



**WYŻSZA SZKOŁA  
INFORMATYKI I ZARZĄDZANIA**  
z siedzibą w Rzeszowie

## **KOLEGIUM INFORMATYKI STOSOWANEJ**

Kierunek: Informatyka

Specjalność: Programowanie

Paweł Skrok w67266

Karolina Rydzik w67264

### ***Aplikacja Desktopowa do Analizy Paragonów***

Prowadzący: dr inż. Leszek Gajecki

*Projekt w języku Python z wykorzystaniem OCR  
i Algorytmów Statystycznych*

**Rzeszów 2024**

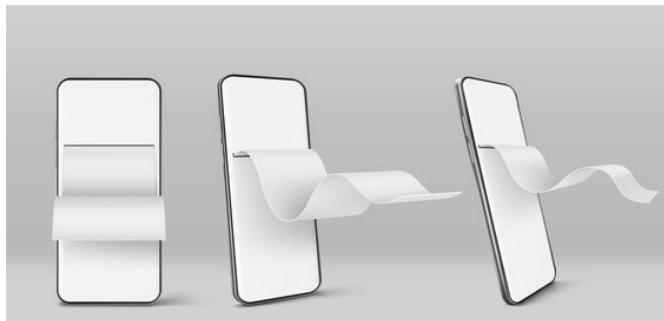
# **Spis treści**

<b>1 Struktura Projektu .....</b>	<b>3</b>
1.1 Menu .....	3
1.2 OCR .....	3
1.3 ImageEditor.....	3
1.4 Mail Sender.....	4
1.5 Draw .....	4
<b>2 Wymagania Systemowe.....</b>	<b>5</b>
2.1 Wymagania funkcjonalne .....	5
2.2 Wymagania niefunkcjonalne .....	5
<b>3 Opis Techniczny Projektu .....</b>	<b>6</b>
3.1 Cel Projektu .....	6
3.2 Język programowania .....	7
3.3 Środowisko i Konfiguracja .....	7
3.4 Bezpieczeństwo i Wydajność .....	8
3.5 Wykorzystane Narzędzia .....	8
<b>4 Integracja i Funkcjonalność Systemu .....</b>	<b>11</b>
4.1 OCR_teseract .....	11
4.2 OCR_window .....	17
4.3 Preprocesing_images.....	22
4.4 Mail_sender.....	27
4.5 Draw .....	31
4.6 Menu .....	35
4.7 ImageEditor.....	37

# Rozdział 1

## Struktura Projektu

Projekt aplikacji desktopowej do analizy paragonów został zaprojektowany z uwzględnieniem struktury umożliwiającej klasowy podział na różne funkcjonalności. Każdy z głównych modułów pełni określoną rolę w procesie analizy paragonów, zapewniając integralność i efektywność działania całego systemu.



### 1.1 Menu

Moduł Menu stanowi centralny punkt nawigacyjny aplikacji, umożliwiający użytkownikowi łatwy dostęp do wszystkich funkcji i opcji systemu. Stworzony został przy użyciu biblioteki Tkinter, czyli podstawowego narzędzia do tworzenia interfejsu użytkownika. Tkinter zapewnia intuicyjny interfejs graficzny, który umożliwia użytkownikom łatwe nawigowanie po aplikacji oraz dostęp do wszystkich funkcji bez konieczności zaawansowanej znajomości skomplikowanych poleceń programistycznych.

### 1.2 OCR

Klasy OCR\_tesseract i OCR\_Window są najważniejsze w przetwarzaniu obrazów, pierwsza klasa służy do odczytu zdjęć paragonów przed przystąpieniem do głębszej analizy danych. Wykorzystuje bibliotekę OpenCV do wstępnego przetwarzania obrazów paragonów, oferując zaawansowane funkcje przetwarzania obrazu, takie jak filtrowanie, binaryzacja oraz usuwanie szumów, co przekłada się na poprawę jakości obrazów przed ich analizą za pomocą narzędzia do rozpoznawania tekstu, jakim jest Pytesseract.

### 1.3 ImageEditor

Klasa ImageEditor zawiera prostą strukturę opierającą się na wczytaniu obrazu za pomocą pola tekstowego a następnie odczytaniu najważniejszych informacji o obrazie wymaganych w dalszej części Modułu takich jak wymiary oraz rozszerzenie pliku. Ma to na celu zapewnienie wygody użytkownikowi w obcinaniu przekazanego obrazu do pewnych wymiarów oraz bezpieczny zapis wyciętego fragmentu obrazu.

## **1.4 Mail Sender**

Moduł Mail Sender umożliwia aplikacji wysyłanie danych oraz raportów drogą mailową do określonego odbiorcy. Integruje moduł Smtpplib i ssl, który obsługuje wysyłanie maili z aplikacji po podaniu hasła z aplikacji wymagane do połączenia z serwerem smtp. Jest to kluczowy element dla tej funkcjonalności, który zapewnia szybki i bezpieczny sposób komunikacji oraz udostępniania informacji.

## **1.5 Draw**

Moduł Draw umożliwia dynamiczne generowanie wykresów i grafik na podstawie przetworzonych danych z funkcji matematycznych. Wykorzystuje zaawansowane techniki wizualizacji danych oferowane przez bibliotekę Matplotlib. Pozwala to na dynamiczne generowanie różnych rodzajów wykresów. Umożliwia to użytkownikom łatwe analizowanie wyników i prezentowanie wniosków w atrakcyjnej formie graficznej jak i tekstowej w postaci min, max funkcji oraz miejsc zerowych.

## Rozdział 2

# Wymagania Systemowe

Przed przystąpieniem do projektowania systemu, konieczne jest określenie wymagań funkcjonalnych i niefunkcjonalnych. Wymagania funkcjonalne obejmują operacje, które system musi wykonywać, natomiast wymagania niefunkcjonalne dotyczą jakości tych operacji, takich jak wydajność, skalowalność i bezpieczeństwo wymagania sprzętowe :System operacyjny w architekturze x64, pamięć ram:2 gb (zalecana 4 gb) wymagania programowe Python version: 3.10.11 TensorFlow version: 2.16.1 Tkinter version: 8.6 Tesseract version: 5.3.1.20230401 Pillow version: 9.2.0 OpenCV version (cv2): 4.7.0 Matplotlib version: 3.7.0 rembg version: 2.0.56 .

### 2.1 Wymagania funkcjonalne

- Bezpieczne wczytywanie obrazów paragonów.
- Przetwarzanie obrazów paragonów za pomocą OCR.
- Edycja obrazów (prycinanie, usuwanie tła etc.)
- Zapisywanie przetworzonych danych i obrazów.
- Wysyłanie wyników analizy drogą mailową.

### 2.2 Wymagania niefunkcjonalne

- Wysoka dokładność rozpoznawania tekstu.
- Intuicyjny interfejs użytkownika.
- Bezpieczeństwo przechowywania i przesyłania danych.
- Skalowalność systemu, aby obsłużyć dużą liczbę paragonów.

# Rozdział 3

## Opis Techniczny Projektu

### 3.1 Cel Projektu

- Pokazowe użycie zasobów i funkcjonalności tkintera
- Użycie modelu OCR dla obrazów z pismem ręcznym
- Użycie modelu z pytesseract do zrozpoznawania tekstu za zdjęcia paragonu
- Przetwarzania obrazów przy użyciu widżetów tkinter
- Zaprezentowanie dodatku wizualnego dla tkinter o nazwie ttkbootstrap
- Funkcje eval i exec w połączenie z matplotlib oraz sympy rysowanie wykresu i wyszukiwaniem miejsc zerowych
- Wysyłanie skopiowanego wcześniej wyniku ocr z pytesseract jako wiadomość email ze schowka za pomocą pyperclip oraz smtplib
- Obsługa wyjątków użycia bibliotek, wczytywania obrazów, sprawdzenie poprawności ścierzki, zgodności i validacji
- Odczytywanie paragonów i podział na szczegółowe informacje z prostych paragonów



## **3.2 Język programowania**

Projekt został napisany w języku Python, wykorzystując moduły takie jak Tkinter do budowy interfejsu użytkownika, OpenCV do przetwarzania obrazów, oraz Pytesseract do rozpoznawania tekstu. Dodatkowo, wykorzystano moduł Matplotlib do generowania wykresów oraz moduł smtplib do obsługi wysyłania maili.

## **3.3 Środowisko i Konfiguracja**

Do projektu użyto środowiska Pycharm 2023.2.1, Anaconda Navigator 2.4.0, JupyterLab 3.5.3 do przygotowywania danych i trenowania modelu OCR , z Pythonem 3.10.11 jako głównym językiem programowania. Wersje bibliotek takich jak TensorFlow, Tkinter, Tesseract, Pillow, OpenCV, Matplotlib oraz innych są dokładnie określone w celu zapewnienia kompatybilności i stabilności działania aplikacji.



## 3.4 Bezpieczeństwo i Wydajność

Bezpieczeństwo danych użytkowników, takich jak adresy e-mail, zostało zapewnione poprzez bezpieczne przetwarzanie i przechowywanie danych w aplikacji. Dbano również o wydajność aplikacji poprzez optymalizację procesów przetwarzania oraz obrazów oraz wysyłania maili oraz zaoszczędzenie czasu użytkownika w wykonywaniu prostych czynności takich jak ocr, wysyłanie maila, obcinanie zdjęć przetwarzanie zdjęć i przetwarzanie funkcji matematycznych aby zapewnić szybkie i niezawodne działanie.



## 3.5 Wykorzystane Narzędzia

### - Tkinter

Tkinter to standardowa biblioteka do tworzenia graficznych interfejsów użytkownika (GUI) w języku Python. Jest prosty w użyciu i umożliwia szybkie tworzenie interaktywnych aplikacji.

- \* Tworzenie okien i dialogów.
- \* Implementacja widżetów takich jak przyciski, etykiety, pola tekstowe i listy.
- \* Obsługa zdarzeń i interakcji użytkownika, wyświetlanie messageboxów i ustawianie obejktów w oknie .
- \* Integracja z biblioteką ttkbootstrap dla nowoczesnych stylów GUI.

### - OpenCV

Open Source Computer Vision Library to otwarta biblioteka do przetwarzania obrazu i wideo. Jest szeroko stosowana w aplikacjach wymagających analizy obrazów i wstępnego przetwarzania.

- \* Wczytywanie, modyfikacja i zapisywanie obrazów.
- \* Techniki przetwarzania obrazu takie jak filtracja, binaryzacja, detekcja krawędzi i konturów.
- \* Operacje morfologiczne jak erozja, dylatacja i usuwanie szumów.
- \* zastosowanie różnych przekształceń obrazów za pomocą np KMEANS\_RANDOM\_CENTERS

### **- Pytesseract**

Pytesseract to narzędzie do rozpoznawania tekstu z obrazów, będące interfejsem Pythona do biblioteki Tesseract OCR. Umożliwia automatyczne odczytywanie tekstu z obrazów paragonów.

- \* Konwersja obrazów na tekst.
- \* Obsługa różnych języków i formatów tekstu.
- \* Integracja z OpenCV do wstępnego przetwarzania obrazów przed OCR.
- \* Zapewnia dodatkowe możliwości przetwarzania obrazu przez openCv przed rozpoznaniem tekstu oraz ustwienie języka jaki jest spodziewany.

### **- Matplotlib**

Matplotlib to biblioteka do tworzenia statycznych i interaktywnych wizualizacji w Pythonie. Jest używana do generowania wykresów.

- \* Tworzenie różnorodnych wykresów: liniowych, słupkowych, kołowych, histogramów etc.
- \* Personalizacja wykresów: kolory, etykiety, osie i tytuły.
- \* Zapis wykresów w różnych formatach graficznych.

### **- Smtp**

Smtp to moduł Pythona do obsługi wysyłania e-maili za pomocą protokołu SMTP (Simple Mail Transfer Protocol). Jest wykorzystywany do wysyłania raportów drogą mailową.

- \* Konfiguracja połączenia z serwerem SMTP.
- \* Tworzenie i wysyłanie wiadomości e-mail, w tym załączników.
- \* Obsługa użytego szyfrowania (SSL) dla bezpiecznej komunikacji.

### **- Tkbootstrap**

Tkbootstrap to dodatkowa biblioteka dla Tkintera, która zapewnia nowoczesne i stylowe komponenty GUI. Pozwala na łatwe tworzenie atrakcyjnych interfejsów użytkownika.

- \* Nowoczesne style i motywy dla widżetów Tkintera.
- \* Zwiększoną estetykę i użyteczność aplikacji.
- \* Łatwa integracja z istniejącymi aplikacjami Tkintera.

### **- Pil(pillow)**

Pillow to rozwinięcie oryginalnej biblioteki PIL (Python Imaging Library), oferujące szeroką gamę narzędzi do obsługi obrazów.

- \* Wczytywanie i zapisywanie obrazów w różnych formatach (JPEG, PNG, GIF, TIFF etc.)
- \* Operacje na obrazach: przycinanie, zmiana rozmiaru, obracanie, filtrowanie.
- \* Obsługa grafiki rastrowej i konwersji kolorów.

