ACADEMIA ONE
PEER REVIEWED JOURNALS

Open Academia: Journal of Scholarly Research
Volume 1, Issue 9, December, 2023
ISSN (E): 2810-6377
Website: https://academiaone.org/index.php/4

Исследование информационной системы распознования графических объектов на базе нейронных сетей WTA.

Набижонов Равшанбек Мухаммаджон ўғли – ассистент кафедры «Информационные технологии» Ферганского филиала ТАТУ имени Мухаммада ал-Хорезми
Ахмаджонов Ихтиёржон Ровшанджон угли - магистрант Ферганского филиала ТАТУ имени Мухаммада ал-Хорезми.

Аннотация. ***Процесс разработки проекта, тестирования его функциональности и проведения экспериментальных исследований для проверки гипотез и оценки эффективности проекта. Будет произведен обзор литературы по теме исследования и также будут рассмотрены различные методы распознавания графических объектов, а также методы обучения нейронных сетей и обзор программной части бдут обсуждаться результаты исследования. Будут рассмотрены преимущества и недостатки разработанной информационной системы для распознавания графических объектов на основе нейронной сети WTA.***
Ключевые слова: информационная система, распознавание графических объектов, нейронные сети, механизм победителя-забирает-все (WTA).

Нейросеть WTA для решения задач в пространстве PYTHON

Нейросеть WTA (Winner-Takes-All) является одним из методов распознавания графических объектов и может быть реализована в пространстве? В среде разработке Python с помощью различных библиотек, таких как TensorFlow, Keras или PyTorch можно построить однослойное и многослойное нейронное сети.

С библиотеками TensorFlow можно создать глубокою нейронную сеть, а нейронная сеть WTA находится в модуле Keras при создание искусственного нейронногй сети мы будим действовать следующим образом:

```
python
import tensorflow as tf
# Создание модели
model = tf.keras.Sequential([
  tf.keras.layers.Dense(256, activation='relu'),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
# Компиляция модели
model.compile(optimizer='adam',
```

ACADEMIA ONE
PEER REVIEWED JOURNALS
Open Academia: Journal of Scholarly Research
Volume 1, Issue 9, December, 2023
ISSN (E): 2810-6377
Website: https://academiaone.org/index.php/4

```
        loss='sparse_categorical_crossentropy',
        metrics=['accuracy'])
# Обучение модели
model.fit(x_train, y_train, epochs=10)
# Оценка точности модели на тестовых данных
test_loss, test_acc = model.evaluate(x_test, y_test)
print('Test accuracy:', test_acc)
```

Здесь создается последовательная модель с несколькими слоями Dense (полносвязные слои), последний из которых имеет 10 выходных нейронов с функцией активации softmax для распознавания 10 классов (цифры от 0 до 9). Модель компилируется с оптимизатором Adam и функцией потерь sparse_categorical_crossentropy. Затем модель обучается на тренировочных данных и оценивается на тестовых данных.

Таким образом, нейросеть WTA может быть реализована в пространстве Python с помощью различных библиотек и методов обучения. рассмотрены возможности для дальнейшего улучшения системы.

Аффинное преобразования графических объектов.

Аффинное пре образование одним словам, аффинных преобразований вращения, сдвигов, изменения масштаба картинок: Такие действии поможет вам снизить вероятность повторного сложного переобучения и обеспечит лучшую инвариантность классификатора к трансформациям, таким образом проект в программе будит работать бесперебойно.

Преобразование картинок реализуется следующим образом. Вообще можно очень долго говорить про эту тему, потому что это основы основ. Взять любую графику и сжать на выборочный коэффициент 1,2, а затем растянуть с коэффициентом 2,0, то все точки вернутся в исходное положение.

Например: Преобразованная фотография дорожного знака въезд запрещен. Каждое изображение варьируется по шкале 28x28=784 пиксел.

Аффинное преобразование графических объектов - это изменение их размера, положения, поворота и перспективы с помощью матрицы преобразования. Оно может быть использовано для создания эффектов, таких как зеркальное отражение, поворот, масштабирование и сдвиг объектов.

Цветовая модель — математическая модель описания представления цветов в виде кортежей чисел (обычно из трёх, реже — четырёх значений), называемых цветовыми компонентами или цветовыми координатами. Все возможные значения цветов, задаваемые моделью, определяют цветовое пространство.

Модель, описывающая способ кодирования цвета для цветовоспроизведения с помощью трёх цветов RGB (Красный Синий Зелённый). Цветовая модель задаёт соответствие между воспринимаемыми человеком цветами, хранимыми в памяти, и цветами, формируемыми на устройствах вывода (возможно, при заданных условиях).

Open Academia: Journal of Scholarly Research
Volume 1, Issue 9, December, 2023
ISSN (E): 2810-6377
Website: https://academiaone.org/index.php/4

Для выполнения аффинных преобразований используются матрицы преобразования, которые состоят из коэффициентов масштабирования, поворота и сдвига. Эти коэффициенты могут быть изменены, чтобы изменить форму, размер и положение объекта.

