 

**Prog.city**

ООО «ПРОГРЕСС» - 2024

Программа для распознавания возгораний и задымлений с камер видеонаблюдения в режиме реального времени при помощи искусственного интеллекта

Оглавление

[Описание программы 3](#_Toc164260920)

[Характеристики 3](#_Toc164260921)

[Скорость работы 3](#_Toc164260922)

[Серверный режим 3](#_Toc164260923)

[Технические требования 4](#_Toc164260924)

[Минимальные требования для рабочего места: 4](#_Toc164260925)

[Оптимальные требования для рабочего места: 4](#_Toc164260926)

[Требования к камерам: 4](#_Toc164260927)

[Требования к операционной системе 5](#_Toc164260928)

[Подготовка и установка демо-версии ПО «Оберег Плюс. Распознавание возгораний и задымлений» 5](#_Toc164260929)

[Ограничения демо-версии ПО «Оберег Плюс. Распознавание возгораний и задымлений» 5](#_Toc164260930)

[Установка зависимостей демо-версии программы «Оберег Плюс. Распознавание возгораний и задымлений» для ОС Ubuntu 20.04. 5](#_Toc164260931)

[Установка демо-версии ПО «Оберег Плюс. Распознавание возгораний и задымлений» 6](#_Toc164260932)

[Подготовка к работе демо-версии 6](#_Toc164260933)

[Подготовка и установка ПО «Оберег Плюс. Распознавание возгораний и задымлений» 6](#_Toc164260934)

[Установка зависимостей программы «Оберег Плюс. Распознавание возгораний и задымлений» для ОС Ubuntu 20.04. 7](#_Toc164260935)

[Очистка системы от установленных ранее драйверов и пакетов 7](#_Toc164260936)

[Скачивание локальных версий пакетов 7](#_Toc164260937)

[Установка OpenCV с зависимостями и поддержкой CUDA и CUDNN 8](#_Toc164260938)

[Установка поддержки звука 10](#_Toc164260939)

[Установка поддержки транскодирования видео (опционально) 10](#_Toc164260940)

[Установка поддержки SQL БД (опционально) 11](#_Toc164260941)

[Установка версии ПО «Оберег Плюс. Распознавание возгораний и задымлений» 12](#_Toc164260942)

[Подготовка к работе 12](#_Toc164260943)

[Описание интерфейса пользователя 15](#_Toc164260944)

[Главный экран 15](#_Toc164260945)

# Описание программы

Программа «**Оберег Плюс. Распознавание возгораний и задымлений»** позволяет мгновенно распознавать возгорания и задымления, как в потоке людей в городских условиях, так и на открытой местности при помощи камер видеонаблюдения, которые сегодня используются повсеместно.

После определения угрозы, программа обратит внимание оператора на нужную камеру и выполнит запрограммированные действия.

# Характеристики

## Скорость работы

На бюджетном ГПУ (Mobile RTX 3070 c 8 GB памяти) поддерживается до 16 камер с высоким FPS, с возможностью распознавания возгораний и задымлений на дистанциях донескольких километров.

## Серверный режим

Существует возможность работы в серверном режиме и подбор оборудования под ваши нужды и возможности, в том числе использование имеющегося.

Технические требования

## Минимальные требования для рабочего места:

|  |  |
| --- | --- |
| Центральный процессор  (CPU) | 8 ядер с частотой 2 ГГц |
| Оперативная память (RAM) | 8+ Гб |
| Видеокарта (CUDA) | RTX 2060, 8 Гб |

## Оптимальные требования для рабочего места:

|  |  |
| --- | --- |
| Центральный процессор  (CPU) | 8 ядер с частотой 2 ГГц |
| Оперативная память (RAM) | 16+ Гб |
| Видеокарта (CUDA) | RTX 3070, 8 Гб |

