GUI User Manual

Document status	RELEASED

Version control

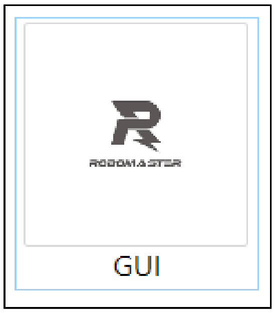
Version	Description	Author	Date (DD/MM/YYYY)
v 1.0.0	GUI information and User Guide	Sejin Kim	31-10-2020

Quick user manual to use object detection GUI

1. Overall image cut

Client customized GUI icon

Instead of the generic Python launcher icon, we used the project icon to create the product's executable in order to deliver a slightly more customized product to the client. The image of the executable file with the customized icon is shown in the figure below.

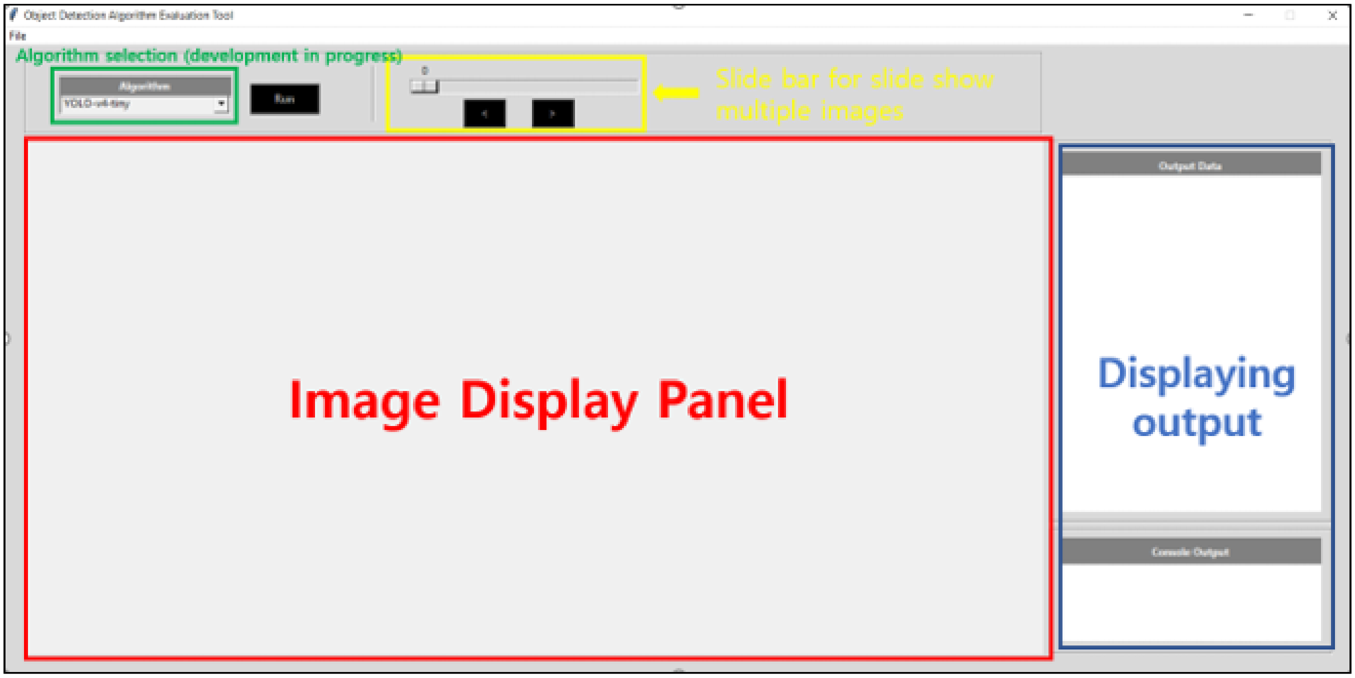


Easy to understand interface

Instead of a complex user interface, we introduced an easy, simple and intuitive interface.

The first screen that appears when you click the program icon is as follows.:

More detailed images and descriptions of the zoomed-in components follow.

