DOI:10.33942/sit1197 **УДК 004.45(045)**

КООПСУЗДУК ЖАНА ТААНУУ СИСТЕМАЛАРЫН ИШТЕП ЧЫГУУДА DLIB КИТЕПКАНАСЫНЫН КОЛДОНУЛУШТАРЫ

[1]Темиров Майрамбек Акбагышович, [2]Акылбекова Айдана Акылбековна, [3]Мамытова Каныкей Турдубаевна, [4]Молдокеев Тынчтык Бекмуратович, [5]Таштемиров Арсланбек Адамович,

[1]Ж.Баласагын атындагы КУУнун Компьютердик технологиялар жана жасалма интеллект институтунун информатика жана эсептөө техникасы кафедрасы, доцент.m.temirov2015@gmail.com,

[2]Ж.Баласагын атындагы КУУнун Колдонмо математика жана информатика программасы, магистрант, Ai.suerkulovava@gmail.com,

[3]Эл аралык инновациялык технологиялар университети, магистрант, Кыргыз Республикасы , Бишкек шаары mamytova@list.ru,

[4]Эл аралык инновациялык технологиялар университети, магистрант, Кыргыз Республикасы , Бишкек шаары, Moldokeev@mail.ru

[5]Эл аралык инновациялык технологиялар университети, магистрант, Кыргыз Республикасы , Бишкек шаары, г.Бишкек, Tashtemirov@mail.ru

***Аннотация**. Макалада таануу системаларынын колдонуу аймактары баяндалган. Таануу жана авто унааны айдоочунун коопмсуздугун көзөмөлдөө системасы dlib китепканасынын колдонулушу менен иштелип чыккан. Таануу жана көзөмөлдөө системасын иштеп чыгуу алгоритми жана системанын иштөө принциптери каралган.*

Ачкыч сөздөр: Маалыматтык коопсуздук, коопсуздук системалары, жүздү таануу технологиялары, биометрия, *алгоритм*.

ПРИМЕНЕНИЕ БИБЛИОТЕКИ DLIB ПРИ РАЗРАБОТКЕ СИСТЕМ БЕЗОПАСНОСТИ И РАСПОЗНАВАНИЯ

[1]Темиров Майрамбек Акбагышович, *кафедра информатики и вычислительной техники Института компьютерных технологий и искусственного интеллекта КНУ им.Ж.Баласагына, доцент, m.temirov2015@gmail.com,*

[2]Акылбекова Айдана Акылбековна, *кафедра «Прикладная математика и информатика» КНУ им.Ж.Баласагына, магистрант, Ai.suerkulovava@gmail.com,*

[3]Мамытова Каныкей Турдубаевна, *Международный университет инновационных технологий, магистрант, Кыргызская Республика, г.Бишкек, mamytova@list.ru*

[4]Молдокеев Тынчтык Бекмуратович, *Международный университет инновационных технологий, магистрант, Кыргызская Республика, г.Бишкек, Moldokeev@mail.ru*

[5]Таштемиров Арсланбек Адамович, *Международный университет инновационных технологий, магистрант, Кыргызская Республика, г.Бишкек, Tashtemirov@mail.ru*

***Аннотация**. В статье приводятся основные принципы создания систем распознавания и обсуждаются примеры практического применения библиотеки dlib в различных областях. Определены области эффективного использования рассмотренной технологии. Разработана система мониторинга за состоянием водителя и система распознование лиц у входа здания с применением библиотеки dlib.*

***Ключевые слова**: Информационная безопасность, системы безопасности, технологии распознавания лиц, биометрия, алгоритм.*

APPLICATION OF THE DLIB LIBRARY IN THE DEVELOPMENT OF SECURITY AND RECOGNITION SYSTEMS

[1]**Temirov Mayrambek Akbagyshovich** – *docent of the Department of Informatics and Computer Science at the Institute of Computer Technologies and Artificial Intelligence of the KNU named after Zh. Balasagyn*, *m.temirov2015@gmail.com,*

[2]**Akylbekova Aidana Akylbekovna -** *Master of Applied Mathematics and Informatics program of the KNU named after Zh. Balasagyn, Ai.suerkulovava@gmail.com,*

[3]***Mamytova Kanykey Turdubaevna - International University of Innovative Technologies, Master's student,*** mamytova@list.ru

[4]***Moldokeev Tynchtyk Bekmuratovich - International University of Innovative Technologies, Master's student,*** Moldokeev@mail.ru

[5]***Tashtemirov Arslanbek Adamovich - International University of Innovative Technologies, Master's student,*** Tashtemirov@mail.ru.

