

TEXNIKA FANLARI

Karimova Zilola, Denov tadbirkorlik va pedagogika instituti stajyor o'qituvchisi, z.karimova@dtpi.uz

Xaydarov Sherali, Denov tadbirkorlik va pedagogika instituti doktranti, sh.haydarov@dtpi.uz

Doniyorova Gulshan, Denov tadbirkorlik va pedagogika instituti stajyor o'qituvchisi, g_doniyorova@dtpi.uz

УДК: 004.891.3

YO'L BELGILARINI ANIQLASHDA PYTHON DASTURLASH TILIDAN FOYDALANGAN HOLDA TASVIRLARNI MANIPULYATSIYA QILISH

Kalit so'zlar:
Yo'l belgi,
OpenCV,
Yo'nalish nuqtasi,
Neyron tarmog',
Yo'l nuqtasi.

Annotatsiya. Yo'l belgisi - yo'lovchilar va haydovchilarga yo'nalishni ko'rsatish uchun ishlatiladigan belgi. Yo'l belgilari yo'llarni boshqarishda, transport oqimini tartibga solishda va harakat xavfsizligini ta'minlashda muhim rol o'ynaydi. Yo'l tarqatuvchi belgilar - ko'rsatilgan yo'nalishni tartibga soluvchi belgilar, masalan, to'g'ri yo'nalish, o'ngga o'tish, chapga o'tish, tekislik, joylashuv va boshqalar. Og'irlik chegarasini taqiqlaydigan belgilar - og'irlik chegaralarini ko'rsatadigan belgilar, ko'p tovuq, kamyonlar va avtomobillarga qo'yiladi. Yo'l markalash belgilar - yo'l ko'rsatilgan yo'nalishlarda ushbu belgilar orqali markalanishi mumkin. Masalan, shahar tashqi yo'llarda va keng ko'layotgan yo'llarda yo'l markalash belgilari ko'rsatiladi. Yo'llantiruvchi belgilar - yo'lga kirish va chiqish joylarini, shuningdek, tashqi yo'ldan kengaytirilgan yo'llarni ko'rsatadigan belgilar. Harorat belgilar - quyidagi belgilar haroratni ta'riflash uchun foydalaniladi, masalan, yo'l shtatsiyasi, temperatur belgisi va boshqalar. O'zgaruvchan harorat belgilar - asosiy harorat belgilaridan farqli ravishda, ushbu belgilar haroratning o'zgarishiga ko'ra ko'rsatiladi.

МАНИПУЛИРОВАНИЕ ИЗОБРАЖЕНИЯМИ С ПОМОЩЬЮ ЯЗЫКА ПРОГРАММИРОВАНИЯ ПРИ РАСПОЗНАВАНИИ ДОРОЖНЫХ ЗНАКОВ

Karimova Zilola, Стажер-преподаватель Денауского института предпринимательства и педагогики

Xaydarov Sherali, докторант Денауского института предпринимательства и педагогики

Doniyorova Gulshan, Стажер-преподаватель Денауского института предпринимательства и педагогики

Ключевые слова:
путевая точка,
OpenCV,

Аннотация. Дорожный знак-это знак, используемый для указания направления движения пассажирам и водителям. Дорожные знаки играют важную роль в управлении дорогами, регулировании транспортных потоков и обеспечении безопасности движения.

нейронная сеть,
путевая точка.

Классификация дорожной разметки обычно выглядит следующим образом: дорожные знаки-это знаки, которые регулируют указанное направление, например, правое направление, поворот направо, поворот налево, уровень, местоположение и т. д. Знаки ограничения веса запрещены. Знаки, указывающие на ограничения веса, размещены на большинстве грузовиков, грузовиков и легковых автомобилей. Знаки дорожной разметки - этими знаками можно размечать дорогу в указанном направлении. Например, дорожная разметка отображается на городских дорогах и автомагистралях. Указательные знаки - знаки, указывающие места въезда и выезда на дорогу, а также проходящие от внешней дороги дороги. Символы температуры - для описания температуры используются следующие символы, например, промежуточная станция, символ температуры и т. д. Символы переменной температуры. В отличие от основных символов температуры, эти символы отображаются в соответствии с изменениями температуры.