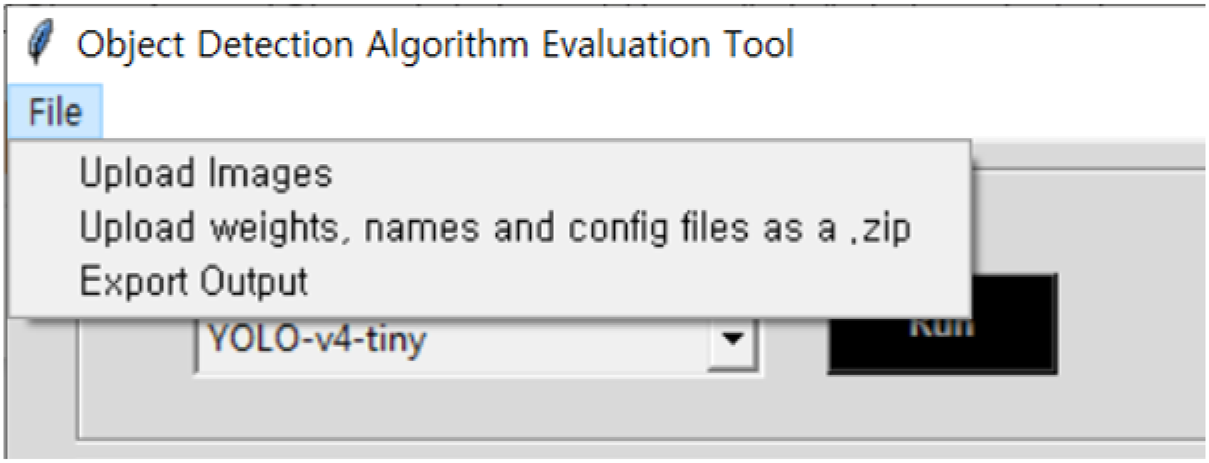

Object Detection Algorithm Evaluation Tool
File
Algorithm selection (development in progress)
Algorithm
YOLO-v4-tiny
Run
0
<
>
Slide bar for slide show multiple images
Image Display Panel
Output Data
Displaying output
Console Output

2. Explanation of the components

The ‘File’ menu

There are three main functions in the file menu: uploading photos, uploading configuration zip files, and exporting output.

These menus appear when you click the file drop-down menu at the top of the application window, and when you click each menu, a new pop-up window for executing the corresponding function appears.

