



Информатика и
Вычислительная техника

Московский институт
электроники и математики

Москва
2022

Разработка веб-приложения ColorFind с использованием технологии искусственного интеллекта

Работу выполнила:
Бессонова Ирина, БИВ181

Преподаватель курса «Разработка веб-приложений»:
Профессор Трубочкина Надежда Константиновна

Актуальность: в современном мире активное развитие получили интернет-технологии, способные упростить жизнь человека во многих аспектах. Разработанное решение позволит решить **проблему подбора цвета** в интерьере, что является **актуальной задачей** в мире дизайна и архитектуры.

Цель: упростить процесс подбора краски в интерьере с помощью средств **веб-разработки** и модифицировать технологию **распознавания образов**.

Задача: разработать **приложение** с применением функций **искусственного интеллекта**, позволяющее **визуализировать** различные цвета в интерьере.

Новизна: **комплексное решение** позволяющее реализовывать поставленные задачи. Пользователь может загрузить **собственное изображение** помещения для выбора цветовой раскладки и **подобрать** предметы мебели на одном веб-ресурсе.

Область применения: технология находит широкое применение в **сфере дизайна интерьеров**. Оно может быть использовано как обычными производителями, так и производителями краски для визуализации использования своей продукции.

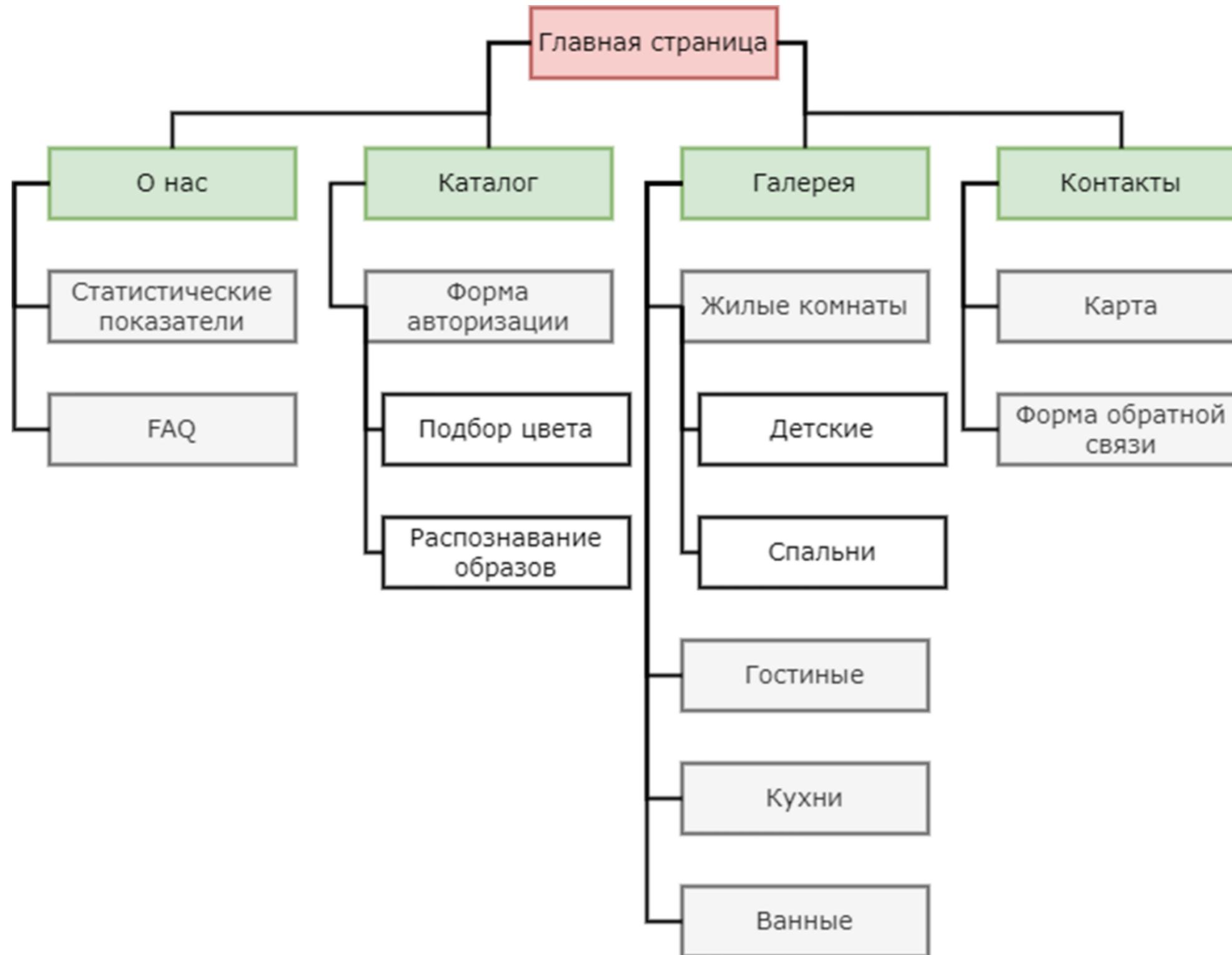
Обзор аналогов

	Палитра	Примеры	Возможность работы со своим изображением	Колористические диаграммы	Ограничения	Возможность поделиться результатом	Интерфейс	Дополнительно
Tikkurila	Большая	Есть	Есть	Есть	Документ JPG или PNG до 15 МБ	JPG и PDF	Понятный на русском языке	-
ColorSnap	Возможность создавать свою	Есть	Есть	Есть	-	По почте или в социальных сетях	Сложный на английском языке	Функция расчета количества краски
Dulux Visualizer	Большая, разбитая на категории	Нет	Есть	Автоматическое составление	Датчик движения на устройстве, хорошее освещение на изображении	По почте или в социальных сетях	-	Необходимость регистрации
Фундамент	Каталог оттенков от различных производителей, менее 22 цветов	Есть	Нет	Таблица, позволяющая гармонично сочетать тона	-	Результат нельзя экспортировать или отправить на электронную почту	-	Настройка отделки
Bergge	Большая коллекция	Только четыре	Нет	Нет	-	Электронная почта	-	-
ColorFind	Большая	Есть	Есть	Есть	-	JPG, PDF почта	Понятный на русском языке	Система распознавания образов