Примеры аффинных преобразований включают в себя поворот объекта на определенный угол, масштабирование объекта в размере и изменение его положения на экране. Эти преобразования могут быть выполнены как на двухмерных, так и на трехмерных объектах.

Аффинные преобразования широко используются в компьютерной графике, играх и анимации. Они позволяют создавать сложные эффекты и анимацию, которые могут быть изменены в режиме реального времени.

ОПИСАНИЕ РАБОТЫ АЛГОРИТМА РАСПОЗНАВАНИЯ

Основными этапами для реализации программы были выделены следующие:

1) ***Сбор нами не обходимых данных для нейронов, увеличение весов нейронов.***
2) ***Обучение нейронной сети для распознавания графических объектов.***
3) ***Определение нужные классов по принципу модульного типа.***
4) ***Разбить изображение на различные виды для точного определения***
5) ***Распознать на изображения и связка с labels.***
6) ***Тестировать проект.***

Для обучения модели была выбрана сверхточная нейронная сеть, которая была обучена различать графические объектов строго определенному классу. Выбор сверочной нейронной сети обусловлено тем, что на данный момент у данной разновидности нейронных сетей один из лучших алгоритмов распознавания и классификации изображений. По сравнению с полно связной нейронной сетью у нее гораздо меньше количество настраиваемых весов.

Рис 1. (Здесь 0 это полностью белой пиксель А 255 это сильно серой цвет это позволяет работать с ними легко и даже можно работать с небольшим мощностью компьютера)

Выбор средств для реализации проекта.

Для того, чтобы разработать любую программу, сначала необходимо выбрать язык программирования, на котором будет она написана. Существует множество языков программирования и на каждом можно написать эквивалентную программу. Отличиями между эквивалентными программами будут являться структура написания кода и средства, используемые для получения нужного результата. Основными языками программирования, на которых решаются задачи, связанные с нейронными сетями, являются:

ACADEMIA ONE
PEER REVIEWED JOURNALS
Open Academia: Journal of Scholarly Research
Volume 1, Issue 9, December, 2023
ISSN (E): 2810-6377
Website: https://academiaone.org/index.php/4

1. ***Python;***
2. ***Java;***
3. ***C++;***
4. ***Matlab.***

Для реализации системы распознавания графического образа был выбран язык Python, так как он имеет ряд плюсов, выделяющих его среди остальных языков программирования:

1. ***Быстрая разработка;***
2. ***Простота в освоении;***
3. ***Большое количество библиотек;***
4. ***Поддержка на любой платформе;***
5. ***И ряд других плюсов.***

Программы, написанные на языке Python выполняются на большинстве современных операционных систем, что позволяет использовать программу на различных устройствах, не совершая глобальных изменений в коде.

Еще одним несомненным плюсом является то, что программы, написанные на Python, имеют высокую скорость выполнения. Это связано с тем, что основные библиотеки Python написаны на языке C++ и выполнение задач занимает меньше времени, чем на других языках высокого уровня.

Определившись с языком программирование, можно начинать думать над тем, какие библиотеки будут использоваться в разработке.

Так как программа будет работать с изображениями, то необходимо обязательно подключить библиотеку, содержащую основные функции по работе с ними. Самыми популярными библиотеками, поддерживающие язык программирования Python, для работы с изображениями являются:

1) ***OpenCV;***
2) ***NumPy;***
3) ***SciPy;***
4) ***Pillow.***

Для работы с изображениями были выбраны библиотеки OpenCV и NumPy, так как совместно они выполняют все необходимые функции для реализации проекта, а именно:

1) ***Перевод изображения в оттенки серого;***
2) ***Перевод изображения в черно-белое;***
3) ***Определение контуров на изображении.***

Также необходимо создать нейронную сеть, для распознавания изображенных на картинке знаков. Для этого было решено использовать библиотеку Tensorflow и надстройку над ней - Keras. С помощью этого сочетания можно создать и обучить нейронную сеть. Функции, встроенные в Keras, позволяют выполнять все необходимые действия для построения нужной модели нейронной сети

Для идентификации дорожных знаков, используются два основных алгоритма. Обработка распознаваемого материала происходит на примерах различных образов и измененной форматов.

ACADEMIA ONE
PEER REVIEWED JOURNALS
Open Academia: Journal of Scholarly Research
Volume 1, Issue 9, December, 2023
ISSN (E): 2810-6377
Website: https://academiaone.org/index.php/4

Распознавание основывается на использовании правил обнаружения признаков, касающихся особенностей конкретного класса. С помощью функции обнаружения программное обеспечение оценивает данные класс в соответствии с правилами о том, как. Например, стоп знак может храниться как две диагональные линии, пересекающиеся с горизонтальной линией посередине.

Особенности.

Программа может посчитать обратно распространять и сохранять образ распознаваемого объекта. В базе образов содержится большинство используемых за ранними графическими образами внутри класса.

Это даже не полноценная программа, а утилита. Установка не потребуется, а исполнительный файл весит всего в несколько килобайт. Процесс распознавания происходит предельно быстро, правда, полученные в его результате выявляется на краю монитора.

