

DOI: 10.12731/2227-930X-2023-13-1-77-89

УДК 004.93



РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ КОНТРОЛЯ КАЧЕСТВА ДЕТАЛЕЙ НА БРАК

*Р.Р. Нурутдинов, Р.Я. Шайхутдинов,
Д.С. Зарипов, Г.А. Гареева*

В статье описывается разработка программного кода для контроля качества деталей. Для решения этой задачи было разработано приложение для сравнения изделий на брак с помощью среды программирования Python и библиотеки компьютерного зрения OpenCV.

***Цель** – разработка эффективной программы для сравнения двух деталей, шаблона правильной детали и сделанного изображения детали в программе для его сравнения с данным шаблоном.*

***Метод или методология проведения работы.** В статье рассматривается программа для сбора и хранения данных, а также их обработка для контроля качества.*

***Результат.** Разработано приложение в среде программирования Python, которая собирает информацию в виде изображения и обрабатывает ее при помощи библиотеки компьютерного зрения OpenCV для сравнения двух изображений.*

***Область применения результатов.** Данное приложение можно использовать в промышленности для мониторинга и контроля качества деталей, так как качество – это важная часть создания точных изделий.*

***Ключевые слова:** библиотека компьютерного зрения, возможности, OpenCV, Python, сравнение изображений*

Для цитирования. Нурутдинов Р.Р., Шайхутдинов Р.Я., Зарипов Д.С., Гареева Г.А. Разработка приложения для контроля качества деталей на брак // International Journal of Advanced Studies. 2023. Т. 13, № 1. С. 77-89. DOI: 10.12731/2227-930X-2023-13-1-77-89

DEVELOPMENT OF AN APPLICATION FOR QUALITY CONTROL OF PARTS FOR REJECTS

**R.R. Nurutdinov, R.Y. Shaikhutdinov,
D.S. Zaripov, G.A. Gareeva**

This article describes the development of software code for quality control of parts. To solve this problem, an application for comparing products for defects was developed with the help of Python programming environment and OpenCV computer vision library.

*The **goal** is to develop an effective program for comparing two parts, a template of a correct part and a made image of the part in a program to compare it to this template.*

***Methodology.** This article discusses a program for collecting and storing data, as well as processing them for quality control.*

***Results.** We developed an application in a Python programming environment that collects information in the form of an image and processes it using the OpenCV computer vision library to compare two images.*

***Practical implications.** This application can be used in industry to monitor and control the quality of parts, because quality is an important part of creating precision products.*

***Keywords:** computer vision library, features, OpenCV, Python, image comparison*

***For citation.** Nurutdinov R.R., Shaikhutdinov R.Y., Zaripov D.S., Gareeva G.A. Development of an Application for Quality Control of Parts for Rejects // International Journal of Advanced Studies, 2023, vol. 13, no. 1, pp. 77-89. DOI: 10.12731/2227-930X-2023-13-1-77-89*

Введение

На любом этапе производства деталей может быть допущен брак. И при своевременном контроле качества детали можно сократить расходы на материалы и увеличить ресурс созданной продук-

ции за счет обеспечения полноценного функционирования на всех режимах эксплуатации. Контроль качества - это одна из основных функций в процессе управления качеством. Значение контроля заключается в том, что он позволяет вовремя выявить ошибки, чтобы затем оперативно исправлять их с минимальными потерями [7].

Одним из методов использования библиотек в программировании является распознавание лиц. Применение технологии распознавания лиц очень актуальна в наше время. Данный метод широко используется для идентификации людей в биометрических системах, например, на смартфонах благодаря встроенному сканеру лица можно снимать блокировку с устройства [3]. Такие системы помогут защитить устройство от злоумышленников. Также метод используется для создания фильтров, изменяющих лицо в приложениях, в робототехнике, медицинских технологиях, промышленных технологиях и для разработки автопилотов.

Для того чтобы создавать такие системы, разработчику необходимы глубокие знания в алгоритмах распознавания лиц, а также правильно применять их в разных средах программирования.

