

РАЗРАБОТКА МОДЕЛИ РАСПОЗНАВАНИЯ ОБЪЕКТОВ НА ИЗОБРАЖЕНИЯХ НА ОСНОВЕ МЕТОДА «TRANSFER LEARNING»

Яхшибоев Рустам Эркинбой угли

Ташкентский университет информационных технологий имени Мухаммада ал-Хоразмий

E-mail: yaxshiboyevrustam@gmail.com

Аннотация – данная статья посвящена изучению “трансфертного обучения” и разработке модели распознавания объектов на изображениях на основе метода “Трансфертного обучения”. Программное обеспечение и пакеты Python для создания квартирных вещей, алгоритм YOLOv5 и roboflow.com используемые данные веб-сайта. В процессе работы была разработана модель для распознавания объектов на изображениях на основе метода “Трансферного обучения”, и полученная модель была протестирована.

Ключевые слова – анализ, алгоритм, обнаружение, метод, трансферная обучения.

Abstract – this article is devoted to the study of “transfer learning” and the development of an object recognition model in images based on the “Transfer learning” method. Apartment creation software and Python packages, YOLOv5 algorithm and roboflow.com used website data. In the process of work, a model was developed for recognizing objects in images based on the “Transfer learning” method, and the resulting model was tested.

Keywords – analysis, algorithm, detection, method, transfer learning.

I. ВВЕДЕНИЕ.

В последние годы, с быстрым развитием компьютерного зрения, обнаружение объектов (OD) широко используется во многих областях как важная часть компьютерного зрения. На основе обработки изображений OD извлекает функции из изображений, а затем извлекает и анализирует информацию об объекте, такую как категория, местоположение и ориентация. OD широко используется в ситуациях, близких к реальному времени, таких как видеонаблюдение, анализ аномального поведения и мобильные роботы [1,2,3]. Этот подход позволяет получить очень ценную информацию путем извлечения и анализа признаков. Однако этот метод сталкивается со значительными проблемами, особенно с точки зрения высоких требований к вычислительным ресурсам и памяти. Традиционные методы машинного обучения извлекают характеристики объектов из изображений, а затем передают их в классификатор. Традиционные методы извлечения признаков включают гистограмму ориентированного градиента (HOG), масштабно-инвариантное преобразование признаков (SIFT) и другие. Методы классификации включают метод опорных векторов (SVM) [4,5,6], байесовский метод, деревья решений и другие. Эти методы опираются в основном на предварительные знания. Они не в режиме реального времени, потому что они постоянно эмулируются. Кроме того, эти методы имеют несколько

характерных точек, и извлечение краевых признаков иногда неясно. В основе этих методов лежит извлечение признаков, и качество выделения признаков напрямую влияет на производительность метода [19,20,21]. Однако в практических приложениях эти методы в основном ориентированы на распознавание конкретных объектов с использованием небольших наборов данных, а способность к обобщению оставляет желать лучшего. Хотя методы машинного обучения постоянно развиваются, от извлечения признаков в реальном времени до рендеринга изображений, наиболее успешным методом является модель деформируемых разделов (DRM) [16,17,18]. Однако этот метод имеет медленное обнаружение и зависит от геометрических характеристик образцов. В настоящее время традиционные методы машинного обучения не могут удовлетворить требованиям эффективности, производительности, скорости и интеллекта обработки данных в технологии OD. С появлением технологии глубокого обучения область компьютерного зрения быстро развивалась. Технология глубокого обучения применялась для распознавания изображений, и в последние годы она добилась больших успехов в распознавании объектов [22,23,24].

Трансферное обучение — это метод машинного обучения, при котором модель, обученная одной задаче, переобучается второй

связанной задаче. Трансферное обучение имеет дело с такими проблемами, как обучение слепым задачам и согласованное обучение, и это не просто область обучения для глубокого обучения. Трансферное обучение популярно в глубоком обучении, учитывая большие ресурсы, необходимые для обучения моделей глубокого обучения, или большие и сложные наборы данных, на которых обучаются модели глубокого обучения [7,8,9].

Технология глубокого обучения может обрабатывать и анализировать функции, изучая и имитируя когнитивные способности человеческого мозга, что оказывает большое влияние на передозировку. В отличие от традиционных методов извлечения признаков, глубокие сверточные нейронные сети могут достигать высокой точности за счет извлечения признаков с использованием слепых многослойных сверточных формулировок.