В дальнейшем именно по этим принципам по этими же методами можно разработать новые бизнес проект и создавать новые слои классов нейронных сетей обучая их интеллектуально автоматизировать многие направление общества.

Для этого нам понадобится рабочая система определённого деятельности. Привожу пример… в Калифорнии в "силиконовым долине" Разработчики на основе нейросетей разрабатывают искусственны интеллект который определяет по медицинскими анализами и снимкам определяет точный диагноз пациента пока этот проект тестовым эксперимент ном этапе.

Для начало загружаем библиотек для работы и обработки изображение, а затем импортируем собранной весь проект в среду Python. (см).Рис 2.

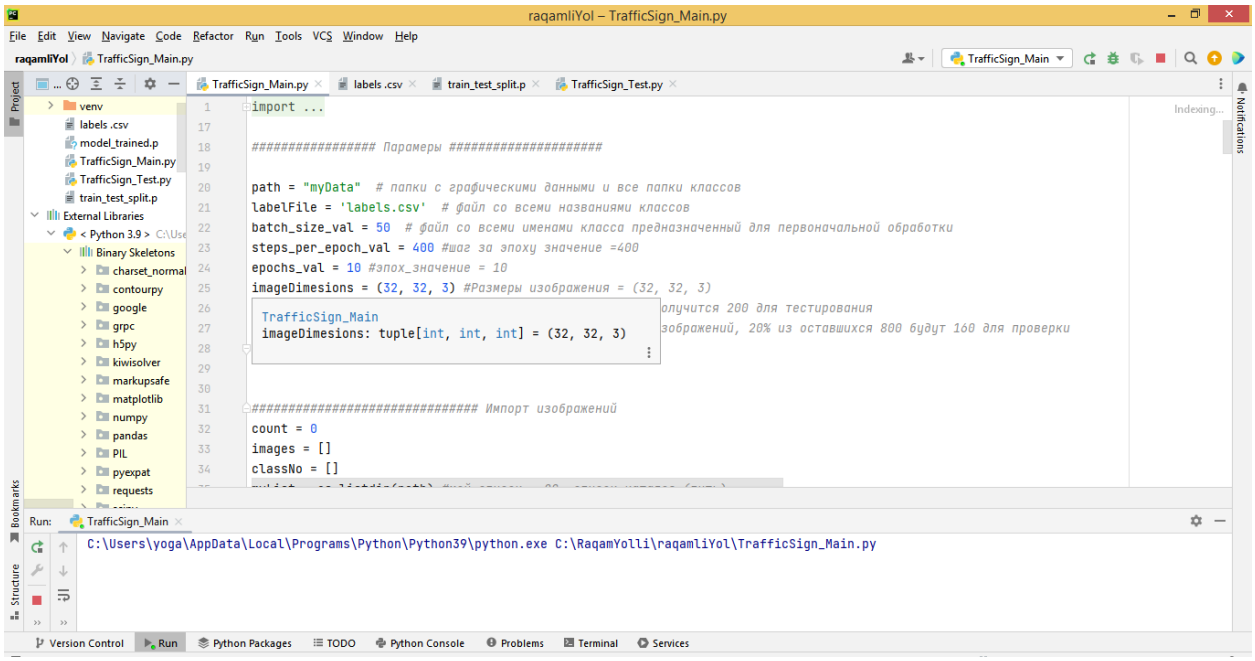

Рис 2. (Импорт необходимых модулей)

```
import numpy as np
import os
import pickle
```

ACADEMIA ONE
PEER REVIEWED JOURNALS
Open Academia: Journal of Scholarly Research
Volume 1, Issue 9, December, 2023
ISSN (E): 2810-6377
Website: https://academiaone.org/index.php/4

```
import random
```

