К сожалению, хоть это и web-приложение, пока что на сервер оно не перенесено(создатель работает над этим в поте лица), однако сейчас можно протестировать на своем устройстве.

1) IDE, в которой необходимо тестировать проект;

Писалось на atom и PyCharm

2) зависимости / фреймворки и т. п., которые необходимо установить;

django

pillow

celery

rabbitmq если windows (<https://www.rabbitmq.com/install-windows-manual.html>)

если иные системы, то там же на сайте есть

opencv

numpy

3) устройство, на которых вы сами тестировали свой проект (если таковые есть);

На ноутбуке, немного на сервере, но в консольном режиме

4) указания на особенности проекта, которые нужно учесть тестировщику (если таковые есть);

Поскольку первое время тестировать придется не на сервере, то в случае тестирования на личных компьютерах будьте осторожны при создании собственного алгоритма, поскольку при загрузки достаточного количества изображений ваше устройство может прийти в негодность из-за нагрузки

5)(На всякий случай заходим в папку с проектом, там где лежит manage.py(через терминал, естественно) и прописываем

python manage.py makemigrations  
python manage.py migrate)

итак, поскольку не сервер то, первое что мы делаем создаем “супер юзера”  
python manage.py createsuperuser

создаем админа

rabbitmq-server

virtualenv -p python3 env && source env/bin/activate && pip install -r requirements.txt && python [manage.py](http://manage.py/) migrate && celery -A image\_handler worker -l info && python [manage.py](http://manage.py/) runserver

celery -A worker -l info

потом пишем   
python manage.py runserver

Заходим на тот адрес(с инкогнито), который появится (localhost) и дописываем “:8000/admin “

<http://localhost:8000/admin>

заходим с тем же логином и паролем, которые создавали в суперюзере

дальше в cascades заливаем по очереди все файлы из папки haarcascdes

После этого все готово для использования :

[http://localhost:8000/](http://localhost:8000/admin)accounts/register.html