Кроме того, они устойчивы к геометрическим изменениям, деформациям и освещению и могут преодолевать проблемы, вызванные изменениями окружающей среды. Методы глубокого обучения могут адаптировать описание признаков с использованием обучающих данных, они очень гибкие и обладают высокой способностью к обобщению [10,11,12]. Методы глубокого обучения включают в себя региональную сверточную нейронную сеть (RCNN), более быструю региональную сверточную нейронную сеть (Faster RCNN), You Only Look Once (YOLO) и One Shot Multibox Detector (SSD), которые являются наиболее популярными в OD. Однако современные методы OD, основанные на глубоком обучении, по-прежнему сталкиваются с проблемами из-за низкой скорости обнаружения и больших затрат времени [13,14,15].

В этой статье мы предлагаем метод OD видео в реальном времени. Мы абстрактно представляем алгоритм Fast YOLO для теории OD, а затем подробно представляем структуру Fast YOLO, включая предварительную обработку, обучение модели и функцию потерь. Далее мы проверим производительность алгоритма Fast YOLO с помощью некоторых экспериментов.

II. ПОДГОТОВКА НАБОРА ДАННЫХ ДЛЯ МОДЕЛИ.

Чтобы подготовить набор данных, мы собираем набор данных и обучаем пользовательскую модель YOLOv5 распознавать объекты в нашем наборе данных. Для этого делаем следующее:

- Скомпонуйте изображения и пометьте нашу Data set.
- Экспорт нашего набора данных в YOLOv5
- Распознавать объекты в нашем наборе данных Train YOLOv5.
- Оценить производительность нашей модели YOLOv5.
- Запустить и протестируем нашу модель, чтобы увидеть

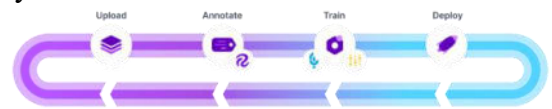


Рис 1. Подготовка набора данных для модели

Шаг 1: Установка Требования:

```
# YOLOv5dan nusxa ko'chirish and clone
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
%pip install -qr requirements.txt # bog'liqliklarni o'rnatish
%pip install -q roboflow
import torch
import os
from IPython.display import Image, clear_output # tasvirlarni ko'rsatish uchun
print(f"Sozlash tugallandi. Torch foydalanish {torch.__version__} ({torch.cuda.get_device_properties(0).name if torch.cuda.is_available() else 'CPU'})")
```

```
Cloning into 'yolov5'...
remote: Enumerating objects: 12251, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 12251 (delta 10), reused 14 (delta 8), pack-reused 12229
Receiving objects: 100% (12251/12251), 12.00 MiB | 10.88 MiB/s, done.
Resolving deltas: 100% (8480/8480), done.
/content/yolov5/yolov5/yolov5
Setup complete. Using torch 1.11.0+cu113 (Tesla T4)
```

Рис 2. Результат копирования YOLOv5 и установки зависимостей

Шаг 2: Соберите нашу сетку данных:

Чтобы обучить нашу пользовательскую модель, нам нужно собрать сетку репрезентативных изображений с аннотациями ограничительной рамки вокруг объектов, которые мы хотим обнаружить. И нам нужно, чтобы наш набор данных был в формате

Route5. В Robotflow вы можете выбрать два пути:

Преобразуйте существующий набор данных в формат YOLOv5. Robotflow поддерживает более 30 форматов обнаружения объектов для преобразования потока. Загрузите необработанные изображения и аннотируйте их в приложение Roboflow с помощью Roboflow Annotate [15].

