

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Р. Е. АЛЕКСЕЕВА  
ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
КАФЕДРА «ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

# ПРОГРАММНАЯ СИСТЕМА РАСПОЗНАВАНИЯ ЗНАКОВ ДОРОЖНОГО ДВИЖЕНИЯ

Выполнил: Новичков В.С. 15-В-2

Научный руководитель: к.т.н., доцент Гай В.Е.

# Цель и задачи исследования

Цель:

Разработка программной системы распознавания дорожных знаков.

Задачи:

Обзор подходов к распознаванию знаков дорожного движения

Разработка алгоритма

Проведение вычислительного эксперимента для установления корректности работы созданной системы

# Известные подходы

Для решения проблемы поиска искомых областей кадра исследователи используют различные цветовые пространства, такие как:

- RGB - используется нормализованный фиксированный диапазон цвета

- HSV - используется для получения информации с кадра, с меньшим воздействием погодных условий и изменением освещенности

Для решения задачи локализации дорожных знаков на входном кадре с помощью детекторов геометрических признаков, используют такие методы, как:

- преобразование Хафа

- построение карты расстояний

- построение гистограммы направленных градиентов

Для классификации объекта используются методы, разделяющиеся на две категории:

- Шаблонные методы

- Классификаторы на основе нейронных сетей

# Архитектура системы



# Обучение классификатора

Предварительный поиск базы изображений, изначально классифицированных как позитивные (должны распознаваться) и негативные (не должны).

Систематизация данных в наборы (тренировочный и тестовый).

Сборка модели.

Оценка точности модели.

# Классификация входных данных

Получение входных данных – получение видеоряда из источника

Формирование системы признаков – формирование набора признаков, на основании которых будет принято решение

Принятие решения – идентификация дорожного знака.

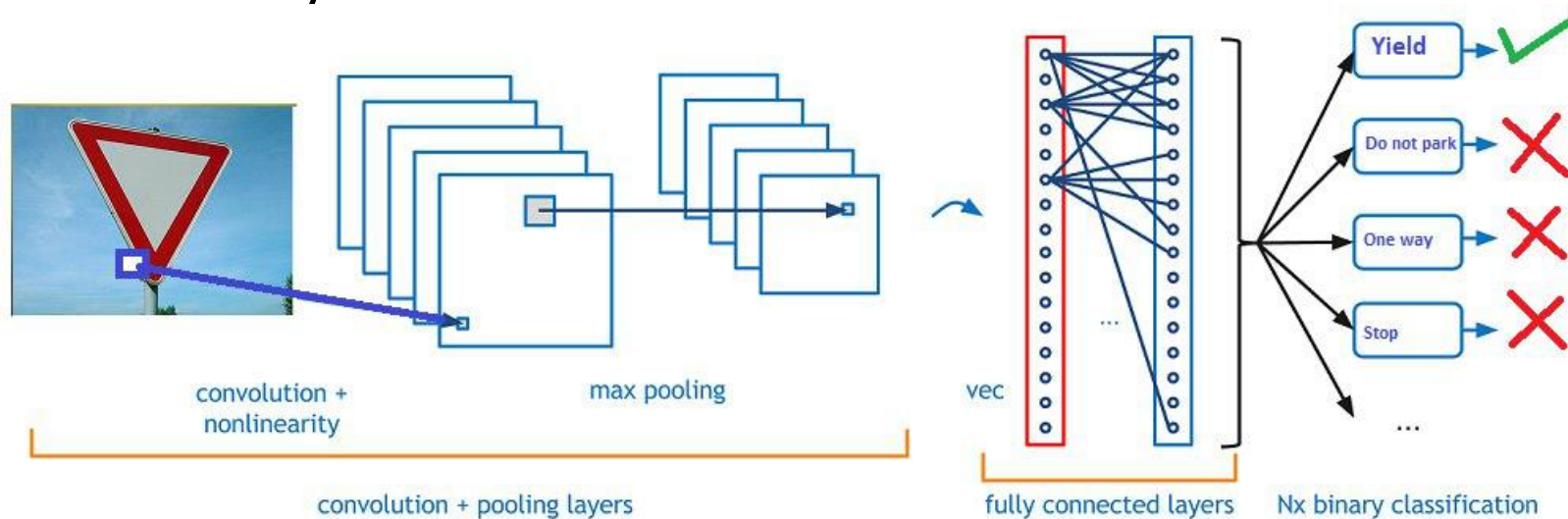
# Структура CNN

Свёрточная нейронная сеть сконфигурирована из трёх слоёв:

Convolution layer

Pooling layer

Fully Connected layer



# Реализация CNN

Код, создающий классификатор и обучающий модель:

```
def prepareClassifier():  
    classifier = Sequential()  
    classifier.add(Convolution2D(32,3, strides=(1,1),input_shape = (64, 64, 3),padding='valid', activation = 'relu'))  
    classifier.add(MaxPooling2D(pool_size = (2,2)))  
    classifier.add(Convolution2D(32, 3,strides=(1,1),activation = 'relu', padding='valid'))  
    classifier.add(MaxPooling2D(pool_size = (2,2)))  
    classifier.add(Flatten())  
    classifier.add(Dense(output_dim = 128, activation = 'relu'))  
    classifier.add(Dense(output_dim = 62, activation = 'softmax'))  
    classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])  
    return classifier  
  
classifier = prepareClassifier()  
train_datagen = ImageDataGenerator(rescale=1./150,shear_range = 0.2, zoom_range = 0.2)  
test_datagen = ImageDataGenerator(rescale=1./150,shear_range = 0.2, zoom_range = 0.2)  
training_set = train_datagen.flow_from_directory("TrafficSigns/Training",  
                                                target_size = (64, 64), batch_size = 8, class_mode = 'categorical')  
test_set = test_datagen.flow_from_directory("TrafficSigns/Testing",  
                                            target_size = (64, 64), batch_size = 8, class_mode = 'categorical')  
classifier.fit_generator(training_set, samples_per_epoch = 4552, nb_epoch =50,  
                        validation_steps=2520, validation_data = test_set )  
classifier.save('./savedmodel/my_model.h5')
```



# Вычислительный эксперимент

Язык программирования: Python 3.7.3

Среда разработки: PyCharm

Программные модули: OpenCV, NumPy,  
TensorFlow, Tkinter

Камера: Brookstone Rover 2.0



# Тестирование системы



**Спасибо за внимание!**