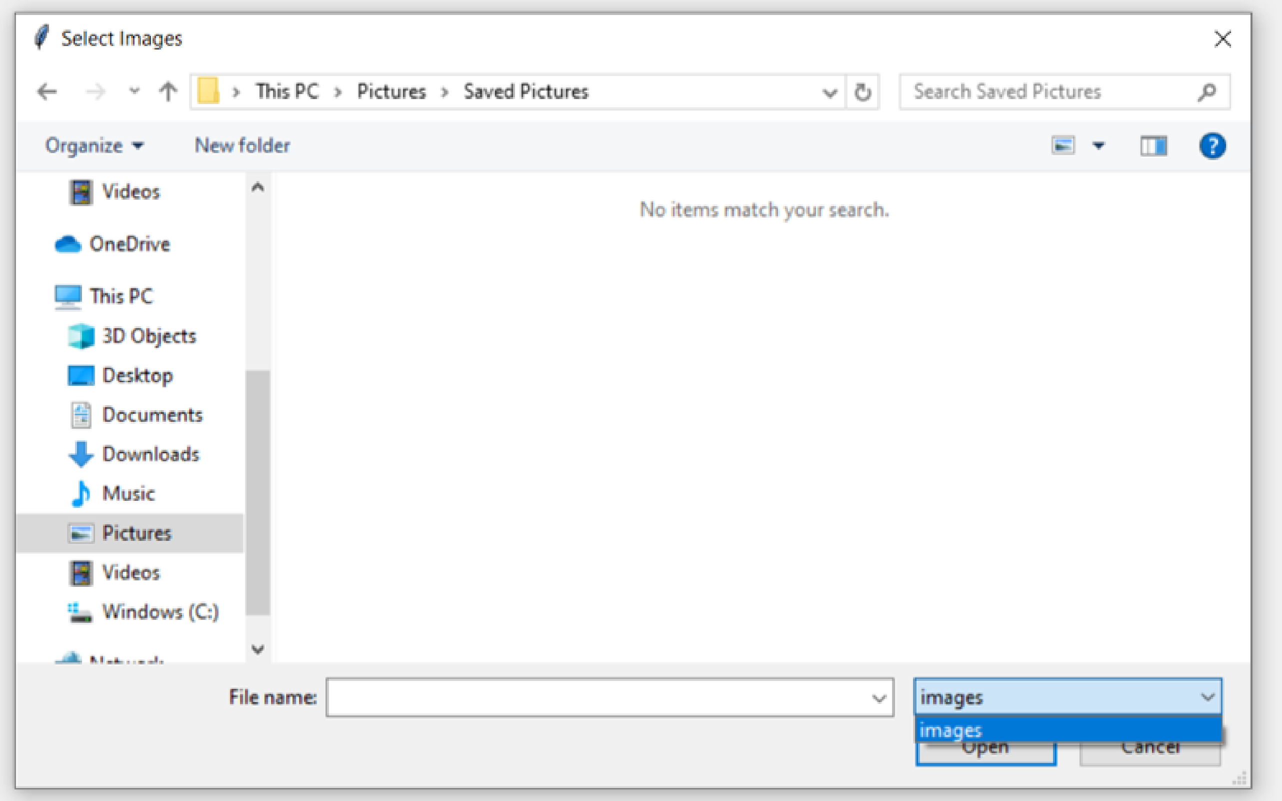

Object Detection Algorithm Evaluation Tool
File
Upload Images
Upload weights, names and config files as a .zip
Export Output
YOLO-v4-tiny
Run

Upload Images

Users can upload images from their local PC to the app using the select images menu. Using this function, the user can upload one or several images at once. The pop-up