

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНСТИТУТ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАТИКИ И МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

Направление подготовки бакалавриата

09.03.04 Программная инженерия

Профиль направления подготовки бакалавриата

“Системное и прикладное программное обеспечение”

Отчет по проекту курса “Разработка для мобильных ОС”

АССИСТЕНТ ВОДИТЕЛЯ: РАСПОЗНАВАНИЕ ДОРОЖНЫХ
ЗНАКОВ

Выполнил:

студент группы 22207

Н. В. Гордеев _____
подпись

Место прохождения практики:

Институт математики и информационных технологий

Период работы над проектом:

01.12.22-16.01.23

Руководитель:

В. М. Димитров, старший преподаватель

подпись

Итоговая оценка

оценка

Содержание

1	Описание	3
2	Архитектура	4
2.1	Общая схема	4
2.2	MainActivity	4
2.2.1	onCreate	4
2.2.2	startNewActivity	5
2.3	CameraActivity	5
2.3.1	onCreate	5
2.3.2	classifyImage	6
2.3.3	onActivityResult	9
2.3.4	goBack	10
3	Описание интерфейса пользователя	11
3.1	Экран приветствия	11
3.2	Экран распознавания	13
	Дневник практики	19
	Заключение	20

1 Описание

О приложении:

Приложение "Ассистент водителя: распознавание знаков" посредством обращения к камере или файлам телефона распознаёт дорожный знак и выводит о нём информацию. Запускается после загрузки фотографии через кнопки «Камера» или «Файлы», останавливается и сбрасывается по нажатию на кнопку «На предыдущий экран», либо повторным выбором изображений для распознавания.

Нейронная сеть обучена заранее. Пользователю не нужно выполнять какие-либо действия для явной загрузки моделей TFLite в проект.

Приложение может запускаться как на устройстве, так и на эмуляторе.

Цель данной работы:

- закрепление и углубление теоретических знаний, полученных в процессе обучения;
- приобретение навыков и опыта по реализации и поддержке жизненного цикла мобильного приложения: управлению процессами разработки требований, оценки рисков, проектирования, конструирования, тестирования, сопровождения мобильного приложения, контролю за ходом реализации мобильного приложения, стратегическому планированию развития мобильного приложения.

Задачи проекта:

1. Обзор приложений по распознаванию объектов.
2. Обзор методов распознавания объектов.
3. Разработка собственного приложения.

Методы, способы достижения поставленных целей и задач.

1. Изучение и анализ статей в интернете;
2. Проведения экспериментов, анализ их результатов.

2 Архитектура

2.1 Общая схема

Программа состоит из двух страниц — приветственной и смысловой. Схему можно увидеть на рис. 1