## **Podsumowanie**

Dzięki zastosowaniu tych narzędzi i bibliotek oraz mniejszych modułów w pythonie, system jest w stanie efektywnie przetwarzać obrazy paragonów, rozpoznawać tekst, generować wizualizacje danych przetwarzając obrazy oraz komunikować się z użytkownikami poprzez intuicyjny interfejs graficzny. Każde z tych narzędzi wnosi unikalne możliwości i funkcjonalności, które wspólnie tworzą zaawansowane i spójne rozwiązanie.



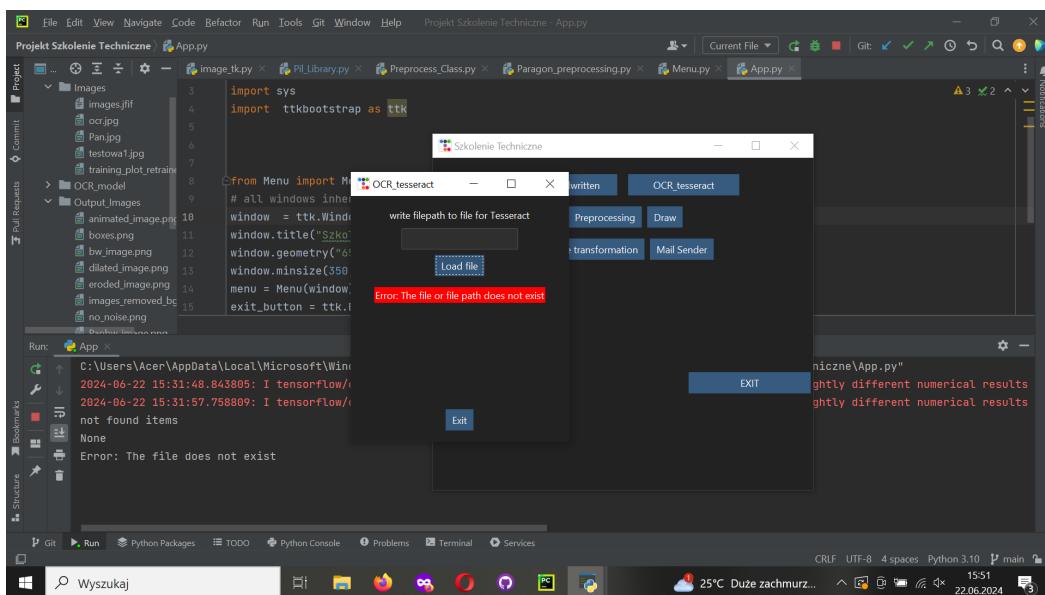
## Rozdział 4

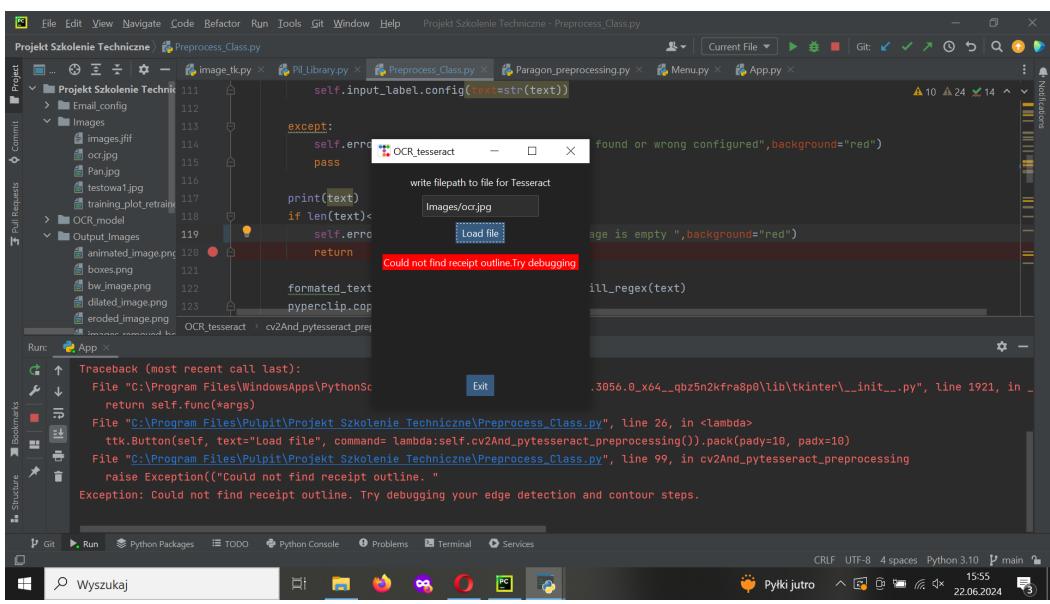
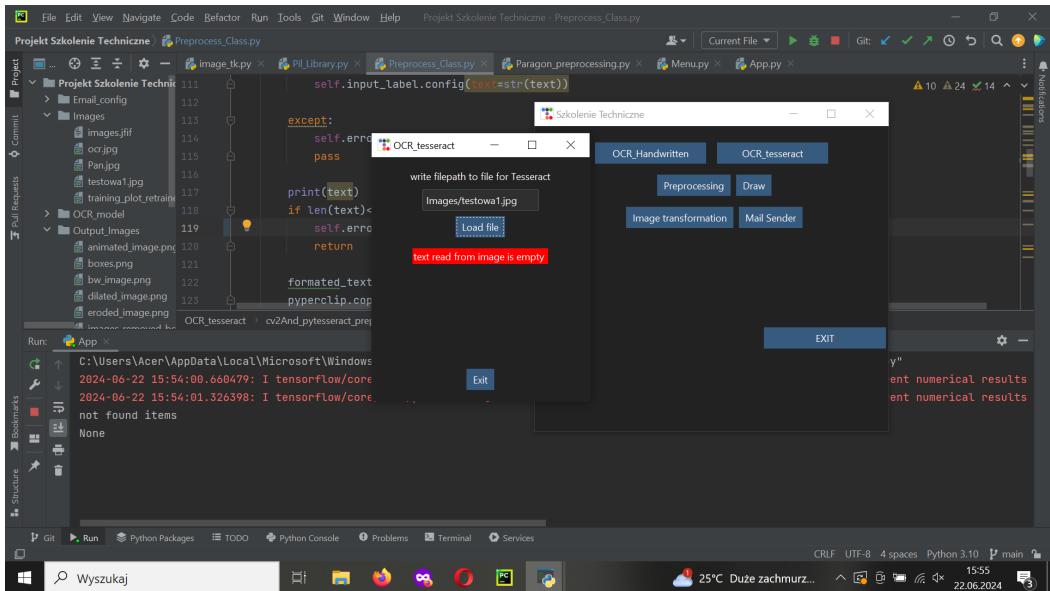
# **Integracja i Funkcjonalność Systemu**

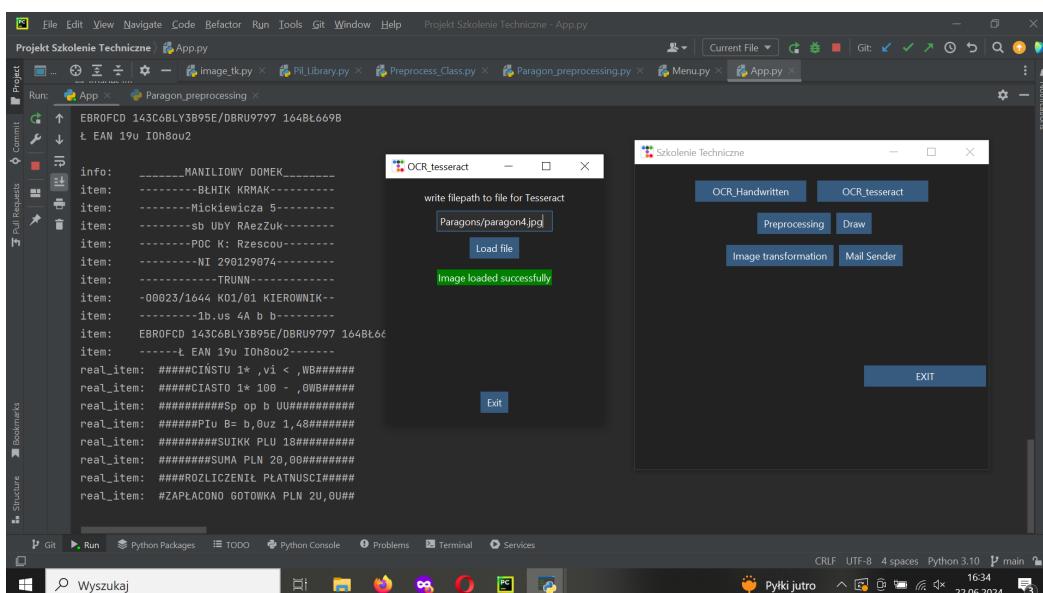
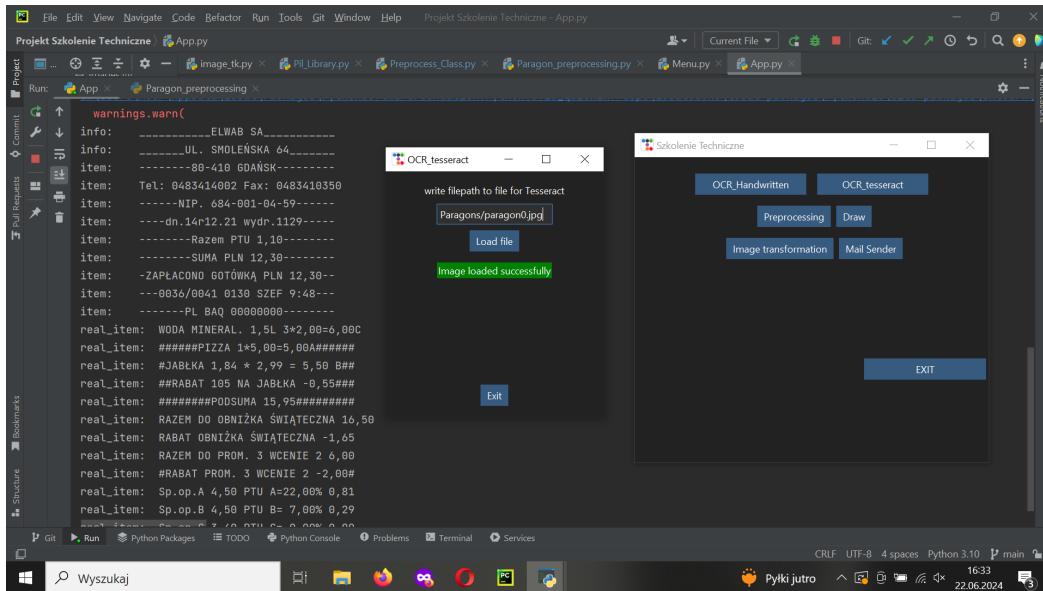
W tym rozdziale opiszemy szczegółowo proces integracji oraz ogólną funkcjonalność systemu rozpoznawania paragonów i edycji obrazów. Zawiera on integrację różnych funkcji takich jak ładowanie, przetwarzanie, analiza i edycja obrazów paragonów, co pozwala użytkownikom wykonywać te operacje za pomocą jednego narzędzia.

