

Руководство к программе для распознавания дефектов на поверхности горячего проката

Ссылка на GitHub проекта <https://github.com/AnLiMan/Recognition-of-defects-on-the-metal-surface>

Датасет для обучения нейросети

https://drive.google.com/file/d/1rDPzh5P2rKIaSV_PduEoMttUndGvpzrL/view?usp=sharing

Модель обученной нейросети

<https://drive.google.com/file/d/1GqPEY834joSFFVflvwcuPNtI3gYBLwNl/view?usp=sharing>

Введение

Задача поиска и классификации дефектов на поверхности горячекатаного металла не нова и до определенного времени выполнялась технологом, и только сейчас технологии достигли достаточного уровня, чтобы автоматизировать данный процесс. При этом самым распространённым, и оптимальным вариантом решения данной проблемы являются алгоритмы машинного обучения, основанные на искусственных нейронных сетях, в частности свёрточных нейронных сетях.

Компьютерное зрение, которое базируется на элементах машинного обучения позволяет автоматизировать контроль качества проката в процессе производства. Соблюдение стандартов качества выходной продукции имеет первостепенное значение в сфере производства, т.к. позволяет на ранней стадии выявить возможный брак партии. Как уже отмечалось ранее, по большей части контроль качества листопрокатной продукции ложится на технолога, которому свойственно ошибаться особенно при выполнении монотонной работы.

Дефекты на поверхности листопроката считаются одной из основных причин увеличения себестоимости продукции и составляют более 90 % от всех дефектов стальной продукции, они могут быть связаны с такими факторами, как например, условия эксплуатации и техническое состояние прокатного оборудования. Среди других недостатков традиционного осмотра человеком можно отметить высокие затраты времени инспекции поверхности, по сравнению с технологиями компьютерного зрения.

О платформе, на которой разрабатывался первоначальный код

Colaboratory, или сокращенно Colab – продукт компании Google Research. Colab позволяет любому писать и выполнять произвольный код Python через браузер и особенно хорошо это подходит для целей машинного обучения, анализа данных и образования. С технической точки зрения Colab – это размещенная на хосте служба Jupyter для скриптов, которая не требует настройки для использования, но при этом предоставляет бесплатный доступ к вычислительным ресурсам, включая графические процессоры.

Скрипты Colab хранятся на Google Диске или могут быть загружены с GitHub. Скриптами Colab можно делиться так же, как документами или таблицами Google. Код выполняется на виртуальной машине, принадлежащей текущей учетной записи. Виртуальные машины удаляются, если они бездействуют в течение некоторого времени.

Типы графических процессоров, доступных в Colab, со временем меняются. Это необходимо, чтобы Colab могла бесплатно предоставлять доступ к этим ресурсам. Графические процессоры, доступные в Colab, включают в себя Nvidia K80s, T4s, P4s и P100s. Нельзя выбрать, к какому типу графического процессора можно подключиться, пользователи, которые заинтересованы в более надежном доступе к самым быстрым графическим процессорам Colab, могут использовать Colab Pro.

Объем памяти, доступной в виртуальных машинах Colab, меняется со временем (но остается стабильным в течение всего срока работы виртуальной машины). Иногда объем может автоматически назначаться дополнительная память, когда Colab обнаруживает, что она может понадобиться.

Ссылки

На ранее упомянуто платформе Google.Colab был произведён процесс обучения свёрточной нейронной сети и первоначальная отладка разработанных дополнительных функций.

Ссылка на GitHub проекта <https://github.com/AnLiMan/Recognition-of-defects-on-the-metal-surface>

Файл проекта имеет название «Computer_vision_v_0_9_3.ipynb».

Итоговая модель обладает следующими качествами:

- модель CNN распознают 6 дефектов листопроката и изображения чистой поверхности с точностью 92,31 %;
- время распознавания составило $0,001384 \pm 5\%$ секунд для всех образцов (тест скорости производился на платформе разработки);
- вынесение рекомендации на основе наиболее распространенных дефектов того или иного типа.

О реализации системы компьютерного зрения с интерфейсом

Работа велась в рамках научно-исследовательской работы гранта фонда содействия инновациям «Умник» код 0081533, заявка У-84718. Данный проект находится в папке "Update with interface" (репозитория проекта). Внутри находятся 3 скрипта - "Main_ML.py", основной код для запуска интерфейса "ProgramInterface.ui", разработанного при помощи PyQtDesigner, а также скрипт для запуска камеры. Список распознаваемых дефектов: вдавленные окаины; пятна; волосные трещины; рябизна; посторонние включения и чистая поверхность.