window that appears when you click the menu is as follows.

In the pop-up window as above, the user can move to the directory containing the images to be uploaded and upload multiple images.

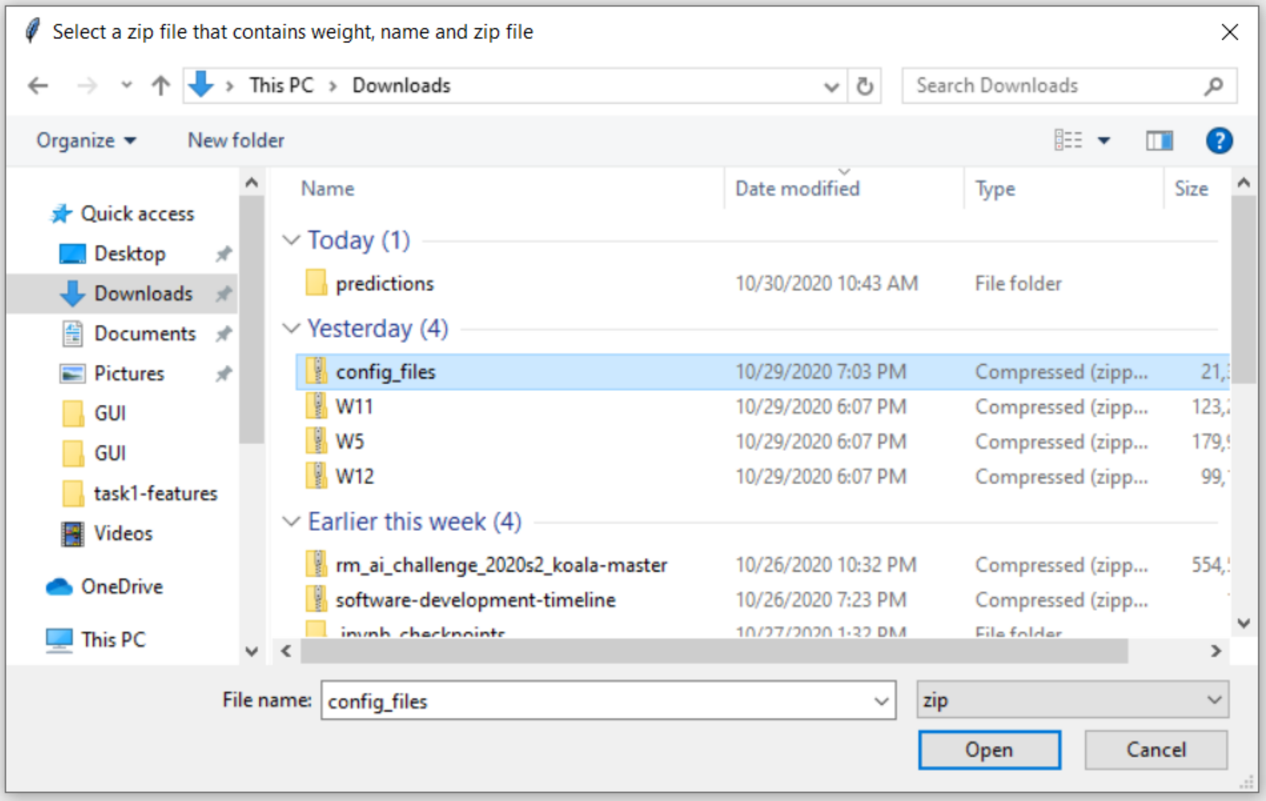
If there is no image type file in the selected directory, a blank window with no files to select is displayed as in the window above.