MANIPULATION OF IMAGES USING PYTHON PROGRAMMING LANGUAGE IN ROAD SIGN RECOGNITION

Karimova Zilola, Trainee-teacher of the Denau Institute of
Entrepreneurship and Pedagogy

Xaydarov Sherali, Doctoral student of Denau Institute of
Entrepreneurship and Pedagogy

Doniyorova Gulshan, Trainee-teacher of the Denau Institute of
Entrepreneurship and Pedagogy

Keywords:
Waypoint,
OpenCV,
Waypoint,
Neural Network,
Waypoint.

Annotation. A road sign is a sign used to indicate the direction to passengers and drivers. Road signs play an important role in managing roads, regulating traffic flow and ensuring traffic safety. The classification of road markings is usually as follows: Road signs are signs that regulate the direction shown, for example, the right direction, turn right, turn left, level, location, etc. No Weight Limit Signs - Signs indicating weight limits are posted on most trucks, trucks, and cars. Road marking signs - the road can be marked by these signs in the indicated directions. For example, road markings are displayed on city roads and highways. Directional signs - signs that indicate the entrance and exit points to the road, as well as the roads extended from the external road. Temperature symbols - the following symbols are used to describe the temperature, for example, a way station, a temperature symbol, etc. Variable temperature symbols - Unlike basic temperature symbols, these symbols are displayed according to temperature changes. For example, motor temperature, battery temperature, etc. Safety signs - used to describe road safety, such as giving the right direction to multiple vehicles, etc. These classifications may depend on the general classification of road signs, the type and location of their use.

Kirish

Tadqiqot metodlari. Python dasturlash tilida, yo'l belgilarini aniqlashda oldindan belgilangan xususiyatlari bilan bir qator boshqa funktsiyalarga ega. Quyidagi metodlar yordamida yo'l belgilari aniqlash mumkin [1]:

os kutubxonasi: Bu kutubxona operatsion tizimga mos keladigan funksiyalarni o'z ichiga oladi. **os** kutubxonasi yordamida quyidagi metodlarni ishlatish mumkin:

os.path.exists(path): berilgan yo'l belgisining mavjudligini tekshiradi.

os.path.abspath(path): berilgan yo'l belgisining mutlaq yo'li (absolute path)ni qaytaradi.

os.path.dirname(path): berilgan fayl yoki direktoriya nomidan o'ng tarafidagi yo'l belgisini qaytaradi.

os.path.basename(path): berilgan fayl yoki direktoriya nomidan chap tarafidagi nomni qaytaradi.

pathlib kutubxonasi: Bu kutubxona yo'l belgilari bilan ishlash uchun oddiy va qulay interfeysni taklif qiladi. **pathlib** kutubxonasi yordamida quyidagi metodlarni ishlatish mumkin:

Path.exists(): berilgan yo'l belgisining mavjudligini tekshiradi.

Path.resolve(): berilgan yo'l belgisining mutlaq yo'li (absolute path)ni qaytaradi.

Path.parent(): berilgan fayl yoki direktoriya nomidan o'ng tarafidagi yo'l belgisini qaytaradi.

Path.name(): berilgan fayl yoki direktoriya nomidan chap tarafidagi nomni qaytaradi.

glob moduli: Bu modul fayl nomlarini tartiblash uchun ishlatiladi. **glob** modulini quyidagi kabi yordamida ishlatish mumkin:

glob.glob(pattern): berilgan pattern kabi mos keladigan fayllarni ro'yxatini qaytaradi [1].

Tadqiqot natijalari. Yo'l belgilarini aniqlash uchun bir nechta yo'l belgilarini aniqlash dasturlari mavjud. Ayrim misollar:

OpenCV: OpenCV - bu Python dasturlash tilidan foydalangan holda tasvirni manipulyatsiya qilish uchun kutubxona. OpenCV yo'l belgilarini tanib olish uchun ko'plab xususiyatlar va algoritmlarni o'z ichiga oladi [2].

TensorFlow Object Detection API: TensorFlow Object Detection API - TensorFlow yordamida ob'ektni aniqlash uchun API. Bu API yo'l belgilari kabi ob'ektlarni tanib olish uchun oldindan o'rgatilgan modellar va funksiyalarni o'z ichiga oladi.