## Требования к камерам:

|  |  |
| --- | --- |
| Видеокодек | H.264 / H.265 / H.265+ |
| Поддержка протокола | RTSP |
| Разрешение матрицы (Megapixel) | 2-8 MP |
| Скорость видеопотока  (Bitrate) | 3-16 MB/s |
| Разрешение выходного потока  (Pixels) | от 1024 х 768 до 2688 х 1520 |
| Количество кадров в секунду (FPS) | От 4 до 60 |

# Требования к операционной системе

Операционная система должна удовлетворять следующим требованиям:

* Linux/GTK
* Bash
* C++ (кроме демо-версии программы)
* CUDA (кроме демо-версии программы)
* OpenCV (для демо-версии устанавливается автоматически, для остальных версий требуется ручная установка)

# Подготовка и установка демо-версии ПО «Оберег Плюс. Распознавание возгораний и задымлений»

## Ограничения демо-версии ПО «Оберег Плюс. Распознавание возгораний и задымлений»

Демо-версия ПО «Оберег Плюс. Распознавание возгораний и задымлений» предоставляется исключительно для оценки пользователем интерфейса программы и не предназначена для распознавания возгораний и задымлений в реальных проектах. Используемая в демо-версии нейросеть урезана по своим рабочим параметрам до возможности работать только на CPU компьютера без использования GPU, обязательного для клиентских и серверных версий ПО «Оберег Плюс. Распознавание возгораний и задымлений»

Искусственно ограничены следующие параметры нейросети:

- дальность распознавания возгораний и задымлений

- качество распознавания возгораний и задымлений

## Установка зависимостей демо-версии программы «Оберег Плюс. Распознавание возгораний и задымлений» для ОС Ubuntu 20.04.

Убедитесь, что в файле */etc/apt/sources.list* указан репозиторий *archive.ubuntu.com* со следующими параметрами:

*deb http://archive.ubuntu.com/ubuntu/ focal-updates main restricted universe multiverse*

если он отсутствует, то добавьте его первым по списку репозиторием.

Обновите установленные пакеты до текущих версий:

*sudo apt update  
sudo apt upgrade*

## Установка демо-версии ПО «Оберег Плюс. Распознавание возгораний и задымлений»

Скопируйте файл **firecheck-demo-1.0\_1.deb** в выбранную папку и выполните команду:

*sudo apt install ./**firecheck-demo-1.0\_1.deb*

# Подготовка к работе демо-версии

Для описания видеоисточника, достаточно указать параметры соединения с камерами.

В командной строке формат параметров:

*--urlN=url*

Для предоставленной демо-версии клиентского комплекса ПО N может быть в пределах от 1 до 4.

**Пример:**

*firecheck ---url1=rtsp://192.168.1.168/onvif1 \  
--url2=rtsp://login:password@192.168.1.169:5554/onvif1 \  
--url3=rtsp://login:password@192.168.1.170 \  
--url4=rtsp://login:password@192.168.1.171/ch01/0*

Далее, если вы пользуйтесь демо-версией ПО «Оберег Плюс. Распознавание возгораний и задымлений», можно переходить к инструкции пользователя клиентской версии.

# Подготовка и установка ПО «Оберег Плюс. Распознавание возгораний и задымлений»

## Установка зависимостей программы «Оберег Плюс. Распознавание возгораний и задымлений» для ОС Ubuntu 20.04.

Убедитесь, что в файле */etc/apt/sources.list* указан репозиторий *archive.ubuntu.com* со следующими параметрами:

*deb http://archive.ubuntu.com/ubuntu/ focal-updates main restricted universe multiverse*

если он отсутствует, то добавьте его первым по списку репозиторием.

Обновите установленные пакеты до текущих версий:

*sudo apt update  
sudo apt upgrade*

Для установки полноценной версии ПО «Оберег Плюс. Распознавание возгораний и задымлений» требуется самостоятельная установка ПО CUDA/CuDNN/TensorRT с сайта NVidia и настройка компиляции OpenCV с указанием доступности этих пакетов.