No items match your search.

Upload weights, names and config files as a .zip file

Various types of input (information) are required to create an object detection model. We did not embed the YOLO algorithm itself into the GUI, as we had the possibility of future expansion of the program in mind (more about this in the algorithm selection drop-down menu). configuration file I needed an input of a name file containing the name of the target class to be labeled. Our program receives these configuration files as compressed files at once. After selecting the configuration upload, the following window appears.



Select a zip file that contains weight, name and zip file

Name	Date modified	Type	Size
Today (1)			
predictions	10/30/2020 10:43 AM	File folder	
Yesterday (4)			
config_files	10/29/2020 7:03 PM	Compressed (zipp...	21,
W11	10/29/2020 6:07 PM	Compressed (zipp...	123,
W5	10/29/2020 6:07 PM	Compressed (zipp...	179,
W12	10/29/2020 6:07 PM	Compressed (zipp...	99,
Earlier this week (4)			
rm_ai_challenge_2020s2_koala-master	10/26/2020 10:32 PM	Compressed (zipp...	554,
software-development-timeline	10/26/2020 7:23 PM	Compressed (zipp...	
ipynb_checkpoints	10/27/2020 1:32 PM	File folder	

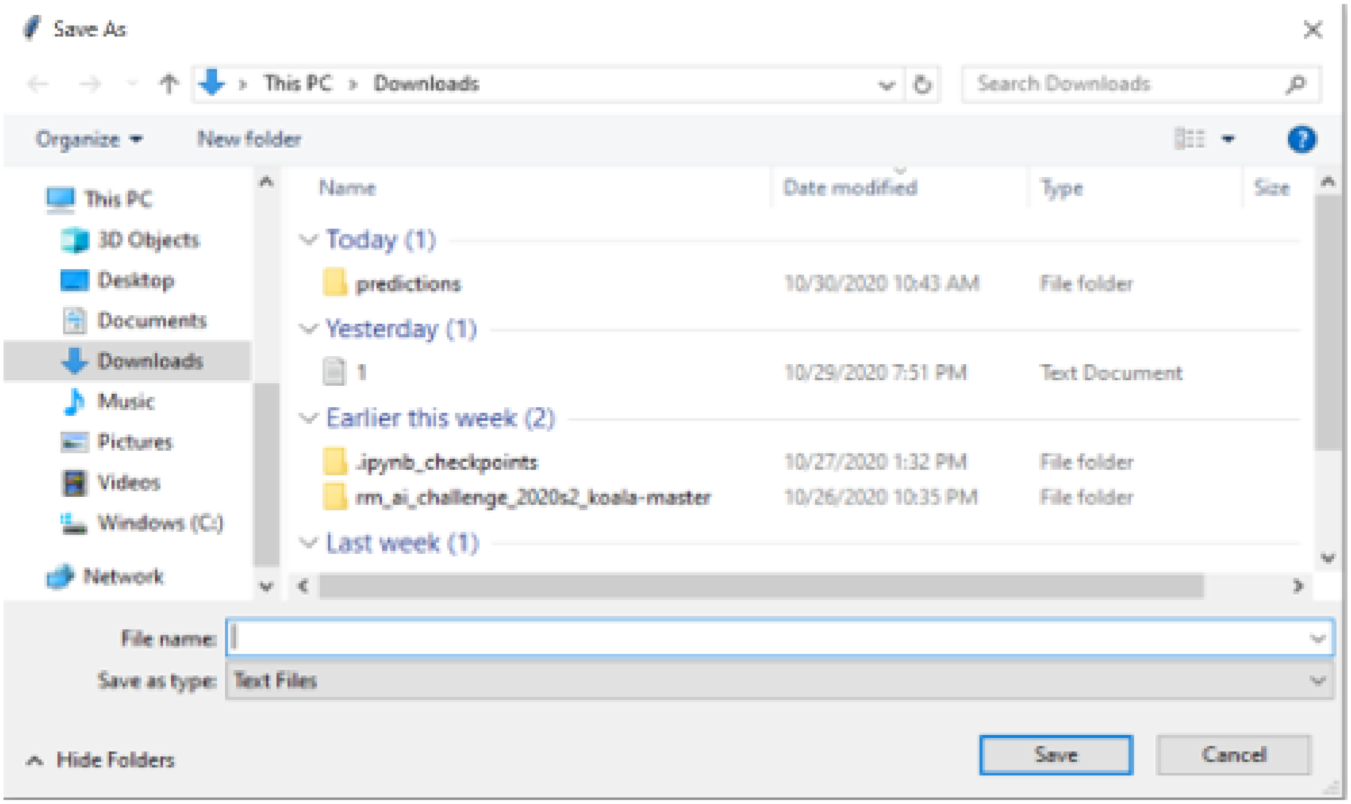
File name: config_files

zip

Export Output

This menu is used to save the model output to the user's local computer in .txt file format. Unlike the other functions mentioned above, this is used after the model has finished detecting objects, and the user can choose not to export the results.

The example window below is a pop-up window that appears when you press the export output button.



Name	Date modified	Type	Size
Today (1)			
predictions	10/30/2020 10:43 AM	File folder	
Yesterday (1)			
1	10/29/2020 7:51 PM	Text Document	
Earlier this week (2)			
.ipynb_checkpoints	10/27/2020 1:32 PM	File folder	
rm_ai_challenge_2020s2_koala-master	10/26/2020 10:35 PM	File folder	
Last week (1)			

File name:

Save as type: Text Files

Other parts of the GUI

Run button

After uploading a configuration file suitable for the algorithm and images for object detection, the object detection model detects the target object in the image by clicking the run button.

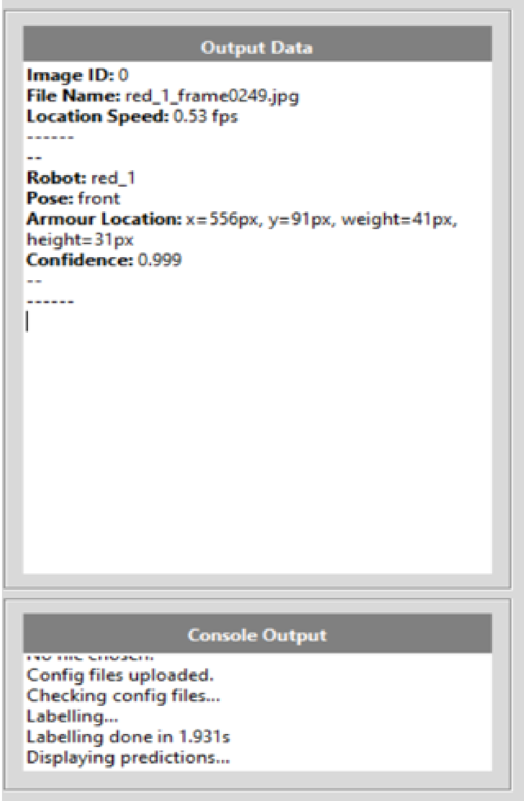
The detected result is displayed on the output panel.