Annotation.*The article provides the basic principles for creating recognition systems and discusses examples of practical application of the dlib library in various fields. Areas of effective use of the technology considered are identified. A driver condition monitoring system and a face recognition system at the building entrance using the dlib library have been developed.*

Keywords*: Information security, security systems, facial recognition technologies, biometrics, algorithm.*

Киришүү. Азыркы дүйнөнү компьютердик технологияларсыз элестетүү мүмкүн эмес. Таануу системалары ар тармакта колдонулуп жүрөт. Таануу системаларын ар кандай фреймворк, китепканалардын жардамы түзүлүп жүрөт. Бирок кайсынысы эффективдүүрөөк жана жеңил? Бул системалар кайсы тармакта кандай колдонулат жана аларды кантип иштеп чыгуу керек? - деген сыяктуу суроолорго, бул макалада кыскача маалымат берилет. Макалада Dlib китепканасын колдонулуп таануу жана коопсуздук системалары түзүлгөн. Dlib башка OpenFace, Keras FaceNet системалары менен иштөө убактысы, таануу тактыгы сыяктуу жыйынтыктары аркылуу салыштырылганда жакшы көрсөткүчтү көрсөткөн [5]. Dlib – объектти аныктоо, жүздү таануу, эмоцияларды талдоо жана башка көптөгөн сүрөттөрдү жана видео мүмкүнчүлүктөрдү камсыз кылган күчтүү компьютердик көрүү жана машина үйрөнүү куралы. Макалада Dlib китепканасынын иштөөсүнүн негизги принциптери, анын артыкчылыктары жана мүмкүнчүлүктөрү, ошондой эле коопсуздук жана таануу системаларында колдонуу мисалдары талкууланат.

Жүздү таануу системасынын маселеси болуп сүрөттөгү адамдын жүзүн автоматтык түрдө белгилеп жана базадагы биометрикалык берилиштер менен салыштырып талдоо жолу менен адамды таануу болуп эсептелет [1]. Жөнөкөй сөз менен айтканда, ар бир адамдын жүзү өзгөчө түзүлүшкө ээ. Таануу программасынын негизги максаты объекти сүрөттөн же видеоагымдан алып, жүктөлгөн маалымат базасы менен салыштыруу жана жолу менен аныктоо болуп эсептелет. Жүздү таануу системалары уурулук, мыйзамсыз иштерди аныктоо жана аларды алдын алуу сыяктуу

иш-аракеттерди аныктоодо көмөк көрсөтөт. Шектүү адамдар таанылып калса, программа дароо укук коргоо органдарына сигнал жиберет.

Мекемелерге кирүүнү же маалымат базаларына кирүүнү башкаруу.

Жүздү таануу технологиясы мекеменин аймагына тез жана контактсыз кирүүнү, аймактын ичиндеги кыймылдарды башкарууну, жумуш убактысын эсепке алуу сыяктуу иш-аракеттерди камсыздайт.

