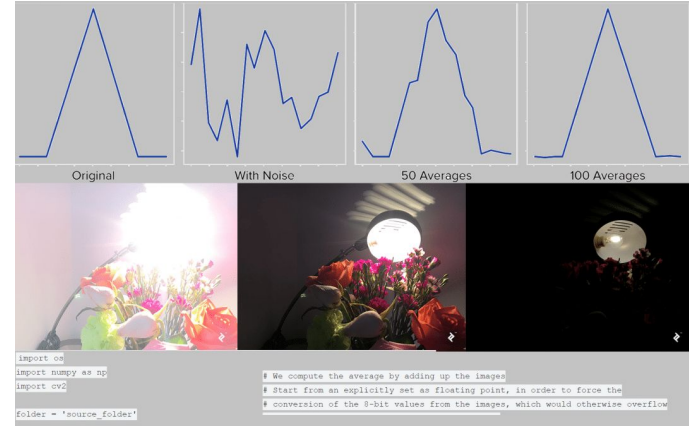


# **Python Final Project Presentation Image Processing Graphical User Interface (GUI)**

David Lara, Jomar Veloso

# Project Description

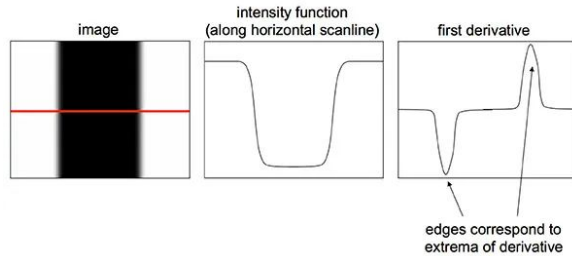
- For the project, a Graphical User Interface (GUI) with Image Processing capabilities was decided upon each selection from the Project
- The image processing capabilities that were decided upon were the ability to invert colors, apply a black and white filter, blur an image, sharpen an image and detect edges on an image
- The GUI library that was used in this project is Tkinter, a GUI library that comes pre-packaged with Python
- For image processing, Python Pillow was used since it allowed for information about an image to be easily revealed such as the Red, Green and Blue values of a pixel, as well as the height and width of the image which allowed for easy processing
- The Graphical User Interface (GUI) is a desktop interface that allows the user to communicate with computers regardless of the device being used
- There was also math involved since height and width were a key factor and the size of the picture or images



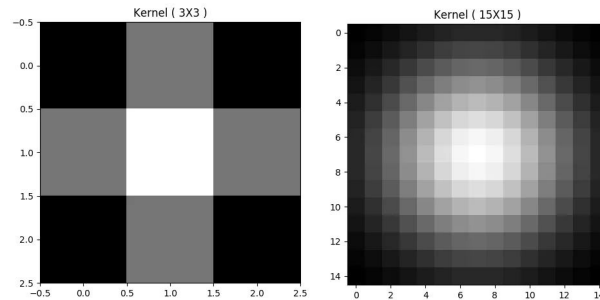
# The major algorithms that gave this project life

- Sobel Edge Detection algorithm was implemented in order to apply edge detection to the image
- The Gaussian Blur algorithm was implemented in order to apply a blur to an image that is a lot more context aware than a 'mean blur' which would only take the averages of the values around the target pixel.
- Finally, a generic sharpening algorithm was used to apply a sharpening filter to an image

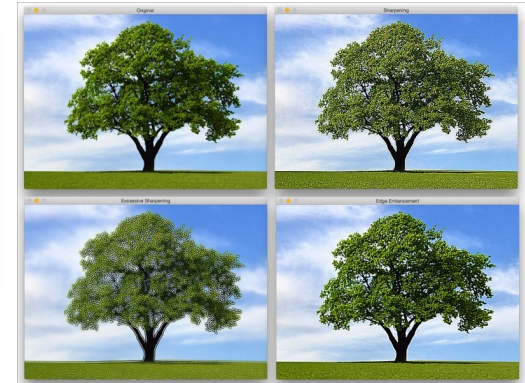
Sobel Edge Detection algorithm



Gaussian Blur algorithm



Generic Sharpening algorithm





## How do these algorithms work?

- Those three aforementioned algorithms all share a similar property of using a 'box' known as a kernel matrix to collect information about the image, in the case of this program, that kernel matrix for each of these algorithms is set to be a 3x3 matrix for performance sake.
- In the case of the Sobel Edge Detection algorithm, we grab a 3x3 of pixels from the target image, and we compute the 'direction' of those pixels in the X and the Y direction so that we can get an idea of what direction a change in pixel will occur in.
- We calculate these directionals by multiplying the 3x3 by another 3x3 matrix, one that is used to compute the X direction of the change in pixels, and another that computes the Y direction of the change in pixels.

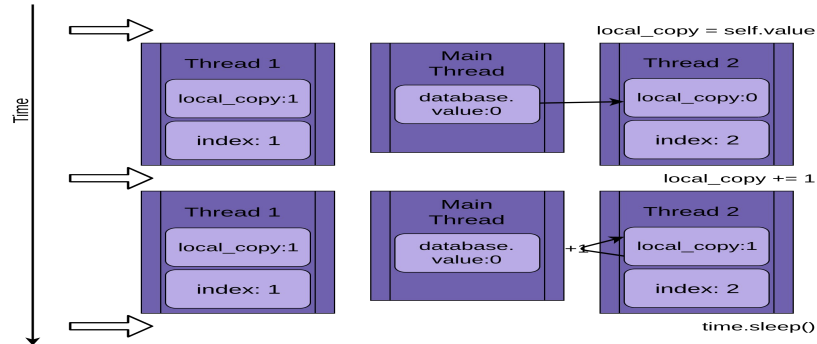


## Cont.

- Then, we normalize the result and then place the normalized result in place of the original pixel we decided to apply the filter to, our pixels  $i$  and  $j$ .
- Notes, this process is made a lot easier if you convert the image to black and white first since calculating a change in color is a lot harder with a colored image as opposed to a black and white one.

