УДК 004.032.26

Создание приложения распознавания текста с изображения при помощи языка программирования C#

Ульянов Егор Андреевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
Целью данной статьи является, применение языка программирования C#, среды разработки Visual Studio 2019 и библиотеки «Tesseract.Net SDK», в создании программы для распознавания текста с изображения. Практическим результатом является разработанное программа.
Ключевые слова: Visual Studio, распознание текста с изображения программа, язык программирования C#

Creating a text recognition application from an image using the C# programming language

Ulianov Egor Andreevich
Sholom-Aleichem Priamursky State University
Student

Abstract
The purpose of this article is to use the C# programming language, the Visual Studio 2019 development environment and the library "Tesseract.Net SDK", in creating a program for recognizing text from an image. The practical result is the developed program.
Keywords: Visual Studio, a program for text recognition from an image, C# programming language

Оптическое распознавание символов — механический или электронный перевод изображений рукописного, машинописного или печатного текста в текстовые данные, использующиеся для представления символов в компьютере (например, в текстовом редакторе). Распознавание широко применяется для преобразования книг и документов в электронный вид, для автоматизации систем учёта в бизнесе или для публикации текста на веб-странице. Оптическое распознавание символов позволяет редактировать текст, осуществлять поиск слов или фраз, хранить его в более компактной форме, демонстрировать или распечатывать материал, не теряя качества, анализировать информацию, а также применять к тексту электронный перевод, форматирование или преобразование в речь. Оптическое распознавание текста является исследуемой проблемой в областях распознавания образов, искусственного интеллекта и компьютерного зрения.

Системы оптического распознавания текста требуют калибровки для работы с конкретным шрифтом; в ранних версиях для программирования было необходимо изображение каждого символа, программа одновременно могла работать только с одним шрифтом. В настоящее время доступно все больше уже готовых библиотек, таких как «Tesseract.Net SDK» которые уже умеют, с высокой степенью точности распознающие большинство шрифтов. Некоторые системы оптического распознавания текста способны восстанавливать исходное форматирование текста, включая изображения, колонки и другие нетекстовые компоненты.

В своей работе Н. Н. Додобоев, О. И. Кукарцева, Я. А. Тынченко рассмотрели вопросы появления различных языков программирования (в частности C#), определения особенностей этих языков, а также составления основных видов и классификаций языков программирования [1]. З. С. Магомадова рассмотрела языки программирования высокого уровня, особенности, недостатки и сложности в изучении, а также описала несколько легких алгоритмов [2]. В своей работе И. Ю. Просвирнина создала приложение «Морской бой», обладающее игровым искусственным интеллектом, в котором предусмотрен режим игры «Игрок против компьютера» [3]. В статье Ф.В. Патюченко, И.С. Слащев, А.В. Клименко, Л.А. Трегубенко были рассмотрены два подхода для создания программ на базе windows, обоснование выбора одного из них [4]. Д.З. Хасаева, А.Ю. Демин в своей работе проанализировали графическую библиотека «3Dbodies» и привели пример ее использования [5]. В статье J. Li, Q. Liu, H. Su описано использование WPF для создания модуля виртуальной реальности портальной системы мониторинга поворотного крана [6]. S. Torsten, S. Toshiyuki, M. Derek в своем отчете описали интегрированную программную систему контроля качества, разработанную для значительного повышения производительности и качества анализа проб в лабораториях по анализу бериллия на территории комплекса Министерства энергетики США [7].

Цель исследования – применяя возможности среды разработки Visual Studio, языка программирования C# и библиотеки «Tesseract.Net SDK», создать программу для распознавания текста с изображения.

Для создания программы будет использовано программное обеспечение Visual Studio, язык программирования C#, и библиотека «Tesseract.Net SDK».

Для проекта использовалась Windows Presentation Foundation (WPF) [8], платформа пользовательского интерфейса для создания клиентских приложений для настольных систем. Платформа разработки WPF поддерживает широкий набор компонентов для разработки приложений, включая модель приложения, ресурсы, элементы управления, графику, макет, привязки данных, документы и безопасность. Эта платформа является частью платформы .NET.

Начнём разработку приложения, с создания WPF приложения. Сразу назовем проект «ImageToText» (рис. 1-2).