Структура проекта



Структура сервиса



Макет страницы

Верхняя панель и меню

Контентная часть

Footer с копирайтом

Выбор средств разработки



WordPress — свободно распространяемая система управления содержимым сайта с открытым исходным кодом; написана на PHP; сервер базы данных — MySQL; выпущена под лицензией GNU GPL версии 2. Сфера применения — от блогов до достаточно сложных новостных ресурсов. Встроенная система «тем» и «плагинов» вместе с удачной архитектурой позволяет конструировать проекты широкой функциональной сложности.



TypeScript — язык программирования, представленный Microsoft в 2012 году и позиционируемый как средство разработки веб-приложений, расширяющее возможности JavaScript. TypeScript является обратно совместимым с JavaScript и компилируется в последний. Фактически, после компиляции программу на TypeScript можно выполнять в любом современном браузере или использовать совместно с серверной платформой Node.js.

Найдите цвета вашей жизни!

Загрузите изображение и мы подберем краску для вашего интерьера

[ПОДОБРАТЬ КРАСКУ](#)



Создай уникальное цветовое решение

С помощью разработанного искусственного интеллекта вы можете загрузить изображение своего помещения на сайт и попробовать подобранные цветовые палитры или создать свою

[ПОПРОБОВАТЬ](#)

ColorFind сотрудничает с лучшими производителями краски по всему миру. Мы тщательно осуществляем выбор партнеров, поэтому вы можете быть уверены в качестве товара.








О НАС

Команда единомышленников

Над платформой трудится команда, состоящая из абсолютно разных людей. Но их объединяет общая цель! Ежедневно программисты, дизайнеры, менеджеры по продукту и аналитики стараются сделать нашу платформу удобнее и функциональнее.



[УЗНАТЬ БОЛЬШЕ](#)

Инновация в мире Дизайна Интерьера

Комфортное Использование

Интерфейс программы не заставит вас ломать голову. Все функции интуитивно понятны и не потребуют тратить время на освоение технологии!

Искусственный Интеллект

Система способна распознавать мельчайшие детали интерьера. Она совершенствуется с каждым днем!



О нас Команда Работы



Наша компания вышла на рынок в 2021 году. Перед этим мы прошли непростой путь становления.

Сначала мы сотрудничали с корпорацией WAB, но нас часто загоняли в жесткие рамки. Для творчества это приговор. Так, в 2017 году сформировалась команда единомышленников, которые позднее создали этот веб-сервис и продолжают развивать его до сих пор.

- 6** дней на доставку вашего заказа
- 9** надежных и проверенных поставщиков
- 10** офисов в Москве и Санкт-Петербурге
- 14** часа в сутки работает служба поддержки

Команда, работающая для вас!

Каждый член нашей команды работает, чтобы сделать сервис удобнее, производительнее и полезнее. Если у вас есть предложения по его модернизации, то обязательно пишите нам!

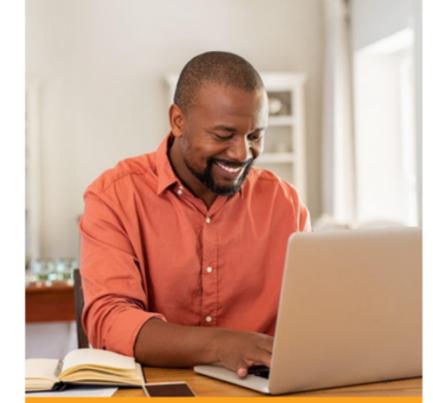
[НАПИСАТЬ](#)




Инновация в мире Дизайна Интерьера

Искусственный Интеллект

Система способна распознавать мельчайшие детали интерьера. Она совершенствуется с каждым днем!

Комфортное Использование

Интерфейс программы не заставит вас ломать голову. Все функции интуитивно понятны и не потребуют тратить время на освоение технологии!




Команда, работающая для вас!