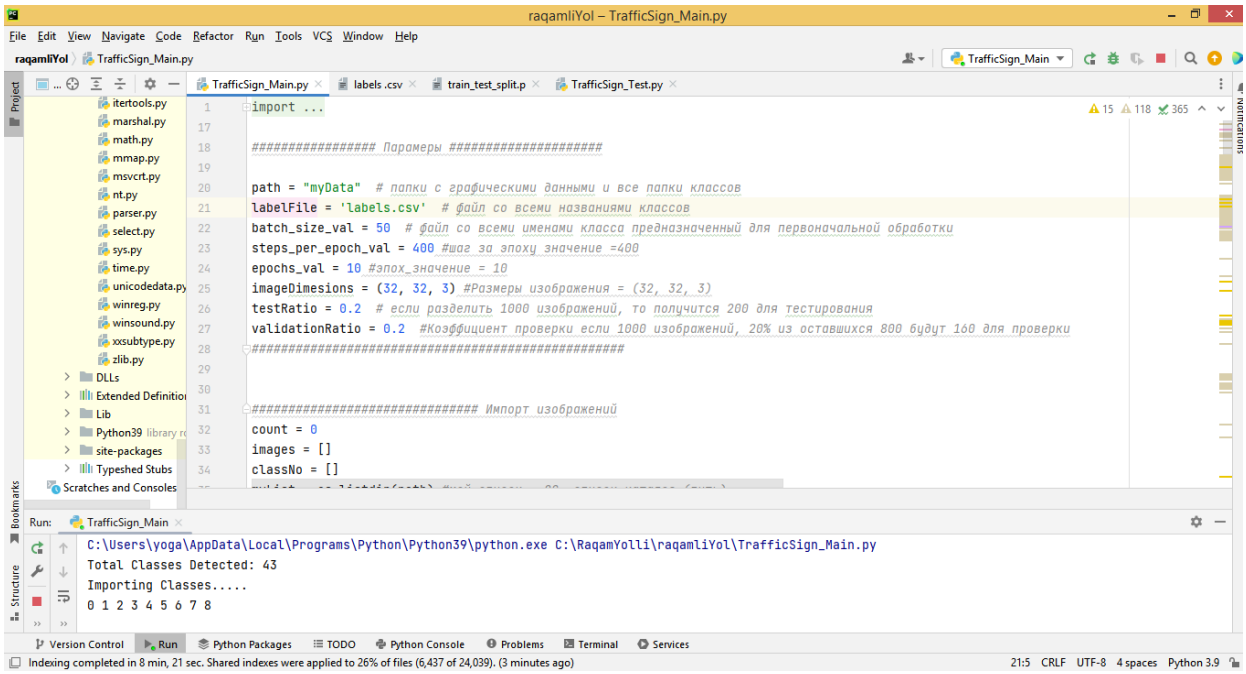

Рис 3. (Импорт и установка необходимых модулей)

```
import cv2
import matplotlib.pyplot as plt
import pandas as pd
from keras.layers import Dense
from keras.layers import Dropout, Flatten
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.models import Sequential
from keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator
from keras.utils.np_utils import to_categorical
from sklearn.model_selection import train_test_split
################### Параметры ######################
path = "myData"  # папки с графическими данными и все папки классов
labelFile = 'labels.csv'  # файл со всеми названиями классов
batch_size_val = 50  # файл со всеми именами класса предназначенный для
первоначальной обработки
steps_per_epoch_val = 400 #шаг за эпоху значение =400
epochs_val = 10 #эпох_значение = 10
imageDimesions = (32, 32, 3) #Размеры изображения = (32, 32, 3)
testRatio = 0.2  # если разделить 1000 изображений, то получится 200 для
тестирования
validationRatio = 0.2  #Коэффициент проверки если 1000 изображений, 20% из
оставшихся 800 будут 160 для проверки
```

ACADEMIA ONE
PEER REVIEWED JOURNALS
Open Academia: Journal of Scholarly Research
Volume 1, Issue 9, December, 2023
ISSN (E): 2810-6377
Website: https://academiaone.org/index.php/4

```
#####################################################
################################## Импорт изображений
count = 0
images = []
classNo = []
myList = os.listdir(path) #мой список = ОС. список каталог (путь)
print("Total Classes Detected:", len(myList)) #print("Общее количество обнаруженных
классов:"
noOfClasses = len(myList) #(мой список))
print("Importing Classes.....")   #Импорт классов
for x in range(0, len(myList)): #для х в диапазоне (0, len (myList))
    myPicList = os.listdir(path + "/" + str(count)) #мой список изображений = os . listdir
(путь + "/" + улица (количество))
    for y in myPicList: #для (у) в моем списке изображений
        curImg = cv2.imread(path + "/" + str(count) + "/" + y)
        images.append(curImg) #картинки . добавить curImg
        classNo.append(count) #Прибовлять номер класса и считать
    print(count, end=" ") #распечатать (количество, конец = "")
    count += 1 #количество += 1
print(" ") #Распечатать(" ")
images = np.array(images) #изображения = пр.массив (изображения)
classNo = np.array(classNo) #№ класса = пр. массив (класс №)
################################## Разделение данных
X_train, X_test, y_train, y_test = train_test_split(images, classNo, test_size=testRatio)
X_train, X_validation, y_train, y_validation = train_test_split(X_train, y_train,
test_size=validationRatio)
# X_train = МАССИВ ИЗОБРАЖЕНИЙ ДЛЯ ОБУЧЕНИЯ
# y_train = СООТВЕТСТВУЮЩИЙ ИДЕНТИФИКАТОР КЛАССА
##### ЧТОБЫ ПРОВЕРИТЬ, СООТВЕТСТВУЕТ ЛИ КОЛИЧЕСТВО ИЗОБРАЖЕНИЙ
КОЛИЧЕСТВУ МЕТОК ДЛЯ КАЖДОГО НАБОРА ДАННЫХ ИМЕННО В ЭТОМ
ЭТАПЕ ОБУЧАЕТСЯ ОБРАТНОЕ РАСПОСТРАНЕНИЕ ОШИБОК
print("Data Shapes") #печать ("Формы данных")
print("Train", end=""); #печать("Конец поезда =");
print(X_train.shape, y_train.shape) #печать (X_ поезд . форма , y _ поезд . форма )
print("Validation", end=""); #print("Конец проверки=");
print(X_validation.shape, y_validation.shape) #печать (форма проверки X_, форма
проверки y_)
print("Test", end=""); #печать("Тест", конец="")
print(X_test.shape, y_test.shape) #печать (X_ тестовая форма, y _ тестовая форма)
assert (X_train.shape[0] == y_train.shape[ #утверждать (X_ форма поезд. форма[0]
== y_ форма поезд[
    0]), "The number of images in not equal to the number of lables in training set"
#Количество изображений не равно количеству меток в обучающем наборе
```