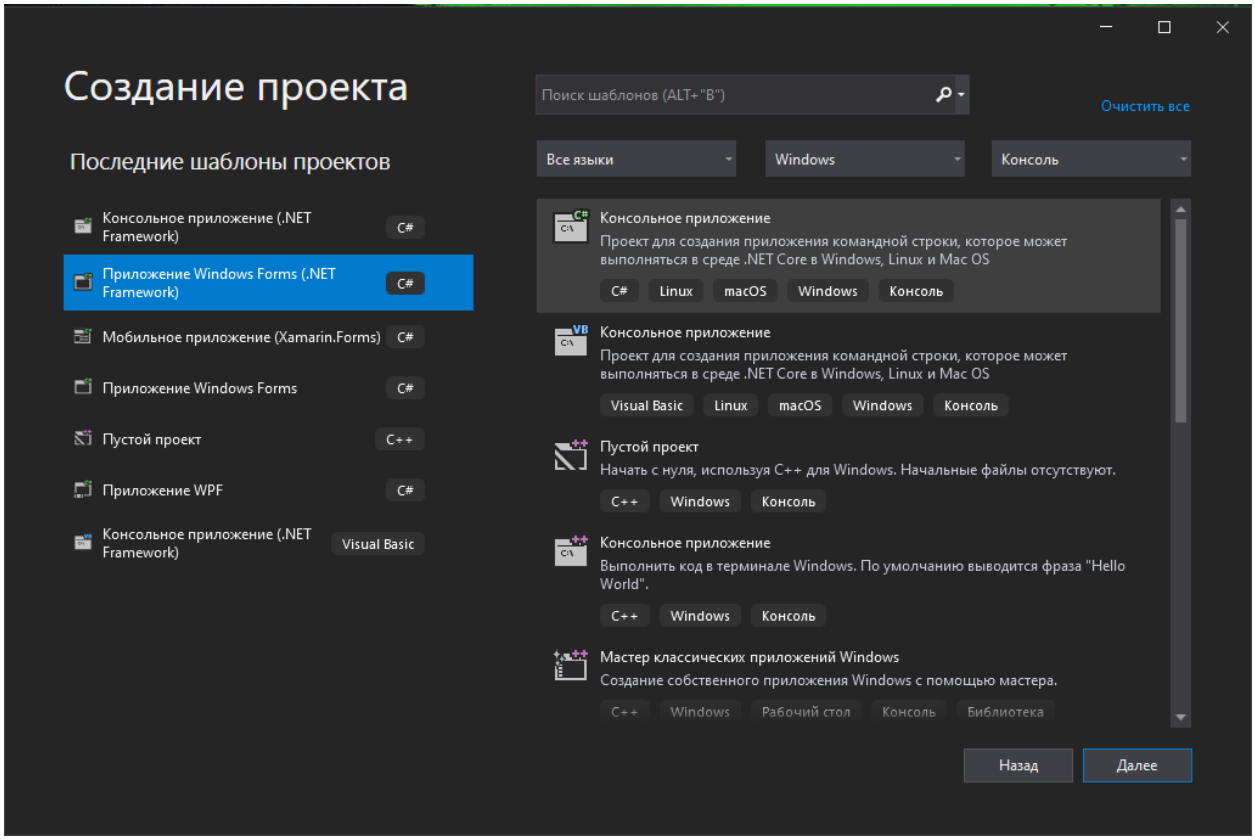

Рисунок 1. Создание проекта

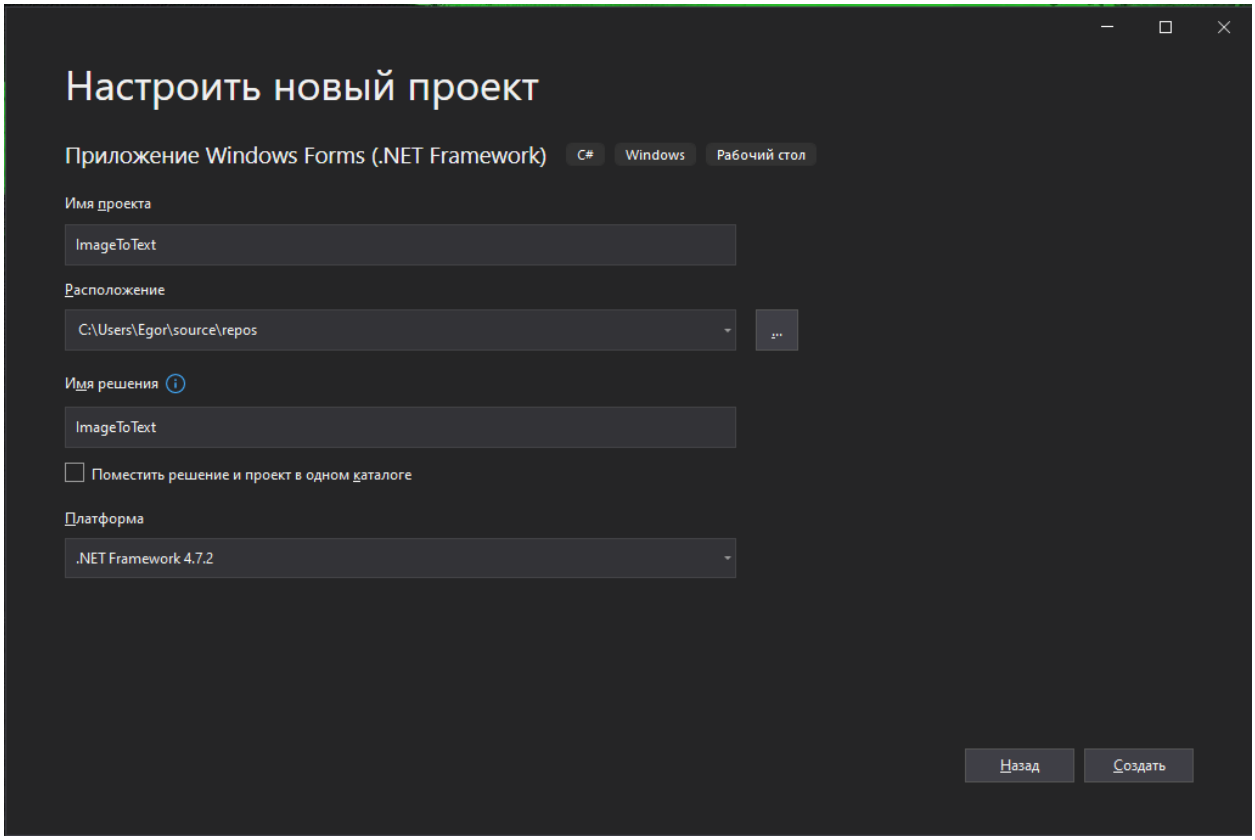

Рисунок 2. Название и пред настройка проекта

Сразу добавим три элемента: «PictureBox» для отображения распознаваемого изображения, «Button» - кнопка для запуска логики программы, «TextBox» для отображения распознанного текста. (рис.3-5).