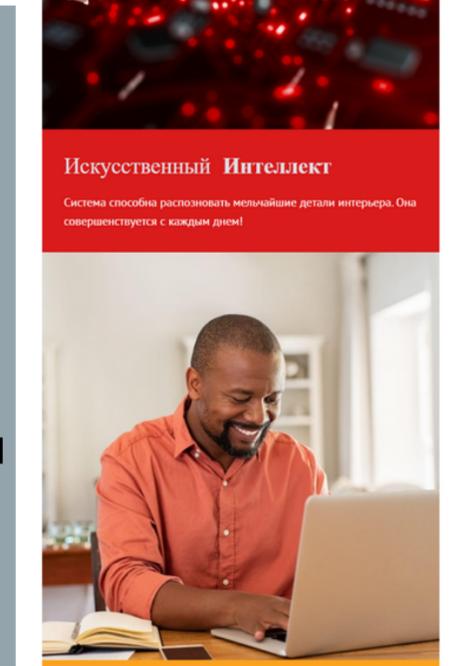
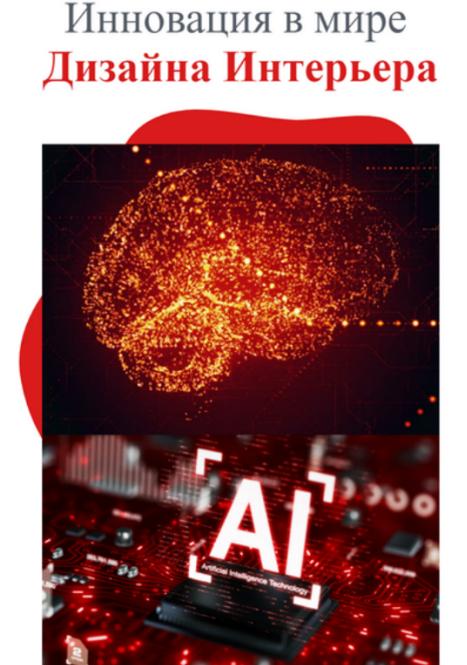
Каждый член нашей команды работает, чтобы сделать сервис удобнее, производительнее и полезнее. Если у вас есть предложения по его модернизации, то обязательно пишите нам!

[НАПИСАТЬ](#)

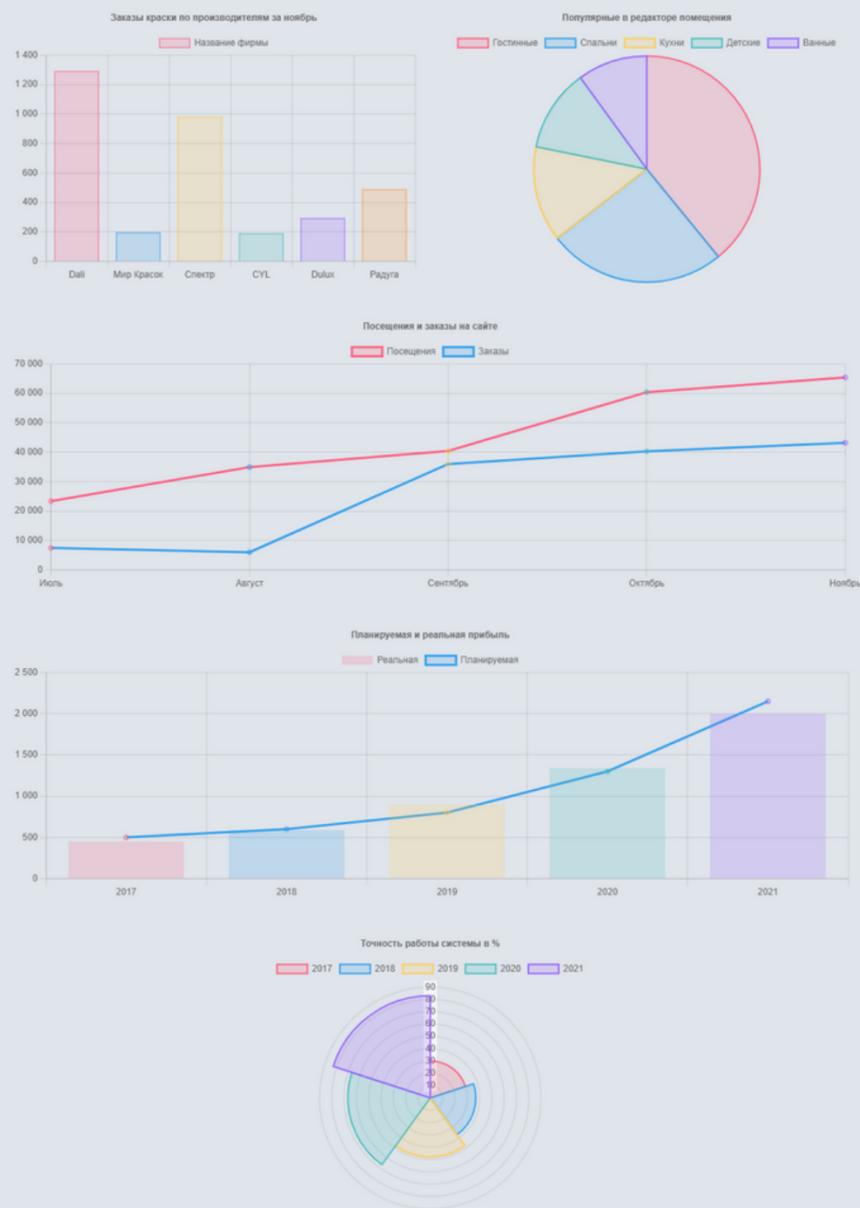



Версия для ПК ←

Мобильная версия →



Наша статистика



Ответы и вопросы

В каком браузере можно использовать редактор? +

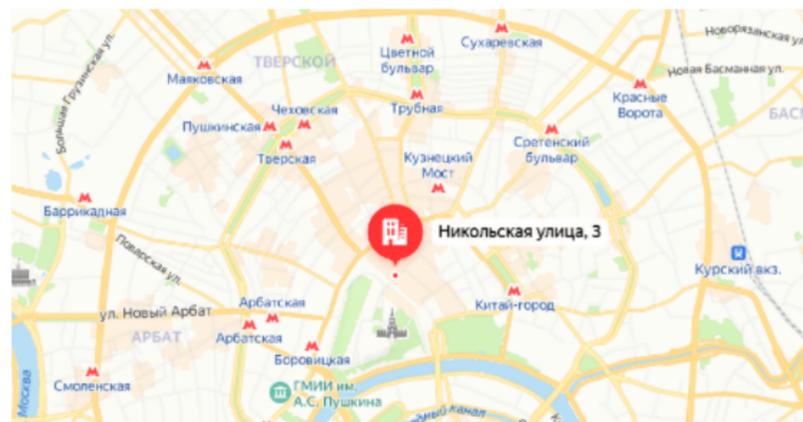
Редактор работает во всех существующих на данный момент браузерах.