Библиотека компьютерного зрения и машинного обучения с открытым исходным кодом поддерживает множество платформ, такие как Windows, Linux, MacOS. Наиболее часто используется в таких языках как C, C++, Java и Python [4]. Чаще всего используется для написания приложений машинного обучения на языке Python, особенно в сфере компьютерного зрения.

Цель работы: разработать приложение для контроля качества деталей, разработанную в среде программирования Python, т.е. программы для сравнения двух деталей, шаблона правильной детали и сделанного изображения детали в программе для его сравнения с данным шаблоном.

Материалы и методы

При разработке приложения для контроля качества деталей использовались следующие библиотеки Python, изображенные на рисунке 1.

```
from tkinter import*  
import tkinter  
import cv2  
from PIL import Image,ImageTk
```

Рис. 1. Список используемых библиотек

Для того чтобы установить библиотеку, необходимо иметь установленную программу Python на компьютере. Далее необходимо открыть программу PowerShell от имени администратора и ввести следующую команду [2]:

```
pip install opencv-python
```

Также понадобится еще одна библиотека Pillow. Библиотека изображений Python добавляет возможности обработки изображений в интерпретатор Python.

```
pip install Pillow
```

Библиотека Tkinter используется для создания графического интерфейса и устанавливается аналогичным способом.

Для создания интерфейса программы используется следующий код, изображенный на рисунке 2.

```
#создание графического интерфейса  
win = tkinter.Tk()  
win.geometry("920x700")
```

Рис. 2. Создание интерфейса при помощи Tkinter

Создается имя интерфейса для дальнейшего использования его в коде программы, а также задается размер окна интерфейса.

Далее идет процесс добавления камеры для создания изображений, показанный на рисунке 3.

Для того, чтобы подключиться к видеоканере нужно использовать команду `cap = cv2.VideoCapture(0)`, где `cap` – имя переменной, а число 0 номер доступной камеры.

```
def openWindow():
    def to_pil(img, label, x, y, w, h):
        img = cv2.resize(img, (w, h))
        img = cv2.flip(img, 1)
        image = Image.fromarray(img)
        pic = ImageTk.PhotoImage(image)
        label.configure(image=pic)
        label.image = pic
        label.grid(row=1, column=1)

    #чтение видео изображения
    cap = cv2.VideoCapture(0)

    while True:
        ret, img = cap.read()

        cv2.imshow('from camera', img)

        k = cv2.waitKey(30)
        if k == 27:
            break

        elif k == ord('s'):
            # сохранение изображения
            cv2.imwrite('newGear.png', img)
            cv2.destroyAllWindows()
```

Рис. 3. Процесс чтения видеоизображения

Для того, чтобы изображение постоянно обновлялось, используется команда `waitKey`, где 30 это время обновления в миллисекундах.

Для вывода информации на ранее созданном интерфейсе используются следующие команды, показанные на рисунке 4.

В интерфейсе была создана кнопка для открытия нового окна с камерой. Также ниже кнопки были расписаны клавиши для управления в режиме камеры. В интерфейс были добавлены два изображения, созданное изображение при помощи камеры и оригинал.

Процесс сравнения двух изображений при помощи библиотеки `OpenCV` изображен на рисунке 5.

```
label = Label(win, text ="MENU")
label.grid(row=1, column=1)

#кнопка для открытия видео камеры
btn = Button(win, text ="Open video capture", command = openWindow)
btn.grid(row=2, column=1)

labe2 = Label(win, text ="Press the S key to take a picture")
labe2.grid(row=3, column=1)

labe3 = Label(win, text ="Press ESC to close window")
labe3.grid(row=4, column=1)

#ввод нового рисунка для сравнения
img1 = cv2.imread('newGear.png')
gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)
_, threshold = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY)
contours, _ = cv2.findContours(
    threshold, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

for contour in contours:
    cv2.drawContours(img1, [contour], 0, (255, 0, 0), 1)

im = Image.fromarray(img1)
imgtk = ImageTk.PhotoImage(image=im)

Label(win, image=imgtk).grid(row=5,column=1, sticky=N)

#ввод оригинального рисунка
canvas = tkinter.Canvas(win, height=500, width=800)
img2 = Image.open("GearPattern.png")
photo = ImageTk.PhotoImage(img2)
img2 = canvas.create_image(0, 0, anchor='nw',image=photo)
canvas.grid(row=5,column=2, padx=10)
```