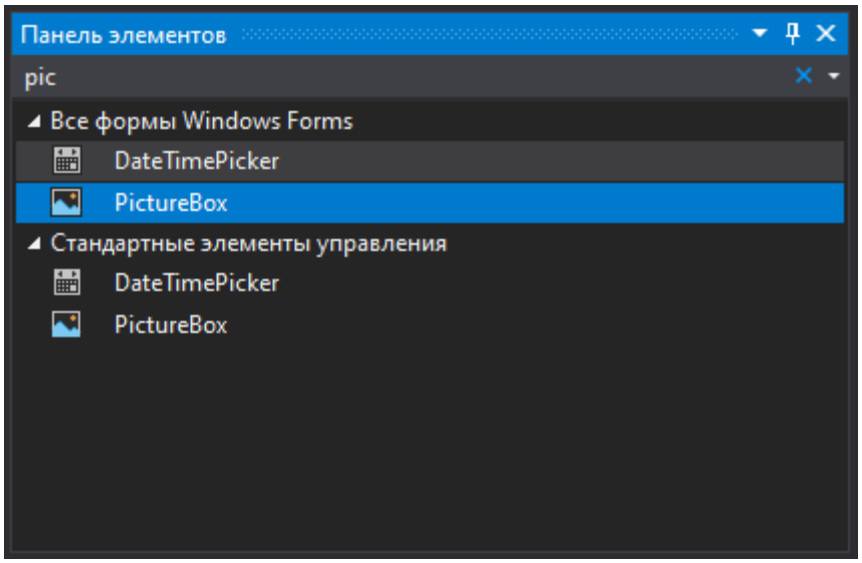

Рисунок 3. Добавление элемента «PictureBox»

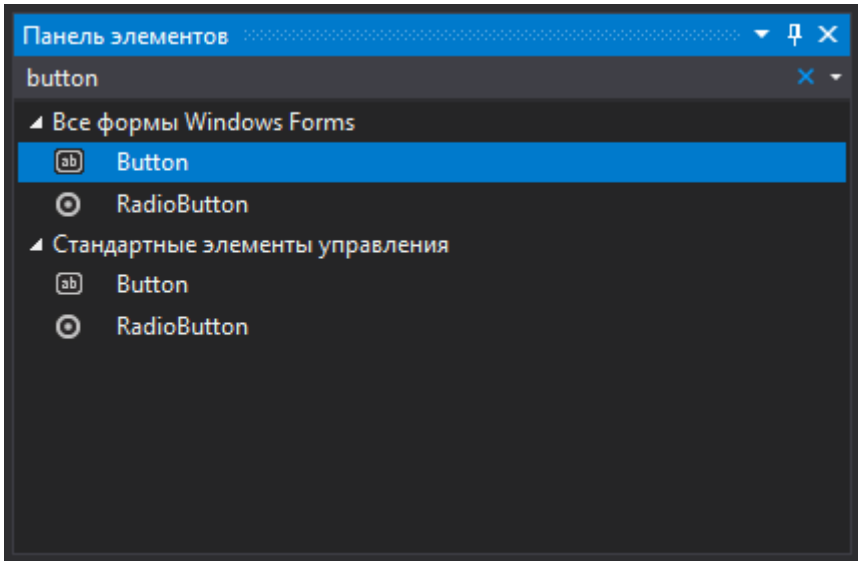

Рисунок 4. Добавление элемента «Button»

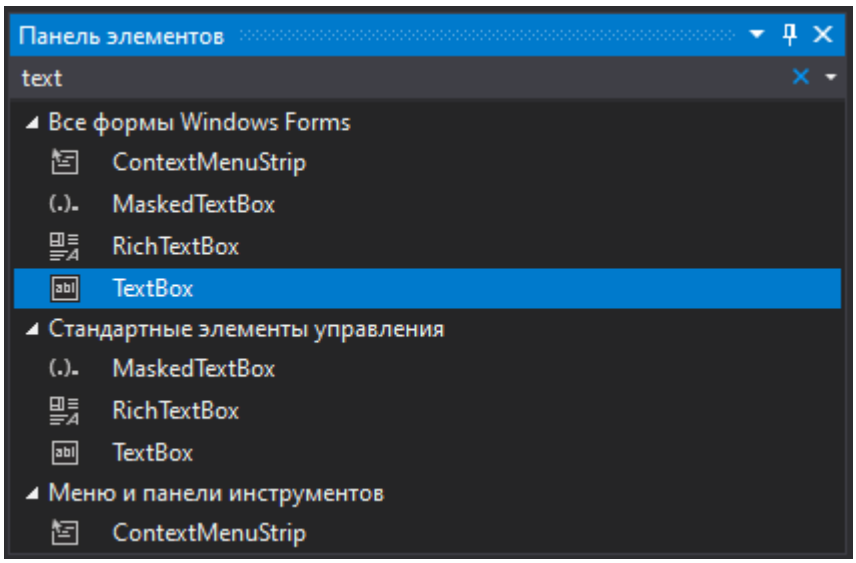

Рисунок 5. Добавление элемента «TextBox»

Далее необходимо для более легкой инициализации в коде: поменять имя элемента «PictureBox» на «PicTarget», имя элемента «Button» на «btnClickHere», а также надпись в кнопке на «Click Here», имя элемента «TextBox» на «txtOutPut», а для добавления свойства многосторонности текста в параметре «Multiline» меняем значение на «True» (рис.6-10).