Могу я создать свой дизайн, или есть возможность использовать только готовые? +

Как происходит оплата заказа? +



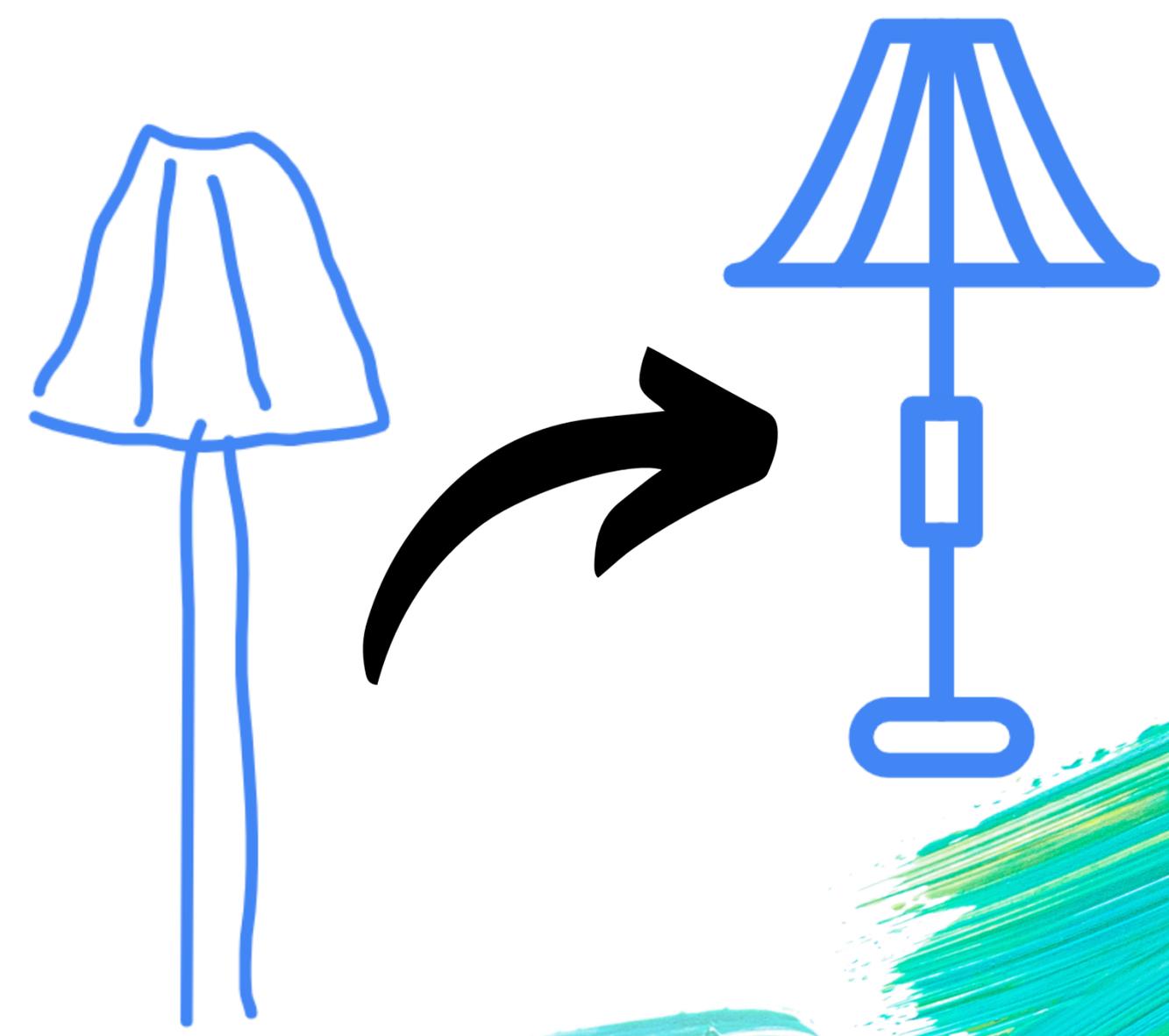
Где нас найти



Офис работает ежедневно с 10:00 до 20:00. Перед посещением свяжитесь позвоните по номеру +7 999 999 99 99

Свяжитесь с нами

Работа интеллектуальных алгоритмов



Описание интеллектуального алгоритма

В качестве основополагающего алгоритма была использована нейронная сеть AutoDraw от Google. Ее основное преимущество заключается в **ПОСТОЯННОМ обучении** на основе пользовательского опыта.

Мы создали службу, которая позволяет нам делать запросы к Google AutoDraw API, а также будет отвечает за загрузку некоторых трафаретов, которые сопоставляются, при разрешении запроса с предложениями по рисованию.