## Очистка системы от установленных ранее драйверов и пакетов

*sudo apt-get --purge remove nvidia  
sudo apt-get --purge remove "\*nvidia\*"  
sudo apt-get --purge remove "\*cublas\*" "cuda\*" "nsight\*"  
sudo rm -rf /usr/local/cuda\*  
sudo apt-get autoremove  
sudo apt-get autoclean  
sudo reboot*

## Скачивание локальных версий пакетов

*- cuda-repo-ubuntu2004-11-6-local\_11.6.2-510.47.03-1\_amd64.deb  
- cudnn-local-repo-ubuntu2004-8.4.0.27\_1.0-1\_amd64.deb  
- nv-tensorrt-repo-ubuntu2004-cuda11.6-trt8.4.0.6-ea-20220212\_1-1\_amd64.deb*

в данной инструкции не рассматривается. Для их получения необходимо перейти на сайт: [https://developer.nvidia.com](https://developer.nvidia.com/). До скачивания уточните в разделе документации, какие версии пакетов CUDA, CUDNN и TensorRT совместимы между собой в настоящий момент.

Подробная инструкция расположена на сайте NVidia:  
<https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html>

Так же следует учитывать, что у вас может быть устаревшее оборудование (например, NVidia K80), поддержка которого может отсутствовать в текущих версиях пакетов. В этом случае необходимо будет найти и скачать их архивные версии.

Ниже показан пример одной из возможных последовательностей команд, ведущих к установке пакетов CUDA, CUDNN и TensorRT:

*sudo apt-get install nvidia-driver-510  
sudo reboot  
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2004/x86\_64/cuda-ubuntu2004.pin  
sudo mv cuda-ubuntu2004.pin /etc/apt/preferences.d/cuda-repository-pin-600  
sudo dpkg -i cuda-repo-ubuntu2004-11-6-local\_11.6.2-510.47.03-1\_amd64.deb  
sudo apt-key add /var/cuda-repo-ubuntu2004-11-6-local/7fa2af80.pub  
sudo apt-get update  
sudo apt-get -y install cuda  
sudo dpkg -i cudnn-local-repo-ubuntu2004-8.4.0.27\_1.0-1\_amd64.deb  
sudo apt-get update  
sudo apt-get install libcudnn8=8.4.0.27-1+cuda11.6  
sudo apt-get install libcudnn8-dev=8.4.0.27-1+cuda11.6  
sudo dpkg -i nv-tensorrt-repo-ubuntu2004-cuda11.6-trt8.4.0.6-ea-20220212\_1-1\_amd64.deb  
sudo apt-get update  
sudo apt-get install tensorrt  
sudo apt-get install python3-libnvinfer-dev  
sudo -H pip3 install -U pip numpy onnx  
sudo apt-get install onnx-graphsurgeon*

## Установка OpenCV с зависимостями и поддержкой CUDA и CUDNN

Далее предполагается, что архивы opencv.zip и opencv\_contrib.zip будут скачаны в каталог ~/Downloads/, что явно учитывается в переменной OPENCV\_EXTRA\_MODULES\_PATH во время сборки OpenCV утилитой cmake.

Поддерживаются версии OpenCV от 4.6.0 до 4.9.0. Сборка для всех указанных версий аналогична. В примере для конкретности использована версия 4.6.0.

*cd ~  
mkdir Downloads  
cd Downloads  
wget -O opencv.zip https://github.com/opencv/opencv/archive/refs/tags/4.6.0.zip  
wget -O opencv\_contrib.zip https://github.com/opencv/opencv\_contrib/archive/refs/tags/4.6.0.zip  
sudo apt-get install libnetplan0  
sudo apt-get install netplan.io  
sudo apt-get install libyaml-cpp-dev  
sudo apt-get install curl  
sudo apt-get install libcurl4-openssl-dev  
sudo apt install build-essential cmake pkg-config unzip yasm git checkinstall  
sudo libjpeg-dev libpng-dev libtiff-dev  
sudo apt install libavcodec-dev libavformat-dev libswscale-dev libavresample-dev  
sudo apt install libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev  
sudo apt install libxvidcore-dev x264 libx264-dev libfaac-dev libmp3lame-dev libtheora-dev   
sudo apt install libfaac-dev libmp3lame-dev libvorbis-dev  
sudo apt install libopencore-amrnb-dev libopencore-amrwb-dev*