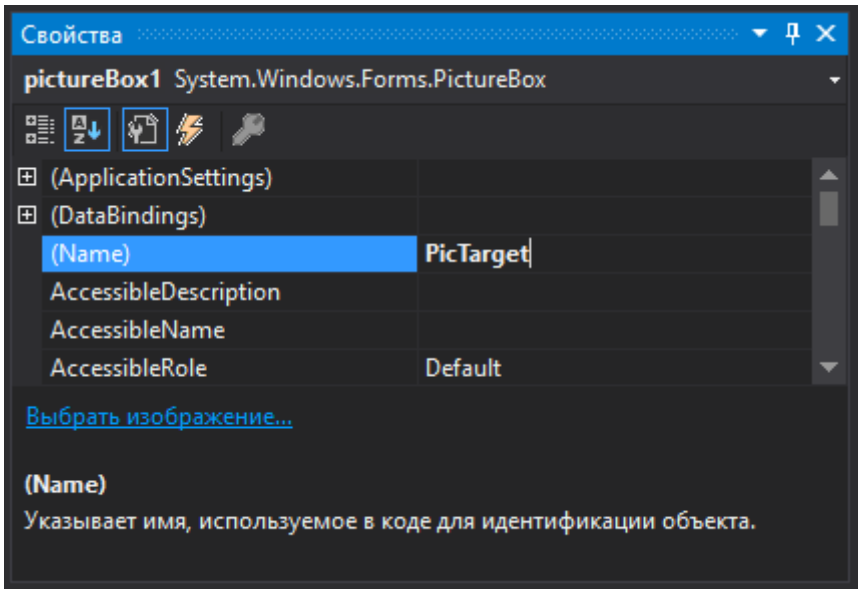

Рисунок 6. Переименование элемента «PictureBox»

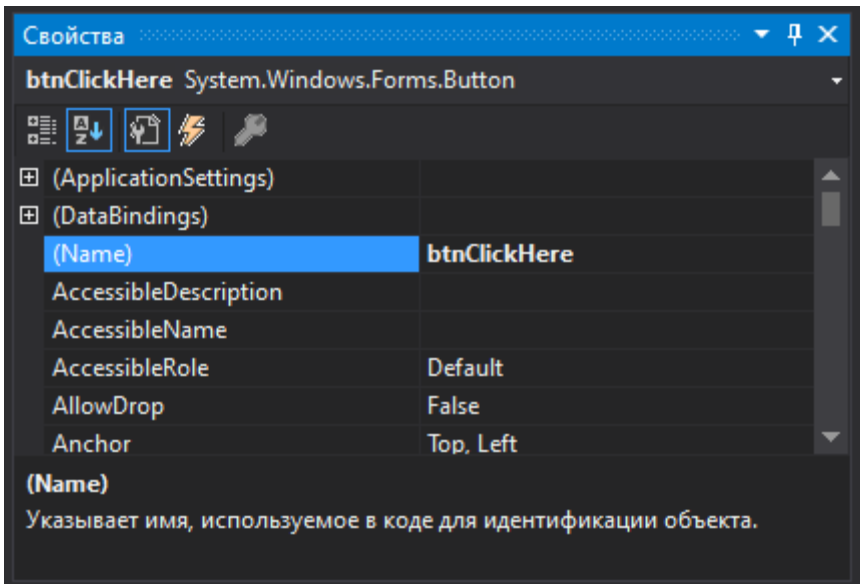

Рисунок 7. Переименование элемента «Button»