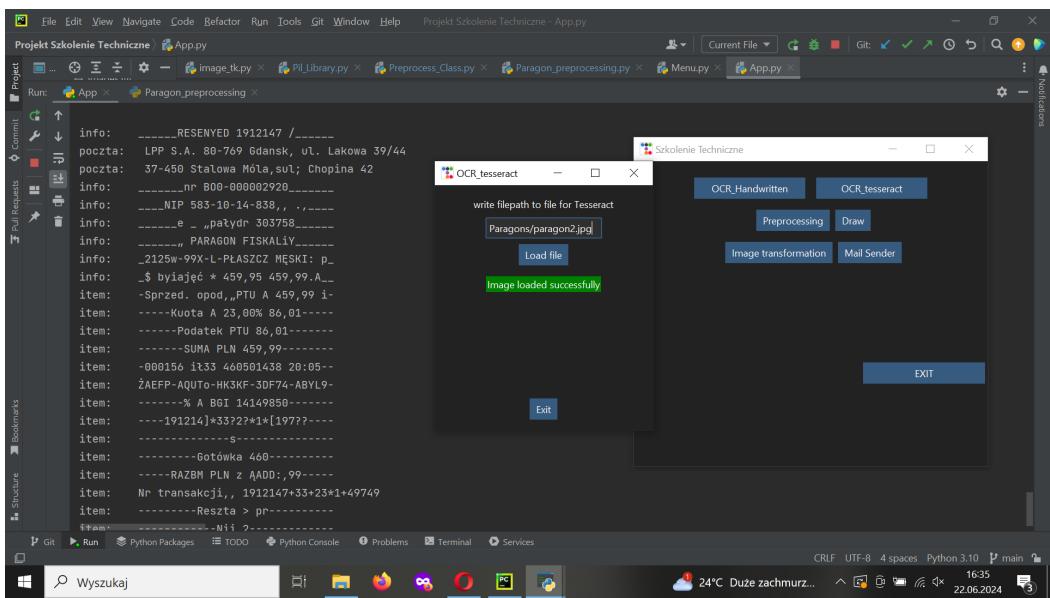
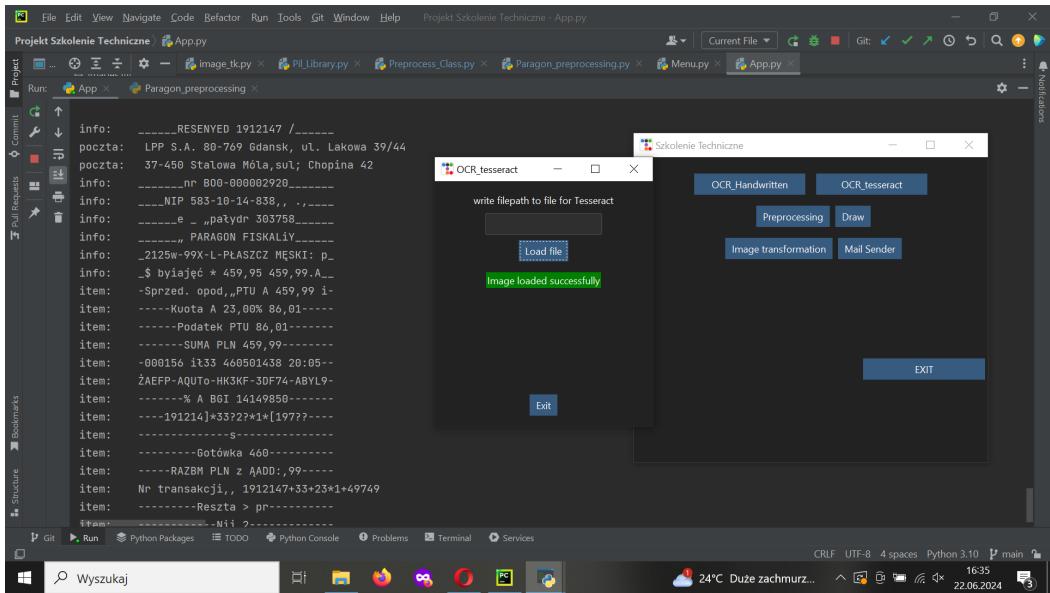
## 4.1 OCR\_teseract

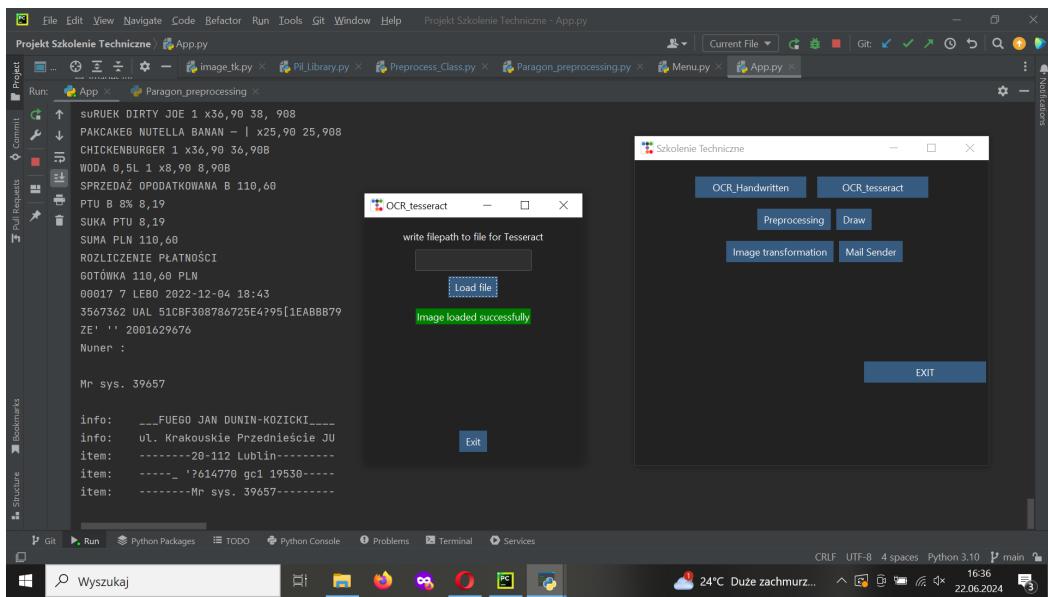
Klasa ta implementuje metodę `Bill_regex`, odpowiedzialną za rozpoznawanie i przetwarzanie tekstu z obrazów paragonów. Składa się z kilka kluczowych funkcji:











- **cv2And\_pytesseract\_preprocessing:**

Wczytuje ona obrazy paragonów z określonego folderu. Przetworzone teksty są następnie zapisywane w pliku tekstowym. Użycie OCR umożliwia automatyczne rozpoznawanie tekstu na paragonach, co jest kluczowe dla dalszej analizy.

## **SZCZEGÓŁY IMPLEMENTACJII**

### » Wczytywanie obrazów

Metoda rozpoczyna od przeglądania określonego katalogu, aby zidentyfikować obrazy paragonów. Każdy obraz jest wczytywany za pomocą biblioteki OpenCV i Pillow, zależnie od formatu i typu obrazu.

### » Zastosowanie OCR Tesseract

Po wczytaniu obrazu, metoda aplikuje OCR Tesseract do każdego obrazu, aby wyodrębnić tekst drukowany. Ustawienia OCR, takie jak język rozpoznawania (np. angielski, polski) oraz parametry przetwarzania obrazu (np. binaryzacja, usuwanie szumów), są dostosowywane w zależności od typu i jakości obrazu paragonu.

### » Zapisywanie wyników:

Przetworzony tekst, uzyskany z każdego paragonu, jest zapisywany do pliku tekstowego lub bazy danych. Każdy tekst jest zapisywany w odpowiednim formacie, aby zachować strukturę danych i umożliwić dalszą analizę.

- **Check\_similarity:**

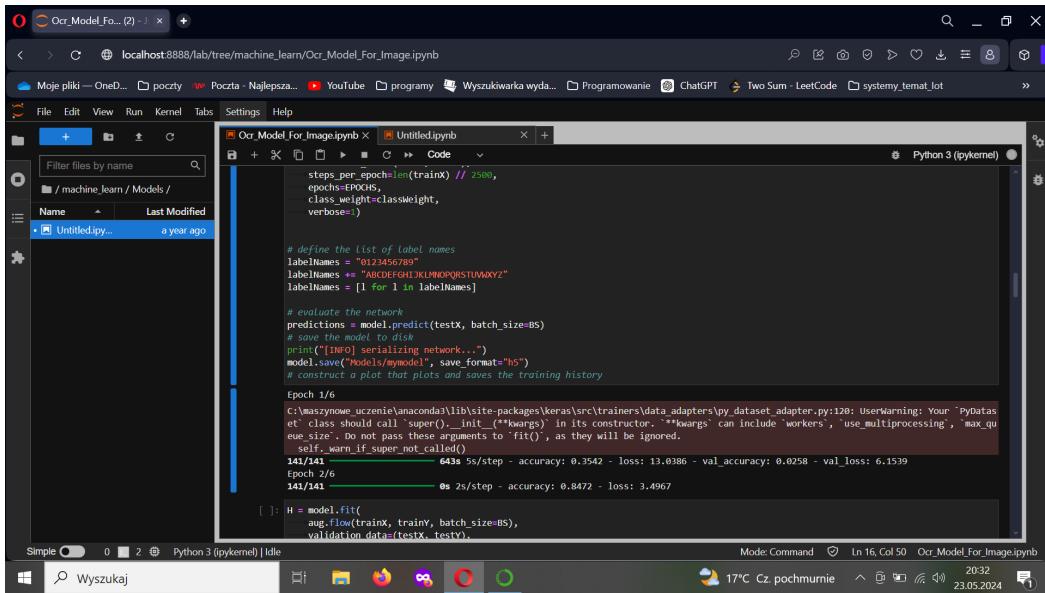
Ta funkcja używa podobieństwa kosinusowego do sprawdzenia, czy dany tekst przypomina standardowy tekst paragonu ("PARAGON FISKALNY"). Wykorzystuje techniki przetwarzania języka naturalnego, takie jak stemming i normalizacja tekstu, aby zapewnić dokładność analizy.

- **Bill\_regex:**

Metoda ta wykorzystuje wyrażenia regularne do wyodrębniania konkretnych informacji z tekstu paragonu. Analizuje tekst pod kątem informacji o sprzedawcy, listy zakupów oraz podsumowania cenowego, dzieląc go na trzy główne sekcje: informacje o sprzedawcy, lista zakupów i lista zakupów po wyrażeniu PARAGON FISKALNY.

## 4.2 OCR\_window

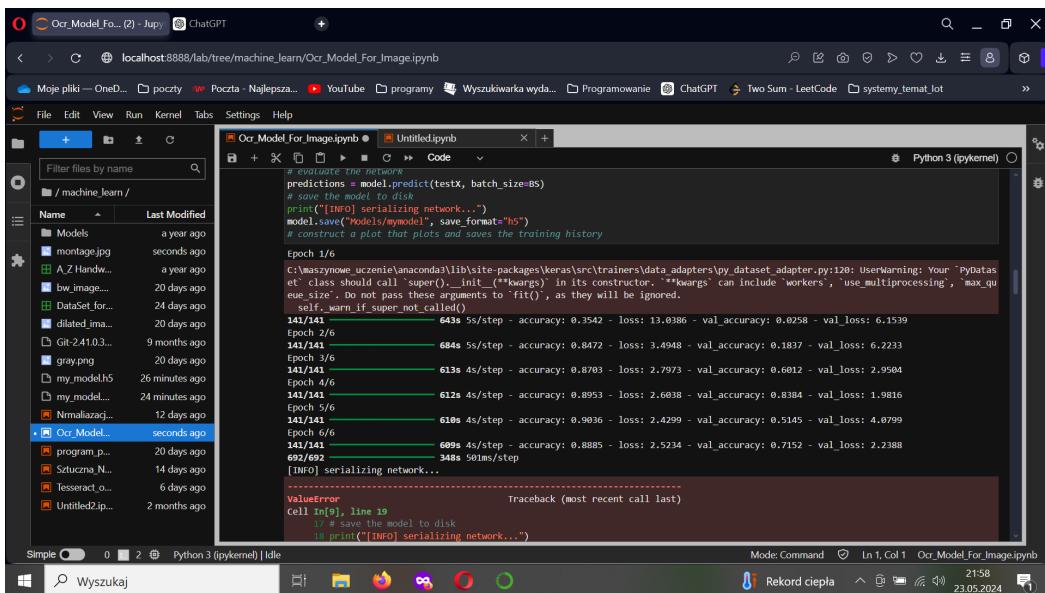
Moduł OCR\_window został stworzony w celu umożliwienia łatwego wczytywania obrazu z ręcznie pisany tekstem, za rozpoznanie i boxing liter odpowida biblioteka openCV następnie odnalezione elementy są transformowane i przetwarzane przez model mymodell\_retrained.h5 który został utworzony na bazie dnaych ręcznie pisanych liter oraz zbiorze mnist. Model można zaklasyfikować jako bardzo nieprzewidywalny ponieważ posiada bardzo słabo zdefiniowany system rozpoznawania i jego skuteczność może być bez problemu poddana pod wątpliwości natomiast niewątpliwym sukcesem jest osiągnięcie ponad 80% accuracy oraz połączenie zastosowania z gui oraz zapewnienie bezpieczeństwa podczas przetwarzania.