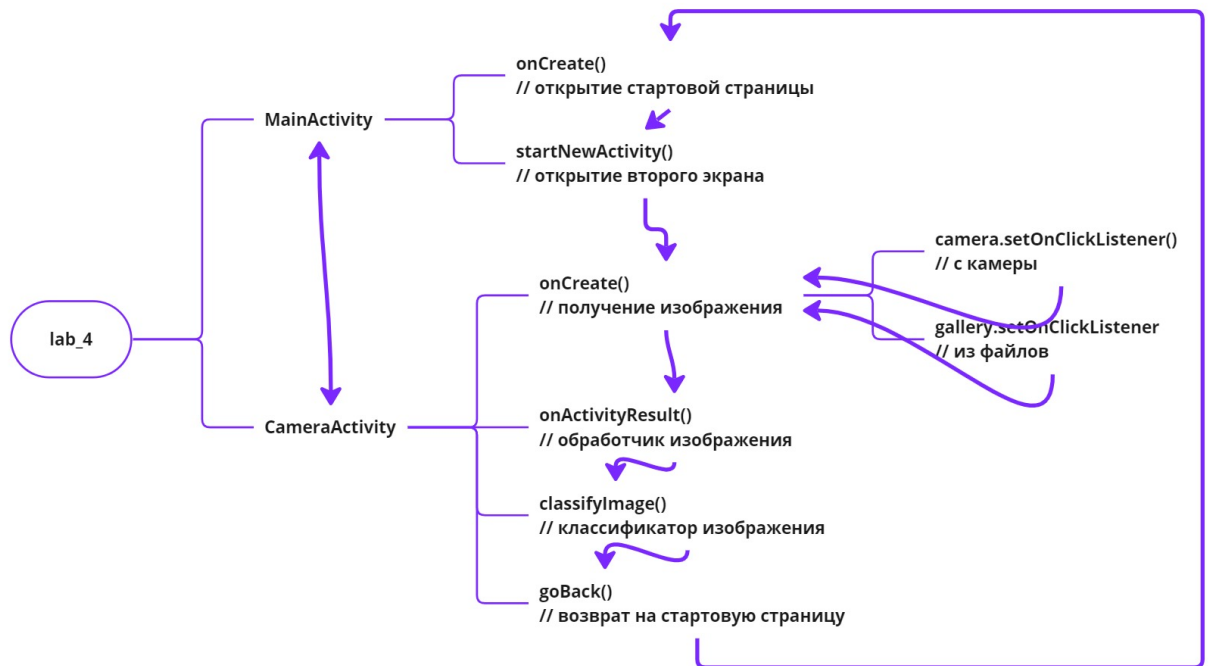


Рис. 1 – Схема программы

2.2 MainActivity

Класс "MainActivity" содержит в себе две функции: "onCreate" и "startNewActivity".

2.2.1 onCreate

Функция, начальной установки параметров при инициализации активности, представлена на языке Java в листинге 1.

Листинг 1 – Начальная установка параметров при инициализации активности "onCreate"

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}
```

2.2.2 startActivity

Функция перехода на второй экран на языке Java представлена в листинге 2.

Листинг 2 – Переход на второй экран "startNewActivity"

```
public void startNewActivity(View v) {  
    Intent intent = new Intent(this, CameraActivity.class);  
    startActivity(intent);  
}
```

2.3 CameraActivity

Класс "CameraActivity" содержит в себе четыре функции: "onCreate", "classifyImage", "onActivityResult" и "goBack".

2.3.1 onCreate

Функция получения изображения с камеры или из файлов представлена на языке Java в листинге 3.

Когда мы запускаем другое действие из текущего действия, чтобы получить для него результат, мы вызываем метод `startActivityForResult(intent, RESPONSE_CODE)`. Он перенаправляет на другое действие, например, открывает камеру, галерею и т.д. После получения изображения из галереи или камеры вернитесь к первому методу текущей активности, который вызывает `is onActivityResult(int requestCode, int resultCode, Intent data)`. В этом методе мы получаем результат, подобный снятому изображению с камеры или галереи.

Листинг 3 – Функция получения изображения с камеры или из файлов "onCreate"

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_camera);  
  
    camera = findViewById(R.id.button);  
    gallery = findViewById(R.id.button2);  
  
    confidence = findViewById(R.id.confidence);  
    result = findViewById(R.id.result);  
    imageView = findViewById(R.id.imageView);  
}
```

```

camera.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            if (checkSelfPermission(Manifest.permission.CAMERA) == PackageManager.
PERMISSION_GRANTED) {
                Intent cameraIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                startActivityForResult(cameraIntent, 3);
            } else {
                requestPermissions(new String[]{Manifest.permission.CAMERA}, 100);
            }
        }
    }
});
gallery.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent cameraIntent = new Intent(Intent.ACTION_PICK, MediaStore.Images.Media.
EXTERNAL_CONTENT_URI);
        startActivityForResult(cameraIntent, 1);
    }
});
}

```

2.3.2 classifyImage

Функция классификации изображения нейронной сетью представлена на языке Java в листинге 4.

