# On Guard Security Assistant

## What is This For/What Does it Do?

Security cameras are getting more and more capable. Detecting motion is one of the most useful features for such cameras. However, the consumer grade cameras are not particularly good at sifting important movement from “junk” movement. Further, movement of one type (people walking) may be important in one area and uninteresting in other areas. Also, it can also be important to know the difference between motion caused by moving cars, or moving people. In these cases AI can be very useful in identifying the type of movement. Getting free software that accomplishes these goals is now possible.

On Guard provides an engine and user interface that takes images from a video source and processes them through an Artificial Intelligence application. It takes the location of objects in video snapshots identified by the AI application and further analyzes them for importance, location, and size. It provides the ability for the user to delineate areas of interest, and the type of objects that are of interest within an area. It can optionally trigger an external application (Blue Iris) to start recording a video. It can also optionally send an email notification to the user that important motion has been detected. It can notify smart devices (Internet of Things) via MQTT as well.

## Dependencies

On Guard has been currently tested using Blue Iris (<https://blueirissoftware.com/>) as a source for motion images. Blue Iris is a very capable Windows PC application for managing security cameras and recording videos from those cameras. Blue Iris has a built-in capability for detecting motion, and some capability of detecting objects (in general).

Blue Iris provides an important service in looking for motion in the cameras it monitors. Blue Iris does need to be setup properly to provide this service. Blue Iris can also be setup to start recording video when On Guard triggers it.

On Guard is **not** strictly speaking dependent on Blue Iris. Any software or IP camera that can provide images of motion/activity and can output a .jpg image of that motion can be used. Even a camera that just takes a snapshot periodically (say once per second or less) can be used. Many inexpensive cameras can do this. However, using an application like Blue Iris that detects motion and takes pictures only when there is motion is a valuable service.

On Guard relies on free Artificial Intelligence software called DeepStack (<https://www.deepstack.cc/>) as the engine identifying objects within the images provided by Blue Iris. DeepStack is an AI can identify a number of objects that we care about in a security camera as well as a number we don’t care about. What we do care about are people, vehicles, and perhaps animals. DeepStack is run as a separate application on a PC or Linux computer. Note that DeepStack exclusively works locally on a computer. It never sends images over the Internet. This is an important security feature.

On Guard is currently functional only with the DeepStack AI. However, the use of other AI processing applications that are free or low cost is also being explored.

## What does this cost?

On Guard itself is free for personal and small business use. However, it does (currently) work closely with Blue Iris, and Blue Iris does require a paid annual support agreement for upgrades (upgrades are important for this type of software). DeepStack is now free for personal use. See their website for any restrictions.

The other cost to consider is the cost of having a reasonably capable Windows PC working 24/7. It is suggested that you use a computer with at least 4 cores and at least 8GB RAM. Adequate hard drive space is also necessary. Beside the cost of the PC (which presumably you have), you will obviously need to pay for some electricity. Depending on where you live this may be cheap or somewhat expensive. Running the AI software can consume minimally more electricity than a standby PC since the AI process is CPU intensive, but events tend to happen relatively rarely. Blue Iris can also be somewhat CPU intensive.

Reasonable quality IP cameras (any camera using WiFi or wired networking can be purchased for between $50 to $300++ per camera. HD cameras are definitely a good idea, but they aren’t required. You can use as many cameras as you wish.

## How this program works…

It would be very nice to have an AI application that could read live video and identify objects of security interest. However, PC software and hardware is not up to that task. Even if the AI could do this, the AI cannot (now) know whether the objects it found are important to you. On Guard is a compromise, but it is a very good compromise for monitoring video cameras and reporting important activity to you.

On Guard works in distinct phases. In the first phase Blue Iris (or a security camera) detects movement. It then starts taking pictures at a rate you define (starting with 1 frame per second to 1 every 3 seconds is suggested). On Guard detects the presence of the picture taken by the camera. It then passes the camera images to the DeepStack AI. The DeepStack AI analyzes the frame. On a relatively fast PC DeepStack can take 1/4 of a second. DeepStack passes the result of the analysis back to On Guard. On Guard takes further look at the picture and the objects returned by the AI. It determines where those objects are. Depending on the type, location and size of those object it decides whether and how to alert you to movement on your camera.

## What your security camera sees may be important (or not).

For security camera purposes we primarily care about people and vehicles, but animals (dogs, cats, etc.) may be considered important to you as well. On Guard allows you to define what you consider important, and in which areas they are important. For instance, cars driving down the street or people walking down the sidewalk are probably not too interesting. However, if a car pulls into your driveway, or if people walk up to your front door, then that event may be quite interesting.

On Guard allows you to define one or more Areas of Interest (or zones as they are sometimes called). Within each area you define if you care about people, cars, trucks, etc.. You define what degree of confidence the AI has that the object it finds has been identified correctly. You define what percentage of an object must be within an area. You define the minimum size required for an object within the area.

## What happens when interesting motion is detected

You decide if you want to be notified by email, or if you want you can send a message to any web address (URL)/web service. Blue Iris has the capability to begin taking videos on the basis of that web site notification (even/preferably if the “web site” is on your local PC). On Guard also allows you to notify smart home devices to start taking some action (turn on lights, etc.) via IFTTT.com (If This Then That). IFTTT can be very powerful (turns on lights, etc.), but it does take some getting used to. You can also decide if you want to just ignore certain objects in certain areas (for example cars are important in your driveway but not the street).

## Getting Started

Download the .zip file from Github at: https://github.com/Ken98045/On-Guard.git. Usually you want to use the latest “Release” version. Extract the directory On-Guard-master. Open the Setup directory and run the Setup.exe program. If all goes well the application will start and the initial setup steps will be started.

On Guard supports multiple cameras. However, for now choose just one camera and get it working. After setting up one camera the others should come easily.

You will need to know the following information underlined and in **red**. First, setup either a camera, Blue Iris, or another source of producing still images when some sort of motion happens. Blue Iris is convenient because you can tell it something like “when you detect motion start taking snapshots at a rate of 1 per second for 10 seconds”. Blue Iris is “standard” for On Guard because it has this capability. From here on when talking about a video source we will be talking about Blue Iris.

When Blue Iris starts taking pictures it “**prefixes**” all the pictures with the **“short” name of the camera**. It then adds on digits representing the time/date of the picture. An example is: aiFront.20200726\_142218821.jpg. You must know what your Blue Iris camera name is. Blue Iris also stores these pictures on your computer hard drive in a location known as the “**path**” to the pictures. You will also need to know this. This path may or may not be where Blue Iris stores videos (it is **much** better if it isn’t). Get setup instructions for Blue “Iris from their web site. Instructions specific to On Guard will be discussed later.

Next, you will need to download and install the DeepStack windows PC (or Linux) program. The current (as of this writing) location for the Windows Download is: <https://deepstack.cc/install/windows>.

For now, use with Windows desktop version (UI) version of the DeepStack program. When you start DeepStack use the checkbox for “Detection API” (not Face API or Scene API). Select any value for **Port** (usually above 80 and always below 65,535). Port 8090 usually works well in most cases and is the program default. Click Start Now and it will open a window beside the main window. For now be sure no errors are displayed. Later in this document we will discuss how to run DeepStack automatically and without the (somewhat annoying) DeepStack windows.

In order for On Guard to display live video from your camera you will need to the **IP Address** (defaults to “localhost” which means “on this PC”) . You will also need the **camera port**. For Blue Iris you will also need the **camera name**, the camera (or Blue Iris) **user name**, and the camera (Blue Iris) **password**.

In order to send email notifications to you in the event of interesting motion events you will need your **outgoing email server address**, the **email server port** (yes a final port), and the **email server password.**

While this may sound like a lot of setting (well, yes it is a lot) there is not much we can do about that. Presumably if you knew enough to find this application, you know enough to find all of that information without too much difficulty. A Checklist of information required will be provided. Also, in many cases the default values will work well or provide you with a good starting place.

## Running On Guard

The first time you run On Guard you will receive a popup message directing you to set some setting values. There are a total of 3 settings pages you will need to fill in. After running for the first time the settings will be saved.

After your settings have been entered you will see either the last picture taken by Blue Iris or a blank screen. If you see a blank screen you should do a “walk test” in front of your camera. Assuming your camera and the On Guard settings are correct you should see a captured picture on your screen **after** you press Refresh. Also try “Snapshot” and “Continuous” image buttons. If your camera settings are correct you should see a current view from the camera. See the sample picture below.