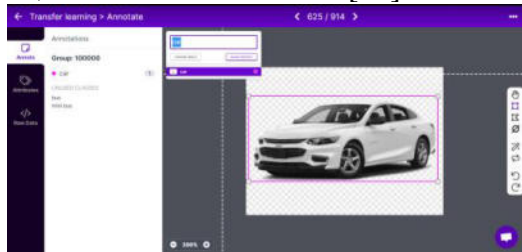


Рис 3. Процесс подготовки набора данных в Roboflow

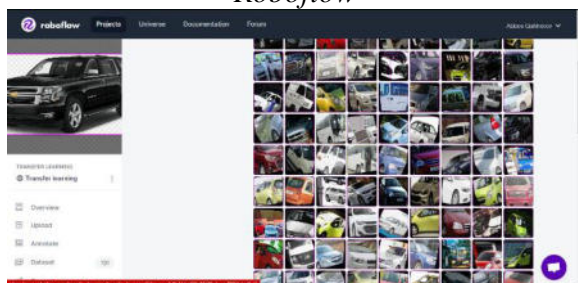


Рис 4. Процесс подготовки набора данных в Roboflow

III. ОБУЧЕНИЕ И ТЕСТИРОВАНИЕ МОДЕЛИ.

Итак, поговорим о самой процедуре. На основе созданной нами модели мы попытаемся распознать объекты на изображениях. Сначала я использовал среду Visual Studio Code для работы с языком программирования Python. Мы назвали инструмент, который мы создаем в Visual Studio Code, «yolov5». Продолжаем работать через терминал Visual Studio Code. Войдите в папку и создайте файл «requirements.txt». В этом файле перечислены библиотеки и инструменты, необходимые для работы программы. Поэтому мы в основном используем «matplotlib», «opencv», «Pillow», «PyYAML», «requests», «scipy», «torch», «torchvision», «tqdm», «scikit-learn», «pandas», необходимые файлы numpy.

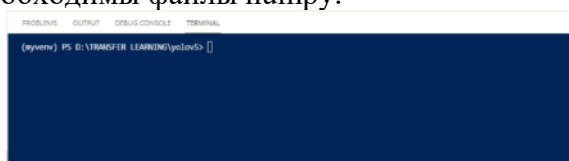


Рис 5. Создадим «Проект».

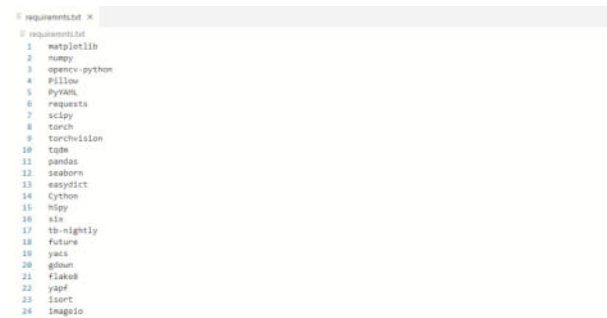


Рис 6. Список необходимых пакетов.

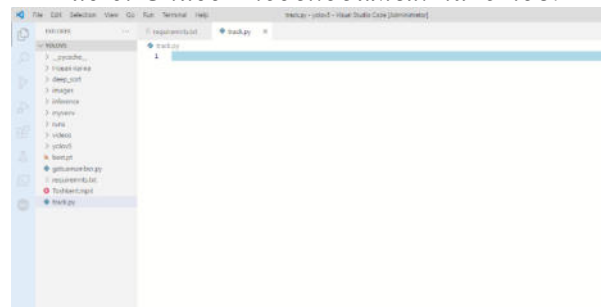


Рис 7. Мы создадим файл track.py.

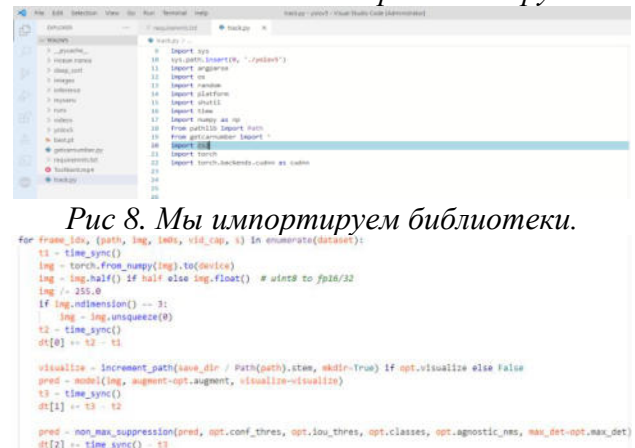


Рис 8. Мы импортируем библиотеки.

Рис 9. Код обнаружения объектов на основе нашей модели.

```
def count_obj(box, w, h, id, name):
    global count, data, carlist, countobject
    if (h-300) > int(box[1]+(box[3]-box[1])/2) > (h-400):
        if id not in data:
            count += 1
            data.append(id)
            if id not in carlist:
                carlist.append(id)
            if name not in countobject.keys():
                countobject[name]=1
            elif name in countobject.keys():
                countobject[name]=countobject[name]+1
```

Рис 10. Код для определения количества объектов на основе нашей модели.