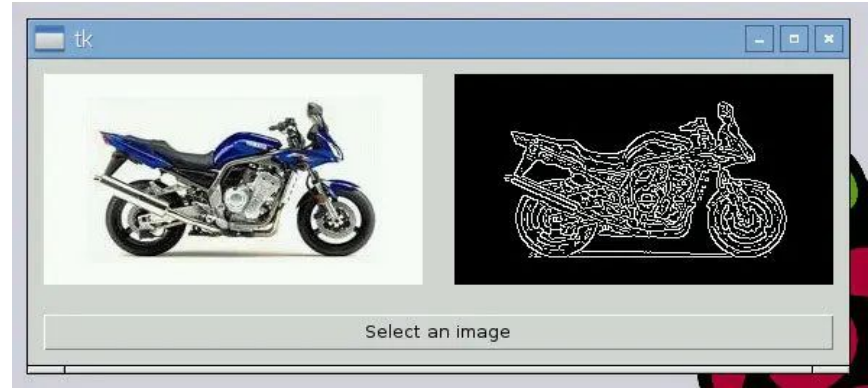
# Threading

- An important feature when using one of the heavier filters as without assigning the image filter to another thread, the app wouldn't crash but it would hang and lead to a generally unfavorable user experience.
- With this feature, hanging would stop as the function responsible for transforming the image could do what it needed to without affecting the end user.
- In python, Threading is used to run multiple threads (tasks, functions, or calls) at the same time
- If the program is at the percentage of 100 at the CPU time, the Python thread will slow down
- It can also run separately too

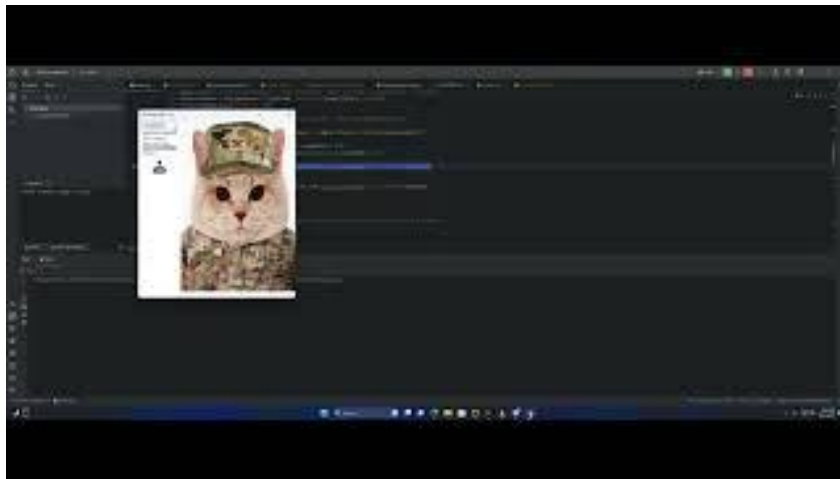


# Tkinter

- Easy to develop a working GUI
- Feature rich and allows for a decent amount of customization in how you want your GUI to look
- Canvas feature of Tkinter allowed for a decent place to place images in for viewing and editing purposes
- The ability to map buttons to images also is a great boon in customizability in the GUI
- When Python and Tkinter combine, Tkinter can create GUI applications easily without delay
- Additionally, it provides a powerful object - oriented interface to the Tk GUI toolkit



# Demonstration





# Work Cited



S, Ravikiran A. "Python GUI: Build Your First Application Using Tkinter." *Simplilearn.Com*, 5 June 2023, [www.simplilearn.com/tutorials/python-tutorial/python-graphical-user-interface-gui#:~:text=A%20graphical%20user%20interface%20\(GUI,erase%20various%20types%20of%20files.](https://www.simplilearn.com/tutorials/python-tutorial/python-graphical-user-interface-gui#:~:text=A%20graphical%20user%20interface%20(GUI,erase%20various%20types%20of%20files.)

Larson, JB. "Python Image Processing: A Tutorial." *Built In*, 18 Apr. 2023, [builtin.com/software-engineering-perspectives/image-processing-python](https://builtin.com/software-engineering-perspectives/image-processing-python).

"Python - GUI Programming (TKINTER)." *Tutorialspoint*, [www.tutorialspoint.com/python/python\\_gui\\_programming.htm](https://www.tutorialspoint.com/python/python_gui_programming.htm). Accessed 6 Aug. 2023.

Computerphile. (2015, November 4). *Finding the Edges (Sobel Operator)* - *computerphile* [Video]. YouTube.

<https://www.youtube.com/watch?v=uihBwtPIBxM>

Admin. (2019). Understand Gaussian Blur Algorithm: A Beginner Guide – deep learning tutorial. *Tutorial Example*.

<https://www.tutorialexample.com/understand-gaussian-blur-algorithm-a-beginner-guide-deep-learning-tutorial/>

projects, Contributors to Wikimedia. "Python Programming/Threading." *Wikibooks, Open Books for an Open World*, 29 Oct. 2022, [en.wikibooks.org/wiki/Python\\_Programming/Threading#:~:text=Threading%20in%20python%20is%20used,to%20look%20into%20parallel%20programming.](https://en.wikibooks.org/wiki/Python_Programming/Threading#:~:text=Threading%20in%20python%20is%20used,to%20look%20into%20parallel%20programming.)

"An Intro to Threading in Python." *Real Python*, 22 May 2022, [realpython.com/intro-to-python-threading/](https://realpython.com/intro-to-python-threading/).