Авто унаа айдоочунун коопсуздугун көзөмөлдөө

Узак сапарларда же түнкү монотондуу айдоодо айдоочунун чарчоосуна алып келет. Бул анын реакциясынын төмөндөшүнө жана айдап бара жатып уктап калуу коркунучу жогорулайт. Кырсыкты алдына алуу максатында олуттуу кырсыкка алып келиши мүмкүн. Мониторинг жана контролдоо үчүн

Таануу жана анлиздөө системасынын жардамы менен айдоочунун чарчоо деңгээлин мониторинг кылса болот. Чарчоону контролдоо үчүн төмөнкү белгилерди көзөмөлдөө керек:

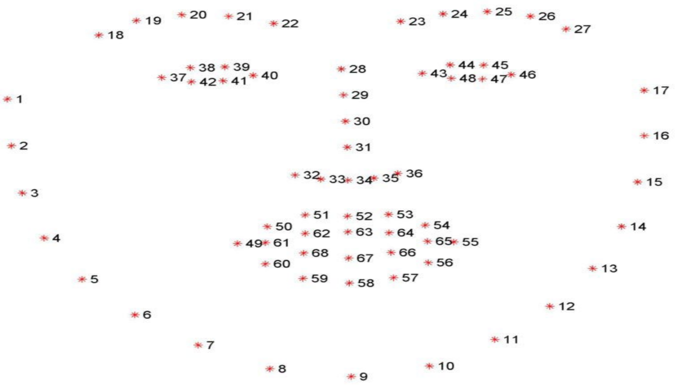
- Көздүн ирмелиш
- Баштын абалы.

Ошондой эле жогорку системаны авто унааны уурдап айдап кетүүдөн сактаныш үчүн дагы колдонсо болот. Эгерде чоочун адам унааны башкаруучу орунга отурган адамдын жүзү аныкталып, база менен салыштырылат, чоочун адам болсо система кыймылдаткычты өчүрүп, бөгөттөп салат.

Жүздү таануунун заманбап ыкмаларына жана алгоритмдерине токтоло кетели. Система төмөнкү алгоритм боюнча адамды аныктайт:

- Сүрөттөгү жүздү аныктоо
- Жүздөгү өзгөчөлүктөрдү аныктоо
- Базасындагы жүздөр менен салыштыруу үчүн алынган берилиштерди трансформациялоо
- Салыштыруу жана жыйынтык чыгаруу

Жүздөрдү аныктоо үчүн көбүнчө учурда Виола-Джонс ыкмасы [2], ал эми жүздү идентификациялоо үчүн – ASM (Active Shape Models) модели колдонулуп жүрөт. Бул ыкмалардын негизги идеясы жүздөгү антропометриялык чекиттердин ортосундагы статистикалык байланыштарды эске алуу болуп саналат. Ар бир сүрөттөгү чекиттер бирдей тартипте номерленет. Алардын жайгашуусуна карата салыштыруу жүгүзүлөт. 1-сүрөттө жүздүн номерленген чекиттери мисал катары көрсөтүлгөн.



1-сүрөт. Жүздү аныктоодо колдонулуучу номерленген чекиттер.

Жүздөрдү салыштыруу үчүн адамдын 2D чекит алкагындагы сүрөттү колдонсок болот. Бирок 3D артыкчылыктуу болуп саналат.

Жүздү таануу системасын иштеп чыгуу үчүн dlib [5] и face_recognition [6] python китепканаларынын колдонулуштарына токтололу.

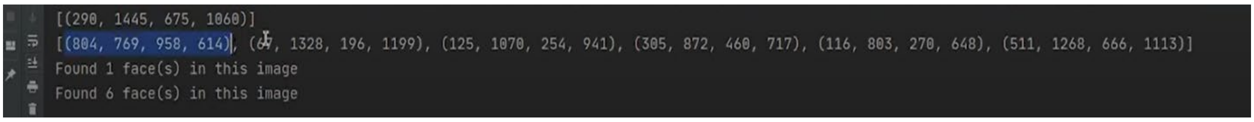
Dlib бул ачык коддуу C++ тилинде иштелип чыккан китепкана. Dlib китепканасы башка OpenFace, Keras FaceNet [6] системалары менен иштөө убактысы, таануу тактыгы сыяктуу жыйынтыктары аркылуу салыштырылганда жакшы көрсөткүчтү көрсөткөн [5]. Ошол себептен бул коопсуздук системасын иштеп чыгууда Dlib китепканасы тандалып алынды. Бул китепкана көптөгөн ар кандай даяр нейрондук тармактардын бир нече ишке ашырылышы, матрицалык геометриялык функциялар, градиенттик түшүү сыктуу ыкмаларды жана аспатарды камтыйт.