Листинг 4 – Функция классификации изображения нейронной сетью "classifyImage"

```

public void classifyImage(Bitmap image) {
    try {
        Model model = Model.newInstance(getApplicationContext());

        // Vkhodnyye dannyye, podgotovka baytovogo bufera
        TensorBuffer inputFeature0 = TensorBuffer.createFixedSize(new int[]{1, 224, 224, 3},
        DataType.FLOAT32);

        ByteBuffer byteBuffer = ByteBuffer.allocateDirect(4 * imageSize * imageSize * 3);
        byteBuffer.order(ByteOrder.nativeOrder());
    }
}

```

```

// massiv znacheniy pikseley
int[] intValues = new int[imageSize * imageSize];
image.getPixels(intValues, 0, image.getWidth(), 0, 0, image.getWidth(), image.
getHeight());

// perebirayem kazhdy piksel i izvlekayem znacheniya RGB. Dobavlyayem eti
znacheniya po otdelnosti v baytovyy bufer
int pixel = 0;
for (int i = 0; i < imageSize; i++) {
    for (int j = 0; j < imageSize; j++) {
        int val = intValues[pixel++];
        byteBuffer.putFloat(((val >> 16) & 0xFF) * (1.f / 255.f));
        byteBuffer.putFloat(((val >> 8) & 0xFF) * (1.f / 255.f));
        byteBuffer.putFloat((val & 0xFF) * (1.f / 255.f));
    }
}

inputFeature0.loadBuffer(byteBuffer);

// Zapuskayet vyvod modeli i poluchayet rezultat
Model.Outputs outputs = model.process(inputFeature0);
TensorBuffer outputFeature0 = outputs.getOutputFeature0AsTensorBuffer();

// Otobrazheniye klassifikatsii s naibolshey dostovernostyu
float[] confidences = outputFeature0.getFloatArray();
int maxPos = 0;
float maxConfidence = 0;
for (int i = 0; i < confidences.length; i++) {
    if (confidences[i] > maxConfidence) {
        maxConfidence = confidences[i];
        maxPos = i;
    }
}

String[] classes = {"2.5 Dvizheniye bez ostanovki zapreshcheno", "2.1 Glavnaya
doroga", "3.1 Vyezd zapreshchen", "3.24 Ogranicheniye maksimalnoy skorosti", "4.1.1
Dvizheniye pryamo", "5.19.1, 5.19.2 Peshekhodnyy perekhod", "6.3.1 Mesto dlya razvorota"
};

result.setText(classes[maxPos]);

```

```

// Oтобrazheniye statisitki po vsem klassam
String s = "";
for (int i = 0; i < classes.length; i++) {
    s += String.format("%s: %.1f%%\n", classes[i], confidences[i] * 100);
}

confidence.setText(s);

// Releases model resources when they are no longer in use
model.close();

} catch (IOException e) {
    // Handle the exception
}
}

```

Для создания нейронной сети была Teachable Machine. Это платформа машинного обучения, которая позволяет тренировать нейросеть. На рис. 2 представлена структура нейронной сети.