YOLO (Faqat bir marta qaraysiz): YOLO ob'ektni aniqlash algoritmi bo'lib, yo'nalish nuqtalarini tanib olish uchun ishlatilishi mumkin. YOLO tasvirni bir marta qayta ishlaydi va bir vaqtning o'zida bir nechta ob'ektlarni aniqlash uchun chuqur o'rganish modelidan foydalanadi [3].

MobileNet: MobileNet kichikroq o'lchamli modellardan foydalangan holda mobil qurilmalarda ob'ektlarni aniqlash uchun chuqur o'rganish modelidir. MobileNet yo'l belgilari kabi ob'ektlarni tanib olish uchun ishlatilishi mumkin[4].

Ushbu dasturlar yo'l belgilarini tanib olish uchun ishlatiladigan turli usullardan foydalanadi. Biroq, barcha dasturlar yo'l nuqtasini aniqlash uchun oldindan tayyorlangan modeldan foydalanadi. Ushbu modellar yo'l belgilari kabi ob'ektlarni tanib olish uchun minglab tasvirlar bo'yicha o'qitiladi va keyin dasturlarga kiritiladi [5]. Shu tarzda, dasturlar yo'l belgilari kabi ob'ektlarni tanib olish uchun aniq natijalarni

berishi mumkin. Yo'l belgilarini aniqlash uchun bir qator yo'l belgilarini aniqlash dasturlari va dasturlari mavjud. Ayrim misollar:

Yo'l belgilarini aniqlash: Bu dastur yo'l belgilarini aniqlash uchun foydalaniladigan tasvirni qayta ishlash dasturidir. Dastur belgilarni shakl, rang va belgi kabi xususiyatlaridan foydalangan holda taniydi va toifalarga ajratadi [6].

signIN: signIN- yo'l belgilarini tanib olish uchun mo'ljallangan mobil ilova. Ilova foydalanuvchi telefonining kamerasi yordamida belgilarni aniqlaydi va belgining ma'nosini aniqlaydi va uni foydalanuvchiga ko'rsatadi. Waveshare OpenCV yo'l belgilarini aniqlash to'plami: Ushbu to'plamdan foydalanib, siz yo'l belgilarini aniqlash uchun Raspberry Pi dan foydalanishingiz mumkin. To'plam OpenCV va Python dasturlash tilidan foydalangan holda ishlab chiqilgan va yo'l nuqtalarini aniqlash va tasniflash uchun tasvirni qayta ishlash usullaridan foydalanadi. Ko'p miqyosli konvolyutsion neyron tarmoqlar bilan yo'l belgilarini aniqlash: Ushbu dastur ob'ektni aniqlash va tasniflash uchun chuqur o'rganish usullaridan foydalanadi. Dastur CNN (Convolutional Neural Network) ni yo'l belgilari tasvirlari yordamida o'qitadi va undan keyin belgilarni tanib olish uchun foydalanadi. Ushbu dasturlar yo'l belgilarini tanib olish uchun turli usullardan foydalanadi. Biroq, ularning barchasi yo'l nuqtasini aniqlash uchun oldindan o'rgatilgan modeldan foydalanadi. Ushbu modellar yo'l belgilari kabi ob'ektlarni tanib olish uchun minglab tasvirlar bo'yicha o'qitiladi va keyin dasturlarga kiritiladi. Shu tarzda, dasturlar yo'l belgilari kabi ob'ektlarni tanib olish uchun aniq natijalarni berishi mumkin [7]. Yo'l belgilarini tanib olish uchun ko'plab algoritmlar va usullar qo'llaniladi. Ulardan ba'zilari:

Rangga asoslangan yondashuv: Bu yondashuv yo'l belgilarining ranglariga qarab belgilarni aniqlaydi. Misol uchun, bu yondashuv qizil belgilarni topish uchun ishlatiladi, chunki qizil belgi ko'zga ko'rinadigan bo'lishi kerak. Rangga asoslangan yondashuvlar, ayniqsa, kam yorug'lik sharoitida yoki tasvir sifati yomon bo'lsa foydali bo'lishi mumkin. **Shaklga asoslangan yondashuv:** Bu yondashuv yo'l belgilarining shakllariga asoslangan belgilarni aniqlaydi. Misol uchun, agar siz aylana belgisi odatda taqiqlash belgisi ekanligini bilsangiz, bu yondashuv aylanma belgilarni topish uchun ishlatilishi mumkin. **Neyron tarmog'iga asoslangan yondashuv:** Bu yondashuv sun'iy neyron tarmoqlari (ANN) yordamida yo'l nuqtalarini aniqlaydi. ANN yo'l belgilari tasvirlarini o'rganish va tanib olish orqali belgilarni tasniflaydi. Ushbu yondashuv yanada murakkab yo'l belgilarini tanib olish imkonini beradi. **Tasvirni qayta ishlashga asoslangan yondashuv:** Bu yondashuv yo'l belgilari tasvirlarini qayta ishlash orqali belgilarni tan oladi. Kenarlarni aniqlash, gistogrammani tenglashtirish, kontrastni yaxshilash va filtrlash kabi tasvirni qayta ishlash usullaridan foydalangan holda tasvirlardagi yo'l belgilarini ajratib ko'rsatish mumkin. Ushbu usullarning ko'pchiligi oldindan tayyorlangan modellar yordamida yo'l belgilarini aniqlash uchun birgalikda ishlatilishi mumkin. Oldindan o'qitilgan modellar yo'nalish nuqtalarini tanib olish uchun minglab tasvirlar bo'yicha o'qitiladi va keyin yo'l nuqtalarini tanib olish uchun ishlatiladi. Ushbu yondashuv yo'l belgilarini aniq tanib olish va tasniflash uchun juda samarali. Yo'nalish nuqtalarini tanib olish uchun tizim yoki dasturiy ta'minotni ishlab chiqish uchun siz quyidagi amallarni bajarishingiz mumkin [8]:

Ma'lumot to'plash: Yo'l belgilarini aniq tanib olish uchun sizga turli xil sharoitlarda olingan etarli miqdordagi yo'l belgilari tasvirlari kerak bo'ladi. Shuning uchun siz turli yo'llarning turli uchastkalarida, turli yorug'lik sharoitida, turli burchaklardan va turli masofalardan yo'l belgilarining rasmlarini to'plashingiz kerak.

Ma'lumotlarni oldindan qayta ishlash: Ma'lumotlarni oldindan qayta ishlash bosqichi ma'lumotlarni tahlil qilish va qayta ishlash uchun tayyorlash imkonini beradi. Ushbu bosqichda siz tasvirlarning o'lchamlari, kontrasti va rang sozlamalarini optimallashtirishingiz kerak bo'lishi mumkin. Yo'nalish nuqtasini aniqlash: Ushbu bosqichda siz tasvirdagi yo'nalish nuqtalarini aniqlash uchun algoritmdan foydalanishingiz kerak bo'ladi. Ushbu algoritm tasvirdagi rang, shakl va belgilar asosida yo'l belgilarini aniqlaydi. Yo'l punkti tasnifi: Yo'nalish nuqtasi aniqlangandan so'ng, siz tasniflash bosqichiga o'tasiz. Ushbu bosqichda siz aniqlangan yo'l belgisi tasvirining ma'nosini aniqlaydigan algoritmdan foydalanishingiz kerak bo'ladi. Ushbu algoritm belgining rangi, shakli va belgisi asosida yo'l belgisini aniqlaydi. Natijani ko'rsatish: Yo'nalish nuqtasini aniqlash va tasniflash bosqichlaridan so'ng siz natijalarni foydalanuvchilarga ko'rsatish uchun interfeys yoki dasturni ishlab chiqishingiz mumkin. Ushbu bosqichlarni bajarish sizga to'g'ri ishlaydigan yo'l belgilarini aniqlash tizimini yaratishga yordam beradi. Biroq, yo'l belgilarini tanib olish texnologiyasi hali ishlab chiqilayotganligi sababli, aniq natijalarga erishish uchun turli xil algoritm va usullardan foydalanish kerak bo'lishi mumkin [9].