*sudo apt-get install libdc1394-22 libdc1394-22-dev libxine2-dev libv4l-dev v4l-utils  
pushd /usr/include/linux  
sudo ln -s -f ../libv4l1-videodev.h videodev.h  
popd  
sudo apt-get install python3-dev python3-pip  
sudo apt install python3-testresources  
sudo apt-get install libgtk-3-dev  
sudo apt-get install libtbb-dev  
sudo apt-get install libatlas-base-dev gfortran  
sudo apt-get install libprotobuf-dev protobuf-compiler  
sudo apt-get install libgoogle-glog-dev libgflags-dev  
sudo apt-get install libgphoto2-dev libeigen3-dev libhdf5-dev doxygen  
sudo apt-get install git  
unzip opencv\_contrib.zip  
unzip opencv.zip  
cd opencv-4.6.0  
mkdir build  
cd build*

Для сборки необходимо указать правильную версию архитектуры CUDA SM в переменной *CUDA\_ARCH\_BIN.* Для вашего GPU вы можете посмотреть её на сайте NVidia: [*https://developer.nvidia.com/cuda-gpus*](https://developer.nvidia.com/cuda-gpus) или кратко, по поколениям: [*https*](https://docs.nvidia.com/cuda/cuda-compiler-driver-nvcc/index.html#virtual-architecture-feature-list)*://docs.nvidia.com/cuda/cuda-compiler-driver-nvcc/index.html#virtual-architecture-feature-list*

*cmake -D CMAKE\_BUILD\_TYPE=RELEASE \  
-D CMAKE\_INSTALL\_PREFIX=/usr/local \  
-D WITH\_TBB=ON \  
-D ENABLE\_FAST\_MATH=1 \  
-D CUDA\_FAST\_MATH=1 \  
-D WITH\_CUBLAS=1 \  
-D WITH\_CUDA=ON \  
-D BUILD\_opencv\_cudacodec=OFF \  
-D WITH\_CUDNN=ON \  
-D OPENCV\_DNN\_CUDA=ON \  
-D CUDA\_ARCH\_BIN=8.6 \  
-D WITH\_V4L=ON \  
-D WITH\_QT=OFF \  
-D WITH\_OPENGL=ON \  
-D WITH\_GSTREAMER=ON \  
-D OPENCV\_GENERATE\_PKGCONFIG=ON \  
-D OPENCV\_PC\_FILE\_NAME=opencv.pc \  
-D OPENCV\_ENABLE\_NONFREE=ON \  
-D OPENCV\_PYTHON3\_INSTALL\_PATH=/usr/lib/python3/dist-packages \  
-D PYTHON\_EXECUTABLE=/usr/bin/python3 \  
-D OPENCV\_EXTRA\_MODULES\_PATH=~/Downloads/opencv\_contrib-4.6.0/modules \  
-D INSTALL\_PYTHON\_EXAMPLES=OFF \  
-D INSTALL\_C\_EXAMPLES=OFF \  
-D BUILD\_PERF\_TESTS=OFF \  
-D BUILD\_TESTS=OFF \  
-D BUILD\_EXAMPLES=OFF ..  
make  
sudo make install*

## Установка поддержки звука

*sudo apt-get install alsa-base alsa-utils gstreamer1.0-alsa  
sudo apt-get install libasound2 libasound2-data libasound2-plugins  
sudo apt-get install libasound2-dev linux-sound-base*

## Установка поддержки транскодирования видео (опционально)

*sudo apt-get install gstreamer1.0-plugins-base gstreamer1.0-plugins-good  
sudo apt-get install gstreamer1.0-plugins-ugly  
sudo apt-get install gstreamer1.0-rtsp gstreamer1.0-tools*

## Установка поддержки SQL БД (опционально)

В случае приобретения ПО **«Оберег Плюс. Распознавание возгораний и задымлений»** с поддержкой работы с SQL БД, вы можете использовать БД для хранения истории сработок, обеспечения доступа к ним через web-интерфейс посредством REST-запросов и поддержки пользовательского механизма нотификации (реализован для PostgreSQL и Oracle).

