Sprawozdanie z laboratorium
Urządzenia Peryferyjne
Temat: “Scanner”

Osoba wykonująca sprawozdanie:

Ivan Hancharyk 264511

Termin zajęć:

PN/P 17:00 gr3

Zadania do wykonania

Napisać aplikację, która będzie umożliwiać:

a). uzyskanie obrazu na ekranie monitora po zeskanowaniu obiektu, oraz zapisanie obrazu do zbioru z możliwością wybrania jednego z kilku podanych formatów.

b). realizację takich opcji jak zmiana parametrów skanowania (rozdzielczość, tryb skanowania), obracanie obrazu o 90, 180, 270 stopni w obie strony, itd.).

Wstęp teoretyczny

Skanery są urządzeniami umożliwiającymi cyfryzację dokumentów, zdjęć oraz innych obrazów płaskich, zamieniając je na dane cyfrowe. Jednym z typów skanerów, z którym mieliśmy do czynienia podczas realizacji zadania, był skaner **Canon LiDE 100**, który korzysta z technologii CIS (dla elementu skanującego) i LED (źródło światła). Skaner ten charakteryzuje się kompaktową budową, energooszczędnością oraz łatwością obsługi, jednakże jego rozdzielczość jest ograniczona w porównaniu do technologii CCD.

Skanery można podzielić na trzy podstawowe typy ze względu na technologię przechwytywania obrazu:

- **CCD** (Charge-Coupled Device) – technologia ta wykorzystuje matrycę fotodiod do przechwytywania światła odbitego od skanowanego obiektu. Skanery CCD charakteryzują się wysoką rozdzielczością i dobrą jakością odwzorowania kolorów, co sprawia, że są stosowane w bardziej profesjonalnych zastosowaniach.
- **CIS** (Contact Image Sensor) – skanery CIS wykorzystują linie czujników światła bezpośrednio stykających się z powierzchnią skanowanego obiektu. Dzięki temu są mniejsze i bardziej energooszczędne, jednakże ich jakość skanowania (szczególnie pod względem głębi ostrości) jest niższa w porównaniu z CCD.
- **LiDE** – technologia LiDE bazuje na technologii CIS, z tą różnicą, że do podświetlania obiektu używane są diody LED. Umożliwia to jeszcze większą energooszczędność, skrócenie czasu skanowania oraz zmniejszenie rozmiarów skanera.

Skanery cyfrowe operują na modelach kolorów, takich jak **RGB** (czerwony, zielony, niebieski), gdzie każdy piksel obrazu jest opisany za pomocą intensywności tych trzech barw. Ważnym aspektem w skanowaniu jest także **balans bieli**, który odpowiada za poprawne odwzorowanie kolorów w różnych warunkach oświetleniowych, aby obraz był możliwie jak najbardziej zbliżony do oryginału.

Rozdzielczość skanowania jest jednym z kluczowych parametrów skanera, określanym w jednostkach DPI (dots per inch). Im wyższa rozdzielczość, tym większa ilość szczegółów może zostać uchwycona, co jest szczególnie istotne przy skanowaniu dokumentów tekstowych oraz obrazów o wysokiej szczegółowości. W naszym projekcie korzystaliśmy z możliwości regulacji DPI, aby uzyskać skany o różnej jakości i rozmiarze.

Filtry w skanerach służą do poprawy jakości obrazu – mogą usuwać kurz i rysy, poprawiać ostrość lub dostosowywać nasycenie i kontrast. **Zoom** w skanerach to zazwyczaj zoom cyfrowy, który polega na przycięciu określonego fragmentu obrazu (crop and zoom) lub zwiększeniu rozdzielczości (DPI) w celu uzyskania bardziej szczegółowego obrazu.

Kolejnym ważnym aspektem skanowania jest **de-mozaikowanie**, czyli proces konwersji informacji z matrycy skanera na pełny obraz kolorowy. W przypadku skanerów CIS i LiDE, gdzie czujniki rejestrują informacje o jasności w różnych kolorach, de-mozaikowanie pozwala na uzyskanie pełnego obrazu o wysokiej jakości.