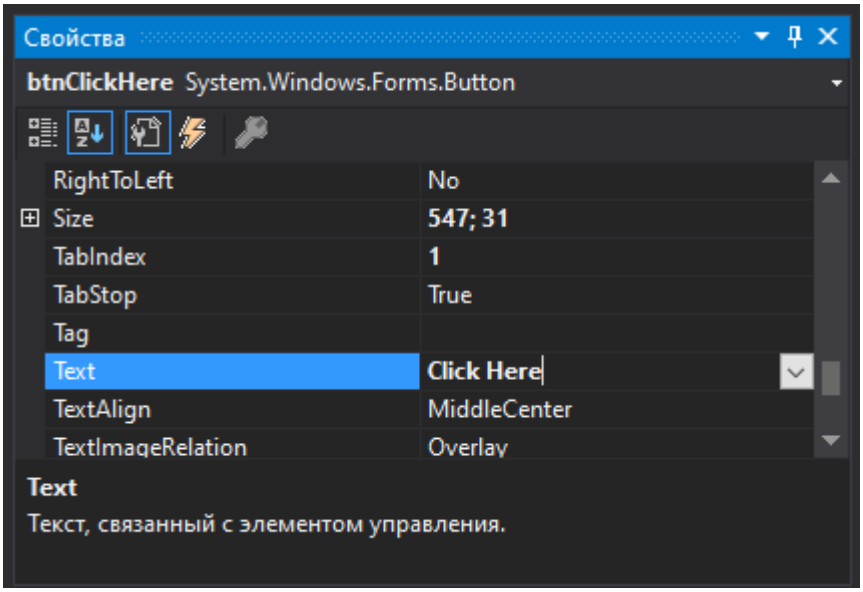

Рисунок 8. Переименование кнопки

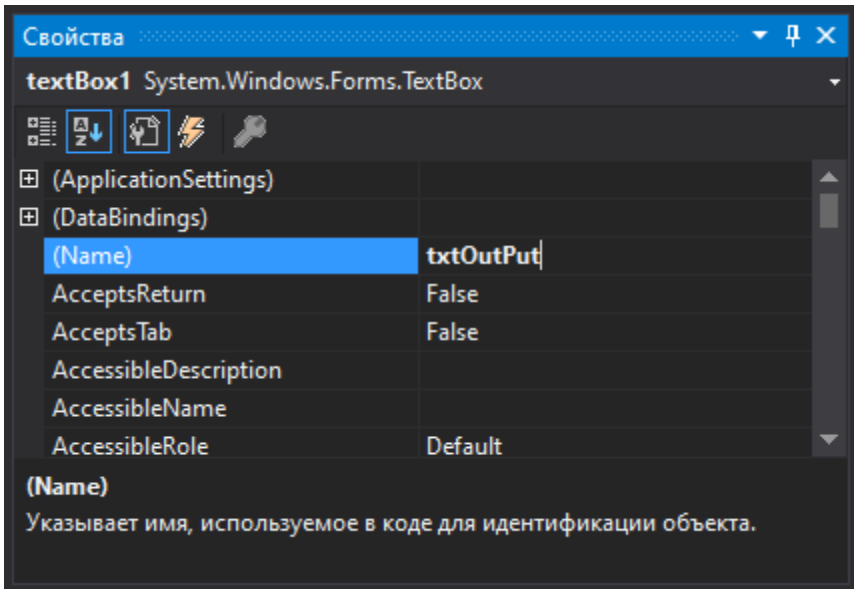

Рисунок 9. Переименование элемента «TextBox»

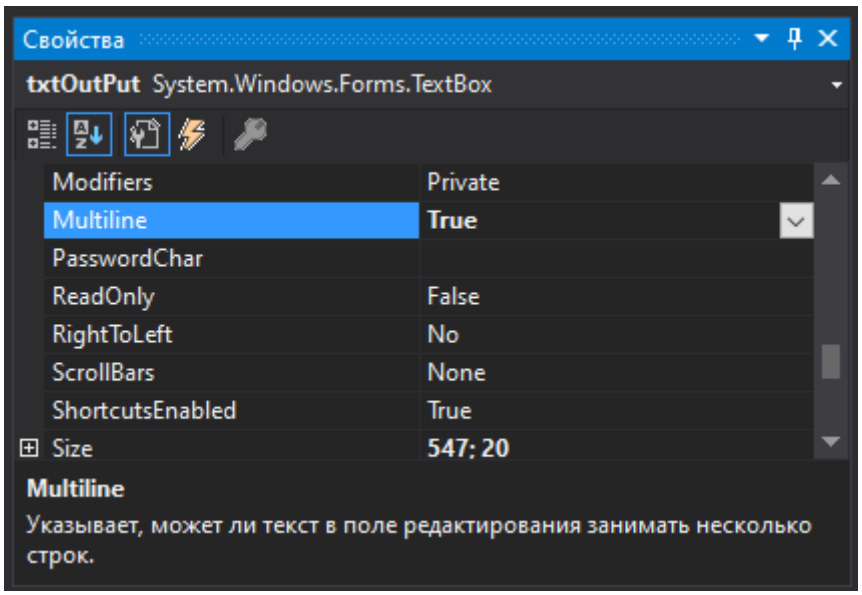

Рисунок 10. Параметр «Multiline»

Теперь необходимо добавить библиотеку «tesseract.Net SDK» — это библиотека классов, основанная на проекте «tesseract-ocr», способная читать самые разные форматы изображений и преобразовывать в текст на более чем 60 языках. (рис.11-12).