When you see a picture your first step is to pick out a prominent point in your landscape (or interior) that you will always be able to visually identify. Hold the control key, and mouse click on that point. This puts a small purple square known as the registration mark overlaying your pictures. In the picture below in is just on the corner of the steps leading down to the lawn. This mark allows you to be certain that your camera is always aimed correctly. Cameras can be bumped, blown by the wind, and/or moved by the Pan/Tilt/Zoom capabilities of the camera. It is important to aim the camera correctly so that the Areas of Interest you set will be correct. There is no sense spending the time to get your areas defined and then have them be “wrong” because the camera moved. You can reset the registration mark whenever you wish. To some degree your Areas of Interest will be automatically adjusted to reflect the change. However due to optics, camera angles, etc. there is only so much On Guard can do to adjust the areas correctly.

### An Important Concept – The Working Set

On Guard displays a list of images called the “working set”. This is all images for the current camera at the time On Guard starts. You can only update this list by pressing the “Refresh” button. You will **not** see new images taken by the security camera without a refresh or a restart. The purpose of the working set is to allow you to review images and use them to create Areas of Interest. Working with a constantly shifting set of images is just too distracting and is more confusing than it is helpful.

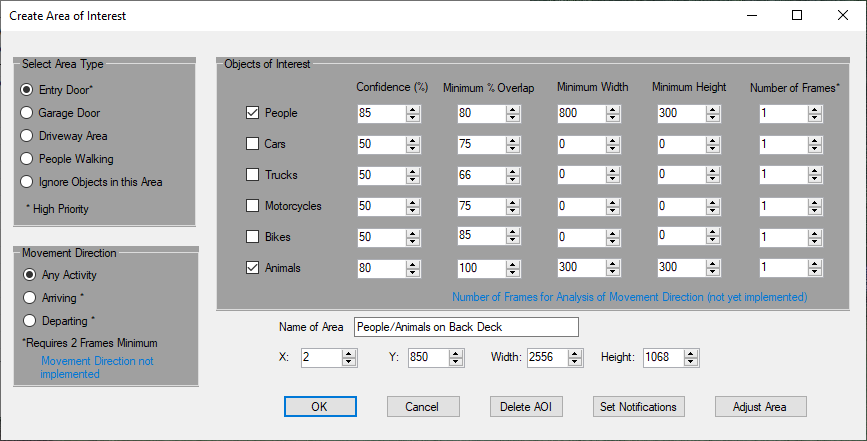
### Red Rectangles!

Objects that the AI has identified and that On Guard recognizes as potentially of security interest are outline with a red rectangle if “Show Object Rectangles” is turned on. The object is also listed in the Objects list in the lower left of the screen. However, this **does not** mean that On Guard considers it to be part of an “Interesting” motion object/event. An Interesting event is one that may trigger a notification that results in a video being recorded or an email notification is sent. In some pictures there may be multiple objects identified by the AI. Some may be interesting, and some may not. If an object is not outlined then the AI has not recognized it, even if you can. An object must be within an Area of Interest to be interesting.

## Areas of Interest

Now, comes the useful part. An **Area of Interest** (zone) is defined by rectangle. Using combinations of rectangles you can almost always define areas that are important for the camera. At the upper/left point of a rectangle that you want to identify as an Area of Interest **right click the mouse button**. A small semi-transparent area is shown on the screen. You can click on the center of the area and drag it anywhere on the screen. You can also “grab” any of the edges and corners to change the size of this area. Once you are satisfied that the area covers the area you are interested in press the “F1” key. This is the “I accept the area as shown” button. If you want to cancel the area definition press the “ESC” key. Note that while you are identifying an area all other buttons (right/left/Go To/etc.) are disable as is the main menu. While you are identifying an area the tools area at the bottom of the screen is shown with a dark background indicating that everything there is disabled. Also the title of the main screen shows you a hint that you may press the Esc or F1 keys to continue.

Your area is now identified.



Once you area is outlined a dialog box window will appear. This dialog allows you to specify whether you are interested in people, cars, trucks, motorcycles, bikes, and/or animals. Specify what type of area this is. Note that Doors are considered a “priority“ area. If you wish to ignore motion in any area, select Ignore Objects. The other options are Driveway, Garage Door, and People Walking. Except for “Doors” the area types are only for classification.

There is also a box labeled Movement Direction. The options in this box are reserved for future use. In conjunction with the Number of Frames box in the future you will get more information when notified via email. For instance, cars arriving may be more important than cars leaving. The Number of Frames box will be used determine something like: “Looking that the last 5 frames there was movement toward the front door”. In the still longer term we may be able to determine something like “A truck arrived followed by a person moving toward the front door. Was this a delivery?”

Name the Area of Interest. This name will be displayed in emails for motion events. It is also used for editing an Area of Interest.

Note that once defined by the mouse the area size/location can only (currently) be altered via settings on this page (X, Y, Width Height). This is an area for future enhancement.

The Objects of Interest section allows you to further specify what you are looking for within an area. Normally you care about people vehicles, and possibly animals.

The Confidence box allows you to specify just how certain the AI is about the identification of an object. DeepStack is usually very good about identifying people with a decent range of the camera (Often 99% certain). On the other hand the AI may not be so confident about vehicles. For instance it may know that there was some sort of vehicle movement, but it may not know if the vehicle was a car or truck. You might get a 50% value for a vehicle as a car and a 60% confidence as a truck. In reality there may be a 99% chance that there was some sort of vehicle.

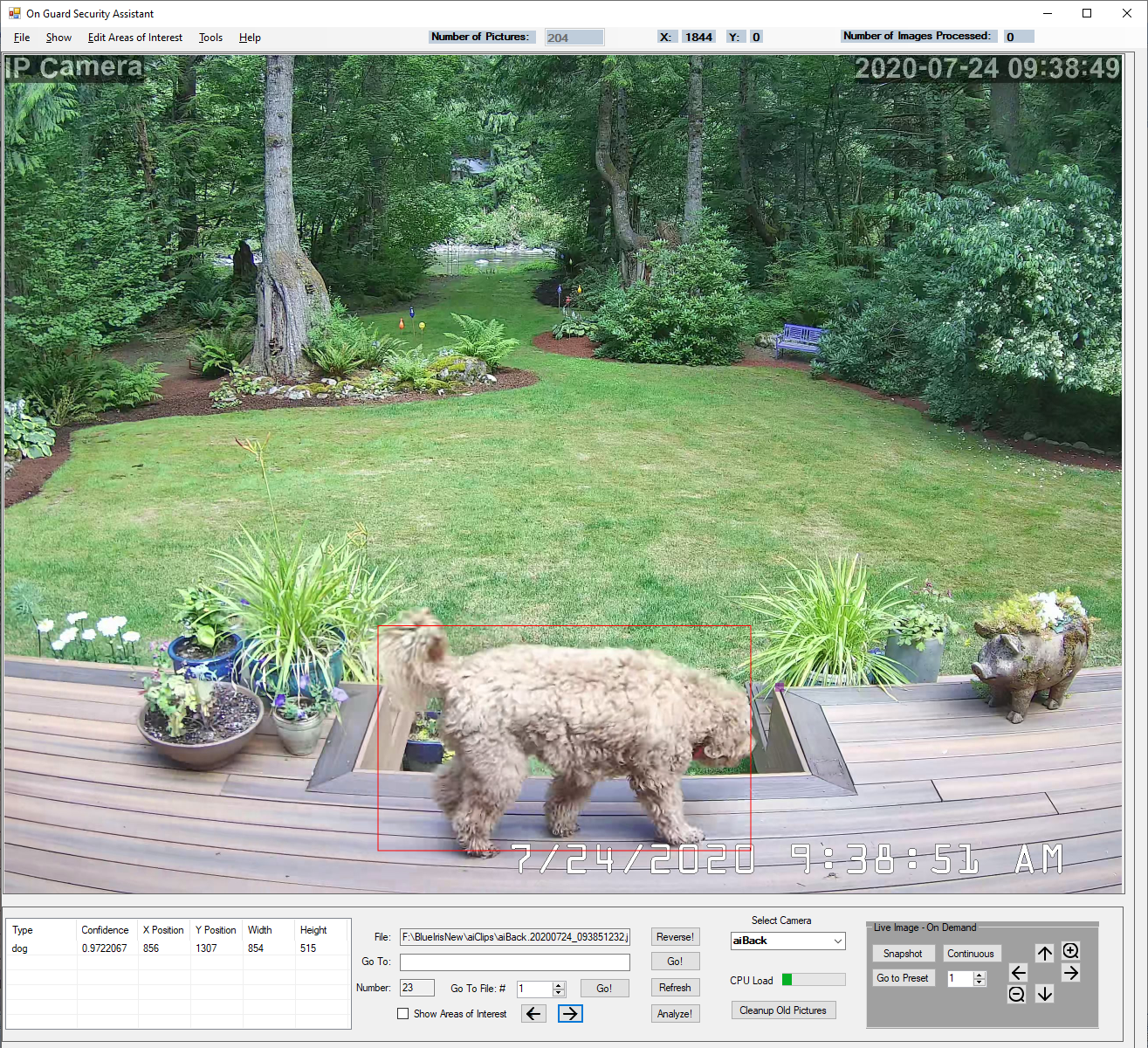
Because of the above described problem in confidence of vehicle type there is a **special rule for vehicles**. Normally On Guard just uses the confidence value returned by DeepStack. However, if there are two (or more) objects identified by DeepStack as any sort of vehicle **and the position of the vehicles overlap by 90%** On Guard fudges these values. On Guard assigns the object type as the object with the highest level confidence (car 66%, truck 52%). In addition, we bump up the priority of the highest confidence by half the difference in confidence between the two values. In the case shown we determine that the object is a car, and we bump up the confidence that it is a car by 7 to 73, that is: (66 – 52)/2. Keep this in mind when you enter a confidence level here for any vehicle. This is designed to give you a more realistic confidence setting. Otherwise, for vehicles you may tend to set the confidence levels too low here.