Podczas realizacji zadania wykorzystywane były różne **formaty zapisu informacji graficznej**, takie jak **JPG, PNG, TIFF, BMP** oraz **RLE**. Każdy z tych formatów posiada różne właściwości:

- **JPEG** – format z kompresją stratną, używany głównie do zdjęć.
- **PNG** – format z kompresją bezstratną, doskonały do grafiki internetowej.
- **TIFF** – format bezstratny, często stosowany w fotografii i archiwizacji obrazów.
- **BMP** – prosty format bitmapowy, wykorzystywany głównie w systemach Windows.
- **RLE** – format z kompresją run-length encoding, stosowany do prostych grafik o małej liczbie kolorów.

Wykorzystanie odpowiednich formatów zależy od celu, w jakim jest tworzony skan. Na przykład, format **JPEG** świetnie nadaje się do przechowywania zdjęć, podczas gdy **TIFF** będzie lepszy do archiwizacji dokumentów.

Realizacja ćwiczenia

W projekcie wykorzystaliśmy bibliotekę **WIA** (Windows Image Acquisition) do obsługi skanera i komunikacji z urządzeniem. **WIA** to standard opracowany przez Microsoft, który umożliwia komunikację między urządzeniami do akwizycji obrazu (takimi jak skanery czy kamery) a systemem operacyjnym Windows.

Omówienie ważniejszych funkcji

Pobranie listy dostępnych skanerów

Pobranie listy dostępnych skanerów odbywa się za pomocą przycisku "Search Devices". Funkcja searchDevicesButton_Click iteruje przez dostępne urządzenia WIA i dodaje wykryte skanery do listy rozwijanej (ComboBox).

```
private void searchDevicesButton_Click(object sender, EventArgs e)
{
    try
    {
        scannersComboBox.Items.Clear();
        for (int i = 1; i <= deviceManager.DeviceInfos.Count; i++)
        {
            if (deviceManager.DeviceInfos[i].Type == WiaDeviceType.ScannerDeviceType)
            {
scannersComboBox.Items.Add(deviceManager.DeviceInfos[i].Properties["Name"].get_Value());
            }
        }
        if (scannersComboBox.Items.Count > 0)
        {
            scannersComboBox.SelectedIndex = 0;
        }
        else
        {MessageBox.Show("No scanners found.");}
    }
    catch (Exception ex)
    {MessageBox.Show("An error occurred while searching for scanners: " + ex.Message);
```

```
    }
}
```

- Funkcja najpierw czyści istniejącą listę skanerów (`scannersComboBox.Items.Clear()`).
- Następnie, iteruje przez dostępne urządzenia i dodaje te, które są skanerami do listy.
- Jeśli żaden skaner nie został wykryty, wyświetlany jest komunikat.

Połączenie ze skanerem

Po wyszukaniu urządzeń, użytkownik może wybrać skaner z listy i kliknąć przycisk "Connect", aby nawiązać połączenie z wybranym urządzeniem.

```
private void connectButton_Click(object sender, EventArgs e)
{
    if (scannersComboBox.SelectedIndex >= 0)
    {
        try
        {
            selectedDeviceInfo =
deviceManager.DeviceInfos[scannersComboBox.SelectedIndex + 1];
            connectedScanner = selectedDeviceInfo.Connect();
            MessageBox.Show("Connected to scanner: " +
scannersComboBox.SelectedItem.ToString());

            // Dezaktywacja przycisku Connect
            connectButton.Enabled = false;

            // Aktywacja pozostałych przycisków po udanym połączeniu
            scanButton.Enabled = true;
            saveButton.Enabled = true;
            rotateLeftButton.Enabled = true;
            rotateRightButton.Enabled = true;

            // Zmiana statusu skanera
            scannerStatusLabel.Text = "Status: Connected to " +
scannersComboBox.SelectedItem.ToString();
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error connecting to scanner: " + ex.Message);
        }
    }
    else
    {
        MessageBox.Show("Please select a scanner from the list.");
    }
}
```

- Sprawdzane jest, czy użytkownik wybrał skaner z listy.
- Po nawiązaniu połączenia, aktywowane są przyciski skanowania i zapisu obrazu.
- Wyświetlany jest komunikat potwierdzający połączenie ze skanerem.

Ustawienie DPI

DPI (dots per inch) jest jednostką rozdzielczości, która decyduje o szczegółowości skanu. Użytkownik może zmieniać wartość DPI za pomocą suwaka (TrackBar).

```
  private void dpiTrackBar_Scroll(object sender, EventArgs e)
  {
      dpi = dpiTrackBar.Value;
      dpiLabel.Text = "DPI: " + dpi;
  }
```

Wartość DPI jest aktualizowana, a etykieta (`dpiLabel`) informuje użytkownika o nowej wartości DPI.