Рис. 4. Вывод информации на экран

Далее происходит ввод изображения для сравнения на брак, где первое изображение это сделанное нами изображение, а второе оригинал (рис. 6).

```

def ImageHash(FileName):
    #Прочитаем картинку
    image = cv2.imread(FileName)
    #Уменьшим картинку
    resized = cv2.resize(image, (8,8), interpolation = cv2.INTER_AREA)
    #Переведем в черно-белый формат
    gray_image = cv2.cvtColor(resized, cv2.COLOR_BGR2GRAY)
    #Среднее значение пикселя
    avg=gray_image.mean()
    #Бинаризация по порогу
    ret, threshold_image = cv2.threshold(gray_image, avg, 255, 0)

    #Рассчитаем хэш
    _hash=""
    for x in range(8):
        for y in range(8):
            val=threshold_image[x,y]
            if val==255:
                _hash=_hash+"1"
            else:
                _hash=_hash+"0"

    return _hash

def CompareHash(hash1,img2):
    l=len(img1)
    i=0
    count=0
    while i<l:
        if img1[i]!=img2[i]:
            count=count+1
        i=i+1
    return count

```

Рис. 5. Сравнение изображений

```

#ввод изображения для сравнения
img1=ImageHash("newGear.png")
img2=ImageHash("GearPattern.png")

#вывод результата на экран
labe4 = Label(win, text = "Сходство: ", font=("Roman", 16))
labe4.grid(row=6, column=1,sticky=E)

labe5 = Label(win, text = CompareHash(img1,img2), font=("Roman", 16))
labe5.grid(row=6, column=2,sticky=W)

win.mainloop()

```

Рис. 6. Ввод изображения для сравнения

Готовый интерфейс программы изображён на рисунке 7. На рисунке представлено поле, показывающее уровень сходства изображения с оригиналом. Чем ниже уровень сходства, тем больше изображения похожи друг на друга.