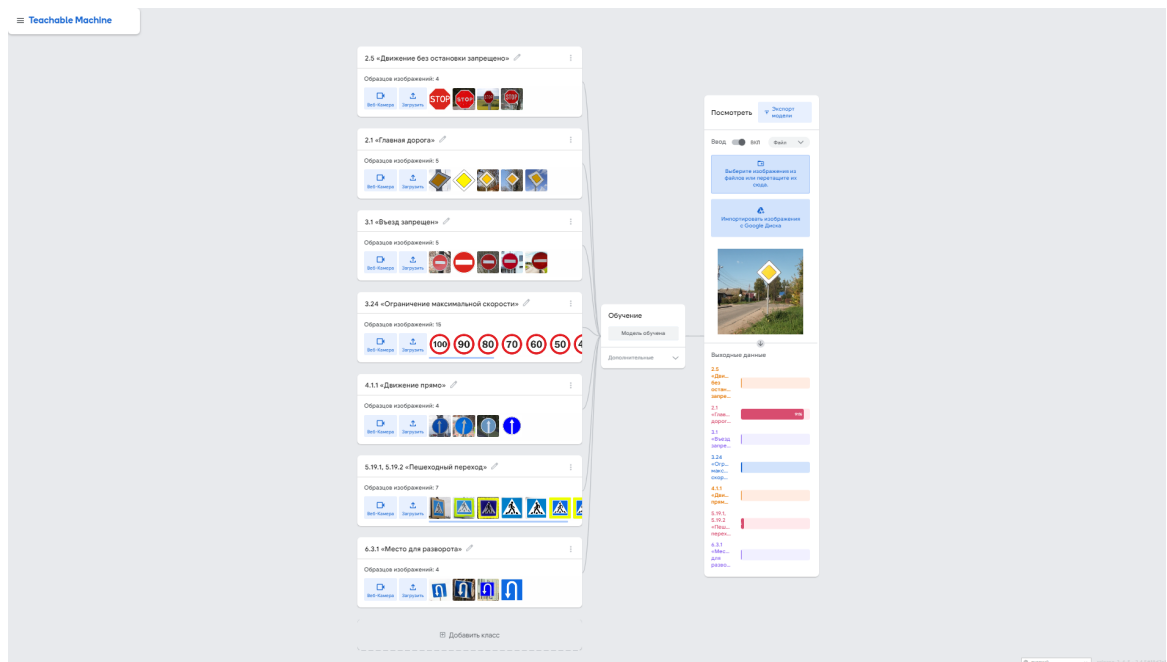


Рис. 2 – Нейронная сеть в Teachable Machine

Параметры, используемые при обучении модели, представлены в таблице 1.

Таблица 1 – Параметры обучения

Параметр	Значение
Число эпох	50
Размер пакета	16
Скорость обучения	0.001

2.3.3 onActivityResult

Функция обработки и отправки на классификацию изображения представлена на языке Java в листинге 5.

Листинг 5 – Функция обработки и отправки на классификацию изображения "onActivityResult"

```

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    if (resultCode == RESULT_OK) {
        // Yesli s kamery
        if (requestCode == 3) {
            Bitmap image = (Bitmap) data.getExtras().get("data");

            // Preobrazovaniye k kvadratu
            int dimension = Math.min(image.getWidth(), image.getHeight());
            image = ThumbnailUtils.extractThumbnail(image, dimension, dimension);

            // Otobrazit v prilozhenii snimok polzovatelya
            imageView.setImageBitmap(image);

            // klassifitsirovat
            image = Bitmap.createScaledBitmap(image, imageSize, imageSize, false);
            classifyImage(image);
        } else {
            // Yesli iz faylov
            Uri dat = data.getData();
            Bitmap image = null;
            try {
                image = MediaStore.Images.Media.getBitmap(this.getContentResolver(), dat);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```
// Otbrazit v prilozhenii fayl polzovatelya
imageView.setImageBitmap(image);

// klassifitsirovat
image = Bitmap.createScaledBitmap(image, imageSize, imageSize, false);
classifyImage(image);
}
}
super.onActivityResult(requestCode, resultCode, data);
}
```

2.3.4 goBack

Функция перехода на начальный экран представлена на языке Java в листинге 6.

Листинг 6 – Функция перехода на начальный экран "goBack"

```
@Override
public void goBack(View v) {
    Intent intent = new Intent(this, MainActivity.class);
    startActivity(intent);
}
```

3 Описание интерфейса пользователя

3.1 Экран приветствия

При входе пользователя встречает "Стартовый экран". Его можно увидеть на рис. 3¹.

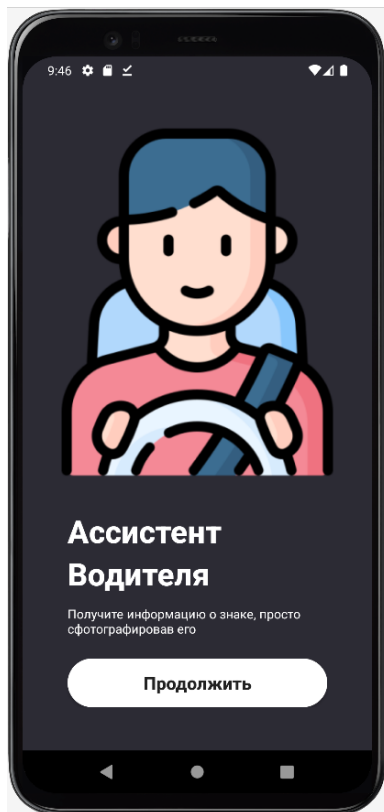


Рис. 3 – Стартовая страница телефона

Реализация стартового экрана представлена на языке XML в листинге 7.