Output Panel

There are two main types of output panels in this GUI: one is the output panel that shows the labelling results and the other is the console output panel that logs what is happening in the console.

Through the output data panel showing the results of object detection, we can see the prediction results in a clearer and more organized form, and through the console output panel, users can easily see what is happening in the backend.

Below is an example of the results displayed in the result output panel using the above program.



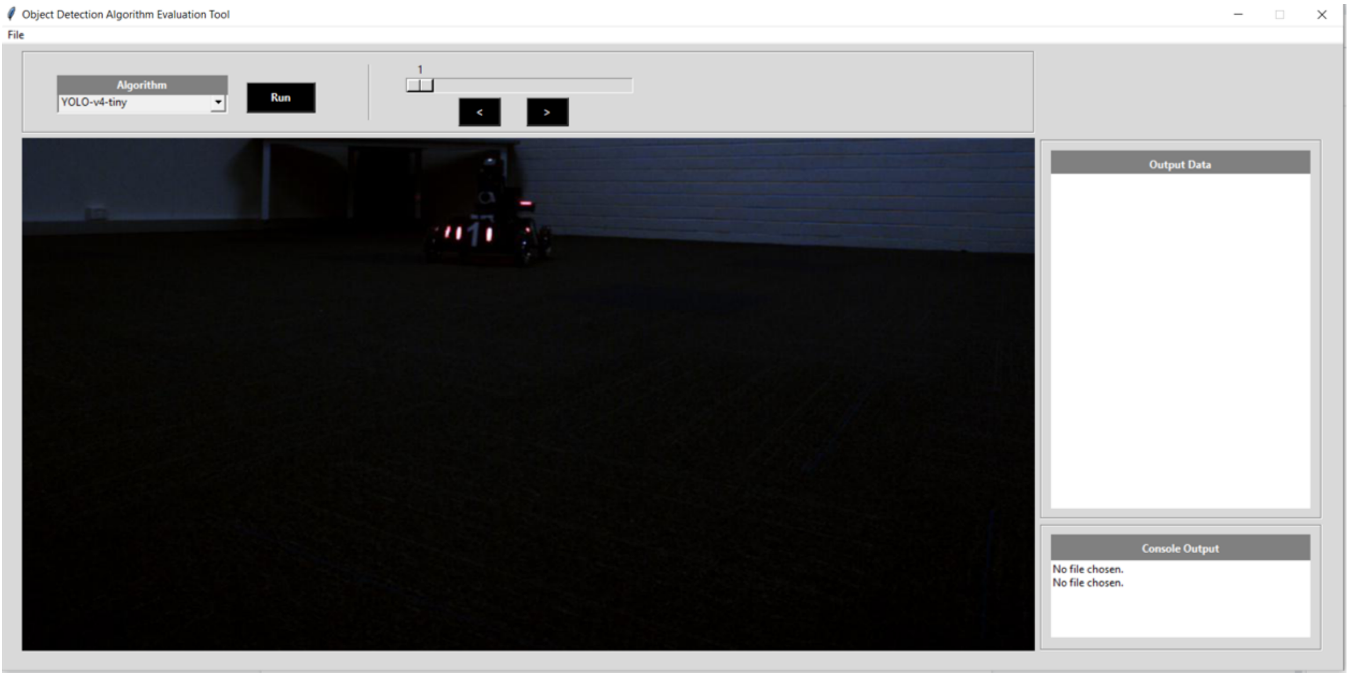
```
Image ID: 0
File Name: red_1_frame0249.jpg
Location Speed: 0.53 fps
------
--
Robot: red_1
Pose: front
Armour Location: x=556px, y=91px, weight=41px,
height=31px
Confidence: 0.999
--
------
```

```
No file chosen.
Config files uploaded.
Checking config files...
Labelling...
Labelling done in 1.931s
Displaying predictions...
```

Display Image Panel

This panel shows the uploaded image before performing object detection. After running the object detection algorithm, the image in which the objects found as a result of detection are displayed on the image as bounding boxes is output.