ПО «**Оберег Плюс. Распознавание возгораний и задымлений»** работает с БД, поддерживающими ODBC-драйвера. Версии с поддержкой Oracle OCI или других высокопроизводительных интерфейсов к БД предоставляются по запросу.

В БД должна присутствовать таблица alarms. В случае использования PostgreSQL, она может быть создана с использованием следующего DDL:

*CREATE TABLE alarms (*

*id serial primary key,*

*cam int not null,*

*tstamp timestamp not null,*

*rec\_class int not null,*

*img bytea not null*

*);*

Если вы планируете использовать механизм поддержки пользовательской нотификации, с нашей реализацией в качестве основы, вам необходима очередь сообщений с именем *new\_alarm\_added.* В случае использования PostgreSQL, она может быть создана с использованием следующего DDL:

*CREATE or REPLACE FUNCTION notify\_new\_alarm()*

*RETURNS trigger*

*LANGUAGE 'plpgsql'*

*as $BODY$*

*declare*

*begin*

*if (tg\_op = 'INSERT') then*

*perform pg\_notify('new\_alarm\_added',*

*json\_build\_object(*

*'id', NEW.id,*

*'cam', NEW.cam,*

*'tstamp', NEW.tstamp,*

*'rec\_class', NEW.rec\_class*

*)::text);*

*end if;*

*return null;*

*end*

*$BODY$;*

*CREATE TRIGGER after\_insert\_alarm*

*AFTER INSERT*

*ON alarms*

*FOR EACH ROW*

*EXECUTE PROCEDURE notify\_new\_alarm();*

## Установка версии ПО «Оберег Плюс. Распознавание возгораний и задымлений»

Скопируйте файл **firecheck-1.0\_1.deb** в выбранную папку и выполните команду:

*sudo apt install ./***firecheck-1.0\_1.deb**

# Подготовка к работе

Все аргументы командной строки, кроме имени нейросети, задаются в формате —*key=value*

Обязательными параметрами для запуска ПО «Оберег Плюс. Распознавание возгораний и задымлений» являются только описания видеоисточников. Для описания видеоисточника, достаточно указать параметры соединения с камерами.

В командной строке формат параметров:

*--urlN=url* для сетевого видеоисточника или

*--inputN=path* для использования видеофайла

Для демо-комплекса ПО, передаваемого с целью оценки пользователем интерфейса программы, N может быть в пределах от 1 до 4.

Для клиентского комплекса ПО N может быть в пределах от 1 до 4...64, в зависимости от версии программы и договора поставки.

Для серверной версии ПО N может быть в пределах от 1 до 4...1024, в зависимости от версии программы и договора поставки.

**Пример:**

firecheck *---url1=rtsp://192.168.1.168/onvif1 \  
--url2=rtsp://login:password@192.168.1.169:5554/onvif1 \  
--url3=rtsp://login:password@192.168.1.170 \  
--url4=rtsp://login:password@192.168.1.171/ch01/0*

В случае, если договор поставки подразумевает версию ПО с несколькими нейросетями, дополнительно в командной строке должно быть указано имя нейросети:

firecheck *---url1=rtsp://192.168.1.168/onvif1 \  
--url2=rtsp://login:password@192.168.1.169:5554/onvif1 \  
--url3=rtsp://login:password@192.168.1.170 \  
--url4=rtsp://login:password@192.168.1.171/ch01/0 \*

*HighResNN*

Дополнительно, для клиентской и серверной версии ПО «Оберег Плюс. Распознавание возгораний и задымлений» доступны следующие параметры командной строки:

*--thr* Уровень тревожности нейросети. По умолчанию 0.3.