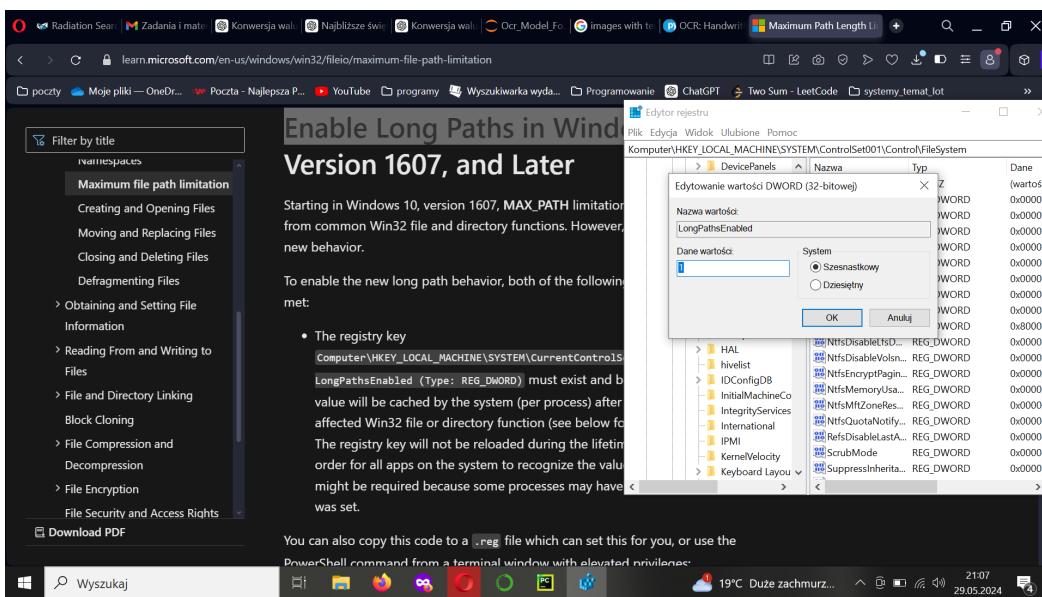
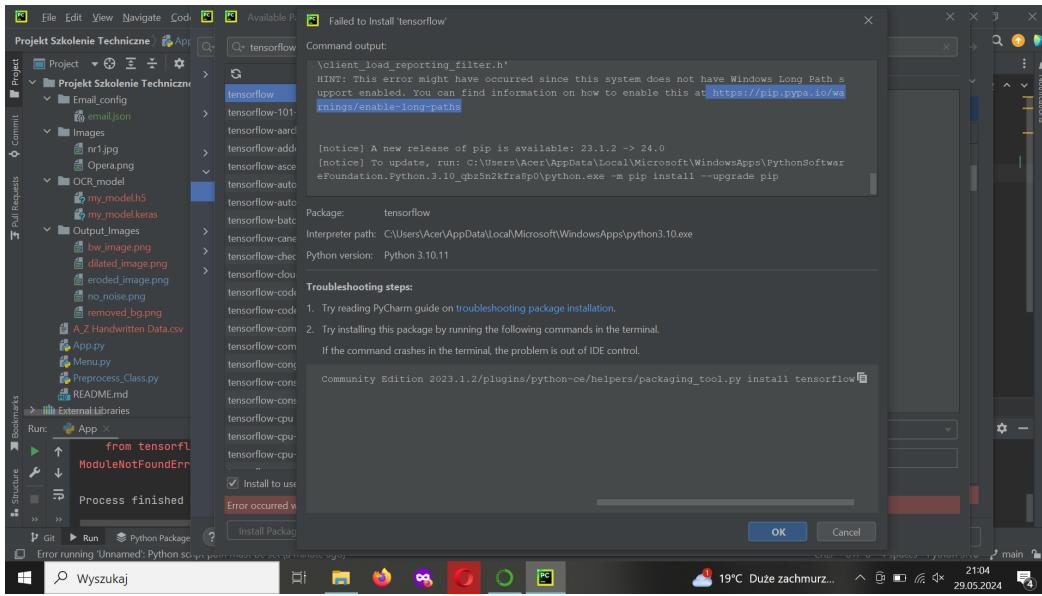
```
steps_per_epoch=len(trainX) // 2500,
epochs=Epochs,
class_weight=classweight,
verbose=1)

# define the list of label names
labelNames = "0123456789"
labelNames += "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
labelNames = [1 for i in labelNames]

# evaluate the network
predictions = model.predict(testX, batch_size=BS)
# save the model to disk
print("[INFO] serializing network...")
model.save("models/mymodel", save_format="h5")
# construct a plot that plots and saves the training history
Epoch 1/6
c:\maszynowe uczenie\anaconda3\lib\site-packages\keras\src\trainers\data_adapters\py_dataset_adapter.py:120: UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in its constructor. `**kwargs` can include `workers`, `use_multiprocessing`, `max_queue_size`. Do not pass these arguments to `fit()`, as they will be ignored.
  self._warn_if_super_not_called()
143/141   643s 5s/step - accuracy: 0.3542 - loss: 13.0386 - val_accuracy: 0.0258 - val_loss: 6.1539
Epoch 2/6
143/141   68s 2s/step - accuracy: 0.8472 - loss: 3.4967
# fit the network
H = model.fit(
    aug.flow(trainX, trainY, batch_size=BS),
    validation_data=(testX, testY),
```



```
# evaluate the network
predictions = model.predict(testX, batch_size=BS)
# save the model to disk
print("[INFO] serializing network...")
model.save("models/mymodel", save_format="h5")
# construct a plot that plots and saves the training history
Epoch 1/6
c:\maszynowe uczenie\anaconda3\lib\site-packages\keras\src\trainers\data_adapters\py_dataset_adapter.py:120: UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in its constructor. `**kwargs` can include `workers`, `use_multiprocessing`, `max_queue_size`. Do not pass these arguments to `fit()`, as they will be ignored.
  self._warn_if_super_not_called()
143/141   643s 5s/step - accuracy: 0.3542 - loss: 13.0386 - val_accuracy: 0.0258 - val_loss: 6.1539
Epoch 2/6
143/141   684s 5s/step - accuracy: 0.8472 - loss: 3.4948 - val_accuracy: 0.1837 - val_loss: 6.2233
143/141   613s 4s/step - accuracy: 0.8703 - loss: 2.7973 - val_accuracy: 0.6012 - val_loss: 2.9504
Epoch 4/6
143/141   612s 4s/step - accuracy: 0.8953 - loss: 2.6038 - val_accuracy: 0.8384 - val_loss: 1.9816
Epoch 5/6
143/141   610s 4s/step - accuracy: 0.9036 - loss: 2.4299 - val_accuracy: 0.5145 - val_loss: 4.0799
143/141   609s 4s/step - accuracy: 0.8885 - loss: 2.5234 - val_accuracy: 0.7152 - val_loss: 2.2388
[INFO] serializing network...
ValueError: Traceback (most recent call last)
cell In[9], line 19
  17 # save the model to disk
  18 print("[INFO] serializing network...")
```



The screenshot shows a Jupyter Notebook interface with several tabs open. The active tab is titled 'Ocr\_Model\_For\_Img.ipynb'. Below the tabs, there's a search bar and a sidebar for file navigation.

The main content area displays a Python code cell containing the following code:

```
precision    recall   f1-score  support
          0       0.14      0.93     0.25    1381
          1       0.95      0.99     0.97    1575
          2       0.81      0.97     0.89    1398
          3       0.89      0.99     0.94    1428
          4       0.94      0.97     0.96    1365
          5       0.75      0.87     0.88    1263
          6       0.58      1.00     0.73    1375
          7       0.93      0.97     0.95    1459
          8       0.93      0.96     0.94    1365
          9       0.93      0.99     0.96    1392
         10      0.99      0.97     0.98    2774
         11      0.98      0.93     0.96    1734
         12      0.99      0.92     0.96    4682
         13      0.98      0.84     0.90    2827
         14      0.98      0.98     0.98    2288
         15      0.97      0.98     0.97    232
         16      0.87      0.84     0.86    1152
         17      0.96      0.95     0.96    1444
         18      0.97      0.97     0.97    924
         19      0.98      0.93     0.95    1699
         20      0.97      0.96     0.97    1121
         21      0.99      0.88     0.93    2317
         22      0.98      0.99     0.99    2467
         23      0.98      0.92     0.94    3892
         24      0.98      0.34     0.58    11565
         25      1.00      0.98     0.99    3868
         26      0.99      0.84     0.91    1162
         27      0.99      0.97     0.98    2313
         28      0.98      0.94     0.96    9684
         29      0.99      0.96     0.98    4499
         30      0.96      0.98     0.97    5802
         31      0.98      1.00     0.99    836
         32      0.98      0.98     0.98    2317
         33      0.99      0.97     0.98    1254
         34      0.96      0.93     0.94    2172
         35      0.97      0.83     0.89    1215

accuracy                           0.97    88491
macro avg       0.92    0.93    0.91    88491
weighted avg    0.95    0.87    0.88    88491
```

Below the code, a 'Confusion Matrix:' section is visible, which likely corresponds to the output of the code above.



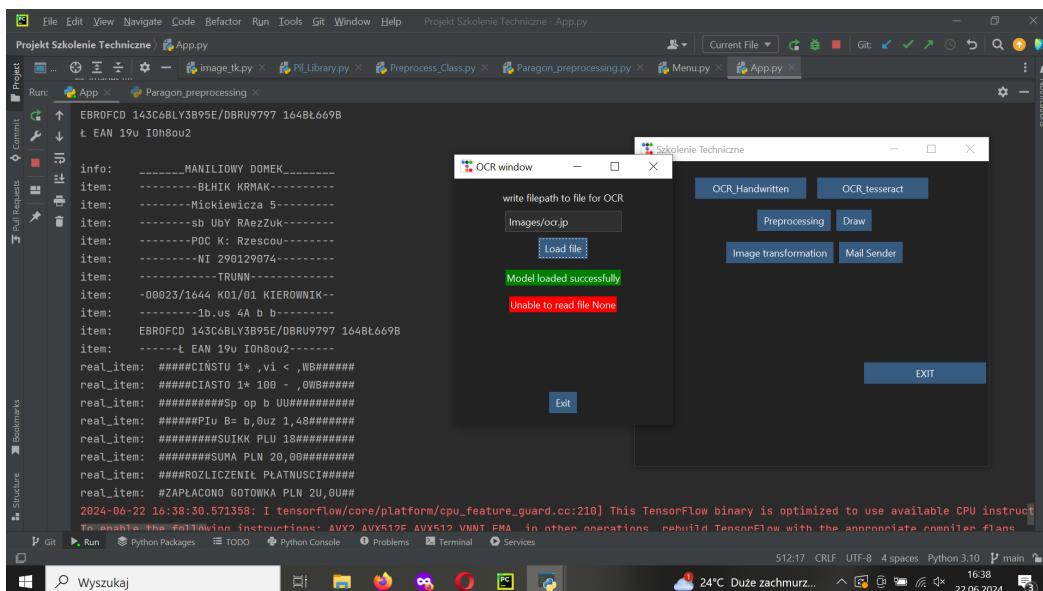
# SZCZEGÓŁY IMPLEMENTACJI

## • Klasa OCRWindow

Klasa OCRWindow jest jedną z głównych klas, która tworzy interfejs użytkownika oraz implementuje funkcje do wczytywania obrazów i rozpoznawania tekstu.

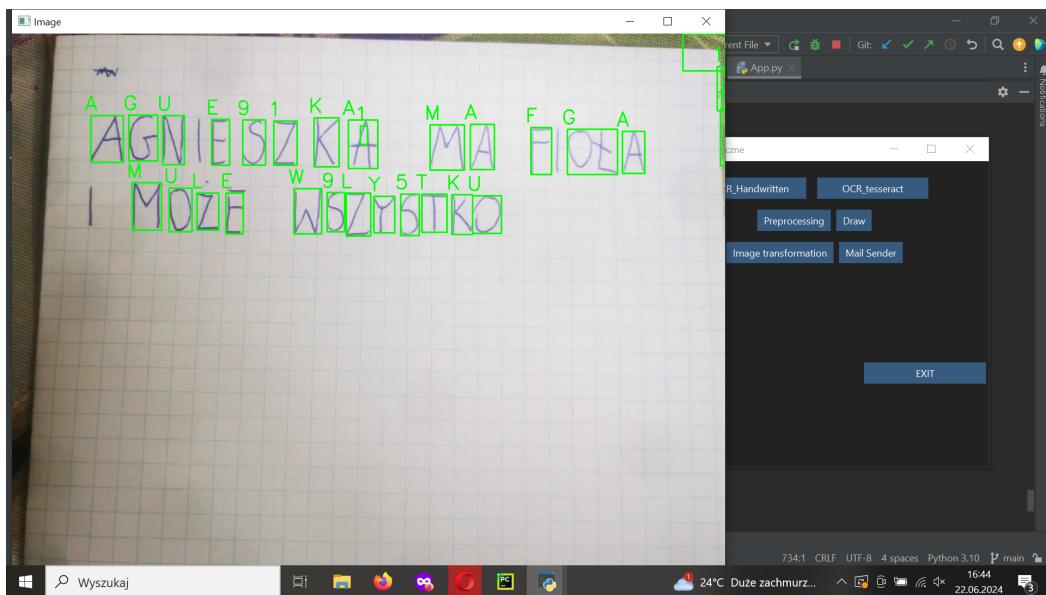
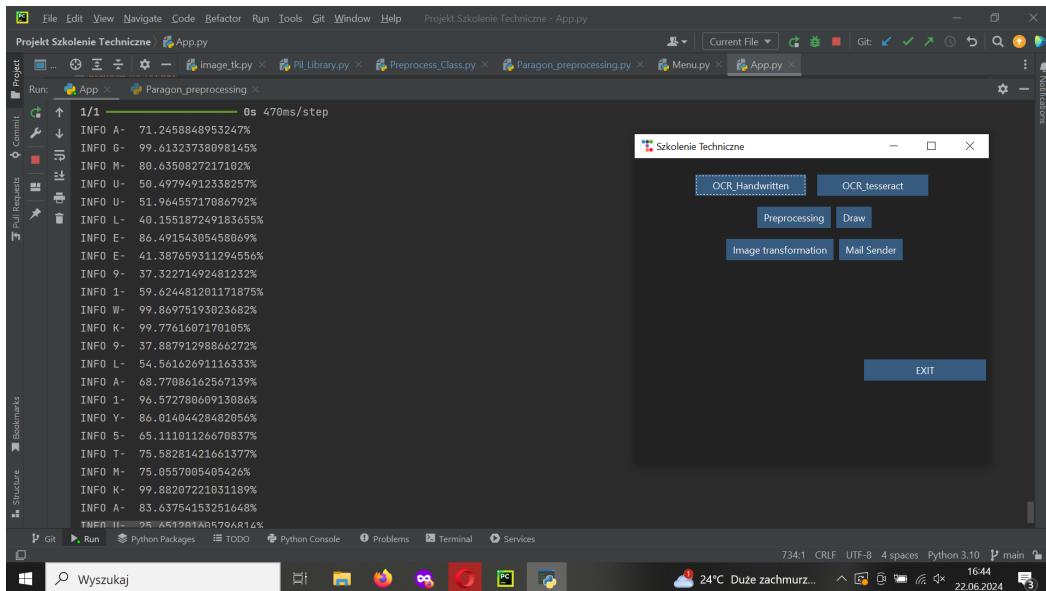
## • Wczytywanie obrazów

Implementujemy funkcję load\_image, która w oknie dialogowym przyjmuje względną ścieżkę obrazu, wczytuje wybrany obraz za pomocą PIL.Image.open nasepnie w bardzo prosty i oczywisty sposób użytkownik dostaje odpowiedź od systemu w postaci tekstu z kolorem zielonym jeśli wszystko poszło pomyślnie lub czerwonym przy wystąpieniu wyjątku np brak możliwości załadowania biblioteki openCV lub nie odnaleziono obrazu oraz drugi informujący czy model do rozpoznawania liter został poprawnie wczytany.