The example image below displays an unlabeled image in a situation immediately after uploading an image and before running the object detection model.

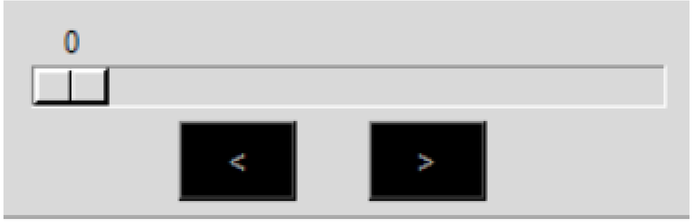


```
No file chosen.
No file chosen.
```

Slider

Since we can upload multiple images, we needed the ability to display multiple images in a slideshow format. The image below is a slide bar that performs such functionality.

A user either slide images by sliding the button on the slide bar or by clicking on < button to back to the previous image and > button to move to the next image. Below is the image of the slide bar in the application.

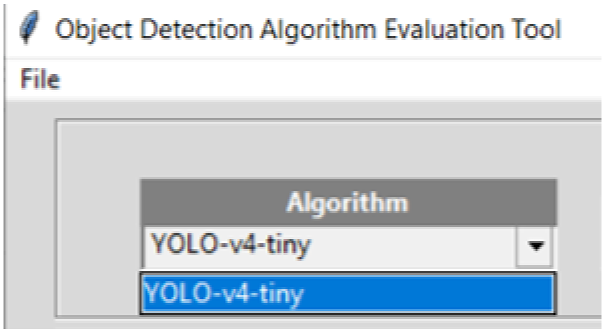


Select Algorithm

This part has not yet fully developed but aims to increase the application's ability on the introduction of additional algorithms. For now, we only have developed one algorithm using YOLO v4 tiny, which was the scope of our project.

However, we are planning to extend this program's usage to general-purpose, so it is ideal if we are able to use different algorithms in this application as well. The rationale behind this option is to increase the reusability and extensibility of the application.

This Combobox option of algorithms idea was introduced with the intention of performing different kinds of object detection algorithm by letting the application load a model that matches the selected algorithm and with the user uploading a configuration file that matches the algorithm they intended to use.



Algorithm

YOLO-v4-tiny

YOLO-v4-tiny

3. How to perform prediction

This section is the base guide to object detection using this application with images (where it is necessary) and potential error messages users might get to help users understand about the product.

Object detection process

- **Upload images**

Click the 'File' menu and select 'Select Images'

Navigate to the directory where the images to upload

Select an image/multiple images and click 'open' button

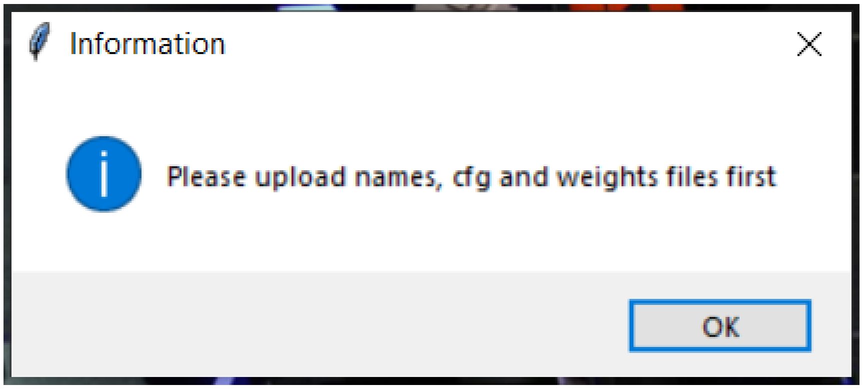
- **Upload configuration files in .zip**

Click the ‘File’ menu and select ‘Upload weights, names and config files as a .zip file’.