ACADEMIA ONE
PEER REVIEWED JOURNALS
Open Academia: Journal of Scholarly Research
Volume 1, Issue 9, December, 2023
ISSN (E): 2810-6377
Website: https://academiaone.org/index.php/4

```
assert (X_validation.shape[0] == y_validation.shape[ #утверждать ( X_ проверка .
форма [0] == y_ проверка .форма [
    0]), "The number of images in not equal to the number of lables in validation
set"#"Количество изображений не равно количеству меток в проверочном наборе"
assert (X_test.shape[0] == y_test.shape[0]), "The number of images in not equal to the
number of lables in test set" #assert (X_ test. shape [0] == y_ test . shape [0]), «Количество
изображений не равно количеству меток в тестовом наборе»
assert (X_train.shape[1:] == (imageDimesions)), " The dimesions of the Training images are
wrong " #(X_train.shape[1:] == (Размеры изображения)), «Размеры тренировочных
изображений неверны»
assert (X_validation.shape[1:] == (imageDimesions)), " The dimesionas of the Validation
images are wrong " #(X_validation.shape[1:] == (imageDimesions)), "Размеры
проверочных изображений неверны"
assert (X_test.shape[1:] == (imageDimesions)), " The dimesionas of the Test images are
wrong" #((X_test. shape [1:] == (размеры изображения)), "Размеры тестовых
изображений неверны"
################################ ПРОЧИТАТЬ CSV-ФАЙЛ
data = pd.read_csv(labelFile) #данные = pd. прочитать _csv (файл метки)
print("data shape ", data.shape, type(data)) #печать ("форма данных", данные .shape,
тип (данные)
################################ ПОКАЖИТЕ НЕСКОЛЬКО ОБРАЗЦОВ
ИЗОБРАЖЕНИЙ ВСЕХ КЛАССОВ
num_of_samples = [] #число _ из _ выборок = []
cols = 5 #столбцы = 5
num_classes = noOfClasses #число _ классов = количество классов
fig, axs = plt.subplots(nrows=num_classes, ncols=cols, figsize=(5, 300)) #рис, топор = пл.
подзаголовки ( nrows = num _ классов, ncols = столбцы , фиговый размер = (5, 300))
fig.tight_layout() #инжир . плотный _ макет ()
for i in range(cols):
    for j, row in data.iterrows():
        x_selected = X_train[y_train == j]
        axs[j][i].imshow(x_selected[random.randint(0, len(x_selected) - 1), :, :],
cmap=plt.get_cmap("gray"))
        axs[j][i].axis("off")
        if i == 2:
            axs[j][i].set_title(str(j) + "-" + row["Name"])
            num_of_samples.append(len(x_selected))
# ОТОБРАЗИТЬ НЕСКОЛЬКО ОБРАЗЦОВ ИЗОБРАЖЕНИЙ ВСЕХ КЛАССОВ
ОТОБРАЗИТЬ СТОЛБЧАТУЮ ДИАГРАММУ, ПОКАЗЫВАЮЩУЮ КОЛИЧЕСТВО
ОБРАЗЦОВ ДЛЯ КАЖДОЙ КАТЕГОРИИ
print(num_of_samples)
plt.figure(figsize=(12, 4))
plt.bar(range(0, num_classes), num_of_samples)
```

