Audion Cars Counter — user guide

Contents

- What you need
- Starting and installing
- The pictures and the folders
- The short way: from pictures to a report
- How the window is laid out
- PICTURES
- VIEW
- COUNT
- A net for your own pictures
- SETTINGS
- Keys
- What accuracy to expect
- Where things are
- What the program does on the network and on the machine
- If something goes wrong

The program counts passenger cars on satellite pictures that are tied to the ground: how many stand on each picture and on the whole set, where each of them is, and how dense they are by district. The result is a number with its bounds, a Word document, an Excel sheet and layers for MapInfo.

The count runs on your machine. The pictures are not sent anywhere.

The program counts with a taught net. The public release carries one, `open-03`: it was taught on free aerial photographs of three open state services and on nothing else, and a new set of pictures is counted with it at once. Your pictures are another source than the ones it has seen - before a count is believed, examine the net on a few patches marked by eye. A net of your own is taught on your own pictures together with an AI agent, or a net file you were given is put into the program: see "A net for

your own pictures". What a net taught this way is worth is said in "What accuracy to expect" - read it before you use a number.

What you need

- Windows 10 or 11, 64-bit.
- An NVIDIA video card with driver 570 or newer is the fast way: in the author's work a picture of 165 megapixels (12490 x 13253) took about a minute on an RTX 5070. Without such a card the count runs on the processor: the count is the same, but such a picture took about twenty minutes on a laptop processor.
- The internet during installation, and disk space: 0.7 to 3.4 GB are downloaded; the installed program takes about 7 GB on a machine with an NVIDIA card and about 3.5 GB without.
- The pictures: a folder of `.jpg` files, each with its MapInfo raster registration `.tab` beside it (in detail: "The pictures and the folders").
- A taught net. The public release carries one, taught on free aerial photographs; a net of your own is taught on your pictures ("A net for your own pictures").
- MapInfo only if you want the layers for a map; the count and the report do not need it.

Starting and installing

Unpack the program folder anywhere and start `Start.exe`. The program keeps everything of its own in this folder. One thing goes into Windows: the Microsoft Visual C++ Redistributable PyTorch needs - and only when it is not there.

At the first start the counting engine is not there yet: the window opens the **SETUP** section by itself and asks whether to install it. The question says that without the engine the program shows the pictures but does not count, how many gigabytes will be downloaded for this machine, and what will be installed into Windows. Nothing is downloaded until you answer **DOWNLOAD AND INSTALL**; after a refusal the button **INSTALL WHAT IS NEEDED** asks the same:

1. The **THIS MACHINE** card shows the processor, the memory, the video cards and what follows from them: whether the count will run on an NVIDIA card or on the processor.
2. Press **INSTALL WHAT IS NEEDED** and confirm. Under the name of the button stands what this machine lacks. As its first step the program looks whether Windows has the Microsoft Visual C++ Redistributable and, when it has not, installs it with winget - Windows asks for the administrator's consent. Then it puts its own Python 3.12 into the folder `runtime`, PyTorch for this machine (with CUDA on an NVIDIA card, for the processor otherwise) and the libraries.

3. The installation is shown in the journal, which opens by itself. It can be stopped with STOP and started again.
4. When everything is installed, the program removes the downloaded installation files, and the **WHAT IS INSTALLED** card says: "Everything is in place: the count can start". In a copy without a net it says instead: "The engine is installed. No taught net: nothing to count with yet", and a row of the card reads "No taught net". That is not a fault of the installation: a net is still to be taught or put in - see "A net for your own pictures".

The other buttons of the INSTALL card repair one part: **THE WHOLE ENVIRONMENT** installs everything anew when the engine does not answer; **PYTORCH FOR NVIDIA** is the count on a video card (about 3 GB); **PYTORCH FOR THE PROCESSOR** is for a machine without an NVIDIA card (about 0.7 GB). Each asks before it downloads.

If the NVIDIA driver is older than 570, the section says so, and the count runs on the processor. Update the driver and press **CHECK AGAIN**: the section then shows that PYTORCH FOR NVIDIA is to be installed.