With respect to confidence levels let experience be your guide. You can also use the Analyze button in the UI to help you make those decisions. Keep in mind that the confidence levels at night may be lower than they are during the day.

The Overlap option is very important. For instance, a car/person may be 60% within an area (defined by the overlap of the object rectangle to the area. The Overlap box allows you to define areas that might be partially in one area and partially within another. This can also be very helpful for two or more areas that may be on top of each other. Again, let experience be your guide.

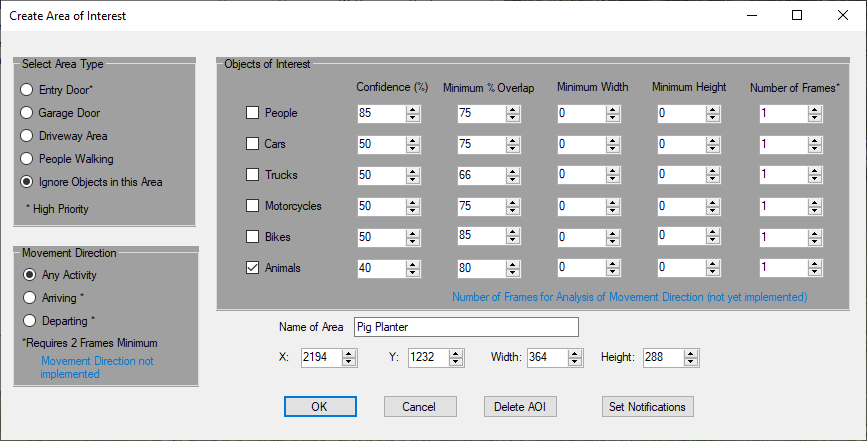
The Minimum Width and Minimum Height boxes are also very important. The same object closer to a camera will appear larger than those further away. Through creative use of the size values you can obtain a quasi 3D location of an object within the picture. In conjunction with the Overlap value you can usually have a fair degree of certainty whether the object is “interesting” in terms of security. Obviously, there are variations in the size of people and objects, and that will affect the determination of the object’s location.

In the “back deck” picture below the people/animals on the deck will be much larger than the people on the lawn. By defining these sizes you can easily have overlapping Areas. In that example people both on the deck can be in the same picture location close up and in the distance. People further away (smaller) can be ignored while larger pictures of people may trigger an email alert.



As an aside, note the three decorative glass blubs in the distance/left (near the big tree trunk). Due to the shape the AI sometimes (not in this picture) thinks that they may be a “person”. However, it generally isn’t too confident that it is a person (say 41% certainty). They are also small/far away, and so aren’t likely to be of interest anyway. The AI is far from perfect, but it is very useful with the proper area definitions.

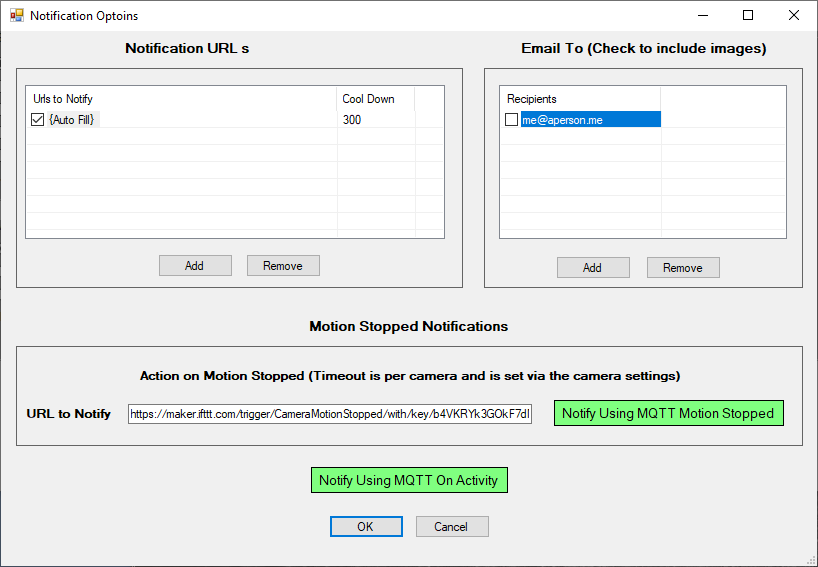
### Ignoring Objects in an Area



If you decide you want to ignore objects within an area you still need to define the type of objects you want to ignore in the area. You probably should define percentage overlap for objects in the area. Object size may also play a role in ignoring objects. At a minimum the object type must match for an object to be ignored. Note the “pig planter” in the photo above. At times the AI may think that that planter is a dog (this varies by the picture). So, the “pig” is defined as within an ignore area. It is also defined as type “animal” an overlap of 80%.

The Adjust Area button allows you to go back to the main screen and visually adjust the affected area. Note that once you have accepted the changes (via the F1 key) you will need to use the Edit Areas of Interest menu item to further define options in the area. It does not automatically go back to the area definition screen. Generally it is much easier/faster to make all area definitions first before you adjust the area.

### OK, I’ve Defined an Area – What Now?



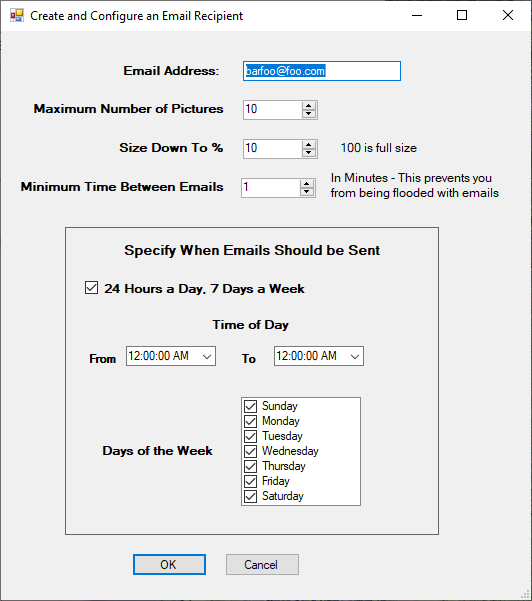
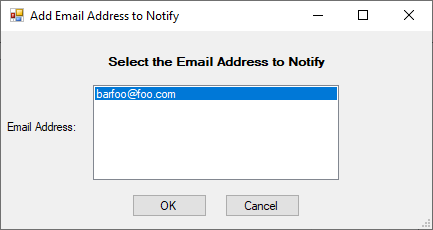
If you haven’t chosen to ignore objects in an area you now need to decide how/if you want to be notified. The two notification primary modes are email and via a message sent to an Internet address. Emails **can include SMS emails/text with pictures sent to your phone, tablet, or computer).** The “Web” address (URL) can be (and usually is) the address of Blue Iris on the same computer. However, there is no restriction on who/where you want notifications “sent” – it is just an address. To define notifications press the “Set Notifications” button.

URL (web addresses) are typically used to trigger the Blue Iris camera to begin recording. First, press the Notifications button from the Area definition. Then press “Add” under the URLs box. Press “Auto Fill” to trigger the Blue Iris camera. For any other Internet destination “just” enter the URL.



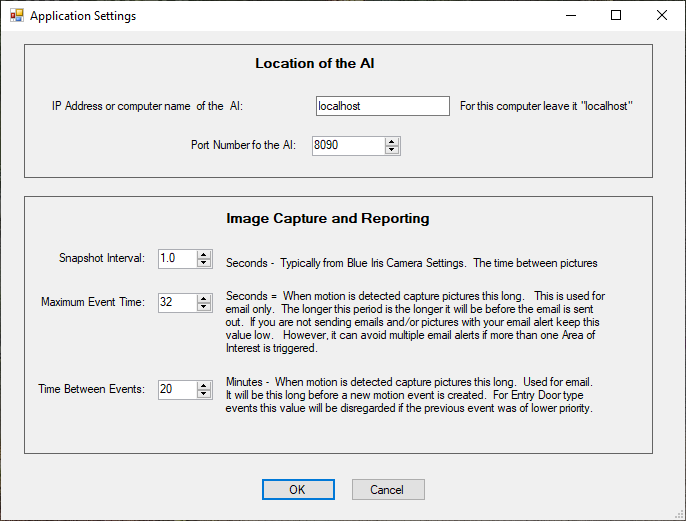
For URLs (Interne addresses) you can also enter the “cool down” time between triggers. This limits the number of videos triggered to a reasonable level. Taking too many video hurts nothing except free disk space so this time is defined in seconds.