ACADEMIA ONE
PEER REVIEWED JOURNALS
Open Academia: Journal of Scholarly Research
Volume 1, Issue 9, December, 2023
ISSN (E): 2810-6377
Website: https://academiaone.org/index.php/4

```
plt.title("Distribution of the training dataset")
plt.xlabel("Class number")
plt.ylabel("Number of images")
plt.show()
################################ ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ
def grayscale(img):
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    return img
def equalize(img):
    img = cv2.equalizeHist(img)
    return img
def preprocessing(img):
    img = grayscale(img)  # ПРЕОБРАЗОВАТЬ В ОТТЕНКИ СЕРОГО
    img = equalize(img)  # ПРЕОБРАЗОВАНИЕ В ОТТЕНКИ СЕРОГО ДЛЯ СТАНДАРТИЗАЦИИ ОСВЕЩЕНИЯ НА ИЗОБРАЖЕНИИ
    img = img / 255  # ДЛЯ НОРМАЛИЗАЦИИ ЗНАЧЕНИЙ ОТ 0 ДО 1 ВМЕСТО 0 ДО 255
    return img
X_train = np.array(list(map(preprocessing, X_train)))  # ДЛЯ РЕДАКТИРОВАНИЯ И ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ВСЕХ ИЗОБРАЖЕНИЙ
X_validation = np.array(list(map(preprocessing, X_validation)))
X_test = np.array(list(map(preprocessing, X_test)))
cv2.imshow("GrayScale Images", X_train[random.randint(0, len(X_train) - 1)])  # ЧТОБЫ ПРОВЕРИТЬ, ПРАВИЛЬНО ЛИ ВЫПОЛНЕНО ОБУЧЕНИЕ
################################ ДОБАВЬТЕ ГЛУБИНУ 1
X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], X_train.shape[2],
X_validation = X_validation.reshape(X_validation.shape[0], X_validation.shape[1],
X_validation.shape[2], 1)
X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], X_test.shape[2], 1)
################################ ДОБАВЬЛЯЕМ ГЛУБИНУ ИЗОБРАЖЕНИЯ В 1 МИЛЛИМЕТР: ЧТОБЫ СДЕЛАТЬ ЕГО БОЛЕЕ ОБЩИМ
dataGen = ImageDataGenerator(width_shift_range=0.1,
                             # 0.1 = 10%, ЕСЛИ БОЛЬШЕ 1, например, 10, ТО ЭТО ОТНОСИТСЯ К № ПИКСЕЛЕЙ, НАПРИМЕР, 10 ПИКСЕЛЕЙ
                             height_shift_range=0.1,
                             zoom_range=0.2,  # 0,2 ОЗНАЧАЕТ, ЧТО МОЖЕТ ВАРЬИРОВАТЬСЯ ОТ 0,8 ДО 1,2
                             shear_range=0.1,  # ВЕЛИЧИНА УГЛА СДВИГА
                             rotation_range=10)  # СТЕПЕНИ
dataGen.fit(X_train)
batches = dataGen.flow(X_train, y_train,
                       batch_size=20)  # ЗАПРАШИВАЮЩИЙ ГЕНЕРАТОР ДАННЫХ ДЛЯ
```

ACADEMIA ONE
PEER REVIEWED JOURNALS
Open Academia: Journal of Scholarly Research
Volume 1, Issue 9, December, 2023
ISSN (E): 2810-6377
Website: https://academiaone.org/index.php/4

```
ГЕНЕРАЦИИ ИЗОБРАЖЕНИЙ РАЗМЕР ПАКЕТА № ОБРАЗОВ, СОЗДАВАЕМЫХ КАЖДЫЙ РАЗ, КОГДА ЕГО ВЫЗЫВАЛИ
X_batch, y_batch = next(batches)
# ЧТОБЫ ПОКАЗАТЬ ФРАГМЕНТИРОВАННЫЕ ОБРАЗЦЫ ИЗОБРАЖЕНИЙ
fig, axs = plt.subplots(1, 15, figsize=(20, 5))
fig.tight_layout()
for i in range(15):
    axs[i].imshow(X_batch[i].reshape(imageDimesions[0], imageDimesions[1]))
    axs[i].axis('off')
plt.show()
y_train = to_categorical(y_train, noOfClasses)
y_validation = to_categorical(y_validation, noOfClasses)
y_test = to_categorical(y_test, noOfClasses)
################################### МОДЕЛЬ СВЕРТОЧНОЙ НЕЙРОННОЙ СЕТИ
def myModel():
    no_Of_Filters = 60
    size_of_Filter = (5, 5)  # ЭТО ЯДРО, КОТОРОЕ ПЕРЕМЕЩАЕТСЯ ПО ИЗОБРАЖЕНИЮ, ЧТОБЫ ПОЛУЧИТЬ ФУНКЦИИ.
    # ЭТО ПОЗВОЛИЛО БЫ УДАЛИТЬ ПО 2 ПИКСЕЛЯ С КАЖДОЙ ГРАНИЦЫ ПРИ ИСПОЛЬЗОВАНИИ ИЗОБРАЖЕНИЯ 32 32
    size_of_Filter2 = (3, 3)
    size_of_pool = (2, 2)  # УМЕНЬШИТ МАСШТАБ ВСЕЙ КАРТЫ ФУНКЦИЙ, ЧТОБЫ БОЛЬШЕ ОПТИМИЗИРОВАТЬ, ЧТОБЫ УМЕНЬШИТЬ ПЕРЕОСНАЩЕНИЕ
    no_Of_Nodes = 500  # КОЛИЧЕСТВО УЗЛОВ В СКРЫТЫХ СЛОЯХ
    model = Sequential()
    model.add((Conv2D(no_Of_Filters, size_of_Filter, input_shape=(imageDimesions[0], imageDimesions[1], 1),
                      activation='relu')))  # ДОБАВЛЕНИЕ МНОЖЕСТВА КОЛИЧЕСТВА СЛОЕВ СВЕРТКИ = МЕНЬШЕЕ КОЛИЧЕСТВО ОБЪЕКТОВ, ПРИВЕДЕТ К ПОВЫШЕНИЮ ТОЧНОСТИ.
    model.add((Conv2D(no_Of_Filters, size_of_Filter, activation='relu')))
    model.add(MaxPooling2D(pool_size=size_of_pool))  # ДОБАВЛЕННЫЙ МОДУЛЬ НЕ ВЛИЯЕТ НА ГЛУБИНУ/ОТСУТСТВИЕ ФИЛЬТРОВ
    model.add((Conv2D(no_Of_Filters // 2, size_of_Filter2, activation='relu')))
    model.add((Conv2D(no_Of_Filters // 2, size_of_Filter2, activation='relu')))
    model.add(MaxPooling2D(pool_size=size_of_pool))
    model.add(Dropout(0.5))
    model.add(Flatten())
    model.add(Dense(no_Of_Nodes, activation='relu'))
    model.add(Dropout(0.5))  # INPUTS NODES TO DROP WITH EACH UPDATE 1 ALL 0 NONE
    model.add(Dense(noOfClasses, activation='softmax'))  # OUTPUT LAYER
    # COMPILE MODEL
```