The pictures and the folders

Which pictures the program takes. A folder of `.jpg` files; beside each lies a MapInfo raster registration `.tab` of the same name. The name of a picture file ends with the column and the row of the picture in the set: `…_2-3.jpg` is the second column, the third row from the top. By these numbers the program lays the pictures out on the map of the set; a single picture is named the same way, `…_1-1.jpg` for example.

From the `.tab` the program takes the size of the picture, the size of a pixel on the ground, the area and the coordinates; it takes the units of the `.tab` for metres. The author's net was taught on pictures of about 0.3 m to a pixel. The program only reads the pictures.

Where to put them. Two ways:

- *Beside the program.* The program looks for a data folder beside its own folder: `Audion Counters Data\Cars Counter`, or `Cars Counter` when that one is not there. Put the pictures into a folder of any name inside `Pictures` of it - `Pictures\MyTown`, say - and the work folder `Work` appears by itself. Nothing has to be set. When `Pictures` holds several such folders, the one counted is the one `Work` was made on - in a new data folder, the first by name; any other is named in SETTINGS.
- *In any folder.* In the SETTINGS section name the folder of the pictures (the PICTURES card) and the work folder (the WORK FOLDER card); an empty work folder will do.

The automated checks of the program went the first way; the folder dialogs themselves were not exercised by them.

The work folder is everything the program makes from the pictures: the counts, the tiles for viewing, the points set by eye, the nets taught on them. It belongs to one set of pictures: the program refuses it for other pictures. Another town, or another shooting of the same place, is another set with a work folder of its own; the numbers of two sets are never added up.

The short way: from pictures to a report

1. SETUP, **INSTALL WHAT IS NEEDED** - once.
2. Put the pictures beside the program or name their folder in SETTINGS. If the window was open meanwhile, press **REREAD**.
3. A net - once, when the program has none: teach it on your pictures together with an AI agent, or put in a net file you were given ("A net for your own pictures").
4. COUNT, **COUNT THE PICTURES**.
5. Click a picture in the BY PICTURE table and look in VIEW at what the marks stand on.
6. COUNT, **REPORT ON ALL PICTURES**; for a map - **LAYERS FOR MAPINFO**.

How the window is laid out

- **At the top** - the name of the program with its version; the three sections of the work, in order: PICTURES, VIEW, COUNT - under the name of each is its state (how many pictures, which one is open, how many are counted); on the right - the language (RU | EN), the dark and the light theme, the font size.
- **Bottom left** - SETUP, SETTINGS and ABOUT; pressed again, a button goes back to the section of the work. **Bottom right** - the **REREAD** button: read the set again when the pictures, the points or the count were changed outside this window.
- One row above - the **F1 DOCS** button (this guide), the messages of the program, the progress of a long operation with its **STOP** button, and the **JOURNAL** button: everything the program does and the output of the engine.
- Every button and card has a detailed tooltip: rest the mouse on it.

PICTURES

The map of the set: every picture stands in its place by the column and the row of its file name. A place shows the name of the picture (2–3); when the picture has points set by eye - how many patches and cars are marked; when it is counted - "net:" and the number of marks. The tooltip of a place tells the size of the picture in pixels, the size of a pixel in metres and the area. A click opens the picture in VIEW.

The small copies of the pictures for the map are made once, when the set is first read; for a set of 31 pictures that is under a minute.

On the right:

- **THE SET** - the totals: how many pictures, megapixels, what area, how much is marked and which nets there are; the **PICTURES FOLDER** button.
- **ANSWERS OF THE NET** - whose count the map and VIEW show ("The count of the net:") and from which sureness an answer of the program is shown as a car ("Threshold of sureness:"). This card is the workplace of whoever checks and teaches a net. The COUNT section does not depend on it: it always shows the count at the sureness the net was accepted with.

VIEW

One picture. In the middle - a piece of it, large; on the right - the whole picture and the PATCH card; on the left - the tools for setting points and the layers. A picture of 165 megapixels is not loaded into the window whole: when it is first opened the program cuts it into tiles, once - about ten seconds and about 100 MB in the work folder for a picture.