Email notifications are a little more complicated because an email can be intrusive (particularly if it is an SMS/text email address). Therefore, when you define an email address (From Tools/Options/Add Edit Email Addresses) you also specify the time of day, day of week, and cool down time for the camera. You also specify the size of the pictures you want sent with an email. Here in the Notifications, however the only thing you get to select is the actual email address you want notified.



### Getting a Clear Picture of Email Notification/Pictures

In the preceding picture you will see an option for the Email Recipient Maximum Number of Pictures. This is a hard and fast maximum limit on the number of pictures you will see in any email notification. Depending on your Blue Iris settings and the duration of the motion you may see fewer emails. Going back to the Application Settings page you will see other settings that affect the number of pictures.



Note the Maximum Event Time setting. This is how long on Guard will wait after the start of the motion to accumulate photos to send. In the example above On Guard will wait 32 seconds (always) before considering an “event” completed. There may be 32 pictures in that interval, there may be only 3.

Once the event completion time expires On Guard will go through all pictures taken in that period. First, it selects “priority” pictures. Priority pictures are those defined as type “Door”. If there are any slots left over in your Maximum Number of Pictures it will then go through the remaining pictures. At this point On Guard selects pictures that have objects (people, etc.) that meet the definitions of other defined areas (such as people in the driveway). Finally, if there are still slots left pictures that fail the previous selection criteria are added. All pictures are shown in time order. So, you may have: 3 pictures of people in the driveway, 2 pictures of people at a door, 4 pictures of people in the driveway, and one picture of someone leaving/not in a defined area.

Another factor to note is that On Guard makes every attempt to avoid bombing you with unwanted email. However, it also doesn’t want to miss an important event. Thus, we need to take a look at the Minimum Time Between Emails setting. Let’s say someone is on the lawn casing the joint. On Guard send an email since she was in an assigned area (Lawn/People Walking). Let’s further say that she decides that it is worth grabbing that nice box on your porch and darts in 10 seconds later. Normally, On Guard would not send a second email since it has only been seconds between the first email set and the motion in the door area. However, since your porch is in a defined area (People at my door/type Door) things are a bit different. Since the previous email was not defined as Door, and the new motion is in a Door area another email will be sent (with attached pictures). The same thing would **not** occur if the first motion was of type Door, and the second motion is of type People Walking or even of type Door. For example, someone walked in your door, and 50 seconds later people were walking on the lawn another email would not be sent for the lawn event. Therefore, be careful to assign type Door to important/protected areas.

But wait, there’s more! In the Application Settings page there is a setting Time Between Events. This settings is a global interval between “events”, and events can lead to emails. The Minimum Time Between Emails is based on the email recipient. The Time Between events is for all email recipients, and for all areas. However, the exception of “Door” priority events still applies. This is yet another attempt to avoid the email spam problem. Let’s assume that you want your phone email to be notified rather rarely (say only cars pulling into your driveway, people at your door, and event that only every 20 minutes). However, it may be less intrusive to send an email more often to your home desktop computer. Thus the Minimum Time Between Emails value may be much higher for the phone. Nevertheless, you really don’t want to come home to 76 emails on your desktop either. Time Between Events setting will prevent this. However, the value shown above (20 minutes) is probably very high in most cases.

## When Motion Stops (Version 1.4)

When defining notifications for an area you can specify what to do when motion stops. You send an URL to a web site and/or you can send an MQTT message. Motion is considered stopped, and motion stopped events are sent when there has been no motion detected for a specified period of time. That period of time is set on a per camera basis since objects may cross multiple areas. If motion happens during that time period then the period is started over and no stopped events are generated. On the other hand, whether or not motions are sent for an area is determined by the area notification options.

Set the time period per camera on the Camera Settings Motion Timeout tab. That period is set in seconds. If you wish to notify a web site via an URL, enter it in the area notifications area. If the URL is completely empty nothing will be sent. If there is anything entered, a web notification will be tried. An invalid web site will result in an error. To send an MQTT message press the notification window button “Notify Using MQTT Motion Stopped” button. It will turn green when MQTT is selected.

## The Motion Only Button (\*New in Version 1.1)

The Motion Only button allows you to go from one picture in which interesting motion has been detected to the next picture with such motion. This can bypass hundreds or thousands of pictures in which no interesting motion has been detected.

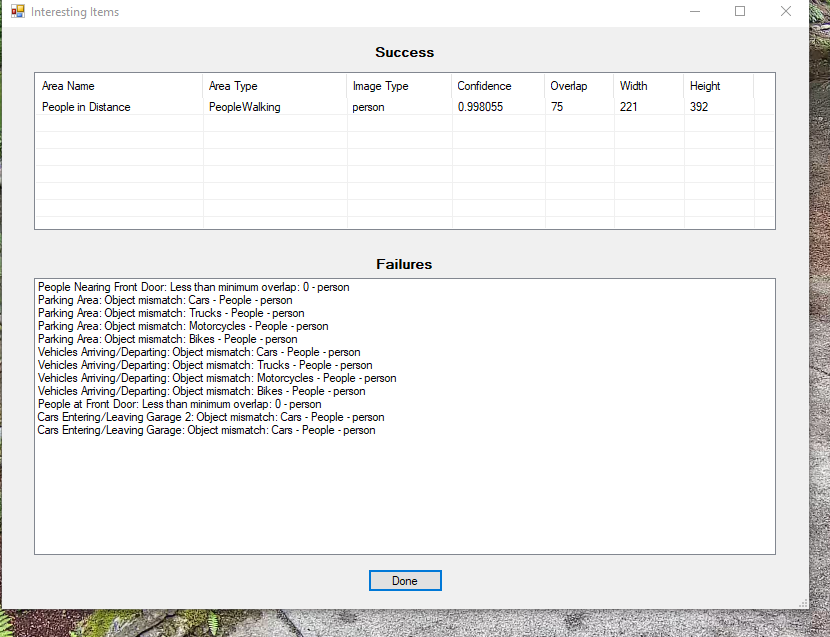
**BUT! …** The Motion Only button only recognizes motion that has been detected while On Guard was running. This is because as new motion is recognized the name of the picture is stored in a database. If On Guard wasn’t running it couldn’t add those pictures to the database. **BUT!....** If you are browsing through pictures **with “Show Object Rectangles” turned on** On Guard will add a picture to the database if it is considered interesting. This is done (1) because we can and (2) because all the work of recognizing interesting objects has already been done. In theory On Guard could browse all pictures in the working set, determine if they are interesting, and add them to the database. However, this would take an enormous amount of computing power if there are a large number of unprocessed pictures. So, the more you browse the more pictures are added to the database.

If you delete pictures in your camera directory (via Blue Iris, Cleanup Old Pictures, or manually) the database has no way of knowing this. Therefore, the pictures may be gone but still in the motion database. If On Guard tries to go to the next interesting picture (the Motion Only button is down) and can’t it find that picture, then the picture name will be deleted from the database. This all happens behind the scenes, so you won’t normally notice this if there are only a few pictures missing/deleted. However, if a lot (as determined by your computer) of pictures have been deleted, the movement forward or backward may be slowed somewhat as the missing pictures are removed from the database. Again, the more you browse (with Motion Only down) the faster things will be in the future.

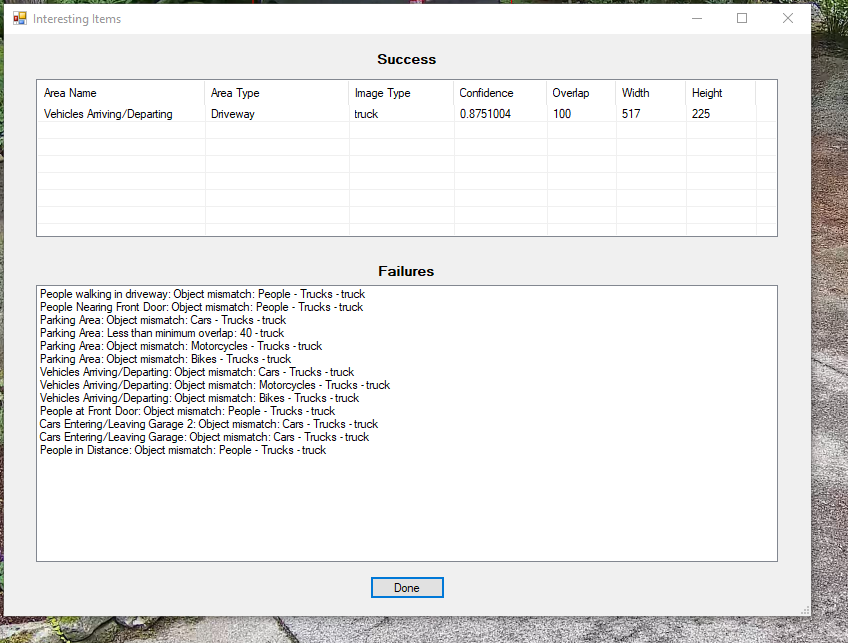
## The Analyze It Button

If you are defining a complicated series of areas it may not always be clear why On Guard chooses to define an event as “interesting”, or not. Go to the image in question using any of the navigation buttons, and then press Analyze It button. See a sample analysis below. Note that On Guard goes through each area defined (there are a number). It shows the result (someone walking in the distance). It also lists the reason another area wasn’t selected. It shows the object type mismatches (person vs. car/truck/etc.).