Ustawienie Jasności i Kontrastu

Jasność i kontrast mogą być ustawiane przez użytkownika za pomocą odpowiednich suwaków. Funkcje te pozwalają użytkownikowi dostosować parametry skanowania.

```
private void brightnessTrackBar_Scroll(object sender, EventArgs e)
 {
     brightness = brightnessTrackBar.Value;
     brightnessLabel.Text = "Brightness: " + brightness;
 }

 private void contrastTrackBar_Scroll(object sender, EventArgs e)
 {
     contrast = contrastTrackBar.Value;
     contrastLabel.Text = "Contrast: " + contrast;
 }
```

- brightnessTrackBar_Scroll i contrastTrackBar_Scroll aktualizują wartości jasności i kontrastu, odpowiednio, kiedy użytkownik porusza suwakiem.
- Etykiety brightnessLabel oraz contrastLabel aktualizują swoje wartości, aby użytkownik mógł zobaczyć obecne ustawienia.

Wywołanie Domyślnej Aplikacji do Skanowania

Wywołanie domyślnej aplikacji do skanowania odbywa się za pomocą przycisku "Scan". Proces skanowania polega na ustawieniu właściwości skanera oraz przeprowadzeniu skanowania.

```
private void scanButton_Click(object sender, EventArgs e)
{
    if (connectedScanner != null)
    {
        try
        {
            // Blokowanie przycisków podczas skanowania
            connectButton.Enabled = false;
            scanButton.Enabled = false;
            saveButton.Enabled = false;
            rotateLeftButton.Enabled = false;
            rotateRightButton.Enabled = false;

            scannerStatusLabel.Text = "Scanning in progress...";

            // Scan the image with the specified parameters
            var scannerItem = connectedScanner.Items[1];
            SetScannerProperties(scannerItem);

            // Transfer scanned image to appropriate format
            switch (formatComboBox.SelectedItem.ToString().ToLower())
            {
                case "jpeg":
                    scannedImage = (ImageFile)scannerItem.Transfer("{B96B3CAE-0728-
11D3-9D7B-0000F81EF32E}");
                    break;
```

```
                case "png":
                    scannedImage = (ImageFile)scannerItem.Transfer("{B96B3CAF-0728-
11D3-9D7B-0000F81EF32E}");
                    break;
                case "bmp":
                case "tiff":
                case "rle":
                    scannedImage = (ImageFile)scannerItem.Transfer();
                    break;
                default:
                    MessageBox.Show("Unsupported format selected.");
                    return;
            }

            // Save scanned image and display in preview
            SaveFileDialog saveDialog = new SaveFileDialog
            {
                Filter = "JPEG|*.jpeg|PNG|*.png|BMP|*.bmp|TIFF|*.tiff|RLE|*.rle",
                Title = "Save Scanned Image",
                FileName = "scanned_image"
            };
            if (saveDialog.ShowDialog() == DialogResult.OK)
            {
                scannedImage.SaveFile(saveDialog.FileName);
                previewPictureBox.ImageLocation = saveDialog.FileName;
            }

            scannerStatusLabel.Text = "Scan complete.";

        }
        catch (Exception ex)
        {
            MessageBox.Show("Error occurred while scanning: " + ex.Message);
        }
        finally
        {
            // Odblokowanie przycisków po skanowaniu
            connectButton.Enabled = true;
            scanButton.Enabled = true;
            saveButton.Enabled = true;
            rotateLeftButton.Enabled = true;
            rotateRightButton.Enabled = true;
        }
    }
    else
    {
        MessageBox.Show("Please connect to a scanner first.");
    }
}
```

- Funkcja `scanButton_Click` ustawia właściwości skanera (`SetScannerProperties(scannerItem)`), a następnie wywołuje proces skanowania.
- Obraz jest zapisywany do wybranego formatu (np. JPEG, PNG).
- Podczas skanowania przyciski są wyłączone, a po zakończeniu zostają ponownie aktywowane.