- The piece moves in steps along the two axes - by the arrow keys or the arrow buttons: looked through in a row, nothing is left unseen. A click on the whole picture brings the large view to that place.
- The size of the piece is chosen by buttons: **CLOSE** (120 x 120 pixels), **PATCH** (240 x 240, about 75 x 75 m), **2 × 2**, **4 × 4**. Around the piece a strip of the neighbouring ground is shown, dimmed.
- **PREVIOUS** and **NEXT** (Page Up, Page Down) lead over the marked patches.
- **TO THE MAP** goes back to the PICTURES section.
- Under the buttons stands where the piece is, how many marked cars are in the frame and how many answers of the net.

The **LAYERS** card - what is drawn over the picture:

- **ANSWERS OF THE NET** - the marks of the program: blue rings, beside each how sure the program was, in hundredths;
- **MARKS** - the points set by eye and the frames of the marked patches: a green dot is a car, an orange square a truck or a bus, a yellow ring - "cannot tell";
- **LIGHTEN** - lightens the picture, so that dark cars in the shadow of a house come apart from the shadow. It does not change the count - only what is seen in the window.

On the whole picture, green squares are marked patches, blue dots are marks of the program, the orange frame is what is shown large.

To check a count by eye no points are needed: open a counted picture, switch the layer ANSWERS OF THE NET on and step through a few places - a car park, a yard, an industrial site, a forest.

Setting points by eye

In this section MARKS are the points a person sets on cars by eye. A net is taught on them and checked against them. To count with a net that is already there, nothing has to be marked.

A patch is a square of 240 x 240 pixels marked **completely**: every car has a point, and whatever else lies in the frame is ground for the net - a place without a car. So a patch once begun is to be gone through whole.

Tool (key)	What a click does
LOOK (1)	sets nothing
CAR (2)	a point on a car: a car, an off-roader, a van, a pickup. The point goes on the middle of the car
TRUCK (3)	a truck, a bus, a trailer, machinery: not counted, and the net is not taught that it is ground
CANNOT TELL (4)	a dark shape in shadow, half under a tree - a car or not: not counted, the net is not taught on it
ERASE (5)	a click on a point takes it away

A click on a point with another tool changes its kind, with its own tool takes it away; the right mouse button erases with any tool. The first point in a square of the grid makes that square a patch.

The buttons of the patch work when one patch is shown (the size PATCH):

- **PATCH WITHOUT CARS** writes the patch as a marked one with no car on it: for roofs, forest, a cemetery - whatever the program takes for cars;
- **TAKE THE NET'S ANSWERS** sets a car point on every mark of the program in the patch, as a draft: what is false is then taken away and what is missed is added;
- **UNDO** (Ctrl+Z) brings the points of the patch back to what they were before the last change;
- **REMOVE THE PATCH** takes the patch out of the marks altogether. The file is not erased but moved into the folder `labels\_removed` of the work folder.

Every change is written into the file at once.

How to mark:

- in doubt set CANNOT TELL: a wrong point teaches worse than a missing one;
- a point goes on the middle of a car, not on its nose; in a row of parked cars count along the row;
- dark cars in the shadow of a house are there: switch LIGHTEN on and look for the pairs of light dots of the glass;
- not cars: gravestones in rows, vents and hatches on a roof, garden beds and greenhouses, stacks of pallets, boats, the roofs of sheds and garages, single light tree crowns.

COUNT

On the left are the **COUNT** and **REPORT** cards, on the right the **BY PICTURE** table; only the table scrolls.

COUNT THE PICTURES counts every picture of the set that is not counted yet: the program goes over each picture and puts a mark on every car. Pictures counted before are not counted again: after new ones are added only they are counted. The number of a picture appears in the table and on the map as soon as it is counted. STOP at the bottom of the window stops the count: what was counted stays, and the next start goes on with the pictures not counted. While a count runs the computer does not fall asleep by itself.

Without a net the button is dimmed and the card says: "No taught net in the work folder or in the program: nothing to count with." See "A net for your own pictures".

The large numbers of the COUNT card:

- **marks of the program** - how many places on the pictures the program took for cars;
- **cars, within these bounds** - from ... to ..., see "How to take the number";

- **pictures counted**;
- **km² covered** - the area of the counted pictures;
- **marks per km²** - the density.

The **BY PICTURE** table is the same for each picture: marks, cars from and to, the area, the density per km². A click on the name of a picture opens it in VIEW with its marks.