*--ui* Выбор фронтэнда. Возможные варианты: *none* для работы без GUI, *sciter* для работы с Sciter UI, opencv тестовый режим для отладки и разработки, не предназначен для использования клиентами.

*--alarm-path* Путь к каталогу, где будут храниться кадры сработок (в случае хранения сработок в файловой системе).

*--sql-url* Параметры соединения с SQL БД, где будут храниться кадры сработок (в случае хранения сработок в SQL БД).

*--ceph-url* Параметры соединения с Ceph-хранилищем (в случае хранения сработок в Ceph).

--fs-alarms разрешение использовать файловую систему для хранения сработок.

*--sql-alarms* разрешение использовать SQL БД для хранения сработок.

*--ceph-alarms* разрешение использовать Ceph-хранилище для хранения сработок.

*--alarms-rate* Ограничение скорости записи сработок в SQL БД, Ceph или файловую систему. Задаётся в миллисекундах. Например, *--alarms-rate=100*, означает записывать не более 10 сработок в секунду.

*--bot-id* Позволяет указать идентификатор бота, для посылки сообщений Telegram-чат.

*--bot-chat* Позволяет указать идентификатор чата, для посылки сообщений Telegram-чат.

*--delay* Позволяет ограничить скорость обработки кадров архивных видеопотоков ускорителем, задаётся в миллисекундах на кадр.

*--out-file* Позволяет записать распознанный видеопоток с метками сработок в файл.

*--out-url* Позволяет записать распознанный видеопоток с метками сработок в сетевой видеопоток по протоколу RTSP.

*--out-fps* Позволяет задать желаемую частоту кадров для выходного видеопотока.

*--scada-delay* В случае интеграции со СКУД, позволяет задать задержку перед сработкой сценария СКУД. Значение по умолчанию 30 секунд.

*--scada-port* В случае интеграции со СКУД, порт для передачи команды на запуск сценария в СКУД.

*--sms* В случае интеграции с IP-телефонией или GSM-шлюзом, разрешение посылать SMS сообщение о сработке на указанный номер или номера.

*--sound* Путь к файлу, который будет проигрываться в случае сработки нейросети. Файл должен удовлетворять следующим требованиям: WAV, 44100Hz, mono, 8 bit, длительностью не менее 8 секунд.

Для удобства пользователя, часть параметров можно хранить в файле конфигурации. Файл конфигурации по умолчанию называется model.yml

Переключаться между конфигурационными файлами можно с помощью опции — zoo.

# Описание интерфейса пользователя

## Главный экран

После запуска, программа **«Оберег Плюс. Распознавание возгораний и задымлений»** сразу готова к работе.

Искусственный интеллект в режиме реального времени обрабатывает входной видеопоток с камер наблюдения и анализирует его на предмет присутствия объектов, похожих на возгорания и задымления.

После определения угрозы, программа обратит внимание оператора на нужную камеру и сделает скриншот (на фото слева). Если оператор в определённый промежуток времени не предпримет никаких действий, то программа самостоятельно выполнит запрограммированное действие, например, отправит скриншот возгорания в мессенджер ответственному лицу, подаст сигнал тревоги на пульт пожарной охраны, отправит сигнал на СКУД.



*На изображении главный экран программы*

Слева находится область для отображения скриншотов, которые автоматически попадают в базу данных при обнаружении возгораний и задымлений в кадре и могут быть просмотрены позже. Если включена опция отправки, данные скриншоты, отправляются адресатам.

Основное пространство экрана занимает сетка с трансляцией видеопотока с камер наблюдения. Зелёными прямоугольниками выделяются обнаруженные программой объекты похожие на возгорание и задымление. Текст наименования камеры становится красным, для привлечения внимания оператора.