У этой сервиса есть **два основных метода**: loadStencils(), который загружает все шаблоны, которые сопоставляем с ответом на запрос, и drawSuggestions(), который отвечает за выполнение необходимого запроса к Google AutoDraw API и возвращает совпадающие шаблоны, которые являются предложениями нарисованного изображения.

```
import { Headers, RequestOptions, Http } from '@angular/http';
import { Injectable } from '@angular/core';
import 'rxjs/add/operator/map';
const API_ENDPOINT = 'https://inputtools.google.com/request?ime=handwriting&app=autodraw&';
const STENCILS_ENDPOINT = 'src/data/stencils.json';
@Injectable()
export class AutoDrawService {
  stencils;
  constructor(
    private http: Http
  ) { }
  loadStencils () {
    this.http.get(STENCILS_ENDPOINT).subscribe(response => this.stencils = response.json())
  }
  drawSuggestions (
    shapes: Array<Array<number[]>>, drawOptions: {
      canvasWidth: number,
      canvasHeight: number
    }) {
    let headers = new Headers({
      'Content-Type': 'application/json; charset=utf-8'
    });
    let options = new RequestOptions({ headers });
    return this.http.post(
      API_ENDPOINT,
      JSON.stringify({
        input_type: 0,
        requests: [{
          language: 'autodraw',
          writing_guide: {
            "width": drawOptions.canvasWidth,
            "height": drawOptions.canvasHeight
          },
          ink: shapes
        }
      ]
    ),
    options
  ).map(response => {
    let data = response.json();
    let results = JSON.parse(data[1][0][3].debug_info.match(/SCORESINKS: (.*) Service_Re
      .map(result => {
        return {
          name: result[0],
          icons: (this.stencils[result[0]] || []).map(collection => collection.src)
        }
      });
    return results;
  });
}
```

Описание алгоритма

Этот компонент вводит службу `AutoDraw`, также он создает переменную экземпляра `canvas`, указывающую на наш `#canvas` `ViewChild`, который был определен в пользовательском интерфейсе. В жизненном цикле компонента `ngOnInit` загружаются трафареты `autodrawservice` и подписываются на некоторые интересующие действия мыши, передавая всю информацию об этих событиях основному методу `draw()`, который в зависимости от типа `MouseEvent` вызывает другие методы для выполнения определенной логики.