How to take the number

A mark of the program is a place on a picture the program took for a car. The number of marks is not yet the number of cars:

- at this resolution a person cannot tell many dark blobs either - a car or a shadow - and the program marks some of them;
- it misses some of the cars in shadow and in dense rows;
- single marks stand on trucks and on things that are no car at all.

So the number is always told with its bounds: **cars from "the marks less a quarter" to "the marks plus a half"**. A thousand marks is from 750 to 1500 cars. One number without its bounds is not to be told.

The density is marks per square kilometre of the area covered; districts are compared by it: built-up land gives hundreds of marks a square kilometre, forest and water a few.

A picture shows the cars under the open sky at the minute it was taken.

The report: Word and Excel files

The **REPORT** card writes the result into a folder you name:

- **REPORT ON ALL PICTURES** - on every counted picture;
- **REPORT ON THIS PICTURE** - on one picture, the one open in VIEW;
- **REPORTS FOLDER** - opens the folder of the last export.

There are two files, both written for a transport engineer, in the language of the window.

The Word document ("Cars count report ...") begins with the main figures - pictures, marks of the program, cars at least and at most, the area, marks per km² - and the paragraph "How to take the number". Then come the layout of the pictures as they lie on the ground, with the number of marks on each; the table by picture with its total; and a page a picture: the picture with a dot on every mark and

eighteen marks taken at random and enlarged, so that one sees what the program took for a car. The last page is "How to read the report".

The Excel sheet ("Cars count sheet ..."): the sheet "Overview" - the table by picture with its total (a picture with points set by eye has the number of cars checked by eye beside it); the sheet "Sensitivity" - how many marks would stay at another strictness of the program; the sheet "Marks" - every mark with its place in the picture and its coordinates on the ground, to lay over a map; the sheet "Service" - what the pictures were counted with, for whoever maintains the program.

The name of a file carries the date and the time, so a new report does not replace an earlier one. A report counts nothing anew - the numbers are taken from the count, so it can always be written again. A set of 31 pictures gives a document of 46 pages, about 15 MB, in 20 seconds.

On the page of a picture look at the sheet of samples. If many of them are not cars - roofs, gravestones, glints on water - the number of that picture is not to be trusted: see "What accuracy to expect".

Layers for MapInfo

LAYERS FOR MAPINFO writes the count for a map in the coordinate system of your pictures' registration - it is taken from their `.tab` files, so the layers lie over the pictures as they are.

Before the export choose the **Grid square**: 100, 250, 500 or 1000 m. Smaller - single car parks and yards show, but the map flickers; larger - districts show. The place of a mark does not depend on it.

File	What is in it
`Cars_density_raster_….png` and `.tab`	the density map as a picture with a registration of its own: MapInfo opens it at once (File > Open). A square is coloured by its marks; white - covered and empty, grey - no picture there
`Cars_marks_….mif` and `.mid`	every mark as a point: the picture, how sure the program was, the place in the picture
`Cars_grid_250m_….mif` and `.mid`	the grid of squares: the number of the square, marks, cars from and to, the share covered by pictures in percent, the area of the covered part, the density per km^2
`Cars_pictures_….mif` and `.mid`	the outlines of the counted pictures with their totals
`Cars_density_map_….png`, `Cars_density_map_dark_….png`	the same map with its scale of colours and a caption: on a light ground for reports and letters, on a dark one for a screen and slides

File	What is in it
`Cars_layers_readme_….txt`	what the layers and the columns are

The name of the grid file carries the chosen side of the square, and every name ends with the date and the time of the export. With the window in Russian the names of the files and of the columns are Russian. The `.mif` layers are the interchange format of MapInfo: Table > Import, choose a `.mif` file (the `.mid` of the same name must lie beside it).

- The squares of the grid are tied to the ground, not to the pictures: two exports with the same side of the square have the same squares, and they compare one to one.
- A square on the edge of the set is covered by pictures in part: the density is by the covered part, and the column "Covered_pct" says how much that is.
- Empty squares are in the grid: "no cars here" is an answer too.
- A square with two or three marks may be empty and one with a hundred a car park: districts and blocks are judged more safely than a single square.
- A picture without a `.tab` file is left out of the layers.