Zachowanie Obrazu w Odpowiednich Formatach

```
        private void saveButton_Click(object sender, EventArgs e)
        {
            if (scannedImage != null)
            {
                // Inicjalizacja dialogu zapisu
                SaveFileDialog saveFileDialog = new SaveFileDialog
                {
                    Filter =
"BMP|*.BMP|JPG|*.JPG|GIF|*.GIF|PNG|*.PNG|TIFF|*.TIFF|RLE|*.RLE",
                    FileName = "scanned_image"
                };

                // Pokazanie dialogu zapisu
                if (saveFileDialog.ShowDialog() == DialogResult.OK)
                {
                    try
                    {
                        // Próba zapisania pliku
                        scannedImage.SaveFile(saveFileDialog.FileName);
                        MessageBox.Show("Image saved successfully to: " +
saveFileDialog.FileName);
                    }
                    catch (Exception ex)
                    {
                        // Obsługa błędów podczas zapisu pliku
                        MessageBox.Show("An error occurred while saving the image: "
+ ex.Message);
                    }
                }
            }
            else
            {
                // Brak zeskanowanego obrazu do zapisu
                MessageBox.Show("No image to save. Please scan an image first.");
            }
        }
```

- Użytkownik wybiera format pliku do zapisu, korzystając z `SaveFileDialog`.
- Aplikacja umożliwia zapis w formatach: BMP, JPG, GIF, PNG, TIFF, oraz RLE.
- Każdy z tych formatów ma inne cechy, np. JPEG jest formatem stratnym i kompresuje obraz, natomiast TIFF oferuje kompresję bezstratną i jest bardziej odpowiedni dla skanów o wysokiej jakości.

Wybór Filtru (Kolorowy, Czarno-Biały, Szary)

Użytkownik może wybrać jeden z trybów skanowania: kolorowy, odcienie szarości, lub czarno-biały. Tryb skanowania decyduje o sposobie rejestrowania kolorów w skanowanym obrazie.

```
private void SetScannerProperties(IItem scannerItem)
{
    const string WIA_HORIZONTAL_SCAN_RESOLUTION_DPI = "6147";
    const string WIA_VERTICAL_SCAN_RESOLUTION_DPI = "6148";
    const string WIA_HORIZONTAL_SCAN_START_PIXEL = "6149";
    const string WIA_VERTICAL_SCAN_START_PIXEL = "6150";
    const string WIA_HORIZONTAL_SCAN_SIZE_PIXELS = "6151";
    const string WIA_VERTICAL_SCAN_SIZE_PIXELS = "6152";
    const string WIA_SCAN_BRIGHTNESS_PERCENTS = "6154";
    const string WIA_SCAN_CONTRAST_PERCENTS = "6155";
    const string WIA_SCAN_COLOR_MODE = "6146";

    try
    {
        // Walidacja DPI
        if (dpi < 100 || dpi > 1200)
        {
            MessageBox.Show("DPI value is out of range (100-1200). Setting default
DPI to 150.");
            dpi = 150;
        }

        // Ustawienie parametrów skanowania
        SetProperty(scannerItem.Properties, WIA_HORIZONTAL_SCAN_RESOLUTION_DPI,
dpi);
        SetProperty(scannerItem.Properties, WIA_VERTICAL_SCAN_RESOLUTION_DPI, dpi);
        SetProperty(scannerItem.Properties, WIA_HORIZONTAL_SCAN_START_PIXEL, X);
        SetProperty(scannerItem.Properties, WIA_VERTICAL_SCAN_START_PIXEL, Y);

        // Automatyczne przeliczenie szerokości i wysokości na piksele na podstawie
DPI
        width = (int)((210.0 / 25.4) * dpi);   // A4 width in pixels
        height = (int)((297.0 / 25.4) * dpi);  // A4 height in pixels
        SetProperty(scannerItem.Properties, WIA_HORIZONTAL_SCAN_SIZE_PIXELS, width);
        SetProperty(scannerItem.Properties, WIA_VERTICAL_SCAN_SIZE_PIXELS, height);

        // Ustawienie jasności i kontrastu
        SetProperty(scannerItem.Properties, WIA_SCAN_BRIGHTNESS_PERCENTS,
brightness);
        SetProperty(scannerItem.Properties, WIA_SCAN_CONTRAST_PERCENTS, contrast);

        // Ustawienie trybu kolorów
        int colorMode = 1; // Default to Color
        if (colorModeComboBox.SelectedIndex == 1)
        {
            colorMode = 2; // Grayscale
        }
        else if (colorModeComboBox.SelectedIndex == 2)
        {
            colorMode = 4; // Black & White
        }
        SetProperty(scannerItem.Properties, WIA_SCAN_COLOR_MODE, colorMode);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error setting scanner properties: " + ex.Message);
    }
}
```