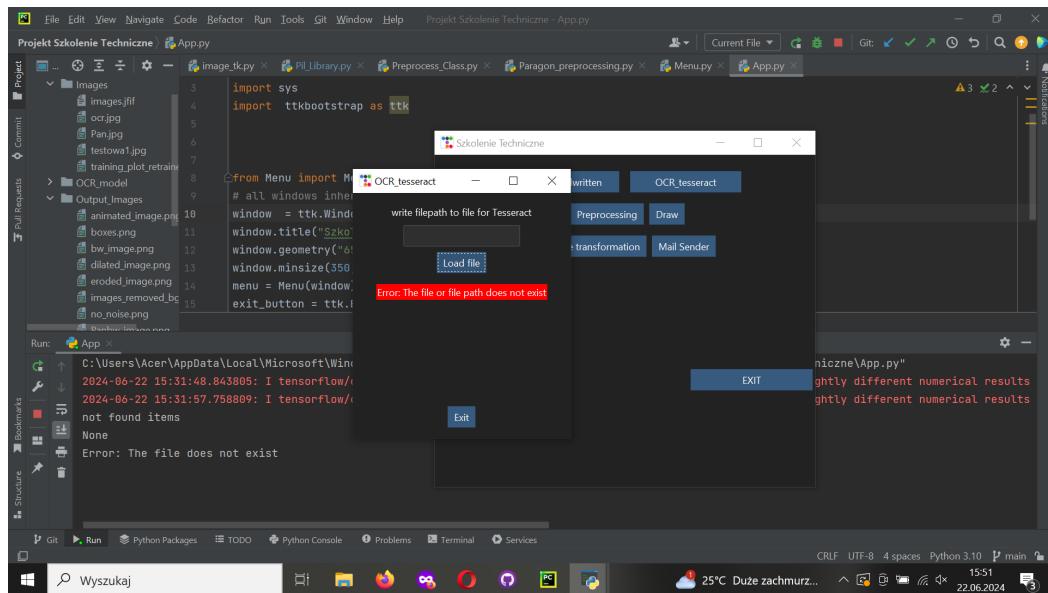
- **Rozpoznawanie tekstu**

Funkcja process\_with\_ocr wykorzystuje bibliotekę pytesseract, aby przetworzyć wczytany obraz na tekst i wyświetlić wynik w polu tekstowym (Text).



## 4.3 Preprocessing\_images

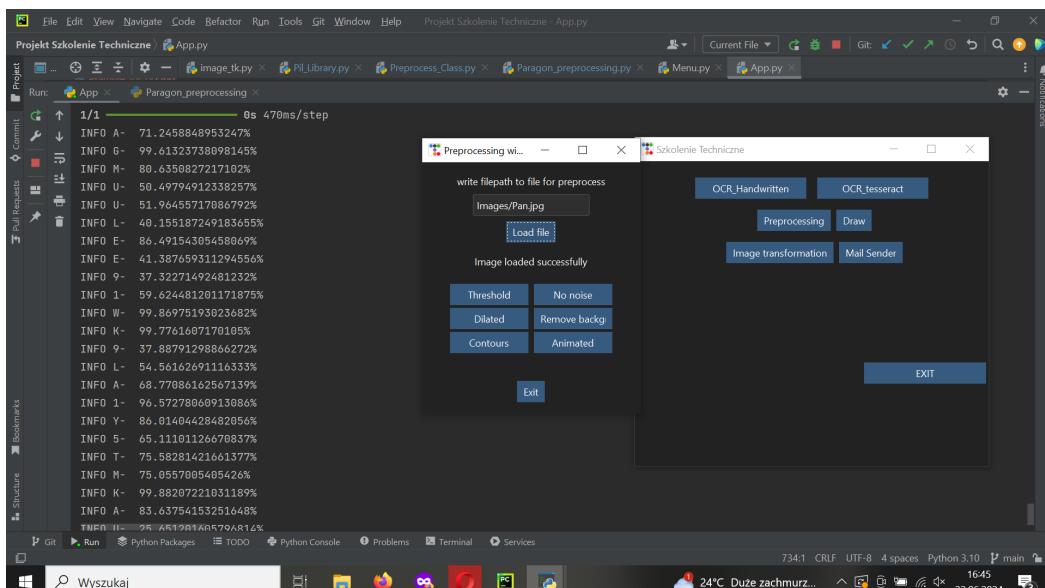
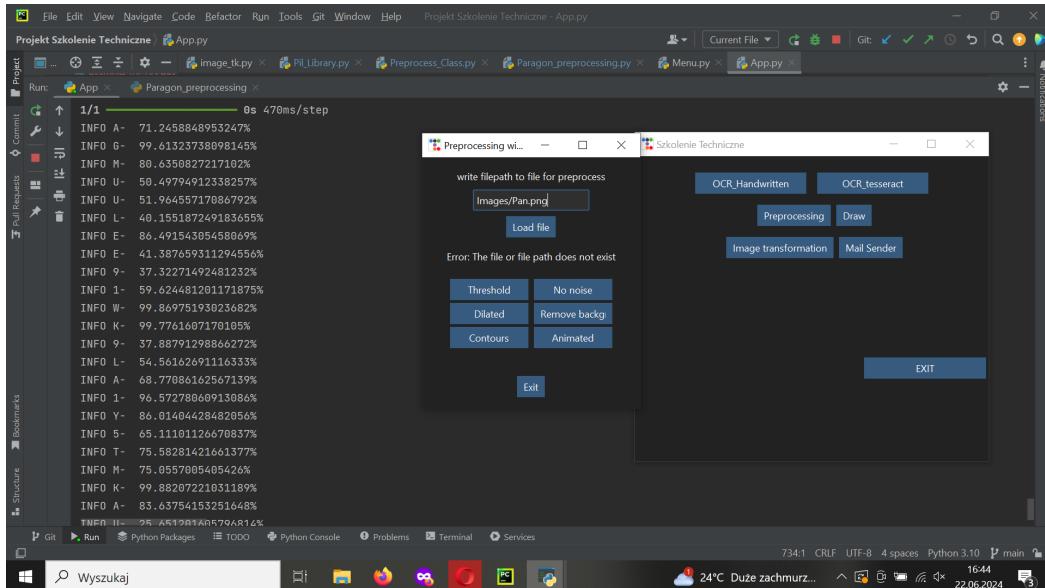
Moduł Preprocessing\_images zajmuje się przetwarzaniem obrazów. Dzięki zastosowaniu odpowiednich technik przetwarzania obrazu, takich jak binarizacja, usuwanie szumów, dylatacja, wykrywanie konturów, animacja obrazu oraz usuwanie tła, możemy dowolnie manipulować obrazami należącym zauważać tak jak w przykładowych wykorzystaniach że może wystąpić błąd w przetwarzaniu obrazu związany który w różnych momentach powoduje zacięcie się obrazu.



## SZCZEGÓŁY IMPLEMENTACJI

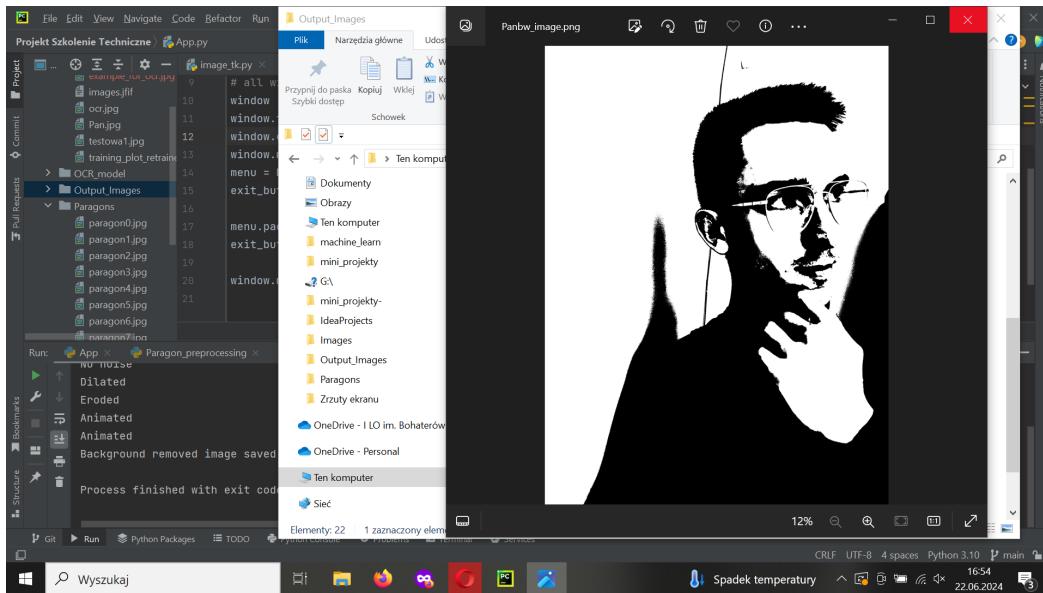
- **Process\_image:**

Metoda process\_image jest kluczowym elementem modułu preprocessing\_images odpowiedzialnym za wczytywanie obrazu przez OpenCV z określonego folderu.



- **Binaryzacja obrazu**

Binaryzacja polega na przekształceniu obrazu w dwuwymiarowy obraz czarno-biały, co upraszcza jego analizę. Proces ten można zrealizować za pomocą metody Otsu.



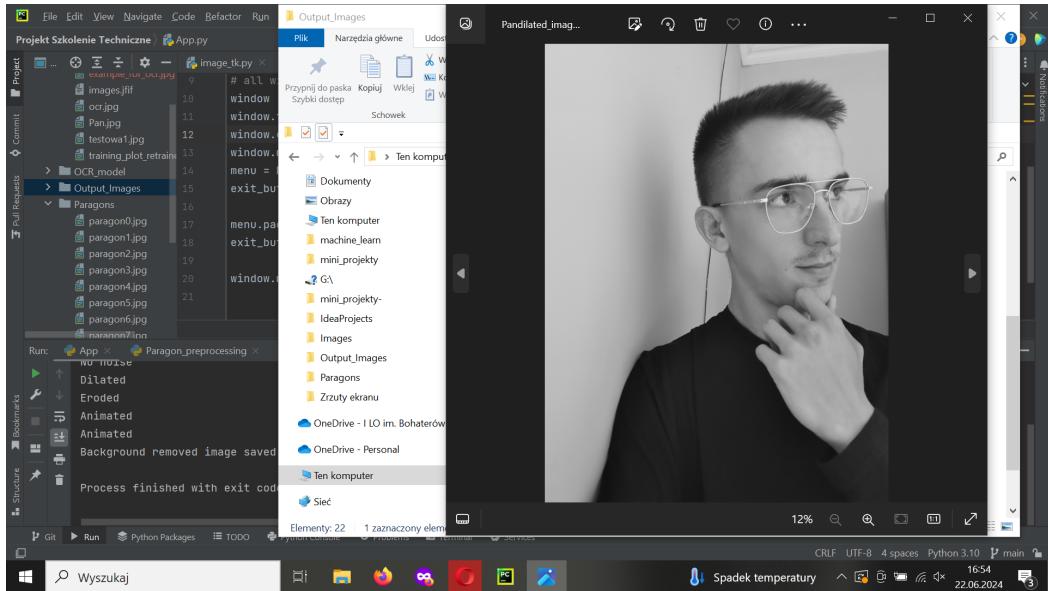
- **Usuwanie szumów**

Szумy mogą zakłócać proces rozpoznawania tekstu, dlatego usuwamy je za pomocą filtra medianowego, który jest skuteczny w redukcji szumów bez utraty ważnych informacji o krawędziach.