What is not checked. The layer files were read back and checked by their numbers, but they have never been opened in MapInfo itself. If an import does not go through, or the layers lack something, that is the first thing to put right in the program.

A net for your own pictures

The public release of the program carries the net `open-03`. It was taught from a random start on marks set by eye on free aerial photographs - 2334 cars on 22 photographs of three open state services (the Netherlands, Estonia, Germany), 0.31 m a pixel - and on nothing else. On three photographs of those services it was not taught on it counted +2 % against 296 cars marked by eye (92 % of them found); on satellite pictures of a fourth source it had never seen, +1 % against 511 cars (89 % found). Those are its numbers on those pictures. Yours are another source: a net that holds on four may still fail on a fifth, so mark a few patches and examine it first - the order below begins with that.

A net of your own comes in one of two ways.

- *It is taught on your own pictures*, together with an AI agent - the order is below. The taught net lies in the work folder of the set and counts that set.

- *A net file you were given* (`<name>.pt`) is put into the folder `system_core\nets` of the program and named in the file `config\counting_default.json`: `{"net": "<name>", "least": 0.20}` - the name of the file without `.pt`, and the sureness from which a mark is counted, as whoever gave you the net names it. If the window was open meanwhile, press **REREAD**, and in the SETUP section **CHECK AGAIN**. A net that lies in the program counts every set that has no accepted net of its own: a new set of pictures is counted at once. A net you taught yourself is put there the same way, to count other sets with it.

A net knows the pictures it was taught on - one town and one source, as a rule. On another source, or on ground of a kind it has not seen, it has to be examined first, and when it fails the exam - a net is taught on your pictures. The window has no buttons for that: teaching a net and checking it are commands of the engine. The points can be set by yourself in the VIEW section, or left to an agent.

The order is this (in detail, with the commands: the guide for the agent):

1. **An exam of what there is** - when there is a net. Count two or three pictures with it and look at samples of the marks; mark a regular grid of patches by eye on two or three pictures and set the count beside the points. Without a net there is nothing to examine: the work begins with the next step, and every point is set by eye.
2. **Check pictures.** A few pictures are marked only for checking, and a net is never taught on them: a check would otherwise show what the net has already seen.
3. **Points for teaching** - on other pictures: a regular grid and the places where a net goes wrong (car parks in shadow, dense rows, yards; roofs, cemeteries, water - as patches without cars).
4. **Teaching.** Four nets are taught on the same points and joined into one file: one teaching differs from another by itself (by up to 13 % of the count on the same points), and four together answer more evenly. For you it is one net - one file, one name. On an RTX 5070 the four teachings take about ten minutes; on a laptop processor one takes about two and a half hours.
5. **The check of the new net** on the check pictures, two tables: is the error the same from place to place, and does the count hold when a picture looks different - lighter, softer, pressed harder.
6. **A net that is accepted** is written into the work folder of the set, and COUNT THE PICTURES counts with it. A net that lies in the program, if there is one, stays where it is.

What we learned:

- what decides is not the length of the teaching and not the size of the net, but the points: how many there are and how varied;
- more points of the same kind add nothing beyond a certain amount: teach what the exam failed;
- ground of a new kind gives false marks by the hundred until it is marked as patches without cars;

- when the pictures are blurred in parts, no teaching gives a steady count on them.

Working with Claude and Codex

The program is made to be worked on together with an AI agent - Claude Code or Codex - the way its authors worked on it. The guide for the agent is `Docs\AI_AGENT_GUIDE_EN.md` (in Russian: `AI_AGENT_GUIDE_RU.md`): the commands of the engine, the file formats, the order of the work and what has already been measured.

How to start. Install the program (SETUP, INSTALL WHAT IS NEEDED), name the folder of the pictures, open the program's folder in Claude Code or Codex, and ask in your first message to read `Docs\AI_AGENT_GUIDE_EN.md`. Then speak plainly:

- "There is no net yet: lead me to the first one; you may look at the pictures" - or, when there is a net, "Examine how the net counts my pictures; you may look at the pictures";
- "Mark check patches by a grid on pictures 2-3 and 4-1, I will look through them in VIEW";
- "Teach a net for these pictures and tell me whether its error holds";
- "Count the whole set with the new net and write the report into such a folder".