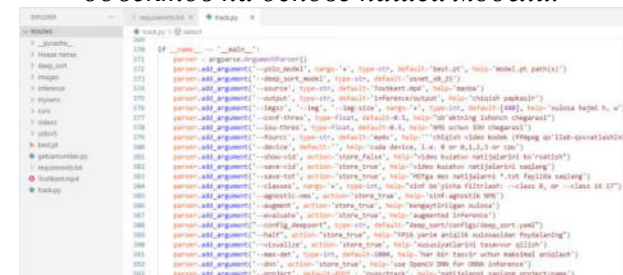


Рис 11. Скормите нашу модель и данные программе и запустите ее.


```
for j in range(1, len(carcenterlist[id])):
    if carcenterlist[id][j-1] is None or carcenterlist[id][j] is None:
        continue
    thickness = int(np.sqrt(25 / float(j+1)) * 2)
    cv2.line(img, (carcenterlist[id][j-1]), (carcenterlist[id][j]), (carlinecolor[id]), thickness)
```

Рис 12. Код для отрисовки направления движения объекта.

```
def speedtestfunc(id):
    global startcarlist, endcarlist, vehicleguid
    if id in startcarlist.keys() and id in endcarlist.keys():
        if (startcarlist[id][2] - endcarlist[id][2]) > 0:
            v = 3.6 * ((abs(endcarlist[id][2] - startcarlist[id][2]) / abs(endcarlist[id][3] - startcarlist[id][3])) / abs(endcarlist[id][2] - startcarlist[id][2]))
            return v
    return 0
```

Рис 13. Код определения скорости автомобиля.

```
def totakeapicture(copyframe, id, left, top, right, bottom):
    copyframe = copyframe[top:bottom, left:right]
    if id not in carimagelist:
        carimagelist.append(id)
        cv2.imwrite('images/car'+str(id)+'.jpg', copyframe)
        cv2.imshow("nasm2", copyframe)
```

Рис 14. Сохранение и отображение изображений объектов.

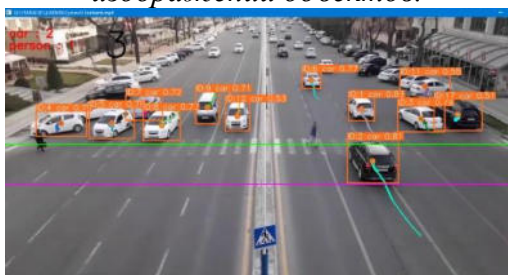


Рис 15. Процесс идентификации объектов на видео с помощью нашего программного обеспечения.

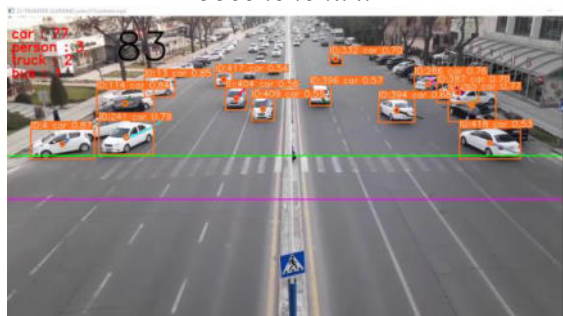


Рис 16. Процесс определения количества объектов, проходящих между двумя линиями, с помощью нашего программного обеспечения. В данном случае 77 автомобилей, 3 человека, 2 грузовика и 1 автобус, то есть всего наша программа определила 83 объекта.

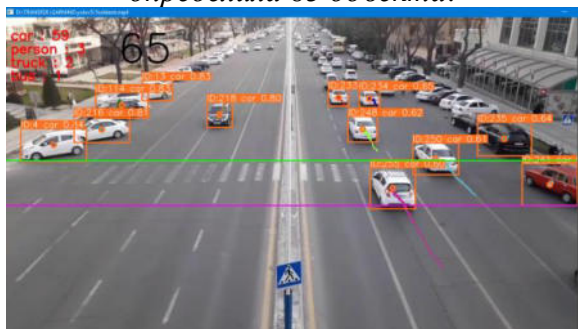


Рис 17. С помощью нашего программного обеспечения рисуется направление движения

объекта, то есть прорисовывается путь, пройденный объектом.