Yo'nalish nuqtalarini aniqlash uchun Python yordamida dastur yaratish mumkin. Buning uchun quyidagi amallarni bajarishingiz mumkin:

- 1) Kerakli kutubxonalarni o'rnatish: OpenCV va Numpy, yo'l nuqtalarini aniqlash uchun zarur bo'lgan tasvirni qayta ishlash kutubxonalarini o'rnatish.

import cv2

import numpy as np

- 2) Rasmlarni o'qish: Yo'nalish nuqtasini aniqlash uchun avval rasmlarni o'qishingiz kerak. Misol tariqasida, quyidagi kod "stop.jpg" nomli rasmni o'qiydi:

img = cv2.imread("stop.jpg")

- 3) Kulrang tasvirni yarating: Rasmga ishlov berishdan oldin uni kul rangga aylantiring:

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

- 4) Yo'nalish nuqtasini aniqlash: Yo'nalish nuqtasini aniqlash uchun algoritmdan foydalaning. Masalan, CascadeClassifier sinfidan foydalanib, "stop_sign.xml" nomli XML faylidan foydalanishingiz mumkin:

cascade = cv2.CascadeClassifier('stop_sign.xml')

signs = cascade.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=4, minSize=(30, 30), flags=cv2.CASCADE_SCALE_IMAGE)

- 5) Yo'l nuqtasini ramkaga qo'ying: yo'l nuqtasi aniqlangandan so'ng, ramkalash amalga oshirilishi mumkin:

for (x, y, w, h) in signs:

`cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)`

- 6) Natijalarni ko'rsatish: Nihoyat, natijalarni ko'rsatish uchun quyidagi koddan foydalanishingiz mumkin:

`cv2.imshow('Stop sign detection', img)`

`cv2.waitKey(0)`

`cv2.destroyAllWindows()`

Python-da yo'llarni boshqarish uchun ishlatiladigan bir nechta turli xil modullar mavjud. Ushbu modullar orasida eng keng tarqalganlarini os, pathlib va glob modullari deb hisoblash mumkin. Ushbu modullardan quyidagi tarzda foydalanish mumkin:

1. OS moduli:

```
import os
# Joriy ishchi katalogni oling
current_directory = os.getcwd()
# yangi katalog yarating
os.mkdir("yangi_katalog ")
# Kataloglarni birlashtirish
path = os.path.join(current_directory, " yangi_katalog ")
# Faylni o'chirish
os.remove("dosya.txt")
```

2. Pathlib moduli:

```
from pathlib import Path
# Joriy ishchi katalogni olish
current_directory = Path.cwd()
# yangi katalog yarating
new_directory = current_directory / " yangi_katalog "
new_directory.mkdir()
# Kataloglarni birlashtirish
path = current_directory.joinpath("yangi_katalog ")
# Faylni o'chirish
file_to_delete = current_directory / "dosya.txt"
file_to_delete.unlink()
```

3. Globus moduli:

```
import glob
# Muayyan naqsh bo'yicha fayl yo'llari ro'yxatini olish
file_list = glob.glob("/path/to/files/*.txt")
# Barcha fayllar ro'yxati
all_files = glob.glob("*")
```

Python dasturlash tilini foydalanib, Yo'l belgilarini tasvirlash uchun "opencv" nomli kutubxona yoki "Pillow" nomli kutubxonalardan foydalanishingiz mumkin.