Листинг 7 – Программа стартового экрана "activity_main"

```
@Override
<?xml version="1.0" encoding="utf-8"?>

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#2C2B34">

    <androidx.constraintlayout.widget.ConstraintLayout
```

¹Источник изображения водителя: <http://www.robgendlertropics.com/M31Page.html>

```

android:layout_width="match_parent"
android:layout_height="wrap_content"
android:padding="1dp"
tools:context=".MainActivity">

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="false"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="false"
    android:layout_marginTop="12dp"
    android:maxLength="292sp"
    android:text="@string/about"
    android:textColor="@color/white"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="false"
    android:layout_centerHorizontal="true"
    android:layout_centerVertical="false"
    android:layout_marginTop="31dp"
    android:fontFamily="@font/nunito_extrabold"
    android:text="@string/activity_main_app_name"
    android:textColor="@color/white"
    android:textSize="35.2sp"
    android:textStyle="bold"
    app:layout_constraintStart_toStartOf="@+id/textView"
    app:layout_constraintTop_toBottomOf="@+id/imageView2"
    tools:ignore="MissingConstraints" />

<Button
    android:id="@+id/button2"
    android:layout_width="292dp"

```

```

        android:layout_height="55dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="24dp"
        android:background="@drawable/button_main_screen"
        android:fontFamily="@font/nunito_extrabold"
        android:maxLength="900px"
        android:minWidth="900px"
        android:onClick="startNewActivity"
        android:text="@string/button_continue"
        android:textAllCaps="false"
        android:textSize="20dp"
        app:layout_constraintEnd_toEndOf="@+id/textView"
        app:layout_constraintStart_toStartOf="@+id/textView"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        tools:ignore="MissingConstraints" />

<ImageView
    android:id="@+id/imageView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="44dp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/lab_4_driver" />

</androidx.constraintlayout.widget.ConstraintLayout>

</ScrollView>

```

3.2 Экран распознавания

При переходе на "Экран распознавания" пользователя встречают две кнопки "Камера" и "Файлы". При выборе "Камера" можно сделать снимок дорожного знака и приложение определит его тип. При выборе "Файлы" можно выбрать уже скачанный файл со знаком, чтобы приложение определило его тип. Интерфейс этой страницы можно увидеть на рис. 4

Реализация страницы распознавания знаков представлена на языке XML в листинге 8.