This is an essential tool in defining your areas. Note (not shown) that the determination of whether or not to send an email vs. just recording a video depends on the area this person is in. Since he is a person in the distance an email is not sent. However, a video recording is started.



Let’s look at another case:



In this case the identified truck (we are 87% certain it is a truck) is in the Driveway and in an area defined as Vehicles Arriving/Departing. It is not in the parking area with only a 40% overlap. If it were in the parking area we’d ignore it. Since it is in an important area we do in fact send an email with 10 photos attached.

Note that there often many reasons why an event is not interesting (worthy of a notification). First, the AI may not have identified it. The AI may not have enough confidence. The object overlap may not be as high as the area requires. Areas defined as “Ignored” are a special case. First, the object found is analyzed. During the first pass of analysis it may “pass” all requirements. Then, there is a second pass to compare the object overlaps to any ignored areas. If the object is then found to overlap the ignored area by the percentage required and it is of the correct type, the object is ignored.

## Boxes and More Boxes!

Moving back to the main screen ---- In the bottom left of the main screen there will be a list of all objects the AI has identified on this stored picture. This includes the size and position of the objects. You can use see the AI’s confidence that it is correct in identifying the object. The size information can help you set the minimum size of objects in the Area of Interest dialog. In the sample picture below there is one object of type “dog”. The AI is 98% certain it is a dog (although sometimes the AI thinks she is a “sheep”, and who can blame it really - no Covid haircut).

### Viewing stored pictures

At the bottom/center of the screen you are given a variety of ways to navigate to stored pictures. This includes Forward/Back, Go To a specific picture file name, and Go To a file number (in the stored list). The list of pictures is presented in an oldest to newest order by default. If you want to change that press the “Reverse” button.

### Viewing Live Camera pictures/video

The bottom/right of the screen allows you to look at live or snapshot views from the camera. The Snapshot” button is self explanatory. The “Continuous” button will feed you a live stream from the camera. In addition there are arrow buttons to move Pan/Tilt cameras and Zoom cameras with that capability. You can also press the “Go to Preset” button will allow you to move to camera preset position (also depending on camera capability). Note that the Preset button waits five seconds before displaying another “live” image. This is to give the camera time to move to the preset. This may not be enough for all cameras and all positions.

### Camera Selection

You may have multiple cameras. However, you can only see images from one camera at a time. The pictures displayed are there for the primary purpose of creating Areas of Interest. If you want an application that can display multiple cameras at once, please use Blue Iris. You can select the current camera from two places. The first is under Tools/Camera Settings. The more convenient method is by selecting the camera from the dropdown box at the bottom mid-right of the screen.

### CPU Digestive Problems

The current CPU load is shown near the camera selection drop down. If you are running the DeepStack application on this computer it is important to note the CPU load when there is continuous motion. Normally the CPU load will spike to 100% (all green) while there is motion, and maybe for a while (maybe much longer than a while). Motion frames are kept in a queue/list. They are parceled out to the DeepStack AI as CPU resources are available. This is done to make sure that your computer is not so overloaded that it becomes non-responsive. As an aside, note that it is **not** a good idea to run a screen saver or other background task on the computer with DeepStack running.

A backlog in processing by DeepStack may mean that On Guard will trigger Blue Iris so that your video clip will start “late”. Therefore, you may also wish to set the Blue Iris pre-trigger video buffer (Blue Iris - Camera Settings/Record tab) to something in the range of 10 to 30 seconds. This enables Blue Iris to look back to a time before it told by On Guard to start recording. This will almost always allow you to see video of the event that triggered the video.

It is also suggested than when testing motion activity that you take a look at the DeepStack output/black window. It shows you how long DeepStack is taking to process frames. If you see the time as excessive you might be forced to reduce the rate that Blue Iris outputs motion capture frames. If you have a slow/older computer it may, in fact, take DeepStack 5 seconds to process just one frame. Time for an upgrade! However, you could tell Blue Iris to only take 1 picture every 5 seconds.

### Cleanup of Old Pictures

Motion in front of your cameras may result in an accumulation of a **lot** of old pictures, particularly outdoors. These can consume significant disk space. The Cleanup Old Pictures button will allow you to delete pictures before the specified time interval. If you are sure you don’t want these pictures, start pressing this button. Note that only pictures for the current camera are deleted (this is a change in versions 1.3+).

On Guard does not automatically cleanup old pictures because these may provide a good backup of important motion events. However, Blue Iris can/may delete them depending on how it is configured. It can, in fact, be less that straight forward preventing Blue Iris from doing so.

Please be aware that some anti-virus software may prevent On Guard from deleting pictures. Your anti-virus software should have a means of excluding On Guard from this protection feature. Also, note that you may need to re-do this exclusion if/when On Guard changes version number or gets reinstalled.

When sending email with attached pictures you can specify that the pictures are sized down on a per recipient basis. On Guard does create temporary pictures that it automatically cleans up after an email is sent. In the unlikely event On Guard dies when attempting to send emails you might see some pictures starting with the name “resized”. Your email client may display the name of the resized pictures as email attachments.

### Information at the Top of the Screen

There is a lot of data to show, and limited screen area to show it. Therefore, at the top of the screen (in line with the menu selections) you should see the following. The “Number of Pictures” is the number of pictures you can select from with the navigate buttons. This number does **not** change even if there is motion detected. Pressing the bottom “Refresh” button will load any new pictures into the working set.

The top also shows the X and Y location of the mouse cursor. Note that this is in picture/pixel coordinates. This is not the absolute mouse position on your screen. Depending on the resolution of your camera you may see the mouse positions number jump no matter how careful you are in moving the mouse.

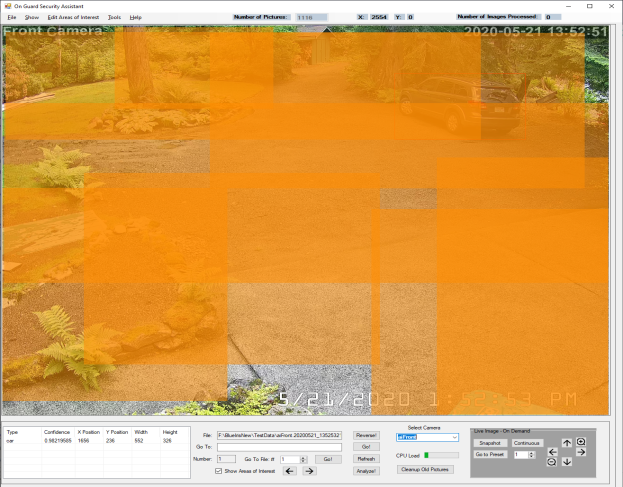
The Number of Pictures Processed keeps a running count of the images processed by On Guard since you started it. If this number seems excessive you may want to check the sensitivity of the Blue Iris motion detection properties. Outdoor cameras tend to have a large number of images because lighting conditions changes may result in “motion”. Also, rain, snow, and bugs can cause motion to be detected.

You will also see the rather cryptic areas labeled X Res and Y Res. This is the number of pixels in width and height contained in the camera snapshot used for analyzing pictures. This is not the width and height of the image displayed on your screen. It is used for reference purposes. Note that the Test Image photos will not have the same width and height as your normal camera pictures. Ideally On Guard does not care about the width and height of the image. Your areas should all be in the appropriate area relative to the top and left edge of the display window.

