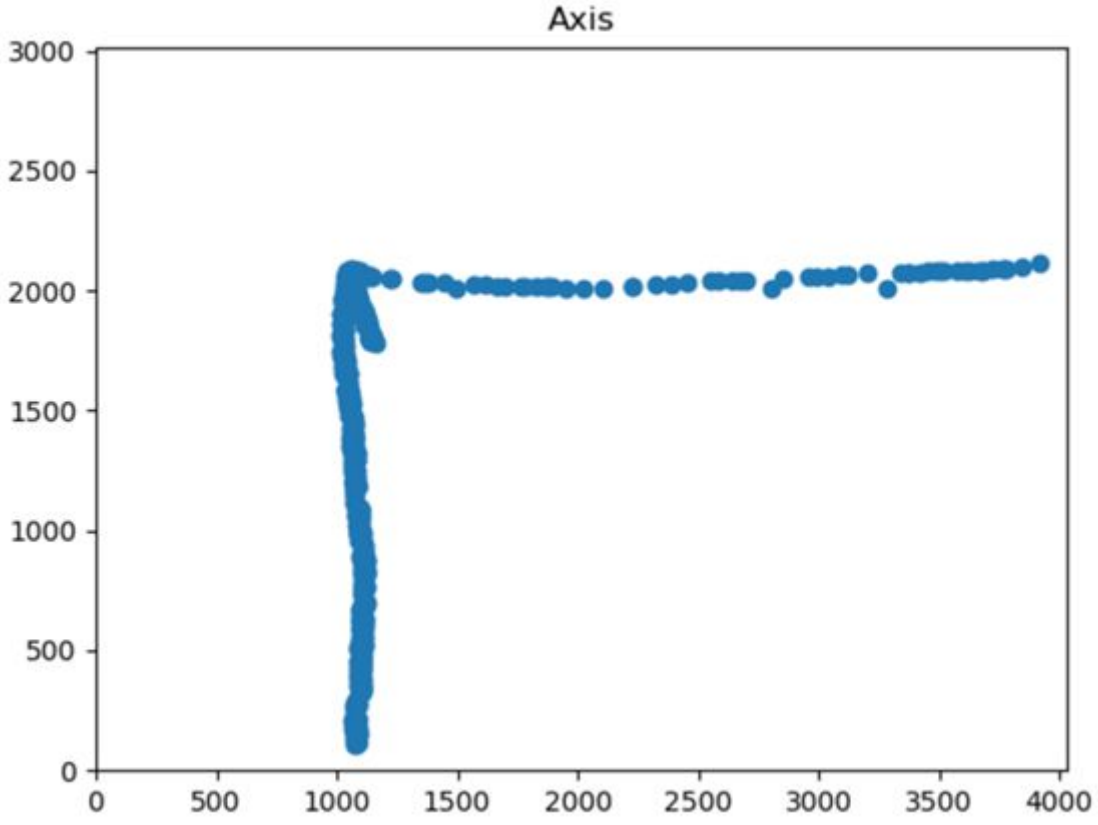
- Log level 3 - This log level will create graphs and requires user interaction. When running this log level, you will see a new window that will be opened. GTT will stop it's run, until you will close the window.
  For example you may see the following window opened:



So, what we actually see here?

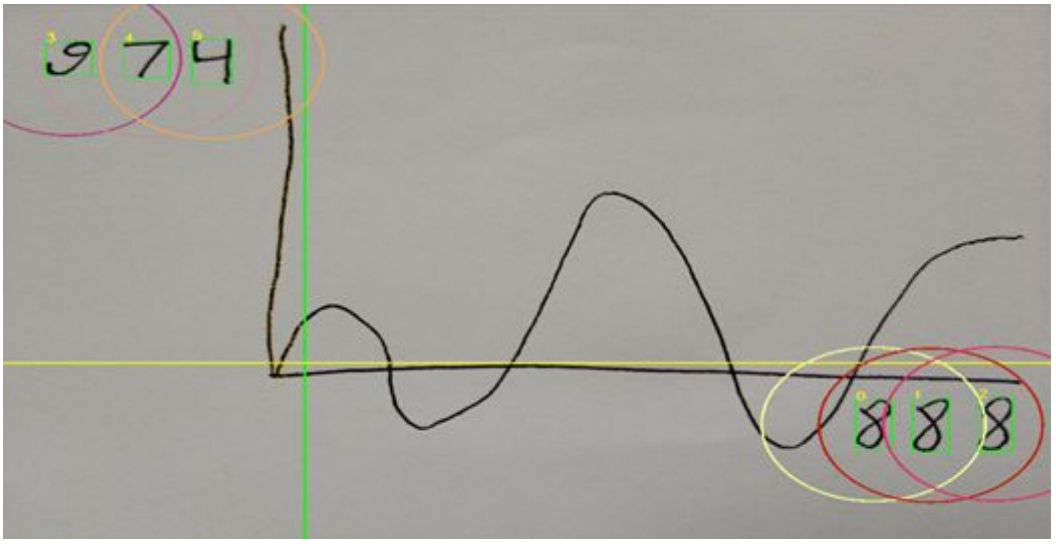
- ❏ Linear lines -
  - ○ Green horizontal line - represents the maximum X value of the detected Y axis.
  - ○ Yellow vertical line - represents the maximum Y value of the detected X axis.
- ❏ Squares - each square is a suspected digit. The index near the square is the index that appear in the log file (and also image index).
- ❏ Circles - each digit has an epsilon circle area. When we union the digits to numbers we look at the epsilon area of the digit and search for another digit in it. All the digits with coinciding epsilon areas are united into a single number.