Navigate to the directory where the compressed configuration file to upload.

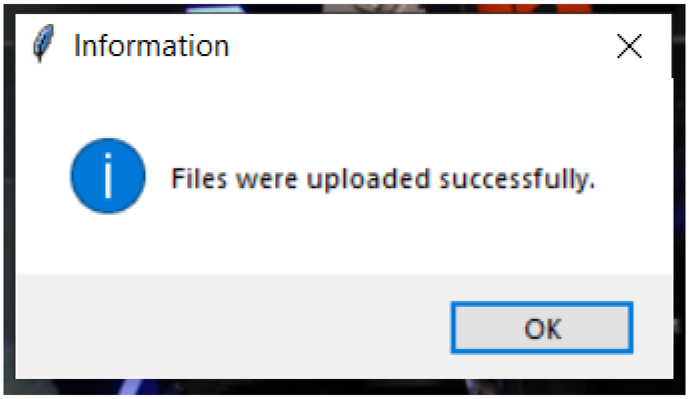
Select a configuration file in .zip format and click ‘open’ button.

NOTE: this is a MUST DO thing before hitting the ‘run’ button. If a user didn’t supply configuration files or the compressed file doesn’t have sufficient files needed(.cfg file, weight file and name file), he/she would get the following error.



Please upload names, cfg and weights files first

Once a user uploaded a proper configuration file, he/she would get the following message.



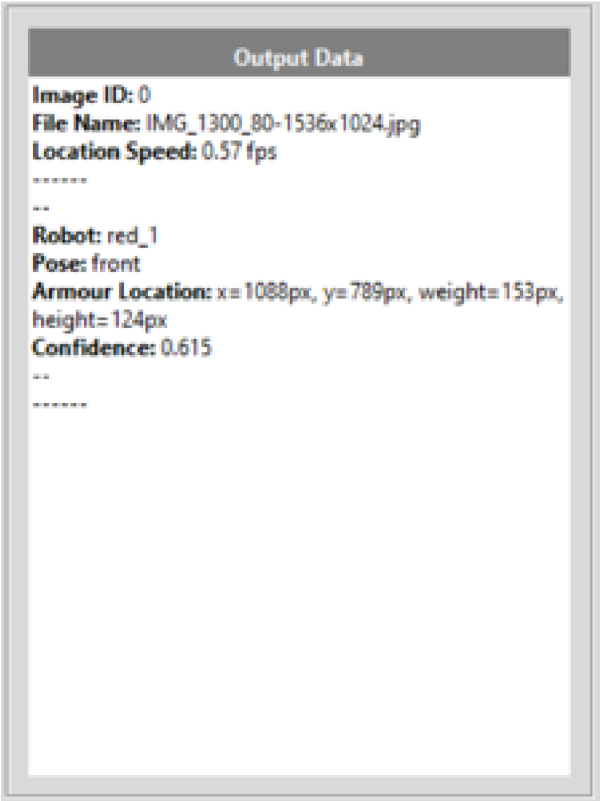
Files were uploaded successfully.

- Hit run button
- Use slider to get a ‘slide show’ view of images(optional)

This applies only when the user uploaded multiple images to do object detection.

- Check the result on output panel & export results

Once a user hits the run button and the application finishes predicting, the user would get labelled image on the image displaying panel as a result and text form of output for a clearer view and detailed information. This is an example output on the output panel of the application.



```
Image ID: 0
File Name: IMG_1300_80-1536x1024.jpg
Location Speed: 0.57 fps
------
--
Robot: red_1
Pose: front
Armour Location: x=1088px, y=789px, weight=153px,
height=124px
Confidence: 0.615
--
------
```

A user can export this result to a local machine in .txt format as well.

To do this, click the ‘File’ menu and click ‘Export Output’ option

Navigate to the directory wish to save the output

Set a name of the output file and click the ‘save’ button.