ACADEMIA ONE
PEER REVIEWED JOURNALS
Open Academia: Journal of Scholarly Research
Volume 1, Issue 9, December, 2023
ISSN (E): 2810-6377
Website: https://academiaone.org/index.php/4

```
    model.compile(Adam(lr=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
    return model
################################## ПОЕЗД ИЛИ ЛОКОМОТИВНЫЙ МОДУЛЬ
model = myModel()
print(model.summary())
history = model.fit_generator(dataGen.flow(X_train, y_train, batch_size=batch_size_val),
                              steps_per_epoch=steps_per_epoch_val, epochs=epochs_val,
                              validation_data=(X_validation, y_validation), shuffle=1)
################################## СЮЖЕТНАЯ ЧАСТЬ
plt.figure(1)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.legend(['training', 'validation'])
plt.title('loss')
plt.xlabel('epoch')
plt.figure(2)
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.legend(['training', 'validation'])
plt.title('Acurracy')
plt.xlabel('epoch')
plt.show()
score = model.evaluate(X_test, y_test, verbose=0)
print('Test Score:', score[0])
print('Test Accuracy:', score[1])
# СОХРАНЕНИЕ МОДЕЛЬ КАК ОБЪЕКТ ПИКСЕЛЕЙ
pickle_out = open("model_trained.p", "wb")  # НАПИСАТЬ БАЙТ
pickle.dump(model, pickle_out)
pickle_out.close()
cv2.waitKey(0)
```

Заключение

В данном исследовании была разработана информационная система, основанная на использовании нейронных сетей с механизмом победителя-забирает-все (WTA) для распознавания графических объектов. Целью исследования было изучение эффективности и точности данной системы в задаче распознавания.

В результате проведенных экспериментов было показано, что информационная система, основанная на нейронных сетях WTA, обладает высокой точностью в распознавании графических объектов. Механизм WTA позволяет системе выделять наиболее значимые признаки и образы, что способствует улучшению качества распознавания.

Исследование также позволило определить оптимальные параметры и конфигурацию нейронных сетей WTA для достижения наилучших результатов. Были использованы различные наборы данных, что подтвердило универсальность и применимость данной

Open Academia: Journal of Scholarly Research
Volume 1, Issue 9, December, 2023
ISSN (E): 2810-6377
Website: https://academiaone.org/index.php/4

информационной системы в различных областях, связанных с распознаванием графических объектов.

Основываясь на полученных результатах, можно сделать вывод, что информационная система на базе нейронных сетей WTA представляет собой эффективный инструмент для задачи распознавания графических объектов. Дальнейшее развитие и оптимизация данной системы могут привести к еще более точным и быстрым результатам распознавания.

Таким образом, исследование подтверждает важность и перспективы применения нейронных сетей WTA в области распознавания графических объектов и открывает возможности для дальнейших исследований и разработок в этой области.

Использованная литература:

1. Обухов, В., Ходжиматов Ж., & Набижонов , Р. (2023). Развитие блокчейн технологий в узбекистане: современные вызовы и перспективы. Research and Implementation. извлечено от https://fer-teach.uz/index.php/rai/article/view/768
2. Обухов , В., Хамидов Э., & Набижонов , Р. (2023). Поэтапное внедрение блокчейн технологий в Республике Узбекистан. Research and Implementation. извлечено от https://fer-teach.uz/index.php/rai/article/view/770
3. Xonto'rayev, S. (2023). Oliy ta'lim muassasalarida Web resurslarda mavjud dasturiy, texnik va uslubiy muammolarni bartaraf etish. Scientific-technical journal (STJ FerPI, ФарПИ ИТЖ, НТЖ ФерПИ, 2023, Т. 27. спец. выпуск№ 2).
4. Nabijonov , R., & Rasulov , A. (2023). Zamonaviy media portal imkoniyatlaridan unumli foydalanish. Research and Implementation. извлечено от https://fer-teach.uz/index.php/rai/article/view/767
5. Sobirov Muzaffarjon Mirzaolimovich, Nabijonov Ravshanbek Mukhammadjon Ugli, & Khaitboev Elbekjon Iminjon Ugli (2023). Development of automated management system in technical processes. Science and innovation, 2 (A4), 195-198. doi: 10.5281/zenodo.7868406
6. Khonturaev , S. I., Fazlitdinov , M. X. ugli, & Mamayeva , O. I. kizi. (2023). Empowering education: The impact of AI in learning management systems. Educational Research in Universal Sciences, 2(11), 348–350. Retrieved from http://erus.uz/index.php/er/article/view/3985
7. Qadamova, Z., Khakimov, A., & Sotvoldieva, D. (2023). APPLICATION OF LIST METHODS IN PRACTICE AND ITS ADVANTAGES. Лучшие интеллектуальные исследования, 7(2), 43-47.
8. Maxmudov , A., & Nabijonov , R. (2023). WDM TEXNOLOGIYASINING AFZALLIK VA KAMCHILIKLARI. Research and Implementation, 1(2), 45–49. извлечено от https://fer-teach.uz/index.php/rai/article/view/680
9. Nabijonov , R., Ergasheva , A., Ibrohimova , N., & Azamov , S. (2023). Masofaviy ta’limda internet tizimlari afzalliklari va ulardan xavfsiz foydalanish usullari. Research and Implementation, 1(4), 31–38. извлечено от https://fer-teach.uz/index.php/rai/article/view/881