About the pictures. The agent works on your machine, and the pictures themselves go nowhere. But what the agent looks at - a reduced picture to choose patches on, enlarged patches to mark, sheets of samples - goes to the service it runs on (Anthropic for Claude, OpenAI for Codex). Satellite pictures usually come with their supplier's terms of use: check them. The agent is to ask your leave before the first picture. If the pictures must not be shown, set the points yourself in the VIEW section - the files are the same, and the agent needs only the commands: it can teach, check and count without looking at a picture.

SETTINGS

Where the data lie (the button at the bottom left, between SETUP and ABOUT). Every kind of data has a card of its own. Pictures and reports can be had again - the pictures copied anew, a report written once more. A net has no card: one taught on your pictures lies in the work folder of their set, beside the points it was taught on; one that came with the program or was put into it lies in the program itself.

The **PICTURES** card - which pictures are counted. **BY DEFAULT** - a folder with pictures inside `Pictures` of the data folder beside the program: the one the work folder `Work` was made on, and in a new data folder the first such folder by name (or `Pictures` itself, when the pictures lie straight in

it). **IN THE USER'S FOLDER** - any folder of pictures: another town or another shooting; the **CHOOSE A FOLDER...** button names it. The program only reads the pictures.

The **WORK FOLDER** card - where what is made from the pictures lies: the points set by eye, the counts, the tiles of the viewer, the nets taught on those points. **BY DEFAULT** - the folder `Work` in the data folder for the default set, and for another set a folder of its own beside it, `Work-<the name of the set>`. **IN THE USER'S FOLDER** - a folder you name; an empty one will do: with a net in the program the count can start at once. Every set of pictures has a work folder of its own: given another set's folder, the program says "made on other pictures" and does not count. The program never erases the points set by eye.

The **REPORTS** card - where the Word and Excel reports and the layers for MapInfo are written: the export buttons of the COUNT section open the choice of a folder at this place. Until a folder is chosen, the first export asks for it.

The **ENGINE** card - where Python and the counting engine lie and whether they are found. **CHECK** asks the engine which folders, which PyTorch and which video card it sees; the answer is in the journal.

The **OPEN** button of a card opens its folder.

The folder dialogs themselves were not exercised by the automated checks of the program. If a dialog does not behave, the folders can be named in the file `config\settings.json` with the window closed: the keys `"Pictures"`, `"Work"`, `"ReportFolder"` (a backslash in a path is doubled).

Keys

Keys	What they do
Ctrl+1 ... Ctrl+6	the sections in order: PICTURES, VIEW, COUNT, SETUP, SETTINGS, ABOUT
Ctrl+J	the journal
F1	this guide
Esc	from SETUP, SETTINGS and ABOUT - back to the section of the work
← → ↑ ↓	VIEW: one step aside
Page Up, Page Down	VIEW: the previous and the next marked patch
A, M, L	VIEW: the layers - the answers of the net, the marks, the lightening

Keys	What they do
1 … 5	VIEW: the tools - look, car, truck, cannot tell, erase
Ctrl+Z	VIEW: undo the last change of the marks

What accuracy to expect

Every number below comes from the author's work: the program counting with the author's net on the set it was made on - satellite pictures of one town from one source, about 0.3 m to a pixel. That net is not part of the public release. The numbers are what a net taught this way gave: what to expect of a net you teach and something to compare it with - not a promise, and not a property of the program you have.