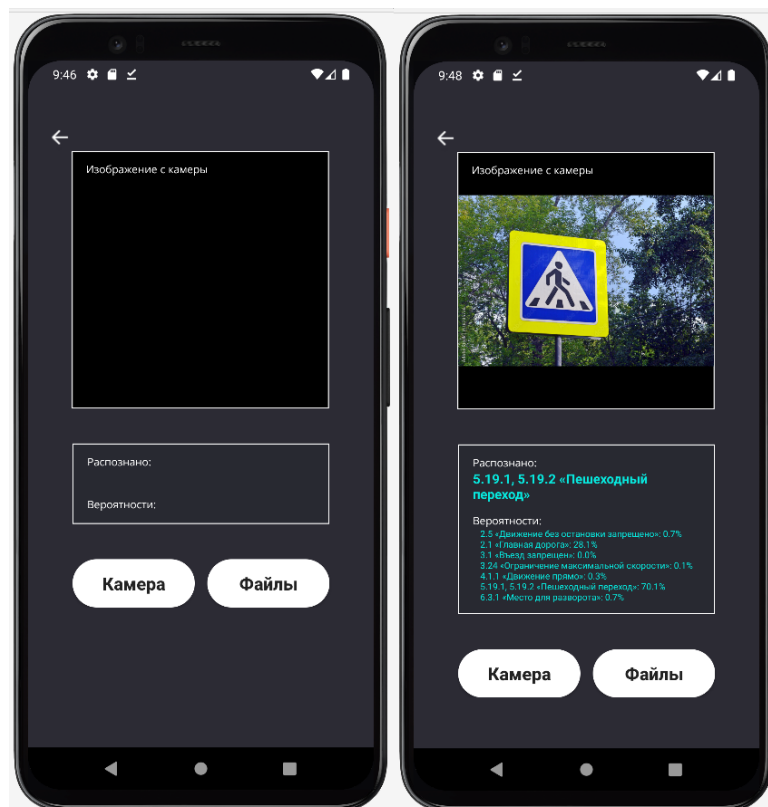


Рис. 4 – Страница распознавания знаков

Листинг 8 – Программа страницы распознавания знаков "activity_camera"

```
<?xml version="1.0" encoding="utf-8"?>

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#2C2B34">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="1dp"
        tools:context=".CameraActivity">

        <ImageButton
            android:id="@+id/button_to_previous_screen"
            android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:layout_above="@+id/button"
        android:layout_marginStart="22dp"
        android:layout_marginTop="44dp"
        android:background="#2C2B34"
        android:onClick="goBack"
        android:src="@drawable/icons_arrow_left"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:ignore="UsingOnClickInXml" />

```

```

<ImageView
    android:id="@+id/imageView"
    android:layout_width="292sp"
    android:layout_height="292sp"
    android:layout_centerHorizontal="true"
    android:background="@drawable/frame_around_image"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.495"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button_to_previous_screen" />

```

```

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="12dp"
    android:fontFamily="@font/open_sans"
    android:text="@string/camera_window_names"
    android:textColor="@color/white"
    android:textSize="12sp"
    app:layout_constraintStart_toStartOf="@+id/imageView"
    app:layout_constraintTop_toBottomOf="@+id/button_to_previous_screen" />

```

```

<androidx.constraintlayout.widget.ConstraintLayout
    android:id="@+id/constraintLayout"
    android:layout_width="292sp"
    android:layout_height="wrap_content"
    android:layout_marginTop="40dp"

```

```

android:background="@drawable/frame_around_statistics"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/imageView"
tools:ignore="MissingConstraints">

<TextView
    android:id="@+id/classified"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginStart="16dp"
    android:layout_marginTop="12dp"
    android:fontFamily="@font/open_sans"
    android:text="@string/recognized"
    android:textColor="@color/white"
    android:textSize="12sp"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/result"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/classified"
    android:layout_centerHorizontal="true"
    android:paddingLeft="16sp"
    android:paddingRight="16sp"
    android:text=""
    android:textColor="#08E8DE"
    android:textSize="15sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/classified" />

<TextView
    android:id="@+id/confidencesText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

```



```

        android:layout_below="@+id/result"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="16dp"
        android:layout_marginTop="12dp"
        android:fontFamily="@font/open_sans"
        android:text="@string/probabilities"
        android:textColor="@color/white"
        android:textSize="12sp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/result" />

```

```
<TextView
```

```

        android:id="@+id/confidence"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/confidencesText"
        android:layout_centerHorizontal="true"
        android:paddingLeft="16sp"
        android:paddingRight="16sp"
        android:text=""
        android:textColor="#08E8DE"
        android:textSize="10sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/confidencesText" />

```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<Button
```

```

        android:id="@+id/button"
        android:layout_width="139dp"
        android:layout_height="55dp"
        android:layout_above="@id/button2"
        android:layout_centerInParent="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="40dp"
        android:background="@drawable/button_main_screen"
        android:text="@string/btn_name_camera"
        android:textAllCaps="false"
        android:textSize="20dp"
        android:textStyle="bold"

```

```
app:layout_constraintStart_toStartOf="@+id/constraintLayout"  
app:layout_constraintTop_toBottomOf="@+id/constraintLayout" />
```