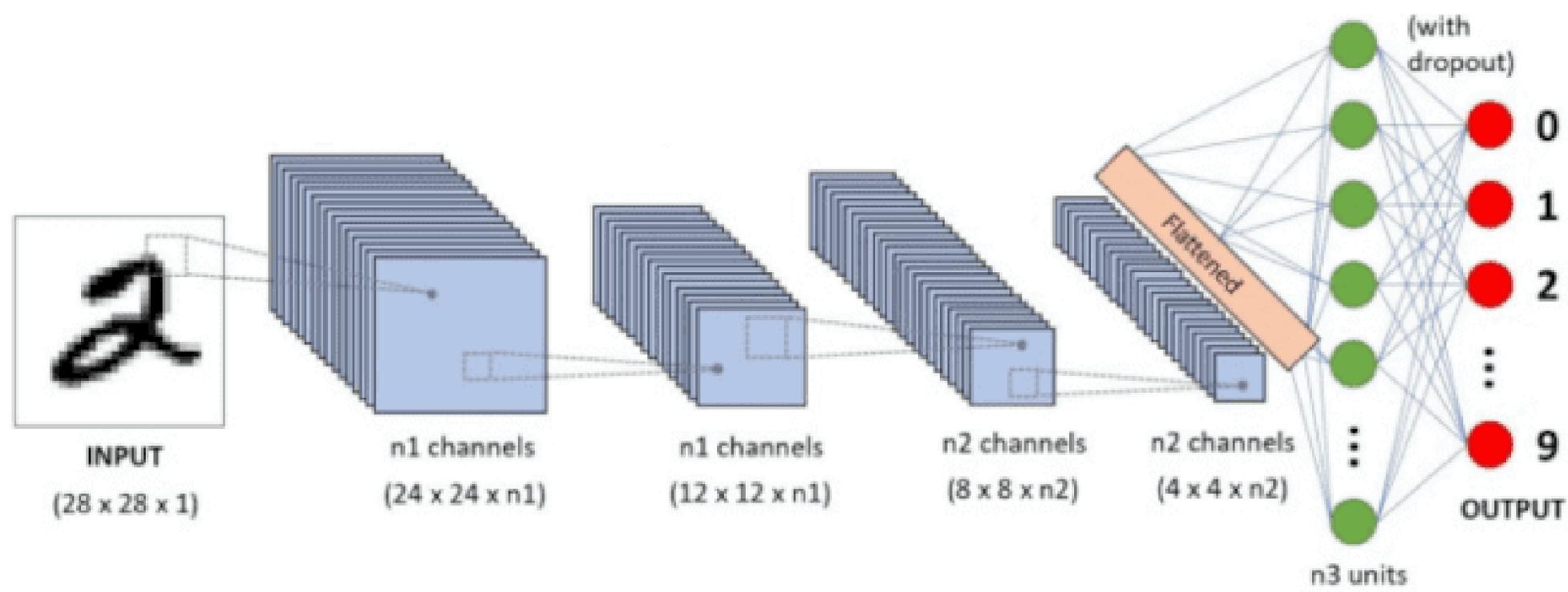
```
import { Component, OnInit, OnDestroy, ViewChild } from '@angular/core';
import { Subscription } from 'rxjs/Subscription';
import { Observable } from 'rxjs/Observable';
import { AutoDrawService } from './services';
import 'rxjs/add/observable/fromEvent';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent implements OnInit, OnDestroy {
  constructor (
    private autoDrawService: AutoDrawService
  ) {}
  @ViewChild('canvas') canvas;
  drawSuggestions: Array<object>;
  canvasMouseEventSubscriptions: Subscription[];
  previousXAxis: number = 0;
  previousYAxis: number = 0;
  currentXAxis: number = 0;
  currentYAxis: number = 0;
  context;
  pressedAt: number;
  pressing: boolean = false;
  currentShape: Array<number[]>;
  shapes: Array<Array<number[]>> = [];
  intervalLastPosition: number[] = [-1, -1];
  ngOnInit () {
    this.autoDrawService.loadStencils();
    this.context = this.canvas.nativeElement.getContext("2d");
    let mouseEvents = ['mousemove', 'mousedown', 'mouseup', 'mouseout'];
    this.canvasMouseEventSubscriptions = mouseEvents.map(
      (mouseEvent: string) => Observable
        .fromEvent(this.canvas.nativeElement, mouseEvent)
        .subscribe((event: MouseEvent) => this.draw(event))
    );
  }
  ngOnDestroy () {
    for (let mouseEventSubscription of this.canvasMouseEventSubscriptions) {
      mouseEventSubscription.unsubscribe();
    }
  }
  eraseCanvas () {
    this.shapes = [];
    this.context.clearRect(0, 0, this.canvas.nativeElement.width, this.canvas.nativeElement.height);
  }
  prepareNewShape () {
    this.currentShape = [
      [], // X coordinates
      [], // Y coordinates
      [] // Times
    ];
  }
  storeCoordinates () {
    if (this.intervalLastPosition[0] !== this.previousXAxis && this.intervalLastPosition[1] !== this.previousYAxis) {
      this.intervalLastPosition = [this.previousXAxis, this.previousYAxis];
      this.currentShape = [
        [...this.currentShape[0], this.previousXAxis],
        [...this.currentShape[1], this.previousYAxis],
        [...this.currentShape[2], Date.now() - this.pressedAt]
      ];
    }
  }
  onDrawingMouseDown (mouseEvent: MouseEvent) {
    let highlightStartPoint, drawColorStartingPoint = 'black';
```

```
    this.previousXAxis = this.currentXAxis;
    this.previousYAxis = this.currentYAxis;
    this.currentXAxis = mouseEvent.clientX - this.canvas.nativeElement.offsetLeft;
    this.currentYAxis = mouseEvent.clientY - this.canvas.nativeElement.offsetTop;
    this.pressing = true;
    this.pressedAt = Date.now();
    highlightStartPoint = true;
    this.prepareNewShape();
    if (highlightStartPoint) {
      this.context.beginPath();
      this.context.fillStyle = drawColorStartingPoint;
      this.context.fillRect(this.currentXAxis, this.currentYAxis, 2, 2);
      this.context.closePath();
      highlightStartPoint = false;
    }
    // Stores coordinates every 9ms
    return window.setInterval(() => this.storeCoordinates(), 9);
  }
  onDrawingMouseMove (mouseEvent: MouseEvent) {
    let drawStroke = 8, drawColor = 'black';
    this.previousXAxis = this.currentXAxis;
    this.previousYAxis = this.currentYAxis;
    this.currentXAxis = mouseEvent.clientX - this.canvas.nativeElement.offsetLeft;
    this.currentYAxis = mouseEvent.clientY - this.canvas.nativeElement.offsetTop;
    this.context.beginPath();
    this.context.moveTo(this.previousXAxis, this.previousYAxis);
    this.context.lineTo(this.currentXAxis, this.currentYAxis);
    this.context.strokeStyle = drawColor;
    this.context.fillStyle = drawColor;
    this.context.lineCap = 'round';
    this.context.lineJoin = 'round';
    this.context.lineWidth = drawStroke;
    this.context.stroke();
    this.context.closePath();
  }
  draw(mouseEvent: MouseEvent) {
    let storeCoordinateInterval;
    if (mouseEvent.type === 'mousedown') {
      storeCoordinateInterval = this.onDrawingMouseDown(mouseEvent);
    }
    if (mouseEvent.type === 'mouseup' || this.pressing && mouseEvent.type === 'mouseout') {
      this.pressing = false;
      clearInterval(storeCoordinateInterval);
      this.commitCurrentShape();
    }
    if (mouseEvent.type === 'mousemove' && this.pressing) {
      this.onDrawingMouseMove(mouseEvent);
    }
  }
  commitCurrentShape () {
    this.shapes.push(this.currentShape);
    let drawOptions = {
      canvasWidth: this.canvas.nativeElement.width,
      canvasHeight: this.canvas.nativeElement.height
    };
    this.autoDrawService.drawSuggestions(this.shapes, drawOptions)
      .subscribe(suggestions => this.drawSuggestions = suggestions);
  }
  pickSuggestion(source: string) {
    this.eraseCanvas();
    let image = new Image();
    image.onload = () => this.context.drawImage(image, 0, 0);
    image.src = source;
  }
}
```