Quyidagi kodni bajarib, tasvirlarni ko'chirib olish, kesish, qayta o'lchash va ranglarni o'zgartirish kabi bir qancha amallarni bajaringiz mumkin [2]:

```
import cv2
from PIL import Image
# Tasvir yuklash
image = cv2.imread("image.jpg")
# Tasvir kesish
cropped_image = image[100:300, 200:400]
# Tasvir o'lchamlarini o'zgartirish
resized_image = cv2.resize(image, (500, 500))
# Tasvirda ranglarni o'zgartirish
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
# Tasvirlarni ko'rsatish
cv2.imshow("Original Image", image)
cv2.imshow("Cropped Image", cropped_image)
cv2.imshow("Resized Image", resized_image)
cv2.imshow("Gray Image", gray_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Bu kod yuklangan tasvirni "image.jpg" nomi bilan yuklaydi. Quyidagi amallarni bajardikdan keyin, tasvirlar quyidagi nomlar bilan ko'rsatiladi:

- "Original Image" - asl tasvir
- "Cropped Image" - kesilgan tasvir
- "Resized Image" - o'lchami o'zgartirilgan tasvir
- "Gray Image" - tasvirda ranglar o'zgartirilgan tasvir (siyoh-oq)

Ushbu kod yordamida, tasvirlarni ko'chirish, kesish, o'lchash va ranglarni o'zgartirish kabi turli to'garaklar amalga oshirilishi mumkin.

Yuqoridagi qadamlar Python yordamida yo'nalish nuqtalarini aniqlash uchun dast yaratishga yordam beradi. Biroq, yo'l belgilarini aniqlash va tasniflash juda murakkab jarayondir, shuning uchun aniqroq natijalarga erishish uchun turli xil algoritm va usullardan foydalanish kerak bo'lishi mumkin.

Xulosa qilib aytganda Python tilining kuchli kutubxonalaridan biri "opencv" va boshqa "Pillow" kutubxonalar tasvirlarni manipulyatsiya qilish uchun foydalaniladi. Siz tasvirlarni yuklab, kesib olish, o'lchamini o'zgartirish va ranglarni o'zgartirish kabi amallarni bajarishni o'rganishingiz mumkin. Ushbu yuoridagi kod misoli yo'l belgilarini tushunishga yordam bergan holda tasvirlarni boshqa dasturlash vazifalari uchun ham foydalanishingiz mumkin.

Foydalanilgan adabiyotlar:

1. O'rinov, D. O., & Maxmudov, O. E. (2022). Improving Traffic Prevention Of Road Traffic Accidents Yol Transport Hodisasi Sodir Bolganda Yollarda Uchraydigan Tirbantlikni Oldini Olish Ishlarini Takomallashtirish. Innovative Technologia: Methodical Research Journal, 3(5), 1-8.
2. Odilov, N., & Muxtorov, A. (2022). Avtomobillar harakatini xavfsiz tashkil etishda GPS tizimlaridan foydalanish samaradorligi. Academic research in educational sciences, 3(2), 298-303.
3. Gaffarov, M. (2023). O 'Zbekistonda Transport Oqimini Tartibga Solish. Yo 'L Harakati Xavfsizligini Ta'minlash Va Tirbandlikni Oldini Olish. Евразийский Журнал Технологий И Инноваций, 1(2), 70-74.
4. Tursunboyev, F. (2022). Avtomobil Yo'llarda Piyodalar Harakat Havfsizligini Ta'minlashning Zamonaviy Yechimlari. Science And Innovation, 1(A8), 548-559.
5. Ziyatovich, M. F., & Islom o'g'li, X. S. (2023). Sun'iy intellekt va uning ta'lim sohasiga alohida murojaat qilgan holda turli sohalardagi qamrovi. Образование наука и инновационные идеи в мире, 16(3), 16-19.
6. Мамажанов, Р. Я., & Хайдаров, Ш. И. (2022). РАЗРАБОТКА Платформы Инновационного Подхода При Подготовке Научных Статей На Международном Уровне. Central Asian Journal Of Mathematical Theory And Computer Sciences,
7. Мамажанов, Р., & Хайдаров, Ш. (2022). Цифровая Экономика В Повышении Экономики И Влияния Вузов Расположение В Центре. Central Asian Journal Of Mathematical Theory And Computer Sciences, 3(12), 270-275.
8. Мамажанов, Р. Я., & Хайдаров, Ш. И. (2022). Разработка Методов И Алгоритмов Для Распознавание Дорожных Знаков. Central Asian Journal Of Mathematical Theory And Computer Sciences, 3(10), 50-57.
9. Хайдаров, Ш. И. (2022). Разработка Программного Обеспечения Qr-Code Для Формирования Электронных Баз Данных И Систем Управления Высшими Учебными Заведениями. Central Asian Journal Of Mathematical Theory And Computer Sciences, 3(1), 3-8.
10. Mamajanov, R. Y., & Haydarov, S. I. (2022). Korxona Va Tashkilotlarda Elektron Tabel Va Samaradorlikning Muhim Ko'rsatkichini Belgilash. Central Asian Research Journal For Interdisciplinary Studies (Carjis), 2(5), 281-289.