Бул иште get_frontal_face_detector() функциясы колдонулган. Бул функция жүздү аныктоочу маалыматтарды кайтарат. detektor() функциясы кадрды кабыл алып, тизме түрүндө чыгарып берет. Face Recognition [6] китепканасы, dlib китепканасынын негизинде python тилинде иштелип чыккан. Жүздү таанууда кеңири колдонулат. Системаны иштеп чыгуу этаптарына токтолсок, биринчи face_recognition китепканасын импорттоп алып, андан соң сүрөттөрдү жүктөөчү ыкманы түзөбүз:

```
import face_recognition


def face_rec():
    gal_face_img = face_recognition.load_image_file("img/gal1.jpg")
    gal_face_location = face_recognition.face_locations(gal_face_img)

    justice_league_img = face_recognition.load_image_file("img/justice_league_actors.jpg")
    justice_league_faces_locations = face_recognition.face_locations(justice_league_img)

    print(gal_face_location)
    print(justice_league_faces_locations)
    print(f"Found {len(gal_face_location)} face(s) in this image")
    print(f"Found {len(justice_league_faces_locations)} face(s) in this image")


def main():
    face_rec()
```

Жогорудагы коддо face-rec ыкмасы түзүлгөнү көрсөтүп турат. Ыкмада load-image-file функциясы gall.img сүрөтүн img папкасынан жүктөйт. Андан соң, биз жүздүн

координаттарды face-locations функциясынын жардамы менен аныктайбыз жана жыйынтыктарды консолго чыгарабыз. Эгер бул кодду иштетсек, биз сүрөттөгү жүздүн координаттарын алабыз



Эми табылган жүздөрдү рамка менен белгилейбиз. Бул үчүн, pillow китепканасын импорттоп жана for циклин колдонобуз. Аягында save() ыкмасынын жардамы менен сүрөткө ат берип сактайбыз. Бул операциянын коду төмөндө келтирилди

```
from PIL import Image, ImageDraw
image = face_recognition.load_image_file("pictures/b2.jpg")
face_locations = face_recognition.face_locations(image, model="cnn")
rectan=Image.fromarray(image)
draw1=ImageDraw.Draw(rectan)
for(top, right, bottom,left) in face_locations:
   draw1.rectangle(((left, top),(right, bottom)), outline=(255,255,255), width=4)
del draw1
rectan.save("pictures/r2.jpg")
# print(face_locations)
```

Кодду иштеткенден кийин сүрөттөгү жүздөр тик бурчтук менен белгиленип калат. Коддо image өзгөрмөсү сүрөттүн жайгашуу жолун көрсөтөт. faces-locations өзгөрмөсү болсо, image өзгөрмөсүндөгү маанилер аркылуу түзүлгөн координаттарды сактайт. Ал эми for циклдык оператору faces-locations өзгөрмөсүнүн маанилери аркылуу чектерди аныктап жана аларды rectan өзгөрмөсүнө ыйгарат. Жыйынтыгында жүздөр тик бурчтук менен белгиленген сүрөттү алабыз.

Эми кийинки этап – салыштыруу. Бизге керектүү адамдардын сүрөттөрү known_pictures папкасында жайгашкан. Ал эми, камерадан келген сүрөт unknown_pictures пакасына түшөт дагы жогорудагы папкадагы негизги сүрөт менен салыштырылат жана жыйынтыгы чыгат. Бул маселени чечүү үчүн known_image жана uknown_image параметрлери менен салыштыруу ыкмасын түзөбүз. Андан кийин img1 өзгөрмөсүн жарыялайбыз. Ал сүрөттүн жайгашкан жолун сактайт. Биз дагы бир өзгөрмө known_encoding өзгөрмөсүн түзөбүз. Бул face_encodings функциясы uknown_image деги берилиштерди коддойт. Ушул эле операцияларды экинчи сүрөткө кайталап жүргүзөбүз. Андан соң, compare_faces функциясын колдонуп, эки функциянын алынган натыйжаларын салыштырабыз. Эгер бул адам негизги сүрөттө болсо, анда программа true маанисин кайтарыш керек, таанылбаса false маанисин кайтарат. Жогорудагы операциянын коду төмөндө келтирилди

```
import face_recognition
```

```
known_image = face_recognition.load_image_file("known_faces/E1.jpg")
unknown_image = face_recognition.load_image_file("unknown_faces/b7.jpg")
known_encoding = face_recognition.face_encodings(known_image)[0]
unknown_encoding = face_recognition.face_encodings(unknown_image)[1]
results = face_recognition.compare_faces([known_encoding], unknown_encoding)
print(results)
if results[0] == True:
   print("Бул адам таанылды")
else:
   print("Бул адам таанылган жок")
```