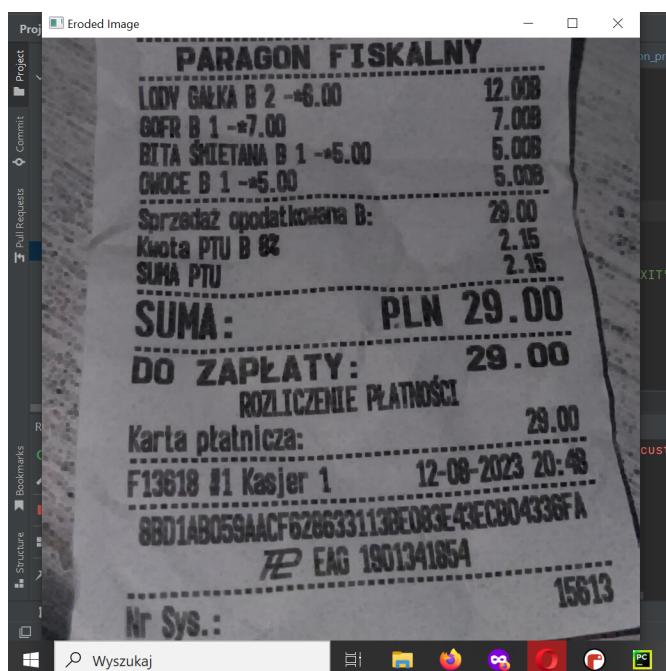
- **Dylatacja**

Dylatacja jest procesem rozszerzania jasnych obszarów obrazu, co może pomóc w połączeniu przerwanych fragmentów liter lub cyfr. Używamy w tym celu prostego elementu strukturalnego.



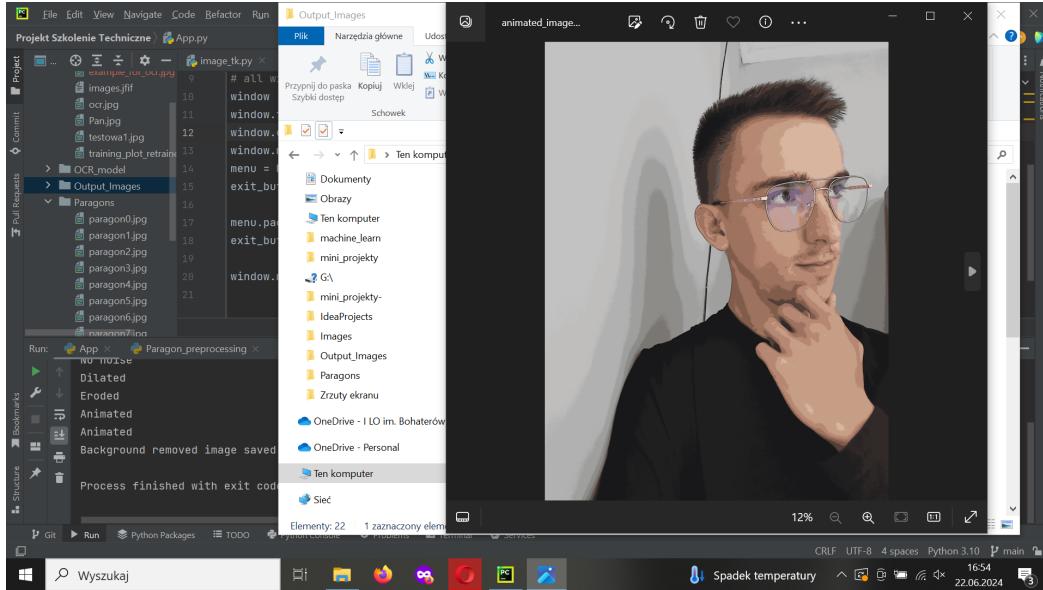
- **Wykrywanie konturów**

Wykrywanie konturów pozwala na identyfikację granic obiektów w obrazie. Jest to przydatne, przede wszystkim gdy chcemy zlokalizować tekst na paragonie jesteśmy w stanie przyciąć zdjęcie na odnalezionych konturach.



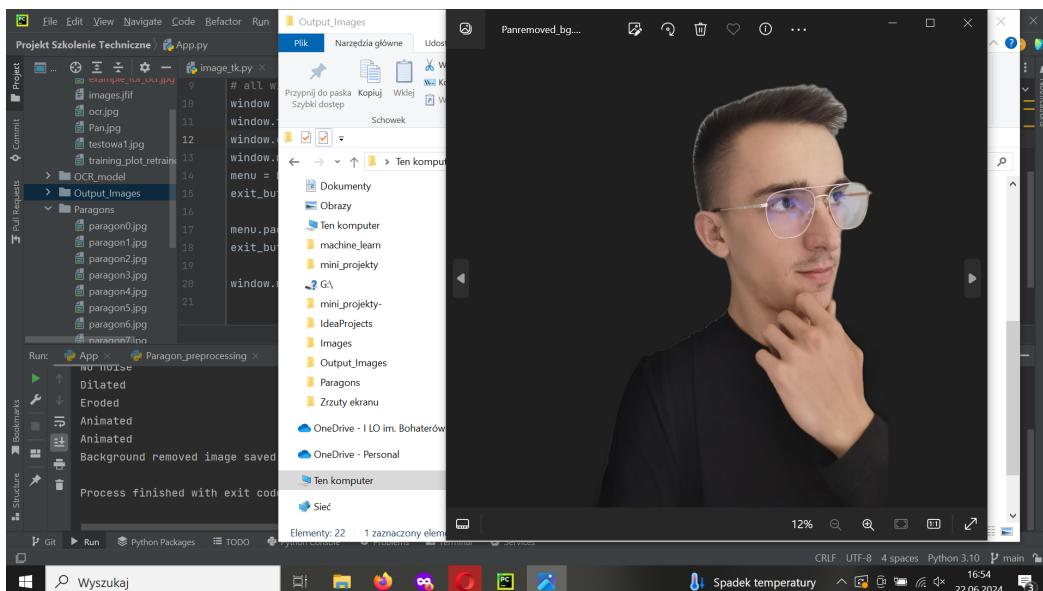
## • Animacja obrazu

Animacje obrazu, metoda która została dodana bez większego użycia może być bardzo skutecznie wykorzystana do zanimowania selfie lub innych zdjęć robiąc bardzo fajny efekt końcowy . Można to osiągnąć poprzez skalowanie wartości pikseli za pomocą algorytmu KMEANS\_RANDOM\_CENTERS.



## • Usuwanie tła

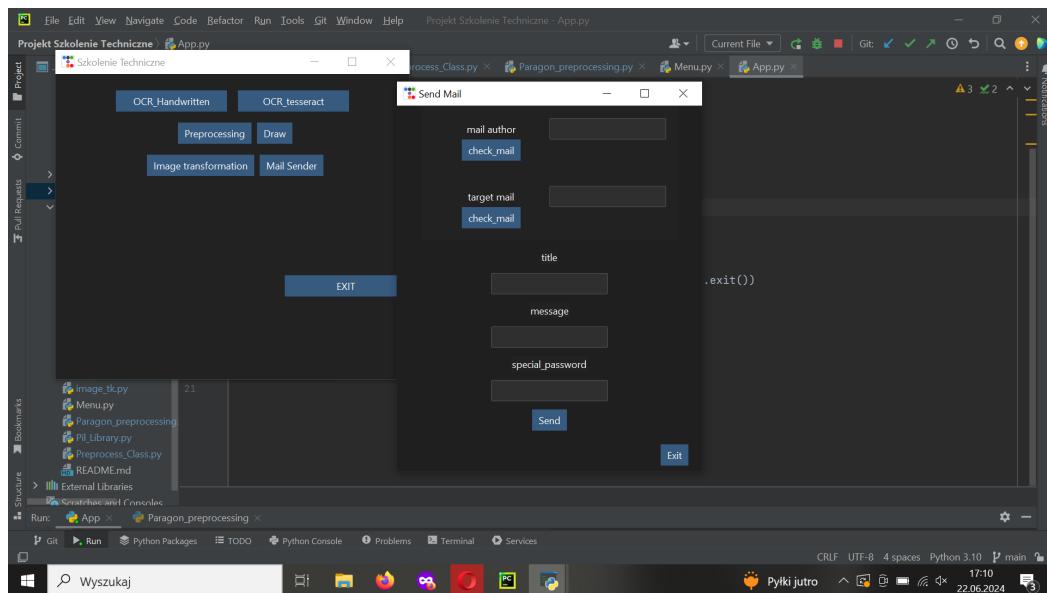
Usuwanie tła polega na eliminacji niepotrzebnych elementów tła, bardzo szybko wycina wszystko poza obręben głównych linii na obrazie . Proces ten można przeprowadzić za pomocą technik maskowania.

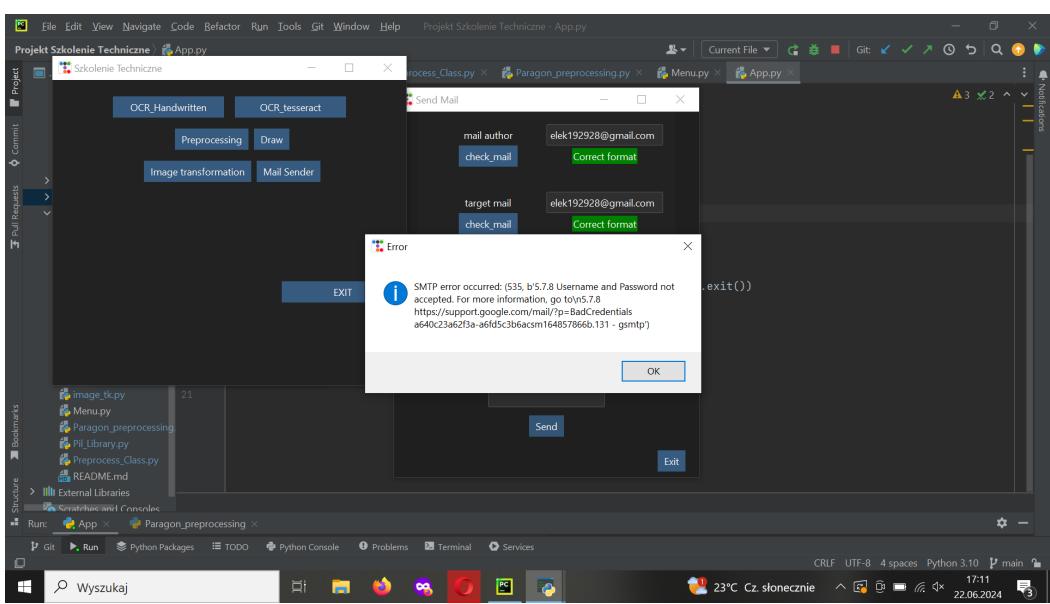
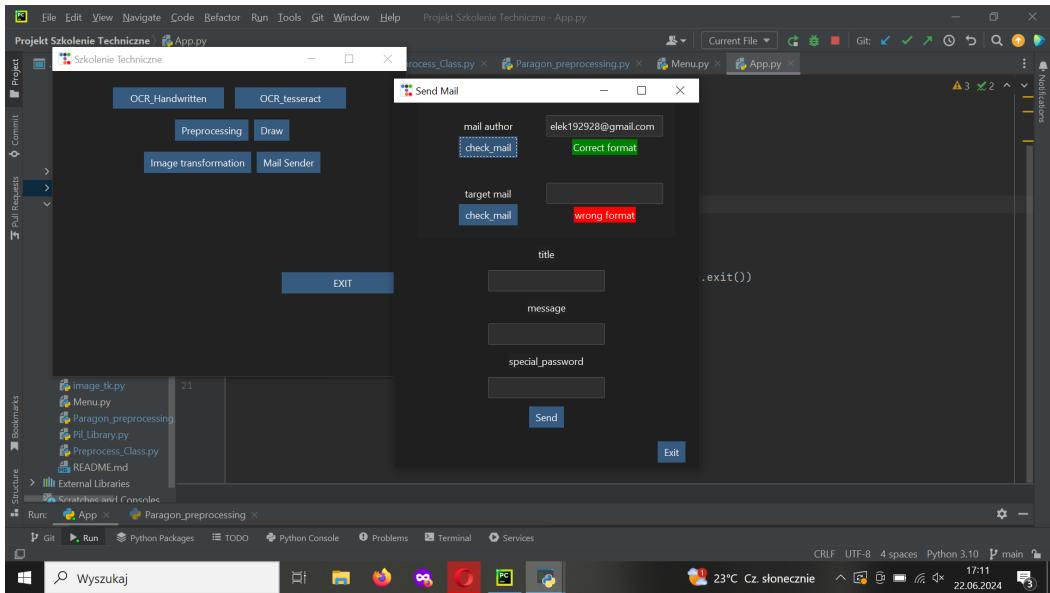