- **On pictures the net had not seen.** On three check pictures 511 cars are marked by eye for certain; with the author's net the program put 509 marks. By kind of ground: a plant with a large open car park -2 %; residential ground (a settlement, a village, a works) +12 %; an old town under long shadows 0 % - but there a quarter of the cars is missed and as many marks are extra. They cancel; elsewhere they may not.
- **When a picture only looks different** - lighter, softer, pressed harder - the count moves: by up to 4 % on open car parks and by up to 16 % in the old town under shadows. Such a difference is to be expected between the strips of one mosaic.
- **A mark is not yet a car.** The number is always told with its bounds: from "the marks less a quarter" to "the marks plus a half". The bounds are fixed in the program and come from the same check pictures; they are not worked out anew for your pictures or your net.
- **Another source of pictures.** On pictures of the same town from another source - of a lower effective resolution, blurred in parts - the author's net found about a third of the cars. A new source or a new kind of ground has to be examined first and is usually taught for: see "A net for your own pictures".
- **Ground a net has not seen** - a cemetery, roof vents of a plant, glints on water - may give false marks by the hundred. Take the number of such a picture only after looking at the sheet of samples on its page of the report.
- **What stands under the marks.** On sheets of 80 marks looked at by eye on two pictures, about 5 to 7 % were not cars, and on one of them about 6 % more were trucks and vans.
- **Neighbouring pictures** are each counted whole: a car on a strip two pictures share is counted twice. On the set the program was made on that strip is a few pixels wide.

- **What the program does not see.** A picture shows the cars under the open sky at the minute it was taken: those under a roof or under tree crowns are not on it. Trucks and buses are not part of the count - a net is taught not to mark them, though single marks do land on them.
- On a video card and on a processor the count may differ by single cars: the check tables came out within a car or two.

Where things are

Folder	What is in it
`Start.exe`, `App\`	the window of the program
`runtime\`	Python with its libraries - installed by the SETUP section
`system_core\`	the counting engine; `system_core\nets\` - the place of a net that came with the program or was put into it (in the public release: `open-03`, taught on free aerial photographs)
`config\settings.json`	the settings of the program: the three folders, the language, the theme - always in its folder
`config\counting_default.json`	which net of `system_core\nets` the program counts with, and from which sureness, when the work folder of the set names none
`logs\`	the journal, a file per day
`Docs\`	the guides
the folder of the pictures	your `.jpg` and `.tab` files; only read
the work folder	everything made from the pictures - the five rows below
`runs\count_<net>\`	the count: a table of marks for each picture and `summary.json` with the totals
`runs\thumbs\`, `runs\pyramid\`	small copies and tiles of the pictures for the window; they can be deleted - they are made again
`labels\<picture>\`	the points set by eye: a file a patch; `labels\_removed\` - patches taken away in the window
`models\`	nets taught on your points; `counting.json` names the one the set is counted with

Folder	What is in it
`set.json`	which pictures this work folder was made on
the reports folder	Word documents, Excel sheets, layers for MapInfo

What the program does on the network and on the machine

It collects nothing and sends nothing. It goes to the network only during installation - to download Python (python.org), the package installer pip (bootstrap.pypa.io), the libraries (PyPI) and PyTorch (download.pytorch.org).

The program keeps everything of its own in its folder; into Windows it installs only the Microsoft Visual C++ Redistributable, when it was not there - that one stays, and many programs need it. To remove the program, delete its folder; the pictures, the work folder and the reports lie apart and stay.

If something goes wrong

What you see	What to do
"The counting engine is not installed"	the SETUP section, INSTALL WHAT IS NEEDED
SETUP says the NVIDIA driver is too old	update the driver and press CHECK AGAIN; until then the count runs on the processor
"The count runs on the processor although there is an NVIDIA card"	the PYTORCH FOR NVIDIA button
"The engine does not answer"	the THE WHOLE ENVIRONMENT button
"No taught net" in SETUP; "No taught net in the work folder or in the program: nothing to count with." in COUNT	not a fault of the installation: no net was taught or put in yet - see "A net for your own pictures"
"There are no pictures in ..."	name the folder of the pictures in SETTINGS; the file names must end with `_column-row.jpg`
"The work folder was made on other pictures"	every set has a work folder of its own: name another one for this set in SETTINGS

What you see	What to do
a picture has no area and no density, the layers are not written	there is no `.tab` file beside the picture
the report buttons are dimmed	nothing is counted yet; for REPORT ON THIS PICTURE open a counted picture in VIEW
the numbers on the map and in COUNT differ	the map shows the count and the threshold chosen in the ANSWERS OF THE NET card of the PICTURES section
the count is slow	it runs on the processor: look at the SETUP section
marks stand where there are no cars - on roofs, gravestones, water	ground the program has not seen: see "What accuracy to expect" and "A net for your own pictures"
the pictures, the points or the count were changed by a command and the window shows the old state	the REREAD button