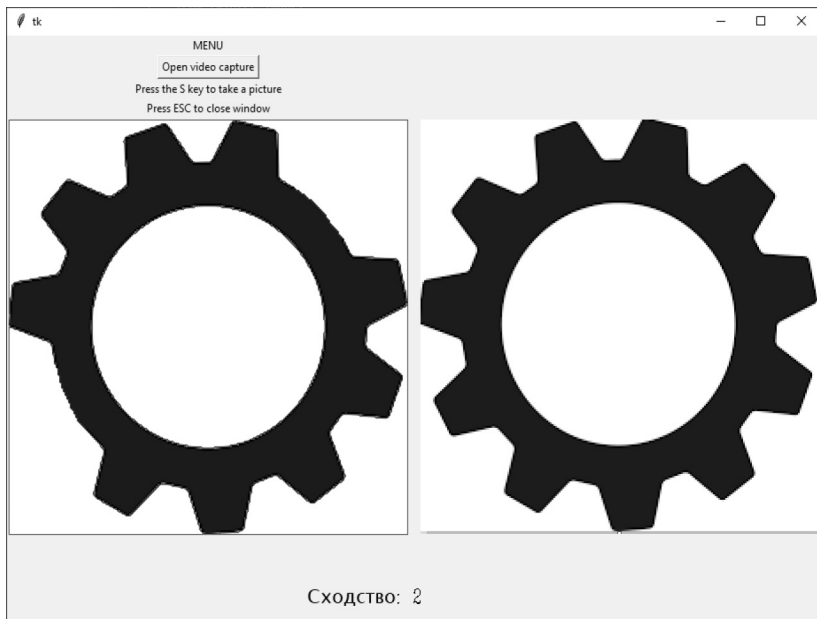


Рис. 7. Интерфейс программы со сходством

Результат работы

Было разработано приложение для контроля качества деталей, который собирает и обрабатывает информацию в виде изображения. В данном приложении в качестве основы использовалась библиотека компьютерного зрения OpenCV. Был реализован вывод камеры при помощи кнопки для создания изображения для её последующей обработки. Создан интерфейс для вывода информации и реализован код для сравнения двух изображений. В результате была создана программа для сравнения двух деталей на брак.

Заключение

Таким образом, разработано приложение для контроля качества деталей на брак с помощью среды программирования Python и библиотеки компьютерного зрения OpenCV. Приложение эффективно для сравнения двух деталей, шаблона правильной детали и сделанного изображения детали в программе для его сравнения с данным шаблоном. Данное приложение можно использовать в промышленности для мониторинга и контроля качества деталей.

Список литературы

1. Тугов В.В. Проектирование автоматизированных систем управления: учебное пособие / В.В. Тугов, А.И. Сергеев, Н.С. Шаров. Санкт-Петербург: Лань, 2019. 172 с.
2. Прохоренок Н.А. OpenCV и Java. Обработка изображений и компьютерное зрение. СПб.: БХВ–Петербург, 2018. 320 с.
3. Рейнхард Клетте. Компьютерное зрение. Теория и алгоритмы / пер. с англ. А.А.Слинкин. М.: ДМК-Пресс, 2019. 506 с.
4. Прохоренок Н.А. OpenCV и Java. Обработка изображений и компьютерное зрение / Н.А. Прохоренок. СПб.: БХВ-Петербург. 2018.
5. Правоткин И.А. Настройка и запуск программ на Python на удалённом хостинге // Приоритетные направления инновационной деятельности в промышленности сборник научных статей по итогам двенадцатой международной научной конференции. Казань: ООО «Конверт», 2020. С. 78-80.
6. Гареева Г.А., Хамидуллин М.Р., Джибладзе З.Г., Ахметов Л.М. Развертывание искусственного интеллекта по распознаванию лиц // Научно-технический вестник Поволжья. № 8. 2022. Казань: ООО «Рашин Сайнс», 2022. С. 50-53.
7. Akhmetov L.M., Bikov D.I., Khamidullin M.R., Gareeva G.A., Gabdullina G.K. Development of a system for analyzing and unloading road traffic using artificial intelligence // Journal of Physics: Conference Series, 2021, vol. 2094.

8. David Love. Tkinter GUI Programming by Example. Packt Publishing, 2018. 340 p.
9. Michael Driscoll. Pillow. Image Processing with Python. Independently Published, 2021. 382 p.
10. Gaurav Leekha. Learn AI with Python. Explore Machine Learning and Deep Learning techniques for Building Smart AI Systems Using Scikit-Learn, NLTK, NeuroLab, and Keras. BPB Publications, 2021. 270 p.