## The Show Menu

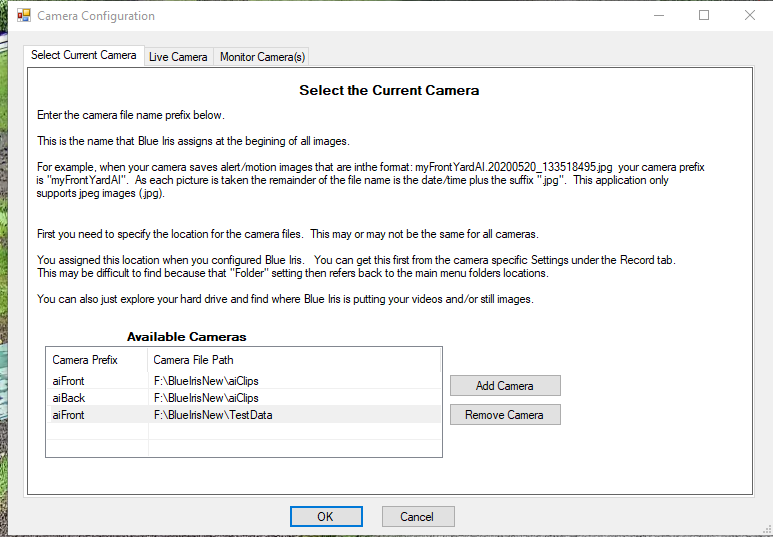
Under the Show menu there are 2 options. Show Object Rectangles turns on/off processing of pictures through the AI. Because the AI can be slow, moving through stored pictures can be slow as well. Turning off Show Object Rectangles is a good way to look through pictures until you see something interesting. Then turn on Show Object Rectangles to see if the AI can recognize anything. You can use the control + “s” key as a shortcut to this menu option. Note that with Show Object Rectangles option on that you can experience the CPU use/DeepStack processing time for just one picture by moving forward/backward within the picture working set.

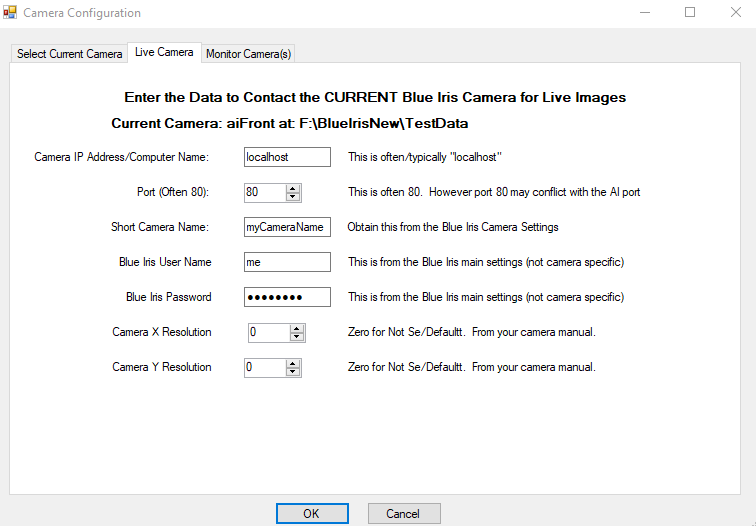
Show Areas of Interest turn on and off the display of areas you’ve defined. Seeing these areas can be important when tuning On Guard to report/record things of interest to you. However, it can also be annoying because these areas partially obscure the pictures. Turn it on and off via the menu or via the control + “a” key. Note the number of areas defined in the picture below.

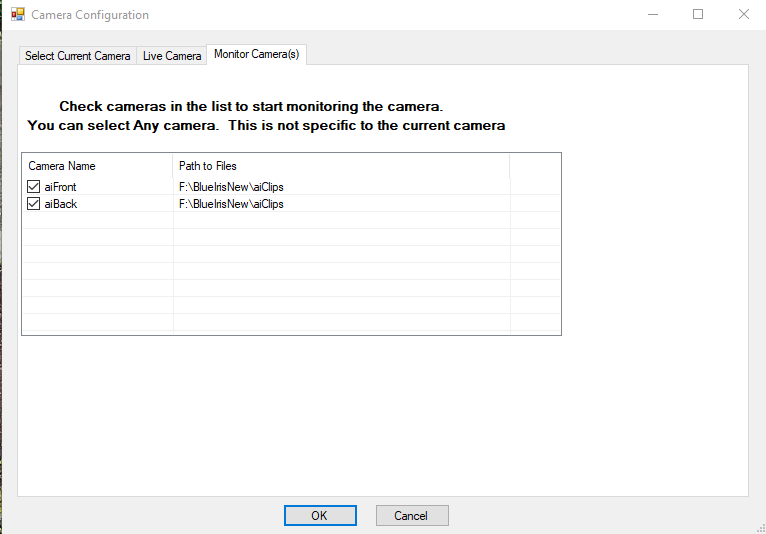


## The Tools Menu

The Tools menu contains the various application related settings. These include Application Settings (original, I know), Camera Settings, Email Server Settings, Email Addresses, and MQTT Settings. Note that the Camera Settings dialog has 3 tabs. These are for Adding/Deleting cameras, defining live access to the current camera, and turning on or off monitoring of the cameras. Sorry if the three tabs are confusing. Otherwise there were just too many dialogs.







### MQTT

MQTT is a means of communication used by many Internet of Things devices. MQTT can be used for such things as turning on a light when motion is detected. On Guard now supports the publishing of motion events to MQTT. Events are published in the format that you define. There is a list of tags surrounded by braces {}. The application will fill in the value for the tag. For instance, the tag {File} will be substituted with the image file name that generated a motion related event. The content of that publication is the file name of the picture triggering the alert.

In order to use MQTT you must first setup the MQTT server address and port. Optionally, you can enter your MQTT user name and password. You can also specify whether the published event will sent to the server via a secure (TLS) link. You either define the topic names and payloads that you want sent or you can just leave the default topics and payloads. Note that anything other than pre-defined tags will be sent exactly as you enter it.

For each area that you want to publish an MQTT event, you must set a flag (button) in the area of interest Notifications page. If the MQTT button is green for that area an MQTT event will be published if you have also setup MQTT under the tools menu.

If you don’t know what MQTT is, don’t worry. It is strictly optional. However, it can’t hurt to do a Google search and find out about it.

### Test Images

Clicking the Test Image menu will cause a series of photos embedded in the application to be sent to the camera directory for all cameras. These images are somewhat random (public domain/internet), and may include people and/or vehicles. This feature helps ensure that On Guard is working properly. If you have areas defined it is more than likely that at one or more of these images will trigger any notifications you’ve defined for your areas.

Note that you can always feel free to use Windows to drag and drop pictures into your camera directories. The only limitation is that these pictures must start with your camera prefix for the camera to recognize it. This just automates the process.

Note that you can get rid of these pictures (and all your others) by using the Cleanup button with a time of zero days and zero hours.

## Special Notes on Object Recognition/Vehicles

Parked vehicles are no longer considered objects when determining motion. For instance, if a car is parked in a driveway and a cat wanders by causing Blue Iris to trigger a series of motion pictures, the car will be filtered out. If you were looking for cats then you would get a notification. If you were looking for cars you won’t. This is new as of Version 1.2. In previous versions just the fact that a car was there would trigger notifications relating to cars. They will still be shown as objects in the image review screen of the main window.

On a different but related matter, occasionally DeepStack might not be sure what type of vehicle you are seeing. You might get 2 (possibly more) outlines around on real life object. It might be considered both a car, a truck, and possibly a bus. All the outlines for these objects might differ slightly. They might all have different confidence levels.

In these cases we now attempt to consider these DeepStack objects as one object (based on the overlap between them). We now select the object type with the highest confidence level. In addition, since DeepStack might not have been sure what **kind** of vehicle it was seeing so the confidence level of any one interpretation is artificially low. Therefore, the On Guard confidence level of the selected vehicles is bumped slightly. Note that this may create problems when multiple vehicles are close together and the camera angle is “just right”. Note that this feature has not been tested in depth since the magical test data for this evaporated. Please report problems.

## Help/About

Under the Help/About menu you can find the all important picture of the fierce mascot of On Guard standing guard over her domain! There is also the obligatory license verbiage. The application is free, but for your personal or small business use. Use it at your own risk too. The license terms may become less restrictive as the project matures.

## Help/Log File

On Guard logs activity and errors to a file called "OnGuard.txt.". When you select Help/Log File an **external** application will be launched to display this file. The specific application that will display your log file is dependent on the application you have set in Windows to display “.txt” files. By default this is Notepad. Note that this file is not update in real time. To see any changes you will need to close the file and re-open it.

## Help/Log Detailed Information

By clicking this menu item you turn on/off the collection of **detailed** log information. All detailed entries (those that normally are not save) are preceded by: (Trace). This detailed information may help in tracking down hard to find problems with On Guard or with your area definitions. However, the amount of detail provided may be annoying.

## Help/Delete Log File

The log file can get quite large over time. Detailed information makes it grow even faster. This menu item allows you to delete the log file. The deletion is tried up to 5 times. If the log file is in heavy use this may fail. Just try again later when things have settled down.