Жогорудагы коддун жыйынтыгында, эгер адам таанылган болсо true маанисин алып жатабыз. Бул маанини алганда эшикти ач, же унааны өчүр, же үн чыгар деген сыяктуу каалаган, бизге керектүү иш аракеттерди аткартсак болот.

Эми жогорудагы технологияларды колдонуу менен, унааны айдап бараткандa айдоочунун уктап кетүүсүн болтурбоо максатында айдоочунун көзүн мониторинг кылган программаны түзүү жолун төмөндө карап көрөлү. Биринчиден керектүү китепканаларды импорттоп алабыз жана коопсуздук сигналын жайгашуу жерин көрсөтөбүз

```
import face_recognition
import dlib
import cv
def runAlarm(path):
        playsound.playsound(path)
```

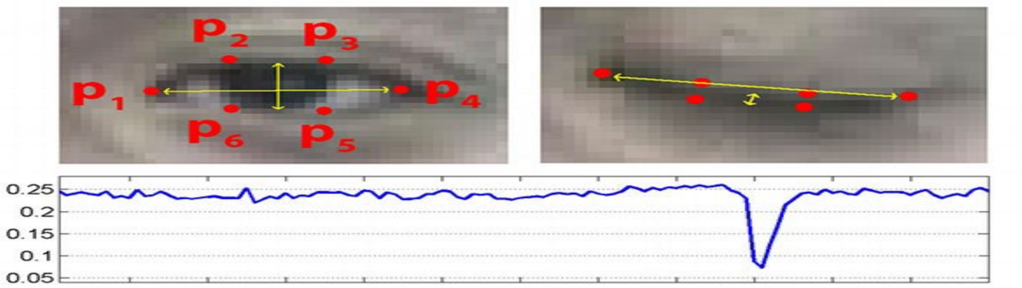
А жана В өзгөрмөлөрүн түзүп бул эки маанинин негизинде евклиддик аралыктарын, ошондой эле (x, y)-координатанын тик маанилери менен көздүн белгилерин эсептейбиз.

```
def findEAR(eye):
        A = dist.euclidean(eye[1], eye[5])
        B = dist.euclidean(eye[2], eye[4])
```

Андай соң, жогорудагыдай кылып туурасынын маанилери менен көздүн белгилерин аныктап алабыз

```
        C = dist.euclidean(eye[0], eye[3])
```

Бул маанилердин керектелиши 3-сүрөттө көрсөтүлүп турат.



3-сүрөт. Көздүн белгилеринин визуализациясы. Жогорку сол тарапта: көздүн ачык абалы. Жогорку оң тарапта: көздүн жабык абалы. Төмөндөгү графикте: Убакыттын өтүшү менен кабактардын катышы көрсөтүлгөн. Сызыктын төмөндөшү көз ирмелип жаткан учурду көрсөтөт.

EAR өзгөрмөсүн эсептөө жана жыйынтыкты кайтаруу

```
    ear = (A + B) / (2.0 * C)
    return ear
```

Эми THRESHOLD_EAR жана FRAMES_EAR деген эки констант аныктайбыз.

THRESHOLD_EAR көздүн ирмелишин көзөмөлдөө үчүн, ал эми FRAMES_EAR болсо ырааттуу кадрлардын санын аныкташ үчүн.