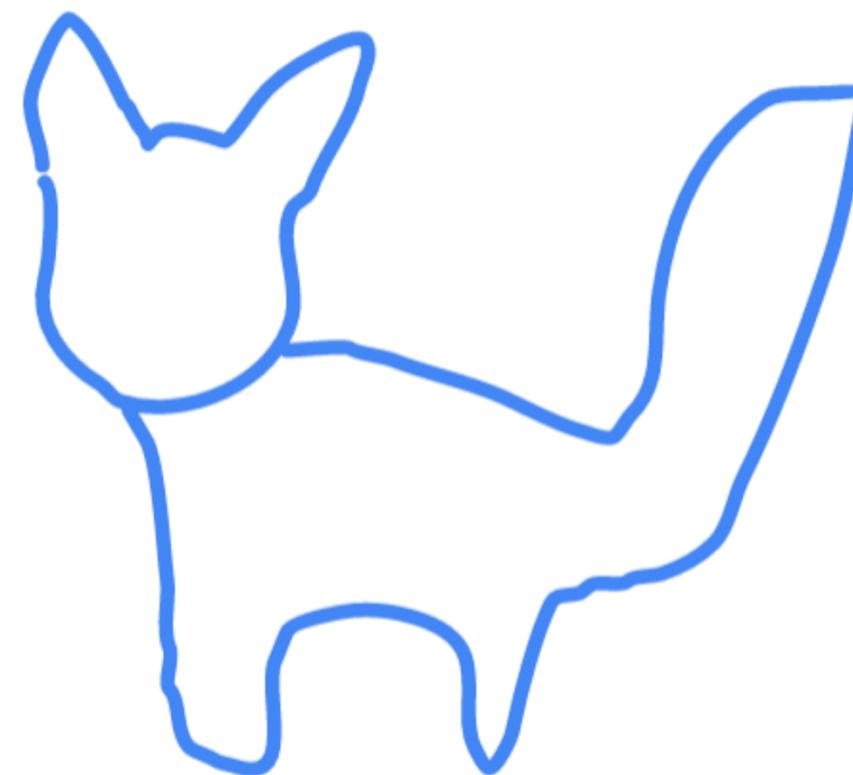
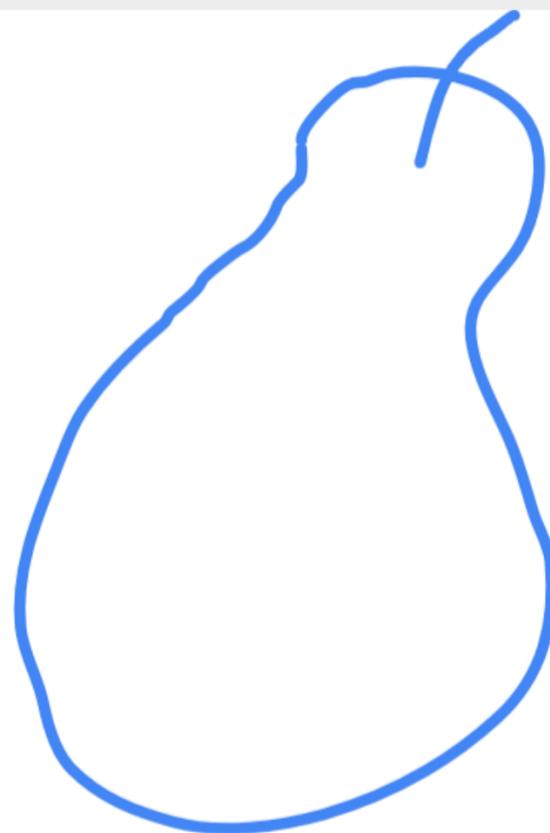
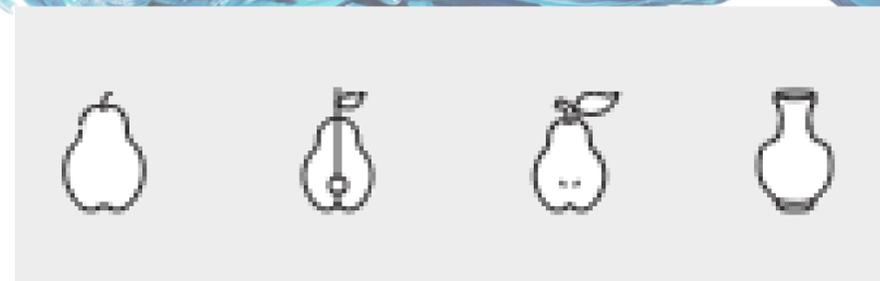
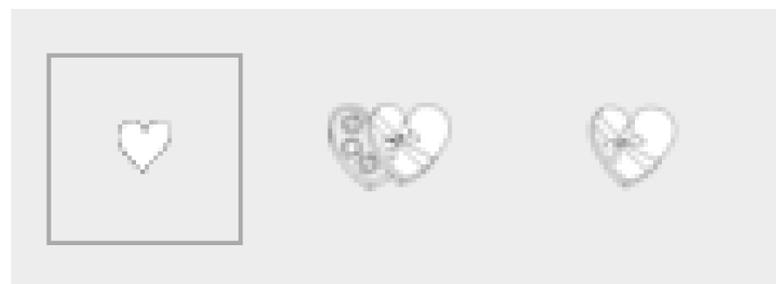
Архитектура нейронной сети



Свёрточная нейронная сеть — специальная архитектура искусственных нейронных сетей нацеленная на эффективное распознавание образов, входит в состав технологий глубокого обучения. Использует некоторые особенности зрительной коры, в которой были открыты так называемые простые клетки, реагирующие на прямые линии под разными углами, и сложные клетки, реакция которых связана с активацией определённого набора простых клеток. Структура сети — однонаправленная (без обратных связей), принципиально многослойная. Для обучения используются стандартные методы, чаще всего метод обратного распространения ошибки. Функция активации нейронов ReLU.