Код разрабатывался для работы в среде PyCharm и содержит в себе следующие библиотеки и версию python на момент окончательной разработки:

- PIL v 9.4.0
- tensorflow v 2.11.0

- matplotlib v 3.7.1
- numpy v 1.24.2
- opencv v 4.7.0.72
- PyQt6 v 6.4.2
- Python v 3.10

Интерфейс программы

Основное окно программы показано ниже

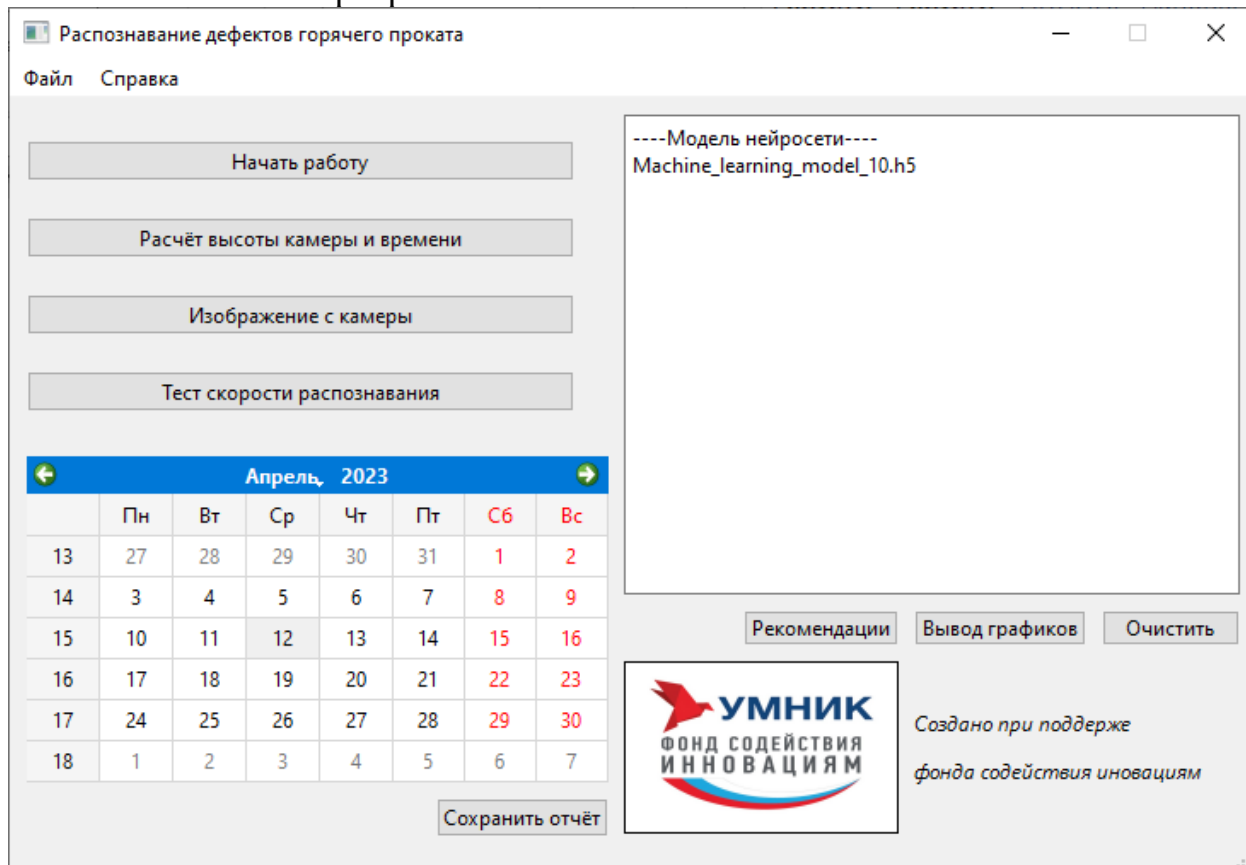


Рисунок 1 – Окно программы

Перед началом основной работы следует произвести инициализацию некоторых переменных для работы программы. Для этого сначала нужно нажать кнопку «Тест скорости распознавания», для верности можно прогнать тест 2-3 раза. После нажать кнопку «Расчёт высоты камеры и времени». После можно нажать кнопку «Начало работы». Важно. Начало работы может производиться на тестовом изображении, расположенном в папке с проектом. Однако для этого предварительно требуется раскомментировать строку в функции «Split_Image()».

```
image = cv2.imread("TestImage2.jpg") #Раскомментировать для тестового прогона
```

Остальные кнопки носят интуитивный характер. Перед запуском рекомендуется также пробежаться по коду для понимания полной картины.