**Filesearch+**

**Design Documents Ideas**

**GROUP MEMBERS**

Hao Wu, Henry Volodin, Jennie Cha

**SOFTWARE OVERVIEW**

The main function of this software is image detection. Windows can only search images by metadata, title, or other words, not being able to search images by the objects in the image. To solve this problem, we will use pre-trained machine learning models in Python to detect the objects in the image. When the users enter a keyword to search, the program starts to detect the object of that keyword and shows the user all the images with that object. Besides, we plan to create a user-friendly GUI that enables the users to search the objects they want in the file they choose and display the images with objects users want.

**ANALYSIS OF THE SOFTWARE**

The core of this software is machine learning image detection. The mechanism for machine learning object detection is to extract different low-level features of the object, such as color, edge, texture, gradient, and frequency-based features, and combine them into different layers. Besides the object characteristics, neural networks add more layers, like activation layer and pooling layers, to improve capabilities. By combining each neuron (function) in each layer into a complex network and determining the flow of information by weight, the neural network can conduct a sophisticated task, such as detecting a complicated object in a real-world setting.

Due to the limitation of CPU capacity of our computer, we are not capable of training a feasible image detection model. Instead, we’ll use a pre-trained model for our purpose. Image detection requires a big amount of computation; as a result, we plan to use the smallest model in the YOLOv5 series model, which can detect only 80 pretrained objects with limited accuracy.

When running our program, a window will show to enable the user to enter the object he/she wants to detect. A suggestion list of object names will show when the user types some relevant letters to improve user experience. After entering the object name, a new window will pop up to allow the user to choose a file that contains the image. After user selects the file, the model starts to evaluate each image in that file. When the detection is over, pictures we want will show in a picture gallery, a new window. Users can shift the pictures by clicking the “Next” or “Previous” button.

**ALGORITHM OF THE SOFTWARE**

Pseudocode:

###1. Import needed libraries

import image detection machine learning, GUI, system file connection and image display libraries

###2. Load image detection mode

load yolov5s image detection model

suggestions = list 80 objects that the image detection model can detect

###3. Main GUI window

create an empty GUI window

set window appear on the top of the screen

set window size, title

add several user prompts in the window

set these prompts automatically pack with each other

###4. Show suggested objects when user types something

# Define a function that show the top 5 suggestion

define show\_suggestions:

query = object\_input that user types

if query is not empty:

store the object names in suggestion\_list if query is a substring of the defined object in suggestions

set suggestion\_list to store up to 5 suggested objects

else:

clear suggestion\_list

# Create a GUI entry that allows user to enter object name

define object\_input that are able to bind with entry widget

create a connection that when user type something, it triggers show\_suggestions function

create an entry widget to allow user to types the object he/she want to detect

bind the text in the widget with object\_input

set the entry size

set the entry to automatically to place below the widgets defined before

# Create GUI list show 5 suggestion

define suggestion\_list that are able to bind with entry widget

create a list box to show suggested objects

bind the text in the list box with suggestion\_list

set the entry size

set the entry to automatically to place below the widgets defined before

###5. Get selected object by clicking

# Define a function that get selected object into entry

define get\_suggestion:

selected\_object = user clicks the object in the suggestion\_list by cursor to select the object

update object\_input as selected\_object

clean suggestion\_list

bind "click" operation in the list box to get\_suggestion function

###6. Get object name and path for the image detection model

# Define a function that gets object name that user wants to detect

define get\_object\_name:

set object\_name as global variable

object\_name = text in the aforementioned entry that allow user to enter object name

close the current window

add disclaimer to the window

set the disclaimer to automatically to place below the widgets defined before

create a button that triggers get\_object\_name function when clicking

start event loop to enable the GUI to interact with user

if object\_name is not empty:

pop up a window to let user to select picture file

directory\_path = the file path of the file that user chooses

### 7. Use image detection model to detect image

image\_paths = an empty list

for (each file in the chosen directory)

if (the file extension is 9 kinds of common-used image extension)

file\_path = join the current image name and directory\_path

open the image based on file\_path

use image detection model to detect the all objects in it

result = only store the detection result we want in specialized data frame (detection result == object chosen)

if the result is not empty:

image\_paths = join the current image name and directory\_path

print object founded

try:

display image

if successful: print image shown successful

if failed:

show the specific error

else:

print object not founded

else:

print image doesn’t exist in the chosen directory.

### 8. Display detected image in a photo gallery

initialize current\_image\_index = 0

# Define a Function that display the image

define update\_image:

define image\_label and image\_display as global variable

if image\_paths is not empty:

img\_path = image\_paths with index 0

resize image

convert image format into a Tkinter-compatible format

display the image in the GUI window

# Define a function that shows the next image

define show\_next\_image:

if current\_image\_index < the total index of image\_paths – 1

increase the current\_image\_index by 1

update\_image

# Define a function that shows the next image

define show\_previous\_image:

if current\_image\_index < the total index of image\_paths – 1

decrease the current\_image\_index by 1

update\_image

#Create an image display window

create and initialize a new GUI window

set window appear on the top of the screen

set window size, title

set the window with a refined theme (appearance).

set the entry to automatically to place at the beginning of the window

create a button named “Previous”, that triggers show\_previous\_image function when clicking

set the position of the button to left corner of the window

create a button named “Next”, that triggers show\_next\_image function when clicking

set the position of the button to right corner of the window

if the images are found

update\_image

else

print “No images found that match the criteria."

start event loop to enable the GUI to interact with user

Discovery path:

We designed the software by first employing a feasible image detection model. After that, we need to build a user-friendly GUI. The first part is a window that can let users enter an object name they want to detect and choose the designated file. A suggestion list of object names will be built to facilitate user input. The second part is a picture gallery that shows the pictures found.

**Document prepared by**

Hao Wu

**Date Created**

November 10, 2024

**Date Last Updated**

December 4, 2024