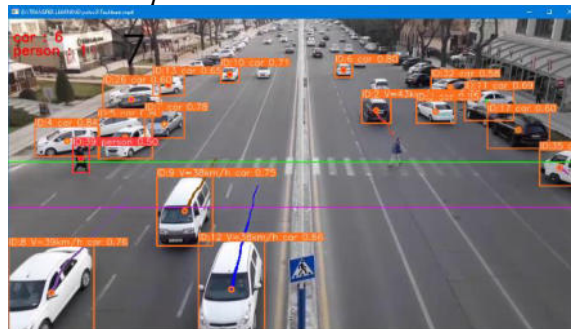


Рис 18. Процесс определения скорости автомобилей с помощью нашего программного обеспечения. Здесь определяется скорость автомобилей, пересекающих две линии.

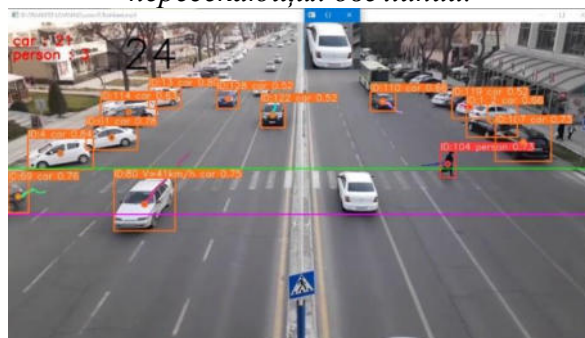


Рис 19. Изображение объектов, переданных между двумя линиями с помощью нашего программного обеспечения, отображается и сохраняется в папке «images».

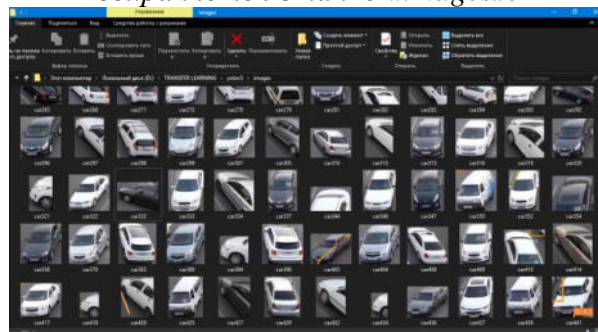


Рис 20. Изображения автомобилей хранятся в папке «Изображения».

IV. ЗАКЛЮЧЕНИЕ

В заключение следует сказать, что в результате данной дипломной работы была разработана модель распознавания объектов на изображениях на основе метода «Трансферное обучение». Наряду с этим были проведены следующие работы по разработке модели распознавания объектов на изображениях.

- проанализирована концепция трансферного обучения и его использование в интеллектуальных транспортных системах, а

также изучены проблемы в интеллектуальных транспортных системах;

- после анализа алгоритмов, необходимых для создания модели, был выбран оптимальный алгоритм;

- Изучены возможности необходимых для создания модели библиотек python

roboflow, torch и использовано создание модели из этих библиотек;

Для контроля движения на дорогах создана система, которая подсчитывает количество автомобилей и выявляет водителей, нарушающих правила;

V. СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУР

1. Постановление Президента Республики Узбекистан, от 17.02.2021 г. № ПП-4996. lex.uz/docs/5297051
2. Предварительная прогнозирование медицинских заболеваний с помощью нейронных сетей. Яхшибоева Д.Э. Material of International students conference.2021
3. Цифровые технологии в диагностике и лечении неврологических заболеваний. Н.В.Петухова, М.П.Фархадов, М.В.Замерград, С.П.Грачев. 2022.
4. Разработка и исследование алгоритмов сегментации и распознавания объектов на медицинских изображениях на основе шпираль-преобразования и нейронных сетей. Хамад Ю.А. 2020.
5. Методы повышения эффективности нейросетевых рекомендательных систем в условиях ограниченных объемов выборок со сложными корреляционными связями (на примере диагностики и прогнозирования сердечно-сосудистых заболеваний человека). Черепанов Ф.М. 2019.
6. V-Net — Volumetric Convolution (Biomedical Image Segmentation).Sik-Ho Tsang. 2019.
7. Andersson J, Ahlström H, Kullberg J (September 2019). "Separation of water and fat signal in whole-body gradient echo scans using convolutional neural networks"
8. Long, J.; Shelhamer, E. & Darrell, T. (2014), Fully convolutional networks for semantic segmentation
9. M. B. Boltaevich, N. R. H. ogli, G. N. S. qizi and M. S. S. ogli, "Estimation affects of formats and resizing process to the accuracy of convolutional neural network," *2019 International Conference on Information Science and Communications Technologies (ICISCT)*, 2019, pp. 1-5, doi: 10.1109/ICISCT47635.2019.9011858.
10. Muminov, B., et al. "Localization and Classification of Myocardial Infarction Based on Artificial Neural Network,(2020) 2020 Information Communication Technologies Conference." (2020): 245-249.
11. R. Yakhshibaev, B. Turaev, K. Jamolov, N. Atadjanova, E. Kim and N. Sayfullaeva, "Development of a mathematical model for balancing the level and device for remote monitoring of groundwater parameters," *2021 International Conference on Information Science and Communications Technologies (ICISCT)*, 2021, pp. 1-4, doi: 10.1109/ICISCT52966.2021.9670022.
12. Yaxshiboyev, Rustam, and Dilbar Yaxshiboyeva. "ANALYSIS OF ALGORITHMS FOR PREDICTION AND PRELIMINARY DIAGNOSTICS OF GASTROENTEROLOGICAL DISEASES." *CENTRAL ASIAN JOURNAL OF EDUCATION AND COMPUTER SCIENCES (CAJECS)* 1.2 (2022): 49-56.
13. Ermetov E. Y. Yaxshiboyev RE Gastroenterologik kasalliklarni KNN algoritmi asosida bashoratlovchi dastur //O ‘zbekiston Respublikasi intellektual mulk agentligi. Elektron hisoblash mashinalari uchun yaratilgan dasturning rasmiy ro‘yxatdan o‘tkazilganligi to‘g‘risidagi guvohnoma.№ DGU. – T. 17014.
14. Ermetov E. Y. Yaxshiboyev RE Gastroenterologik kasalliklarni ANN algoritmi asosida bashoratlovchi dastur //O ‘zbekiston Respublikasi intellektual mulk agentligi. Elektron hisoblash mashinalari uchun yaratilgan dasturning rasmiy ro‘yxatdan o‘tkazilganligi to‘g‘risidagi guvohnoma.№ DGU. – T. 17016.
15. Ermetov E. Y. Yaxshiboyev RE Gastroenterologik kasalliklarni SVM algoritmi asosida bashoratlovchi dastur //O ‘zbekiston Respublikasi intellektual mulk agentligi. Elektron hisoblash

- mashinalari uchun yaratilgan dasturning rasmiy ro'yxatdan o'tkazilganligi to'g'risidagi guvohnoma. № DGU. – T. 17015
16. Yaxshiboyev, R. E., et al. "FORECASTING GROUNDWATER EVAPORATION USING MULTIPLE LINEAR REGRESSION." *Galaxy International Interdisciplinary Research Journal* 9.12 (2021): 1101-1107.
 17. Djumanov, Jamoljon, et al. "Mathematical model and software package for calculating the balance of information flow." *2021 International Conference on Information Science and Communications Technologies (ICISCT)*. IEEE, 2021.
 18. "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython", 2-nashr - Ues Makkinni (O'reilly, 2017).
 19. "Understanding Machine Learning: From Theory to Algorithms" - Shai Shalev-Shvarts va Shai Ben-David (Cambridge University Press, 2014).
 20. "Programming Computer Vision with Python" - Jan Erik Solem (Creative Commons, 2012).
 21. "Обработка изображений с помощью OpenCV" – Gloria Bueno, Ismael Serrano Garsiya, Noeliya Vallez, Oskar Denis Suarez, Xesus Salido, Espinosa Aranda (DMK Press, 2016).
 22. "Computer Vision and Machine Learning based Hand Gesture Recognition" – Paulo Trigeros, Fernando Ribeyro, Luis Paulo Reis (Scholaar Press, 2015).
 23. "Система распознавания жестов из ограниченного набора" - Александр Носов (LAP Lambert Academic Publishing 2012).
 24. Mo'minov B., Dauletov A. CLASSES OF ELECTRONIC DOCUMENT CIRCULATION SYSTEMS AND MATHEMATICAL MODELS OF PROCESSING //CENTRAL ASIAN JOURNAL OF EDUCATION AND COMPUTER SCIENCES (CAJECS). – 2022. – T. 1. – №. 2. – С. 6-16.