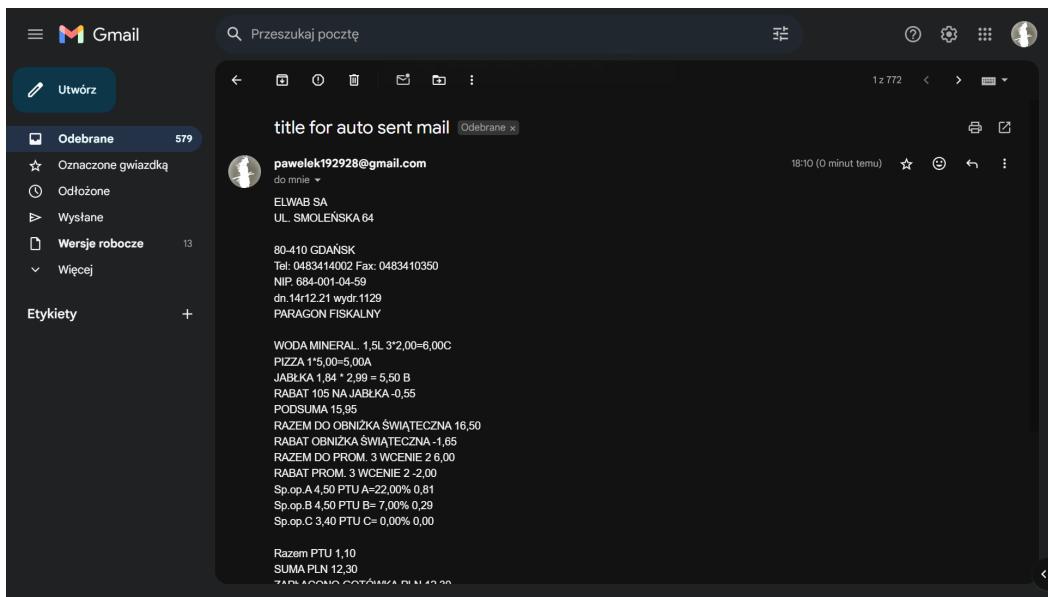
## 4.4 Mail\_sender

Moduł Mail\_sender umożliwia wysyłanie danych i raportów drogą mailową do określonych odbiorców. Jest to kluczowe dla efektywnego zarządzania wynikami analizy paragonów. Dzięki integracji z serwerem SMTP i odpowiedniemu przetwarzaniu wiadomości, moduł ten zapewnia niezawodne i bezpieczne przesyłanie informacji do odbiorców, co jest szczególnie przydatne w kontekście biznesowym i raportowym.)

```
class Mail_sender(tk.Toplevel):
    def __init__(self, parent):
        super().__init__(parent)
        # Kod konstruktora klasy Mail_sender
```







# SZCZEGÓŁY IMPLEMENTACJI

## Konfiguracja serwera SMTP

Przed wysłaniem wiadomości, musimy skonfigurować serwer SMTP. W tym przykładzie używamy serwera Gmail, ale można użyć dowolnego serwera SMTP.

## Logowanie do serwera SMTP

Logowanie do serwera SMTP odbywa się za pomocą adresu e-mail i hasła nadawcy. Upewnij się, że używasz odpowiednich poświadczeń.

## Tworzenie wiadomości e-mail

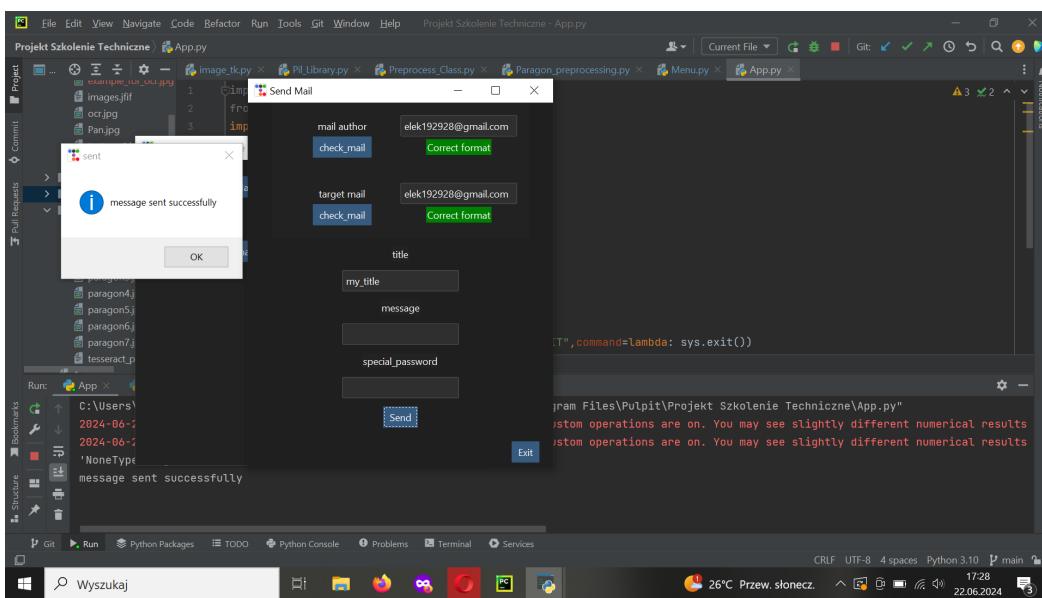
Wiadomość e-mail jest tworzona jako obiekt EmailMessage(), co pozwala na dodanie wielu części, takich jak tekst oraz załączniki.

## Wysyłanie wiadomości e-mail

Po stworzeniu wiadomości e-mail, można ją wysłać za pomocą wcześniej zalogowanego serwera SMTP.

## Funkcja wysyłania maila - mail\_sender

Korzystając z podanych w odpowiednich polach informacji jeśli są podane prawidłowo metoda check\_send\_mail przekazuje te informacje do głównej metody która tworzy obiekt następnie przesyłany przez serwer smtp, w razie wystąpienia jednego z wyjątków AttributeError lub wyjątku smtp zostanie wyświetlona informacja o niepowodzeniu, jeśli operacja się uda użytkownik jest przedstawiony messagebox z informacją że operacja została wykonana.



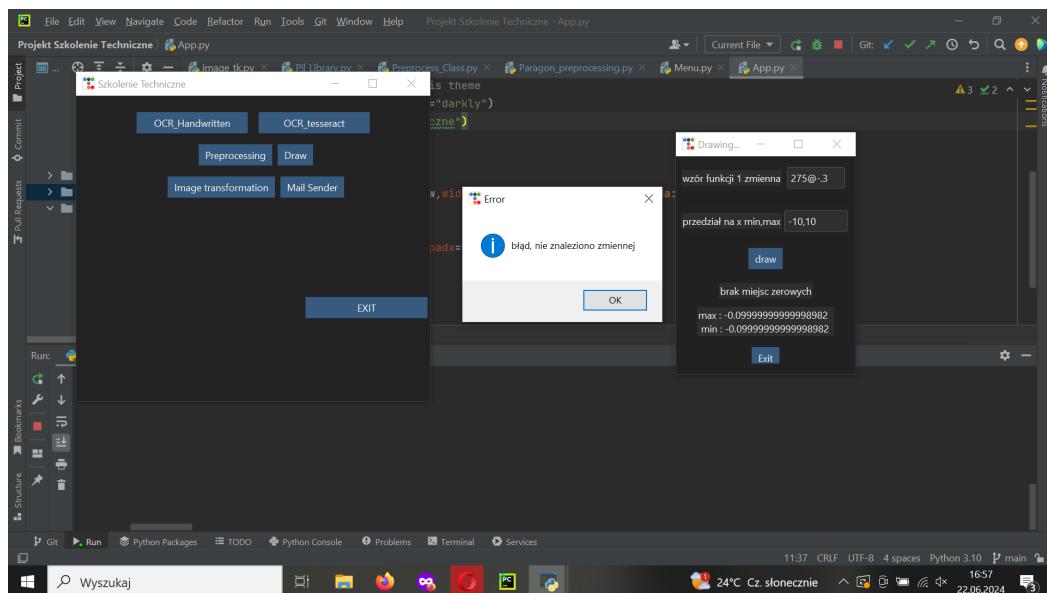
## 4.5 Draw

Moduł Draw umożliwia przeazanie wzoru funkcji jendej zmiennej jako argumentu na podstawie którego metoda tworzy wykres przecięcie z osią x oraz wyznacza charakterystyczne punkty. Dzięki integracji z interfejsem graficznym, moduł jest łatwy w obsłudze i oferuje obsługę dowonego wzoru dla funkcji matematycznej i zakresu dla jakich dane mają być liczone. Warunkiem JEST że przekazana wartość ma postać kodu python czyli podanie `cos(x)` spowoduje błąd składni, poprawne będzie przekazanie `math.cos(x)`, klasa ta jest dostosowana do potrzeb użytkownika i jest prosta w obsłudze.

```
class Draw(tk.Toplevel):
    def __init__(self, parent):
        super().__init__(parent)
        # Kod konstruktora klasy Draw
```

### Process\_function

Funkcja ta sprawdza czy użytkownik poprawnie wpisuje wzór funkcji, a następnie przetwarza ją na kod gotowy do skompli-walania za pomocą eval. Dla podanego przedziału wyznacza wartości funkcji oraz przekazuje ją do odpowiednich etykiet, które następnie prezentowane są użytkownikowi.



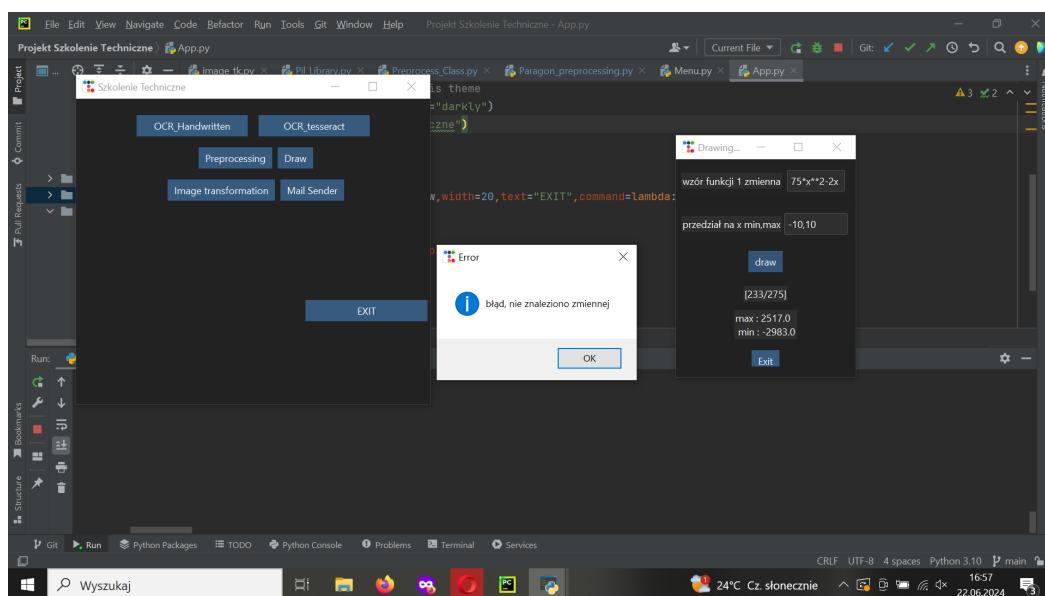
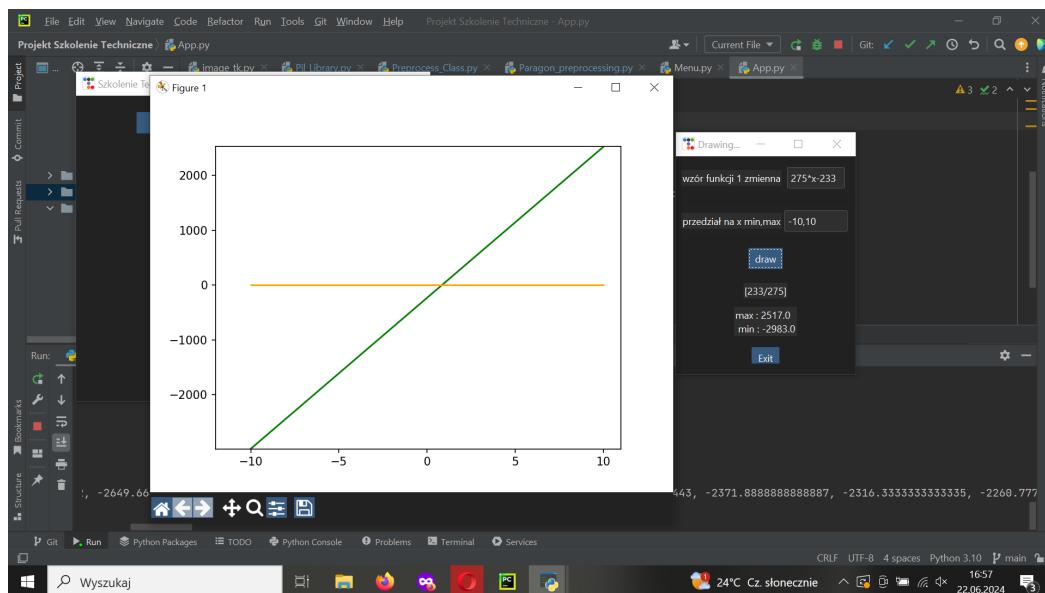
## SZCZEGÓŁY IMPLEMENTACJI