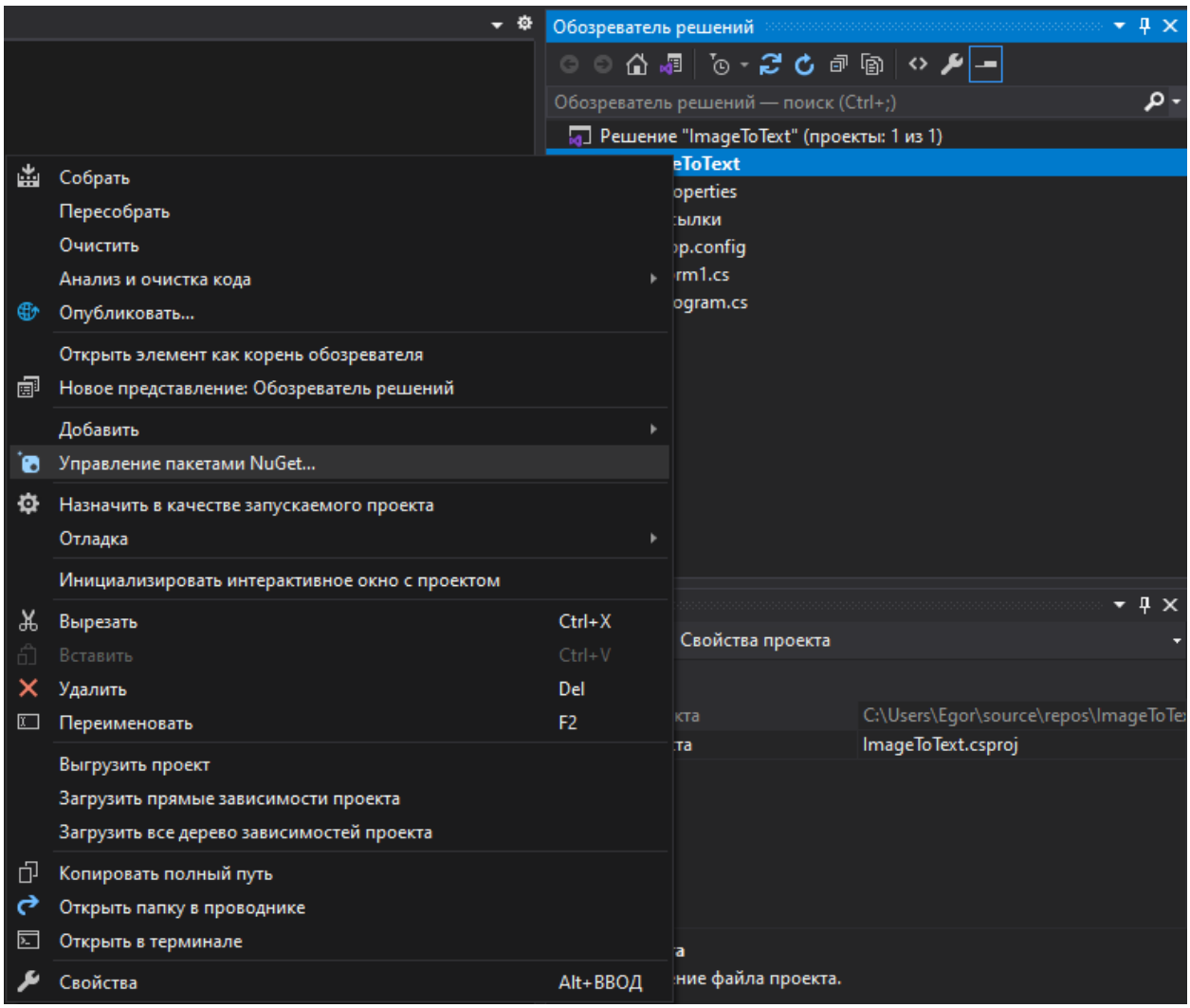

Рисунок 11. Переход в менеджер управления пакетами

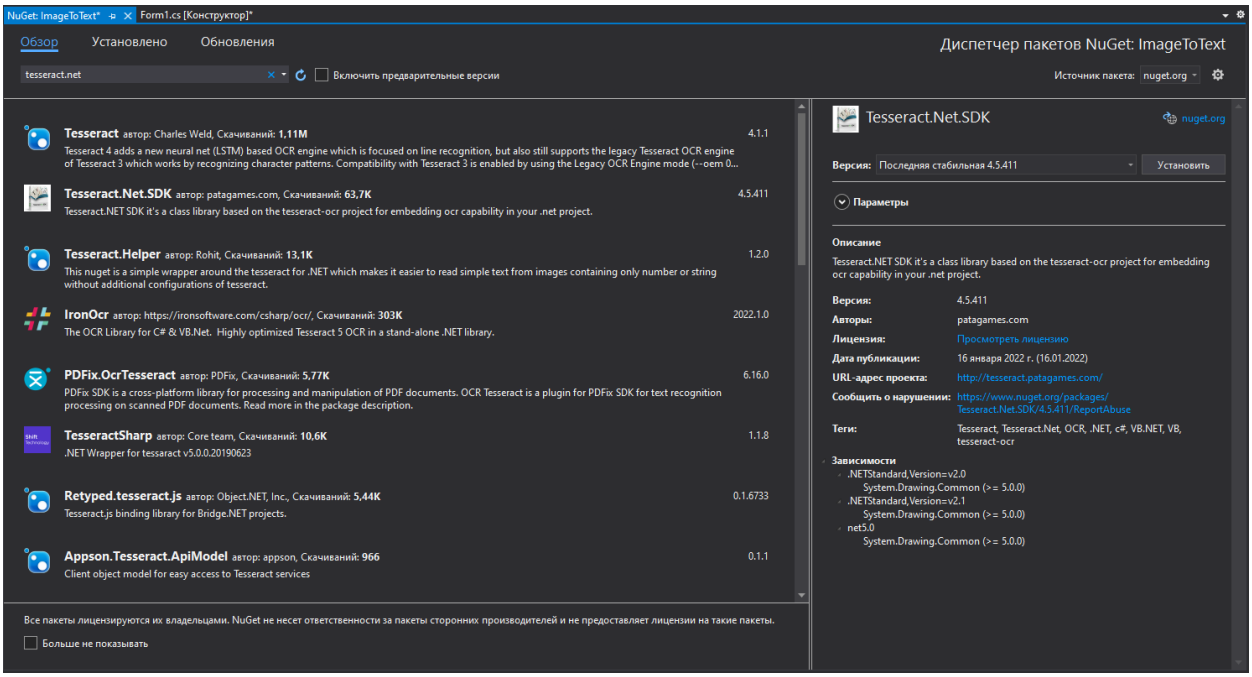

Рисунок 12. Установка библиотеки «tesseract.Net SDK»

Приступаем к написанию кода, начнем с подключения необходимых библиотек: «Patagames.Ocr» для работы библиотеки и «System.Drawing» для работы среды разработки с изображениями (рис.13).