```
<Button
```

```
    android:id="@+id/button2"  
    android:layout_width="139dp"  
    android:layout_height="55dp"  
    android:layout_alignParentBottom="true"  
    android:layout_centerInParent="true"  
    android:layout_marginTop="40dp"  
    android:background="@drawable/button_main_screen"  
    android:text="@string/btn_name_files"  
    android:textAllCaps="false"  
    android:textSize="20dp"  
    android:textStyle="bold"  
    app:layout_constraintEnd_toEndOf="@+id/constraintLayout"  
    app:layout_constraintTop_toBottomOf="@+id/constraintLayout" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
</ScrollView>
```

Дневник практики

Период	Выполняемая работа
01.12.22 - 04.12.22	Изучение материалов по разработке приложений распознавания объектов на android
05.12.22 - 11.12.22	Изучение материалов по классификации данных при помощи нейронных сетей
12.12.22 - 18.12.22	Изучение материалов по созданию нейронных сетей
19.12.22 - 24.12.22	Проектирование интерфейса мобильного приложения
25.12.22	Подготовка данных для обучения нейронной сети
26.12.22 - 27.12.22	Создание и обучение нейронной сети
28.12.22 - 01.12.22	Разработка интерфейса мобильного приложения
02.01.23 - 06.01.23	Интеграция нейронной сети в мобильное приложение
07.01.23 - 08.01.23	Тестирование мобильного приложения
09.01.23 - 16.01.23	Подготовка отчета.
17.01.23	Представление отчёта руководителю

Заключение

Подводя итог, мне удалось выполнить частичный обзор систем по распознаванию объектов, обучить нейронную сеть, спроектировать и создать приложение, умеющее распознавать дорожные знаки.

Список литературы

1. Пишем простейшее приложение под Android для распознавания объектов с камеры // Дзен URL: <https://dzen.ru/a/XmJL-oJKLUN6gQpx> (дата обращения: 02.12.2022).
2. Распознавание объектов на android с помощью TensorFlow: от подготовки данных до запуска на устройстве // Хабр URL: <https://habr.com/ru/company/redmadrobot/blog/488210/> (дата обращения: 03.12.2022). <https://habr.com/ru/post/570052/>
3. How to add image for button in android? // stackoverflow URL: <https://stackoverflow.com/questions/4209582/how-to-add-image-for-button-in-android> (дата обращения: 29.12.2022).
4. Распознавание дорожных знаков с помощью CNN: Spatial Transformer Networks // Хабр URL: <https://habr.com/ru/company/newprolab/blog/339484/> (дата обращения: 07.12.2022).
5. OpenCV. Поиск дорожных знаков методом контурного анализа в Android // Хабр URL: <https://habr.com/ru/post/339506/> (дата обращения: 08.12.2022).
6. Smart IDReader SDK — добавляем распознавание в Android приложения // Хабр URL: <https://habr.com/ru/company/smartengines/blog/332670/> (дата обращения: 11.12.2022).
7. Using TensorFlow Lite on Android // Medium URL: <https://medium.com/tensorflow/using-tensorflow-lite-on-android-9bbc9cb7d69d> (дата обращения: 26.12.2022).
8. Teachable Machine URL: <https://teachablemachine.withgoogle.com/> (дата обращения: 27.12.2022).
9. How to change the color of a button? // stackoverflow URL: <https://stackoverflow.com/questions/32671004/how-to-change-the-color-of-a-button> (дата обращения: zz.zz.zzzz).
10. How to add image for button in android? // stackoverflow URL: <https://stackoverflow.com/questions/4209582/how-to-add-image-for-button-in-android> (дата обращения: zz.zz.zzzz).

11. Camera API // developers URL: <https://developer.android.com/guide/topics/media/camera> (дата обращения: zz.zz.zzzz).
12. How to Make Android Constraintlayout Scrollable Using Android Scrollview // YouTube Coding Demos URL: <https://www.youtube.com/watch?v=DpFNfQzhKQM> (дата обращения: zz.zz.zzzz).
13. Как изменить цвет верхней панели приложения? // Хабр URL: <https://qna.habr.com/q/280050> (дата обращения: zz.zz.zzzz).
14. Border for an Image view in Android? // stackoverflow URL: <https://stackoverflow.com/questions/3263611/border-for-an-image-view-in-android> (дата обращения: zz.zz.zzzz).
15. ConstraintLayout // androidx.* javadoc URL: <https://androidx.de/androidx/constraintlayout/> (дата обращения: zz.zz.zzzz).