## Help! Where Did On Guard Go?

Let’s face it. Once you get On Guard up and running you really don’t want it cluttering up your Windows task bar or your screen. Normally, all active Windows applications are shown in the task bar. When you minimize On Guard it hides itself. It may look like On Guard has crashed (that never happens)! Instead a (very small and difficult to recognize) icon shows up in your System Tray (in the general vicinity of the clock). To restore On Guard double click the icon. BTW, anyone interested may contribute a better icon, and it will be gladly accepted. The disadvantage of relocating the icon to the system tray is that there are often more icons in the tray that can be shown at once. You may need to open up the tray to see it by clicking the “^” or “<” button in the system tray to see On Guard listed.

## Turning Lights (and other things) on and Off

### MQTT

There are a variety of options for using detected motion to turn lights (or any other “smart” devices) on and off. First, go to the area Notification Options screen. If you have devices that are MQTT compatible, you can simply press the Notify Using MQTT on Activity button. To turn the device off when motion stops (set at the Camera Settings screen) press the Notify Using MQTT Motion Stopped button.

You can also turn devices on and off using HTTP/URL notifications. Just enter your URL in the appropriate boxes. Sounds easy! The question now becomes, what URLs should you use? Try IFTTT.

### IFTTT (If This Then That)

IFTTT is a very popular web utility that can be found at IFTTT.com. Unfortunately it is not quite as easy as MQTT. Many inexpensive light switches are IFTTT compatible. Further, Alexa is IFTTT compatible. A variety of switches can be found in the $20 - $40 price range. Just try an Amazon search for IFTTT switch. It is beyond the scope of this document to fully describe getting setup with IFTTT, but here are some hints that worked as of 1/2021.

The first thing you need to do is setup an IFTTT account. Add your device/switch by brand or in accordance with the device instructions. Now comes the considerably less than obvious steps. To turn a device on/off or otherwise trigger a device you need to create a service of type “Webhooks”.

1. Click “Create” to create an action. It will bring up: “**IF** This (Add)”.
2. Click “Add”. It will bring up a set of icons for actions.
3. Type “Webhooks” in the search bar that comes up next.
4. Click on the “Webhooks” Icon.
5. Click “Receive a Web Request”
6. Give your Event a name.
7. Click “Create Trigger”
8. Select the device and device action you want (in my case SmartLife (the switch device name)) plus , “Turn On” – The action I wanted to happen.). Let there be light!
9. Click “Continue”
10. Click “Finish”
11. Go to “Home”, then “My Services” under your account (the head icon)
12. In the “My Services” scroll down to “Webhooks” and click on it.
13. Click “Settings” for “Webhooks” (gear icon)
14. You will see a section called “URL”. Copy the section of the URL after “use/” (apparently random characters, but they aren’t). This is your secret key. Keep it for use with other URLs! It does not change.
15. Create your URL. It will look something like: [**https://maker.ifttt.com/trigger/YourActionNameGoesHere/with/key**/**b4VKRYk3GOkabcdefghijk**](https://maker.ifttt.com/trigger/YourActionNameGoesHere/with/key/b4VKRYk3GOkabcdefghijk). The **red part** up to “trigger/” and “/with/key/” is always the same. Substitute **your action name in the green part, and the secret key you copied in place of the stuff after** “**key/**”.

Yes, that could be much, much easier. However, **after** the first time (and you have save your secret key) it actually goes reasonably fast (5 minutes or less, guaranteed).

# Appendices

## Thanks To:

This application was inspired by Gentle Pumpkin and his work “Free AI Person Detection for Blue Iris” at <https://ipcamtalk.com/threads/tool-tutorial-free-ai-person-detection-for-blue-iris.37330/>. On Guard has taken his basic idea of using Blue Iris and DeepStack to enhance security camera results. On Guard offers some enhancements. Gentle Pumpkin offers some features On Guard does not. In particular he offers notification via Telegram. He has put a lot of effort into explaining exactly how to setup Blue Iris at the link above. These setup steps have been copied directly and/or paraphrased. Again, Thanks!

## Setting Up Blue Iris for Use with On Guard

### A Note on the Need for a Second Hidden/Virtual Camera in Blue Iris

The original “Free AI Person Detection for Blue Iris” tool required the use of a second Blue Iris virtual/normally hidden camera (not an actual physical camera) to generate still .jpg pictures for use in triggering a second camera that takes the actual video. In is my understanding that that tool has migrated away from that mode so that only one camera need be setup.

On Guard was a spinoff from the AI Tool, so in part the use of a second virtual camera on Blue Iris was related to the original concept. However, at this point there are no plans to migrate to any other model, and a second camera will be required in Blue Iris for the foreseeable future.

There are good reasons for maintaining the second camera model. The primary reason is that On Guard is compatible with any IP/network camera that can output still images to a directory. Blue Iris is not required. This includes any camera that can output motion files via FTP. Almost any decent IP/network camera made in the last 5+ years or so can do that with the proper setup. Don’t get me wrong. Blue Iris is a great tool!

The second major reason is that On Guard operates on a different theory. Its job is **not** just to trigger video. While it does do that, it can be used to capture and display motion related file. It is also intended to do things such as (not yet implemented) determine the direction of movement (toward/away from a door, into or out of a driveway). In order to do that it needs the history of the movement.

Finally, On Guard was designed from the ground up to allow for the easy definition of Areas of Interest (zones). As you layout a zone, you need the history of recently captured images to compare those images to that zone. You can look at a picture and say “yeah, that should have triggered a recording, why didn’t it”. Using the Analyze button you can tell exactly why it did or didn’t on any one frame. You can then easily adjust your area settings.

### Setup Steps (sorry, some of these need updating!)

1. Setup Blue Iris and open it in Administrative Settings. Ensure that videos are recorded when motion occurs just to double check everything.
2. **Create or find an 'Input Path' folder:**​ We need an directory where Blue Iris stores all the images possibly containing alerts. You can add a path to Blue Iris by opening the settings of Blue Iris, then 'Clips and archiving', then click on one of aux folders in the list on the left (if you click on p.e. 'Aux\_7' and don't move the cursor for 1-2s, you will be able to change the displayed name). Then create a new folder in the Blue Iris main directory. We can name this folder for example "aiinput". We can furthermore limit the folder size to for example 100Gb, so that old images are automatically removed.​
3. **Enable URL triggering feature in Blue Iris**​: This is the method we use to trigger a video clip when an interesting event occurs. URL triggering is disabled by default, so to be able to trigger a camera in Blue Iris via URL, you have to do the following in Blue Iris:​

**A**: Go to Setting->Webserver->Advanced and disable 'use secure session keys and login page'.​

**B**: Go to Settings->Users and enter and select a user and copy the password, or create a new administrator user. The credentials will be later to make the trigger URL.​

1. **Duplicate a camera**​: Now we have to create a camera **duplicate** whose only purpose is to save a images when a motion is detected. Add a new camera, give it a name that makes sense (e.g. if your original camera was called 'frontyard', call it 'aifrontyard'), and under type select 'copy from another camera' and choose the appropriate one.​
2. **Disable unnecessary stuff in the new/duplicated camera**​: Keep in mind that this camera’s only job is to detect motion and then save a still image into the folder we created in step 2 so disable all features on this camera that are not needed (recording, pretrigger, etc). Because Blue Iris is already prepared to work with camera clones it is not necessary to lower the resolution to save on CPU resources. Quite the opposite: If the camera stream URL isn't changed, there will be zero additional CPU usage. Instead, changing the stream URL to a profile with a lower resolution will cause additional CPU load.​ Additionally you can go to the 'General' tab and check 'Hidden', which will hide this duplicate camera from the Blue Iris UI (otherwise you suddenly have twice as many cameras as before). This is really useful, as it keeps your Live View page tidy.​
3. **Store alert images in 'Input Path'**​: Go to Record, check 'JPEG snapshot each (mm:ss)', select the folder you created in step 2. Check the box 'Only when triggered' and set the interval to 0:02.0 (one image every 2 seconds). You may wish to alter the interval value depending on your experience. A reasonably fast computer can keep up with an interval of 1 second.
4. Go to 'Trigger', check 'Capture an alert list image' and set the Break time 'End trigger unless retriggered' to 10 seconds, so that a causes 5 images to analyzed. If you set the snapshot interval to 1 second (do so if your computer can handle it) then reduce this value to 5 seconds. Note that this differs from Gentle Pumpkins recommendations due to additional processing with On Guard. Make sure that Object Detection is off under Motion Detection/Configure. If it is on you may miss important frames.
5. **Disable motion detection for original camera:** Finally, we have to disable motion detection and other triggers on the original camera ('frontyard'), so that nothing except On Guard triggers the original camera. To do that we open the camera settings of our original camera, go to 'Trigger' and uncheck all boxes in the 'Sources' area.​ It is also suggested that you check the box for “pre-trigger buffer” and enter at least 10 seconds. This will ensure that you see the beginning of the motion if On Guard is backed up processing images.