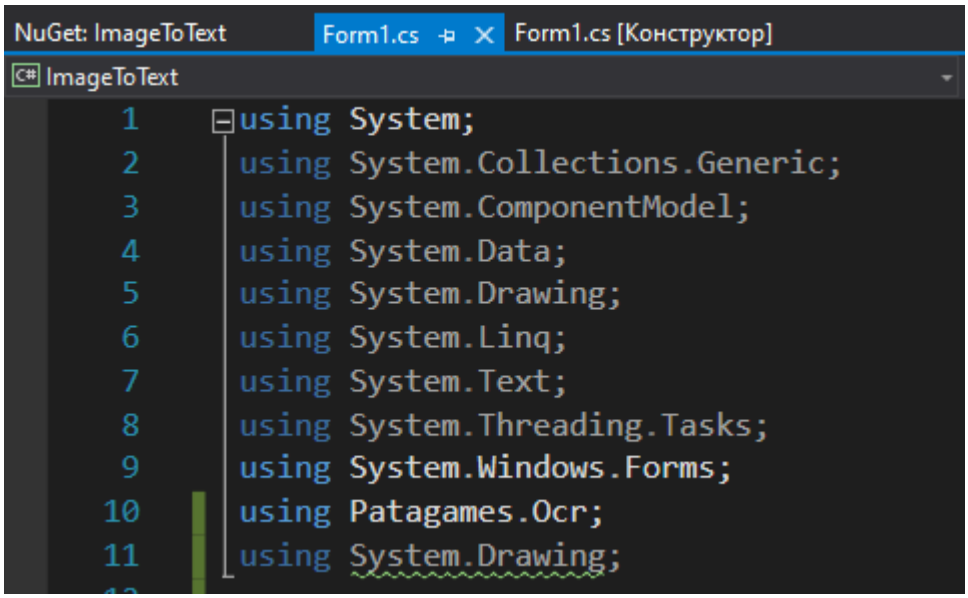

Рисунок 13. Добавление библиотек

Далее прописываем путь к распознаваемому изображению, и добавляем события клика по кнопке, в событии прописываем язык текста на изображении, а также место вывода текста (рис.14).

```
namespace ImageToText
{
    Ссылок: 3
    public partial class Form1 : Form
    {
        ссылка: 1
        public Form1()
        {
            InitializeComponent();
        }

        ссылка: 1
        private void Form1_Load(object sender, EventArgs e)
        {
            PicTarget.ImageLocation = @"C:\Users\Egor\source\repos\ImageToText\ImageToText\img\exm.png";
        }

        ссылка: 1
        private void btnClickHere_Click(object sender, EventArgs e)
        {
            using(var objOcr = OcrApi.Create())
            {
                objOcr.Init(Patagames.Ocr.Enums.Languages.English);

                string plainText = objOcr.GetTextFromImage(PicTarget.ImageLocation);
                txtOutPut.Text = plainText;
            }

        }
    }
}
```

Рисунок 14. Программирование логики программы

Теперь можем проверить программу, для этого запускаем приложение и нажимаем кнопку распознания (рис.15-16).

