

How to use this Autolt image detector:

Point 1:

Install JRE (Java Runtime Environment), without this you can't execute Java Applications (.jar files). Open the Java tool named AISD with double click on it, or "java -jar aisd.jar" in the cmd. In AISD, open the image you want to detect in your script. Remember: the upper left pixel of the image, MUST be a not common color in the background if you want a fast detection. Then you will get a script file if you want, or only a clipboard copied version, full ready to paste into your main script because this one comes with extra help codes. Anyway, the generated code is the \$sign definition of the image (calm down, point 3 explains what is the \$sign and it's easy).

Point 2:

Paste the mentioned code in your main script, and try to follow the sample_usage script. If you know how to write au3 codes, it should be very easy. In general terms, all you need is to include image_search.au3, and then you can use the func image_search(\$sign, \$width, \$height). You need also the get_image_sign() function, that calculates the first parameter for image_search(). The get_image_sign() function is the one you have generated at point 1.

Point 3:

image_search() returns the center point of the target image. The func needs the \$sign from the image, width and height. This \$sign is the "footprint" of the image, in the form of an array of 9 integers. In most of the cases the \$sign should be enough to identify the image, this is useful because using the \$sign instead of the real image let you get a small script and a fast detection. You can also delete the image once you get the \$sign. This \$sign has always the same size (those 9 integers, it does not matter if the original image is small or big). How can you define the \$sign for a certain image?, you don't need to generate the \$sign, AISD does (point 1). Follow the sample_usage script and you will get it.

Files in this distro package:

- * AISD.jar: The already explained Java app.
- * image_search.au3: The library or module with the resources that lets you detect images.
- * sample_usage.au3: The mentioned sample_usage script. It contains the smallest example, but i think it should be ok for learning purposes.
- * sample_usage.exe: The sample_usage.au3 script compiled (au3 -> .exe). You can execute this to see the image_search() func working.
- * sample_target.png: The image to be detected in the sample_usage. You can delete this file and the script will be fine.
- * OffsetCalc.jar: It's only a Java tool to calculate the offset to transform an XY point into another one. It's useful if you want to use apply_offset function (from image_search.au3). Both things are only necessary when it's hard to find a version of the target image with upper left pixel in the way mentioned. As you can imagine, the idea is to detect another image and then apply the offset to get the real coordinates of the original target image.