- Log level 4 - This log level will create two pop-up windows. Again, this log level requires user interaction.
  - ○ First popup will show you the detected X and Y axis. For example:

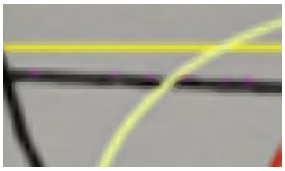


Pay attention that the Y axis is reflected to X axis.

- The second popup is almost identical to the popup from log level 3, but in addition we add dots on the recognized axes:

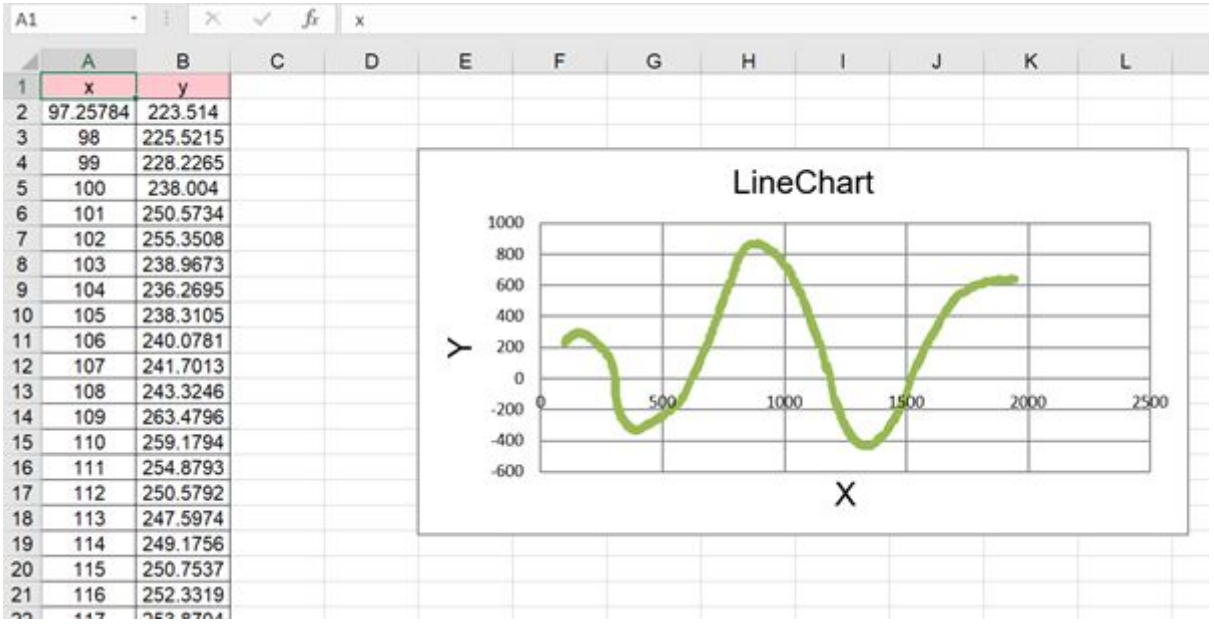


Y-axis -  GTT will create small yellow circles along the found y axis.

X-axis -  GTT will create small pink circles along the found x axis

7. Output

In addition to the created log file (in case of log level >=1), the main output of GTT is the excel/csv file that is placed at the generated “/out” folder (This folder, will be auto-generated by GTT).
Each time, you run GTT, it will create a new excel/csv file.
The name will be: “<image-name>_<id>”, where image-name is the original name of the source image, and id is an internal unique identifier.
Example output:



	A	B
1	x	y
2	97.25784	223.514
3	98	225.5215
4	99	228.2265
5	100	238.004
6	101	250.5734
7	102	255.3508
8	103	238.9673
9	104	236.2695
10	105	238.3105
11	106	240.0781
12	107	241.7013
13	108	243.3246
14	109	263.4796
15	110	259.1794
16	111	254.8793
17	112	250.5792
18	113	247.5974
19	114	249.1756
20	115	250.7537
21	116	252.3319

In the left side of the excel file, GTT creates a table of x and y values that represents the detected graph.
Scale rate is defined by default or by the given user input.
The x values may start at bigger value then zero. This is a direct result of small image dimensions with big maximum number value.
In the center of the excel file, GTT creates a graph that is based on the x and y values from the left side of the excel.

8. Wrong Detection

In case GTT produces a wrong detection of the graph (incorrect numbers, or undetected axis or even a different graph) you may try these options:

- Make sure that the image quality is good enough.
- Try to open the image using your computer and verify that the image isn't corrupted.
- Crop noise background – GTT is very sensitive. Your image should contain only white background and the handwritten graph. Any other objects in the image, may confuse GTT (even a shadow may confuse GTT because of the sensitive image recognition modules).
- Upgrade your log level as explained at 5.Log level information. That will help you to identify where is the problem.
- Take a new image 😊