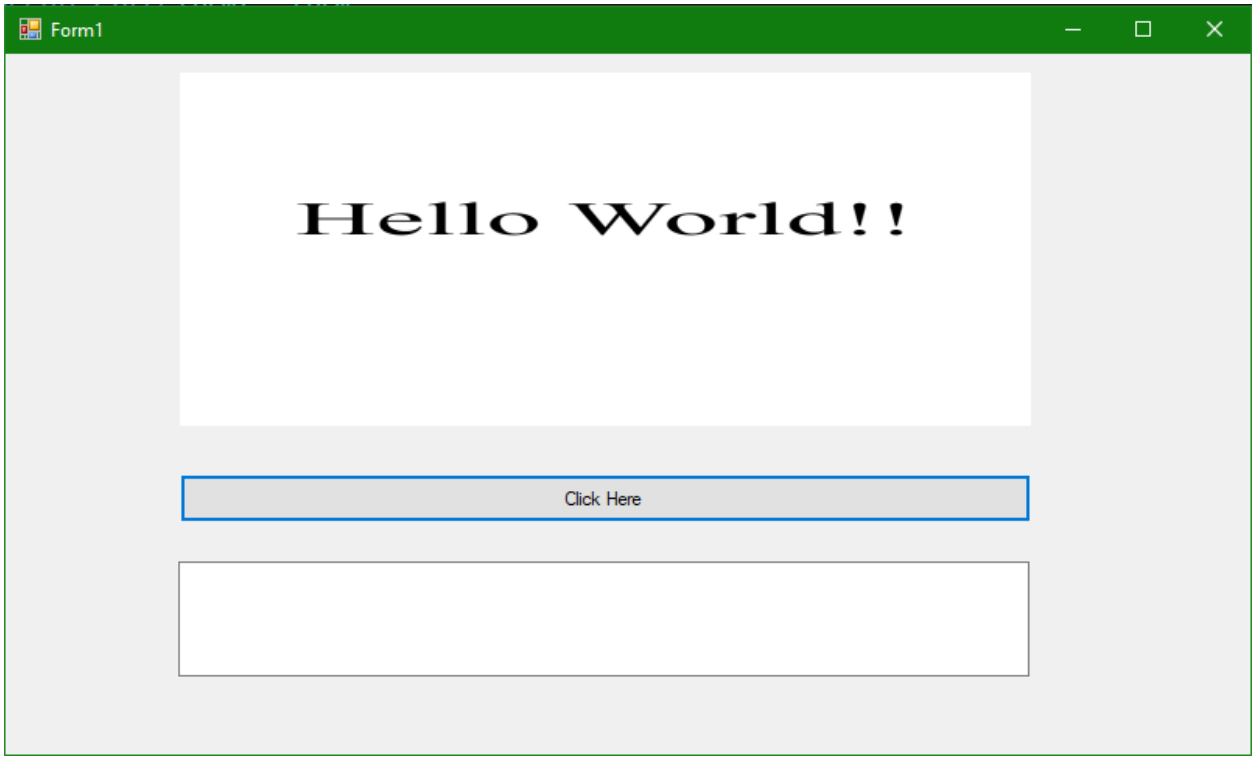

Рисунок 15. Запуск программы

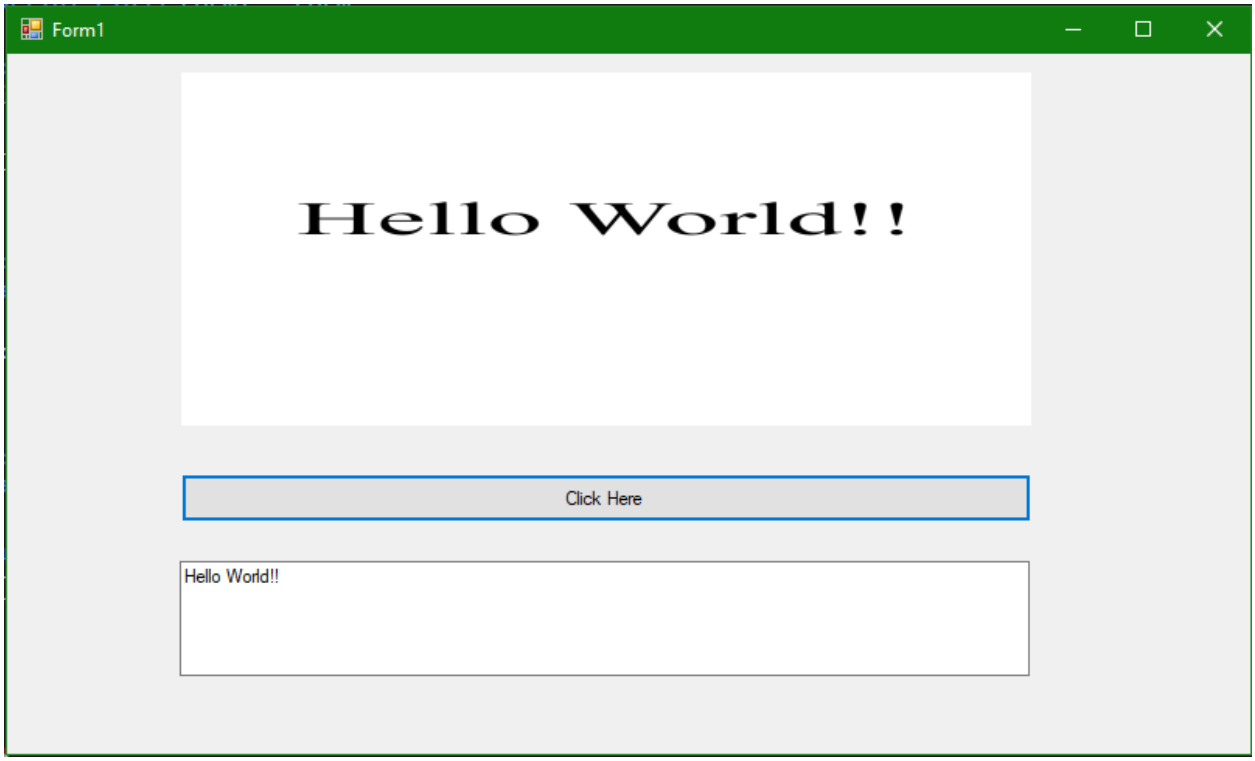

Рисунок 16. Распознанный текст

Были проанализированы существующие аналоги и методы разработки, а также выбрана среда разработки. Для реализации поставленной задачи отлично подошла разработка с помощью Visual Studio и языка

программирования C#. Такой выбор заметно упростил разработку приложения, так как в интернете имеется достаточное количество документации. Во время создания приложения был полученный ценный опыт работы с этим средством разработки. В итоге было разработано приложения для распознавания текста на изображении. Тесты программа прошло хорошо, были исправлены незначительные ошибки. Созданное приложение имеет потенциал к развитию, а именно: добавление новых функций и улучшение интерфейса.

Библиографический список

1. Додобоев Н. Н., Кукарцева О. И., Тынченко Я. А. Современные языки программирования //Современные технологии: актуальные вопросы, достижения и инновации. 2014. №5. С. 81-85.
2. Магомадова З. С. Языки программирования высокого уровня //Разработка и применение наукоёмких технологий в эпоху глобальных трансформаций. 2020. №8. С. 94-96.
3. Просвирнина И. Ю., Егунова А. И., Аббакумов А. А. Среда разработки Microsoft Visual Studio на примере создания игры "морской бой" //Интеграционные процессы в науке в современных условиях. 2017. С. 123-125.
4. Патюченко Ф.В., Слащев И.С., Клименко А.В., Трегубенко Л.А. Windows form или windows presentation foundation // Modern science. 2019. №7-2. С. 318-320.
5. Хасаева Д.З., Демин А.Ю. Разработка графической библиотеки для визуализации объектов робототехники на основе технологии Windows presentation foundation // XXVII Международная инновационно-ориентированная конференция молодых ученых и студентов. 2015. С. 536-539.
6. Li J., Liu Q., Su H. Virtual Reality Method of Portal Slewing Crane Based on WPF // Advances in Mechanical Engineering. 2013. С. 1-5.
7. Torsten S., Toshiyuki S., Derek M. Improving Sample Analysis Throughput and Quality with a .NET™-based, Real-Time QC Decision Support System // Journal of the Association for Laboratory Automation. 2003. №8. С. 107-112.
8. WFP URL: https://docs.microsoft.com/ru-ru/visualstudio/designers/getting-started-with-wpf?view=vs-2022