- `SetScannerProperties` ustawia właściwości skanera na podstawie wyboru użytkownika.
- Tryb kolorowy (`Color`), szary (`Grayscale`), oraz czarno-biały (`Black & White`) może być wybrany w `colorModeComboBox`.
- Tryb kolorowy zapisuje pełne spektrum barw, odcienie szarości zapisują obraz w skali szarości, a tryb czarno-biały zapisuje jedynie czarne i białe piksele, co zmniejsza rozmiar pliku, ale może obniżyć szczegółowość obrazu.

Rotacja Obrazu

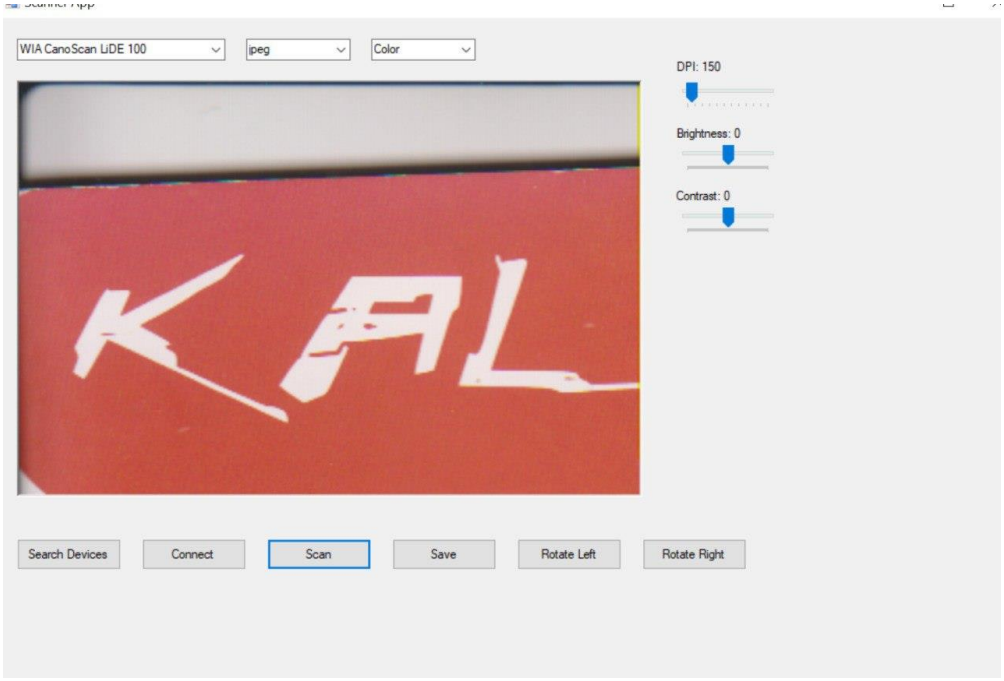
Aplikacja oferuje możliwość obrócenia obrazu po zeskanowaniu. Użytkownik może obracać obraz w lewo lub w prawo o 90 stopni. Może to być szczególnie przydatne, gdy dokument został umieszczony na skanerze w niewłaściwej orientacji.

```
private void rotateLeftButton_Click(object sender, EventArgs e)
{
    RotateImage(270);
}

private void rotateRightButton_Click(object sender, EventArgs e)
{
    RotateImage(90);
}

private void RotateImage(int angle)
{
    if (previewPictureBox.Image != null)
    {
        Bitmap bitmap = new Bitmap(previewPictureBox.Image);
        switch (angle)
        {
            case 90:
                bitmap.RotateFlip(RotateFlipType.Rotate90FlipNone);
                break;
            case 270:
                bitmap.RotateFlip(RotateFlipType.Rotate270FlipNone);
                break;
        }
        previewPictureBox.Image = bitmap;
    }
}
```