References

1. Tugov V.V. *Proektirovanie avtomatizirovannykh sistem upravleniya: uchebnoe posobie* [Designing automated control systems: a textbook] / V.V. Tugov, A.I. Sergeev, N.S. Sharov. Sankt-Peterburg: Lan', 2019. 172 p.
2. Prokhorenok N.A. *OpenCV i Java. Obrabotka izobrazheniy i komp'yuternoe zrenie* [OpenCV and Java. Image Processing and Computer Vision]. SPb.: BKhV–Peterburg, 2018. 320 p.
3. Reynkhard Klette. *Komp'yuternoe zrenie. Teoriya i algoritmy* [Computer Vision. Theory and algorithms] / per. s angl. A.A.Slinkin. M.: DMK-Press, 2019. 506 p.
4. Prokhorenok N.A. *OpenCV i Java. Obrabotka izobrazheniy i komp'yuternoe zrenie* [OpenCV and Java. Image Processing and Computer Vision] / N.A. Prokhorenok. SPb.: BKhV-Peterburg. 2018.
5. Pravotkin I.A. *Nastroyka i zapusk programm na Python na uda-lennom khostinge // Prioritetnye napravleniya innovatsionnoy deyatelnosti v promyshlennosti sbornik nauchnykh statey po itogam dvenadtsatoy mezhdunarodnoy nauchnoy konferentsii* [Setting up and running Python programs on remote hosting // Priority directions of innovation in industry - a collection of scientific articles based on the results of the twelfth international scientific conference]. Kazan': OOO «Konvert», 2020. P. 78-80.
6. Gareeva G.A., Khamidullin M.R., Dzhibladze Z.G., Akhmetov L.M. *Razvertyvanie iskusstvennogo intellekta po raspoznavaniyu lits* [Deploying artificial intelligence for facial recognition] // Nauchno-tekh-

- nicheskiiy vestnik Povolzh'ya. № 8 2022. Kazan': ООО «Rashin Sayns», 2022. P. 50-53.
7. Akhmetov L.M., Bikov D.I., Khamidullin M.R., Gareeva G.A., Gab-dullina G.K. Development of a system for analyzing and unloading road traffic using artificial intelligence // Journal of Physics: Conference Series, 2021, vol. 2094.
 8. David Love. Tkinter GUI Programming by Example. Packt Publishing, 2018. 340 p.
 9. Michael Driscoll. Pillow. Image Processing with Python. Independently Published, 2021. 382 p.
 10. Gaurav Leekha. Learn AI with Python. Explore Machine Learning and Deep Learning techniques for Building Smart AI Systems Using Scikit-Learn, NLTK, NeuroLab, and Keras . BPB Publications, 2021. 270 p.

ДААННЫЕ ОБ АВТОРЕ

Нурутдинов Рияз Ринатович, студент

*Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ
ул. Академика Королева, 1, г. Набережные Челны, 423814,
Российская Федерация
nriyaz@mail.ru*

Шайхутдинов Ринас Ягъфасович, студент

*Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ
ул. Академика Королева, 1, г. Набережные Челны, 423814,
Российская Федерация
iaqfas@mail.ru*

Зарипов Динар Салаватович, студент

Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ

*ул. Академика Королева, 1, г. Набережные Челны, 423814,
Российская Федерация
dzaripov98@mail.ru*

Гареева Гульнара Альбертовна, заведующий кафедрой Информационных систем, канд. пед. наук, доцент
*Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ
ул. Академика Королева, 1, г. Набережные Челны, 423814,
Российская Федерация
gagareeva1977@mail.ru*

DATA ABOUT THE AUTHORS

Riyaz R. Nurutdinov, student

*Kazan National Research Technical University named after
A.N. Tupolev-KAI
1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation
nriyaz@mail.ru*

Rinas Y. Shaikhutdinov, student

*Kazan National Research Technical University named after
A.N. Tupolev-KAI
1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation
iagfas@mail.ru*

Dinar S. Zaripov, student

*Kazan National Research Technical University named after
A.N. Tupolev-KAI
1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation
dzaripov98@mail.ru*

Gulnara A. Gareeva, Head of the Department of Information Systems, Candidate of
Pedagogical sciences, Associate professor
Kazan National Research Technical University named after A.N. Tupolev-KAI
1, Akademika Koroleva Str., Naberezhnye Chelny, 423814, Russian Federation
gagareeva1977@mail.ru
SPIN-code: 3279-8465
Scopus Author ID: 36801593200
ResearcherID: M-1728-2015
ORCID: <https://orcid.org/0000-0002-8539-4541>

Поступила 16.01.2023

После рецензирования 05.02.2023

Принята 10.02.2023

Received 16.01.2023

Revised 05.02.2023

Accepted 10.02.2023