Open Academia: Journal of Scholarly Research
Volume 1, Issue 9, December, 2023
ISSN (E): 2810-6377
Website: https://academiaone.org/index.php/4

10. Nabijonov, R., & Sobirov, M. (2023). ZAMNONAVIY OPERATSION TIZIMLAR. Engineering Problems and Innovations. извлечено от https://fer-teach.uz/index.php/epai/article/view/53
11. Nabijonov Ravshanbek Muxammadjon o‘g‘li, Azamov Shohruhmirzo Alisher o‘g‘li, & Turdaliyev Kamronbek Ilhomjon o’g’li. (2022). Vebinar va multimedia texnologiyalaridan foydalanishning qulayliklari. Proceedings of International Conference on Educational Discoveries and Humanities, 1(2), 86–91. Retrieved from https://econferenceseries.com/index.php/icedh/article/view/243
12. Мамуров, М., & Мамадалиев, А. (2018). ГЕОГРАФИЧЕСКОЕ ПОЛОЖЕНИЕ И ЗЕМЕЛЬНО-ВОДНЫЕ ОТНОШЕНИЙ КОКАНДСКОГО ХАНСТВА В XVIII-XIX ВЕКАХ. Актуальные научные исследования в современном мире, (5-10), 134-136.
13. Мамуров, М. (2022). Из истории земельно-водных отношений Кокандского ханства. Актуальные проблемы истории Узбекистана, 1(1), 356-362.
14. Мамуров, М. (2022). ҚЎҚОН ХОНЛИГИДАГИ ЕР-СУВ МУНОСАБАТЛАРИНИНГ АЙРИМ ЖИҲАТЛАРИ ХУСУСИДА. ВЗГЛЯД В ПРОШЛОЕ, 5(2).
15. Маъмуров, М. М. (2020). РУС ШАРҚШУНОСЛАРИ АСАРЛАРИДА ҚЎҚОН ХОНЛИГИ ХЎЖАЛИГИ ТАРИХИНИНГ АЙРИМ ЖИҲАТЛАРИ. Academic research in educational sciences, (4), 341-346.
16. Melikozievich, M. M. (2022). From the history of irrigation system of the Khanate. ASIA PACIFIC JOURNAL OF MARKETING & MANAGEMENT REVIEW ISSN: 2319-2836 Impact Factor: 7.603, 11(12), 340-345.
17. Функциональная классификация микропроцессоров. (2023). Journal of Technical Research and Development, 1(2), 239-246. https://jtrd.mcdir.me/index.php/jtrd/article/view/82
18. МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ И ИХ ПРОИСХОЖДЕНИЕ. (2023). Journal of Technical Research and Development, 1(2), 32-37. https://jtrd.mcdir.me/index.php/jtrd/article/view/80
19. Обухов Вадим Анатольевич, Тохирова Сарвиноз Гайратжон кизи, & Исахонов Хушнидбек Муродилжон угли. (2023). ПРОГРАММЫ ДЛЯ РАСПОЗНАВАНИЯ ТЕКСТА. Ta’lim Innovatsiyasi Va Integratsiyasi, 7(1), 52–57. Retrieved from http://web-journal.ru/index.php/ilmiy/article/view/749
20. Тошматов, Ш., Исаков, А., & Махмудов, Ш. (2023). Модульные измерительные датчики. Journal of technical research and development, 1(2), 210-218.
21. Тошматов, Ш. (2023). Работа нейронной сети. Формирования Graphic detection detection проекта в языке программирование Python определение используемых библиотек. Journal of technical research and development, 1(2), 297-305.
22. Расулов, А. М., & Тошматов, Ш. М. (2023). Создание базы данных и классификация для graphic detection проекта искусственного нейронного сети. Central Asian journal of mathematical theory and computer sciences, 4(4), 15-21.

Open Academia: Journal of Scholarly Research
Volume 1, Issue 9, December, 2023
ISSN (E): 2810-6377
Website: https://academiaone.org/index.php/4

23. Murotzhonovich, T. S. (2023). Introduction to Artificial Neural Networks. Web of Synergy: International Interdisciplinary Research Journal, 2(4), 198-202.