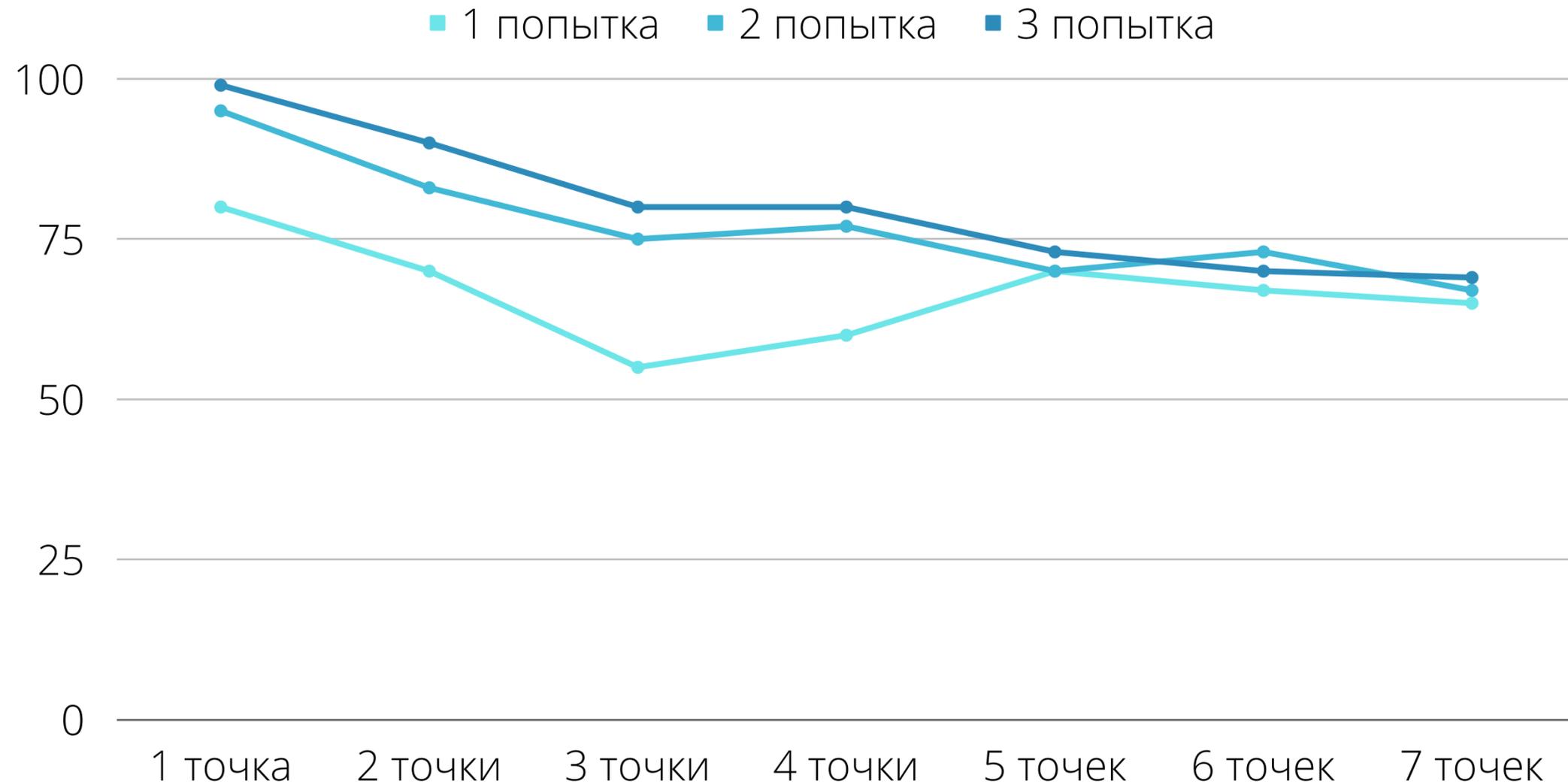


Эксперименты

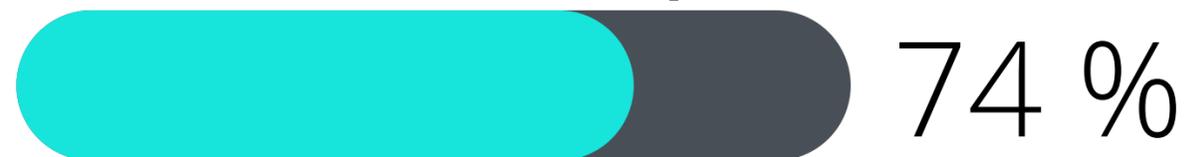


Для тестирования корректности работы ИИ Было проведено 3 эксперимента: на простом объекте и двух более сложных. Результаты эксперимента показали, что на таком простом рисунке как сердце точность распознавания приближена к 100%. При изображении рисунка более высокого уровня сложности - груша и самого высокого - кот точность распознавания составила порядка 75%, что является отличным показателем!

Точность определения изображения в зависимости от количества точек



Средняя точность работы системы



В ходе работы **выполнено** следующее:

- разработано приложение с применением функций искусственного интеллекта, позволяющее визуализировать различные цвета в интерьере
- реализована функция загрузки собственного изображения в приложение
- проведен набор экспериментов, на основе которых установлено, что точность работы ИИ составляет порядка 74%
- разработан адаптивный интерфейс веб-страницы, способной обрабатывать запросы пользователей
- изучен, применен и модифицирован алгоритм ИИ для распознавания визуальных образов

Данный проект может быть доработан и имеет следующие **перспективы развития**:

- упрощение веб-страницы для оптимального перехода к алгоритмам искусственного интеллекта
- реализация функции закраски плоскости поверхности (стены, пола, потолка) одновременно
- загрузка в базу шаблонов ИИ товаров из магазинов мебели для оптимального подбора предметов интерьера



Спасибо за внимание!