аныкталган. Эгерде көздүн тараптардын катышы THRESHOLD_EAR өзгөрмөсүндөгү мааниден төмөн түшүп калса, анда адам көзүн жумган кадрлардын санын санап баштоо керек. Эгерде адам көзүн жумган кадрлардын саны FRAMES_EAR деги мааниден дан ашса, сигнал иштеши керек.

Эксперименталдык түрдө THRESHOLD_EAR өзгөрмөсүнө 0,3 мааниси берилген. Бул маани ар кандай кырдаалдарда анализденип жакшы иштей тургандыгы аныкталган.

FRAMES_EAR мааниси да эксперименталдык түрдө 48ге коюлду - эгер адам 48 кадр катары менен көзүн жумса, сигнал иштейт. FRAMES_EAR көлөмүн азайтуу менен детекторду сезгичирээк кылып же тескерииснче аны көбөйтүү менен сезгичтигин азайтып салса болот.

```
THRESHOLD_EAR = 0.3
FRAMES_EAR = 48
COUNTER = 0
ALARM_ON = False
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor(args["shape_predictor"])
(leftEyeSt, leftEyeEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
(rightEyeSt, rightEyeEnd) = face_utils. FACIAL_LANDMARKS_IDXS["right_eye"]
# камера аркылуу берилген видеоагымдарды тартуу
capture=cv.CaptureFromCAM(0)
```

```
vs = VideoStream(src=capture).start()
time.sleep(1.0)
# видеоагымдын кадрлары боюнча цикл жүргүзүү
while True:
    frame = vs.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    # жүздү detector() функциясынын жардамы менен аныктоо
    rects = detector(gray, 0)
```

Литература

1. *Алгоритм распознавания лиц на основе контрольных точек // URL: https://api-2d3d-cad.com/face_recognition_with_opencv.*
2. *Yadav, Neha, Anupam Yadav, and Manoj Kumar. "History of neural networks." An Introduction to Neural Network Methods for Differential Equations. - Springer, Dordrecht, 2015. 13-15 pp.*
3. *Библиотека OpenCV: OpenCV modules. // Официальная документация. URL: https://docs.opencv.org/master.*
4. *Библиотека dlib C++ Library. // Официальная документация. URL: http://dlib.net.*
5. *E. Kalyashov. Comparative analysis of face recognition systems built using standard architecture blocks. Информационные технологии и телекоммуникации. - Т. 8, № 3.- Telecom IT, 2020. – С.95-101.*
6. *Ageitgey. Face_recognition // Официальный репозиторий. URL: https://github.com/ageitgey/face_recognition.*
7. *Yadav, Neha, Anupam Yadav, and Manoj Kumar. "History of neural networks." An Introduction to Neural Network Methods for Differential Equations. Springer, Dordrecht, 2015. 13-15 pp. 4. OpenCV: OpenCV modules. Retrieved April 14, 2021, from: https://docs.opencv.org/master/ 5. dlib C++ Library. Retrieved April 11, 2021, http://dlib.net/*
8. *Жамалова В.Ж. Информационные системы управления персаналом. Наука и инновационные технологии. № 2/2022 (23). С. 48-53.*
9. *Жамалова В.Ж., Абзалов Ф.С., Синельников В.Ю., Бакасов Т.А. Анализ конструкторов по созданию сайта. Наука и инновационные технологии. 2021. № 2 (19). С. 11-15.*
10. *Жамалова В.Ж., Каримбаев Т.Т., Раймжанова Ф.Р., Сатаров Э.С. Программа тестирования с мультимедийными компонентами на основе WPF. Наука и инновационные технологии. 2020. № 1 (14). С. 55-60.*
11. *Каримбаев Т.Т., Жамалова В.Ж., Муралева А.Ж., Трансформация спирометра ССП в цифровой прибор. Наука и инновационные технологии. 2021. № 1 (18). С. 106.*

Рецензент: к.т.н. Каримбаев Турсунжан Турашевич