## Installation Checklist

1. Follow the steps for setting up Blue Iris.
2. Get camera/cameras prefix (Blue Iris Short Camera Name).
3. Get the AI Camera output directory path (setup step 2 above)
4. Get live camera name (usually the Blue Iris camera name)
5. Get live camera IP Address and Port (usually from Blue Iris) Often: localhost
6. Get live camera user name and password (usually from Blue Iris step 3B above).
7. Get your email server address (usually smtp.xxx.net-or-.com).
8. Get your email server port (preferably the SSL port which may be different from the default port 587). You can usually get look this up online or get it from your Internet Service Provider. Try port 587 first.
9. Get your email server user name and password. The user name is usually your email address.
10. Download and install the DeepStack Windows (or Linux) application. Start it using the port you decide to use (port 80 is **not** a good idea).

## Installation Note (Database)

As of version 1.1 On Guard installs a database service from Microsoft called “LocalDB”. This provides support for storing the list of interesting/motion related pictures. Therefore, you may notice this installation process during setup. It is normal, and nothing to worry about. All files are stored locally. Nothing is ever **sent** over the Internet. That said, Setup does use the Internet to **download** the necessary support files.

## Installation of DeepStack as a Windows Service

**This information is not directly applicable to the DeepStack “Beta” released 12/20.**

Let’s say you are tired of seeing the DeepStack window on your desktop, or you want a little bit better performance analyzing images. You can do this by installing DeepStack as a Windows service. If you are not familiar with Windows services, think of them as background applications that run without a user interface/window.

In particular, DeepStack is more than a little annoying because the Windows desktop application does not have a “minimize” button (why?). That means it can frequently/usually is in the way. Note that experience has found that running DeepStack as services can speed up image processing a little, sometimes up to 10% (depending on your computer). If you have a lot of motion on your cameras this savings can add up.

It is strongly suggested that you start off by running DeepStack as a Windows desktop application. Once you are comfortable with the application you can get it out of the way. That said, installing it as a service is not necessarily straight forward even for people comfortable with Windows. This is made worse by the fact that you need to install 3 separate services.

A good tutorial on installing DeepStack as a background services can be found here: <https://ipcamtalk.com/threads/tool-tutorial-free-ai-person-detection-for-blue-iris.37330/page-58#post-471089>. It should be emphasized that by default a Windows service does not have access to the motion images. This is a Windows security feature. It is relatively easy to get around this by (1) setting up the services to log on with your user name and password, or (2) giving the service access to your files. The first alternative is considerably easier. So, under the services application “Log On” tab click the “This account” button and enter your user name and password. Yes, this is secure. If you install DeepStack as services and you aren’t getting objects identified, this is one of the first things to suspect.

You also need to be very careful that the port value in the server.exe service is set to **your** port number. For (no particularly good reason) On Guard uses port 8090 by default. The desktop version of DeepStack uses port 80 (this is **not** particularly good since other applications like Blue Iris tend to use port 80 as well).

The first thing you need to do after installing it as services is to use a browser and in the address bar type in “localhost:yourPortNumber” (yes port=8090 by default) . You may need activate DeepStack depending on the version you are installing. After you press Enter the browser will either say “DeepStack Activated” or it will ask you to activate it. If you need to re-activate you must find the email from DeepQuest AI with the activation code. If you can’t find that email you will need to ask for another one or get it in some other way. If the browser does not show one page or the other you probably are using the wrong port, so this is a good double-check on your services setup.

It is possible that On Guard itself could be setup as a service so that you don’t need to have the window on your desktop. In future versions that may be possible. However, since you can minimize On Guard easily and it magically disappears from your task bar this isn’t a priority.

## Using localhost

In several of the settings for On Guard the default address for your local PC will be shown as “localhost”. This should work in 90+% of the cases. However, there have been reports that “localhost” does not work with **some** On Guard computers. If “localhost” does not work it is suggested that you try using an IP address of “127.0.0.1” (no quotes). If that still does not work you may need to use the IP (V4) address of your computer. This can be found in several spots in Windows. One of the easiest is to type cmd from your Windows Start Menu. Then, type “ipconfig” (no quotes, plus return). You will see a listing that may be somewhat cryptic you should see something that says something like: (IPv4 Address. . . . . . . . . . . : **192.168.0.2**) . That numeric part with dots should be the local address of your machine. You can use that in place of localhost wherever you find it. Note that you **do not** normally use anything ending with “.1” like: 192.168.0.**1** because that refers to your local router. Also, note that the address of your computer does not necessarily start with “192.”, but it often does on home devices.

On Guard does not currently support IPV6 addresses. That may change, but for now in spite of IPV4 limitations, IPV4 is widely understood and the addresses are relatively easy to enter. If this is a priority for anyone please leave feedback regarding this.

## Running DeepStack using An NVidia Graphics card

The GPU version of DeepStack uses the processing power of the compatible video cards to offload the workload from the main computer CPU. You could go from analyzing one frame per second to 10 frames per second! That is, with many/most Nvidia video cards you can cut the processing time per frame up to 90%. Also, this frees up the CPU to do other work on your computer. This is relatively big deal because On Guard can/should/must process multiple frames in order to do its work.

The latest Windows version of DeepStack (at this point a Beta version at: <https://github.com/johnolafenwa/DeepStack/releases/tag/0.1-beta> (as of 12/24/2020)) adds support for running the GPU (NVidia card) version of DeepStack. This version is much faster than the previous Windows version. However, the normal Windows (CPU) version is **also much** faster than previous versions as well. If you have an NVidia graphics card it is suggested that you try the Windows GPU version first.

If you are a Linux fan you can run the DeepStack AI on Linux. Linux with NVidia cards also supports the GPU version of DeepStack. On Guard doesn’t really care where DeepStack is located as long is there a network connection. It is possible that the On Guard application will be supported on Linux in the future (via .NET Core to get technical). If there is interest please say so and the priority of this may be bumped.

## Bugs and Feature Requests

Yes, there are bugs! Hopefully it works well enough to be useful.

On Guard is provided on GitHub along with the C# code and build environment. If you spot a bug fix it! Just (mostly) kidding. Please report bugs via Github. The more community interest there is the more likely it is that bugs will be fixed and features added.

A word of apology: Yes, the User Interface is using the plain old ugly Windows Forms environment. This was the fastest way (for me) to code the project. It may be ugly, but it is functional. If the pandemic lasts much longer (now 1/21) maybe there will be time to make it pretty. However, that comes after exploring new features and alternative AI engines, so it isn’t happening soon.

## Bugs Fixed (Not up to date!)

Version 1.1.0

1. When On Guard starts or you press the Refresh button a list of the currently available pictures is loaded. If you keep On Guard running and delete pictures outside of On Guard (via Blue Iris or manually) then the list that On Guard keeps and reality won’t match. This could cause problems while browsing pictures so that On Guard stopped running or shutdown. This problem has now been fixed.
2. The Outgoing Email Server “Test” feature has been fixed.

## New Features (Not up to date!)

Version 1.1.0

1. Added the ability to browse just “interesting” pictures via the “Motion Only” button.

Version 1.2.1

1. Added the ability to publish MQTT on qualifying motion.
2. Added the capablity to screen out parked cars from being picked up as moving cars. Cars that haven’t moved significantly will no longer be reported as in motion.
3. Bumped up the confidence level on overlapping vehicle for instance when the same object is reported as both a car and a truck we choose one and bump the confidence that it is a car somewhat. Otherwise vehicle confidence tended to be reported as too low.
4. Parked/Unmoving vehicles of all types are now filtered out of the found objects list for the purposes of determining motion. They will still be shown within the UI as a vehicle. This prevents false notifications when a car is parked and some other object causes motion.

Version 1.41.

1. MQTT can now be setup to provide custom messages.
2. Motion Stopped events can be sent via web site URL notification or via MQTT.
3. The data files containing settings have changed in format. The location of those files have also moved. Old camera/area definitions are no longer compatible with older versions.
4. Email notifications now included the object type causing the motion for each area.
5. There is better error handling. At least errors are displayed differently!

Version 1.4.7

1. The old picture cleanup process now works in a background thread. Thus, you can continue working while pictures are being deleted. However, once the background process completes the working set will be refreshed.
2. You can now delete the log file or increase the log details from the help menu.
3. The processing of pictures has changed slightly to track down bugs where “interesting” pictures were not being processed correctly. This is a work in progress.
4. There were some unknown circumstances where area definitions were occasionally moved way off the screen. This caused a breaking bug for the area. Now, areas cannot be move left/up more than 5,000 pixels off the top/left of the screen. If you see your X and/or Y positions at -5,000 please try to report what seems to have caused the problem.
5. The X and Y resolution of the picture are shown at the top of the screen. This refers to the underlying picture, not what is displayed. Every effort has been made to ensure that your area definitions are resolution agnostic. In a picture of taken of the same scene at different resolutions your area should be almost identical. Please report any problems.