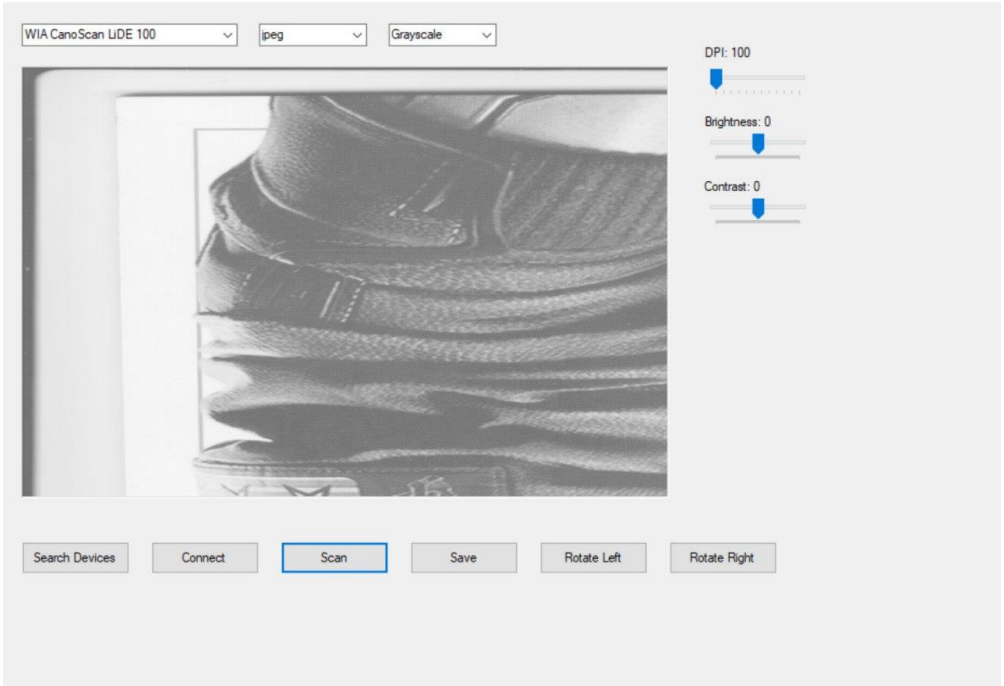
- Użytkownik może kliknąć przyciski `rotateLeftButton` lub `rotateRightButton`, aby obrócić obraz o 90 stopni w lewo lub w prawo.
- Funkcja `RotateImage(int angle)` wykorzystuje `Bitmap.RotateFlip()`, aby obrócić obraz o zadany kąt.
- Przycisk `rotateLeftButton` obraca obraz o 270 stopni, co odpowiada obrotowi o 90 stopni w lewo, natomiast `rotateRightButton` obraca obraz o 90 stopni w prawo.

Screen’y

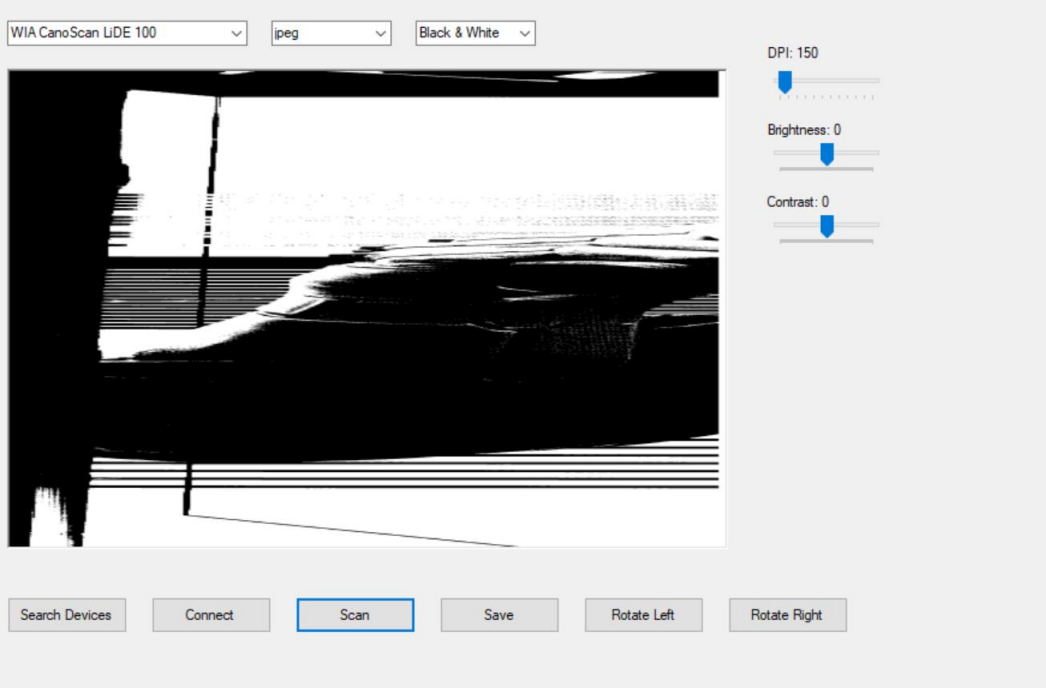
Interfejs aplikacji oraz przykładowe skany zrobione przez naszą aplikacje:



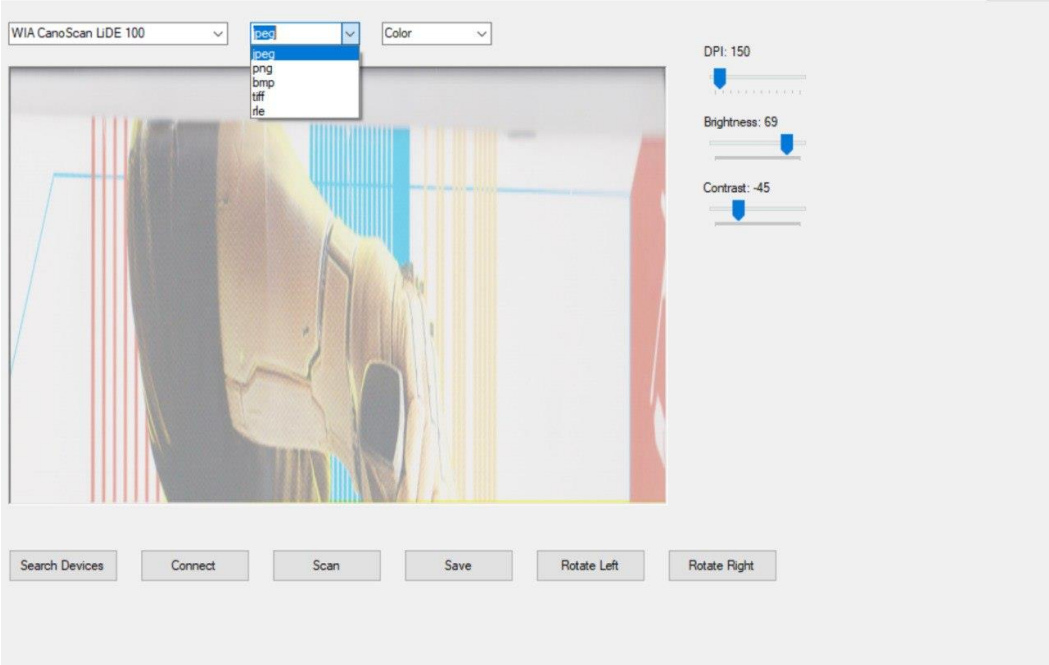
Rysunek 1: skan – obraz kolorowy, domyślne ustawienia jasności i kontrastu



Rysunek 2: skan – obraz szary, domyślne ustawienia jasności i kontrastu



Rysunek 3: skan – obraz czarno-biały, domyślne ustawienia jasności i kontrastu



Rysunek 4: skan – obraz kolorowy, przy wybranych ustawieniach jasności i kontrastu

Wnioski

Użytkownik ma możliwość regulacji parametrów skanowania, takich jak rozdzielczość (DPI), jasność i kontrast, co pozwala na uzyskanie optymalnej jakości obrazu. Program wspiera różne formaty zapisu, takie jak JPEG, PNG, BMP, oraz TIFF, co umożliwia elastyczne przechowywanie zeskanowanych dokumentów. Dodane funkcje, takie jak rotacja obrazu oraz wybór trybu kolorów, zwiększają funkcjonalność aplikacji i ułatwiają użytkownikowi edycję

wyników skanowania. Niestety nie udało się spełnić wszystkich wymagań i aplikacje nie jest w stanie zrobić skanu pełnego obrazu przez problemy w frontendzie naszej aplikacji.