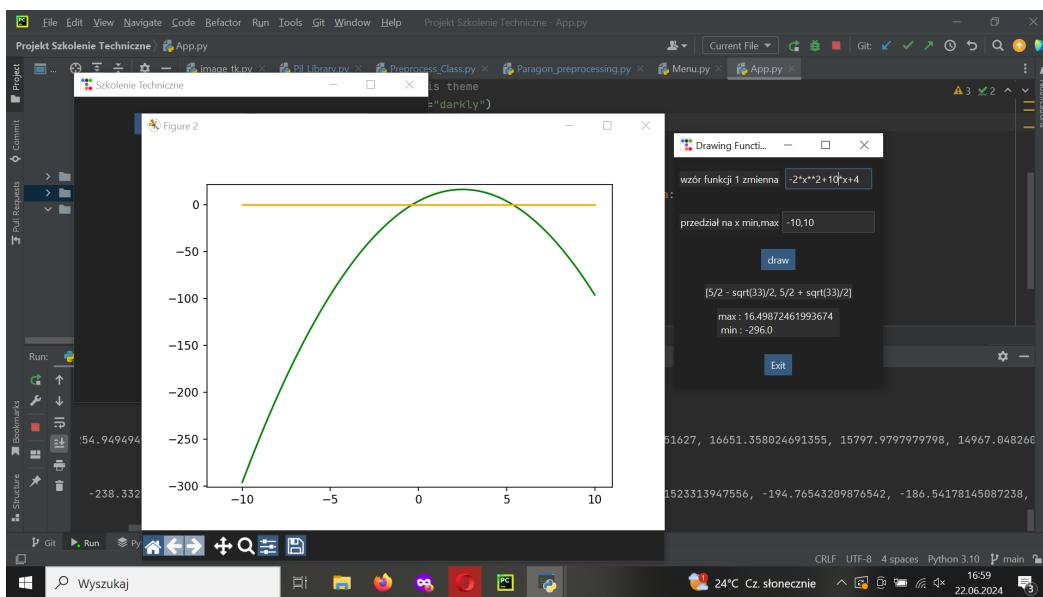
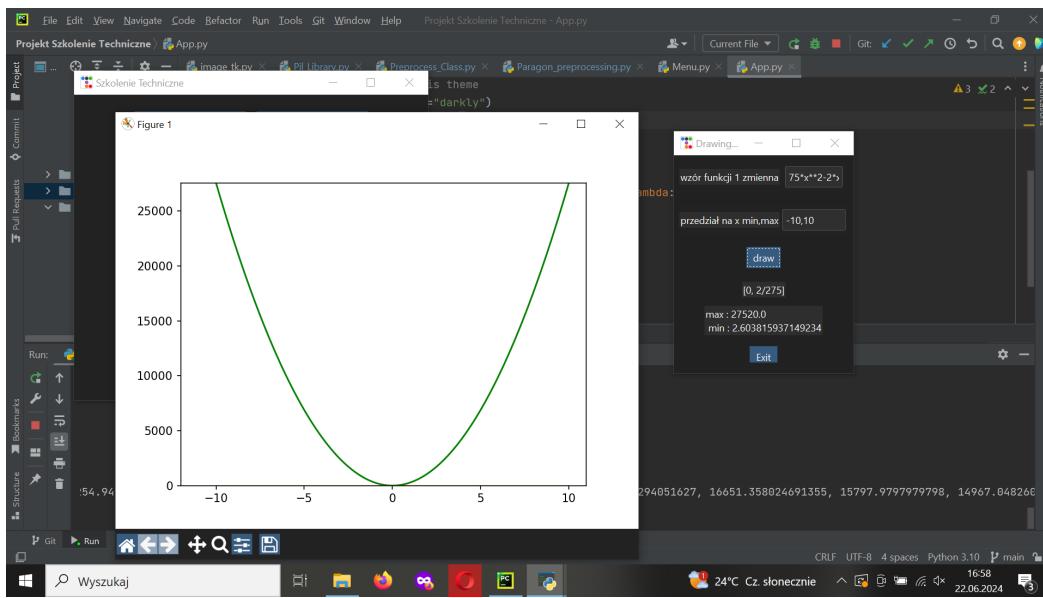
### Tworzenie narzędzi rysowania

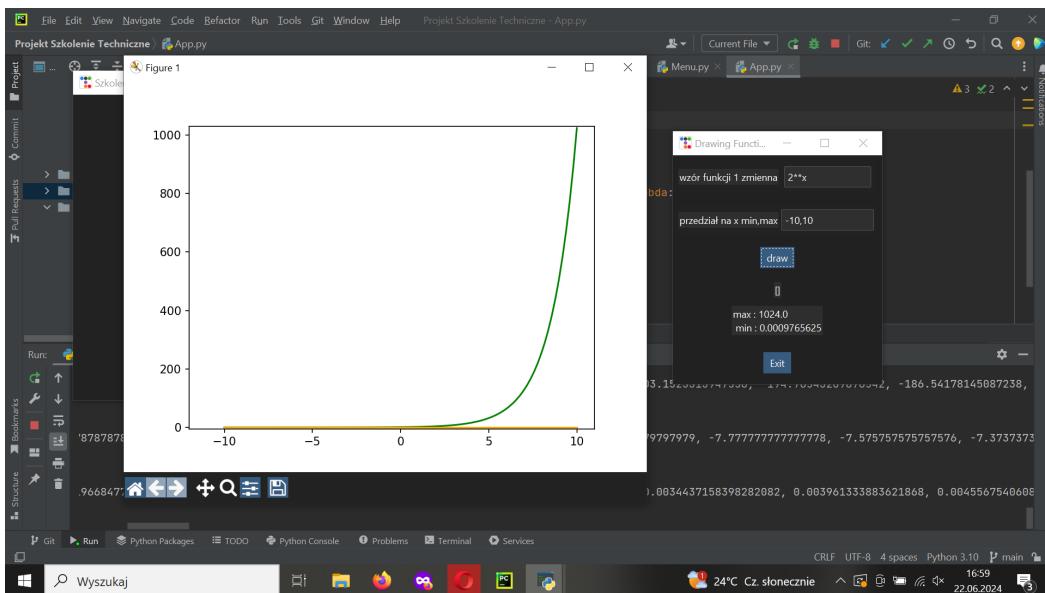
Moduł zapewnia narysowanie funkcji matematycznej w 2 wymiarowym układzie współrzędnych z zaznaczoną osią 0 oraz wyodrębnia ze wzoru miejsca zerowe i inne wartości.

### Integracja narzędzi z interfejsem użytkownika

Narzędzia rysunkowe są dostępne w interfejsie za pomocą przycisków i opcji wyboru. Każde narzędzie jest odpowiednio oznaczone i umieszczone.







## 4.6 Menu

Moduł Menu jest integralną częścią aplikacji, zapewniającą użytkownikom łatwy dostęp do wszystkich funkcji związańych z rozpoznawaniem paragonów i edycją obrazów. Dzięki przejrzystemu i intuicyjnemu interfejsowi, użytkownicy mogą szybko i efektywnie korzystać z aplikacji, wykonując różnorodne operacje w sposób uporządkowany i bezproblemowy.

```
class Menu(ttk.Frame):
    def __init__(self, parent):
        super().__init__(parent)
        # Kod konstruktora klasy Menu
```

## SZCZEGÓŁY IMPLEMENTACJI

### Tworzenie Menu

Menu główne aplikacji jest tworzone za pomocą biblioteki tkinter. Menu jest dodawane do głównego okna aplikacji jako klasa główna która w oknie głównym aplikacji dziedziczy z modułu ttkbootstrap theme oraz jego właściwości, a następnie są dodawane do niego poszczególne podmenu, zawierające różne opcje i funkcje. Każde podmenu zawiera odpowiadającą jej metodą uruchamianą na przycisk , które są wywoływanie po kliknięciu przez użytkownika.

```
class Menu(ttk.Frame):
    def __init__(self, parent):
        super().__init__(parent) # Wywołanie konstruktora klasy nadzędnej ttk.Frame
        # Ustawienie szerokości i wysokości oraz marginesów ramki
        self.config(width=230, height=170, padding=20)

        # Tworzenie ramki dla pierwszego wiersza przycisków
        new_row = ttk.Frame(self)
        # Przycisk "OCR" otwierający okno do obsługi funkcji rozpoznawania tekstu
        ttk.Button(new_row, text="OCR", command=self.open_OCR_window, width=18).pack(side='left', padx=10)
        new_row.pack(pady=10) # Umieszczenie ramki w głównej ramce menu

        # Tworzenie ramki dla drugiego wiersza przycisków
        new_row2 = ttk.Frame(self)
        # Przycisk "Preprocessing" otwierający okno do przetwarzania wstępnego obrazu paragonu
        ttk.Button(new_row2, text="Preprocessing", command=self.open_Preprocessing_window).pack(side='left', padx=10)
        # Przycisk "Draw" otwierający okno do rysowania
        ttk.Button(new_row2, text="Draw", command=self.open_Drawing_window).pack(side='left')
        new_row2.pack(pady=10) # Umieszczenie ramki w głównej ramce menu

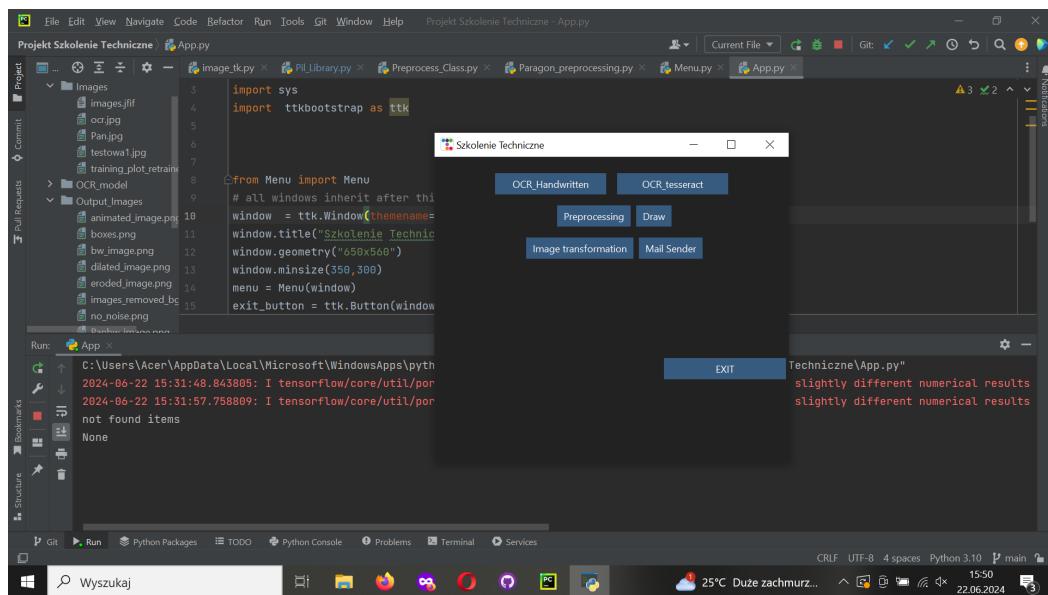
        # Tworzenie ramki dla trzeciego wiersza przycisków
        new_row3 = ttk.Frame(self)
        # Przycisk "Algorithms" otwierający okno do obsługi algorytmów
        ttk.Button(new_row3, text="Algorithms", command=self.open_Algorithms_window).pack(side='left', padx=10)
        # Przycisk "Mail Sender" otwierający okno do obsługi wysyłania maili
        ttk.Button(new_row3, text="Mail Sender", command=self.open_MailSender_window).pack(side='left')
        new_row3.pack(pady=10) # Umieszczenie ramki w głównej ramce menu
```

## Integracja z główną aplikacją

Menu jest zintegrowane z główną aplikacją, co umożliwia użytkownikom dostęp do wszystkich funkcji aplikacji z poziomu jednego miejsca. Menu jest dodawane do głównego okna aplikacji, a poszczególne funkcje są przypisywane do odpowiednich opcji menu.

## Obsługa błędów i komunikaty

Moduł Menu zawiera również mechanizmy obsługi błędów oraz wyświetlania komunikatów informacyjnych dla użytkownika. W przypadku problemów z otwieraniem lub zapisywaniem plików, użytkownik jest informowany o błędach za pomocą okien dialogowych.



## 4.7 ImageEditor

Moduł ImageEditor jest kluczowym komponentem aplikacji, który zapewnia funkcjonalność edycji obrazów paragonów. Umożliwia on użytkownikom wczytywanie, przetwarzanie i zapisywanie obrazów za pomocą graficznego interfejsu użytkownika (GUI). Wykorzystuje on biblioteki tkinter i ttkbootstrap, aby dostarczyć intuicyjny i przyjazny interfejs.

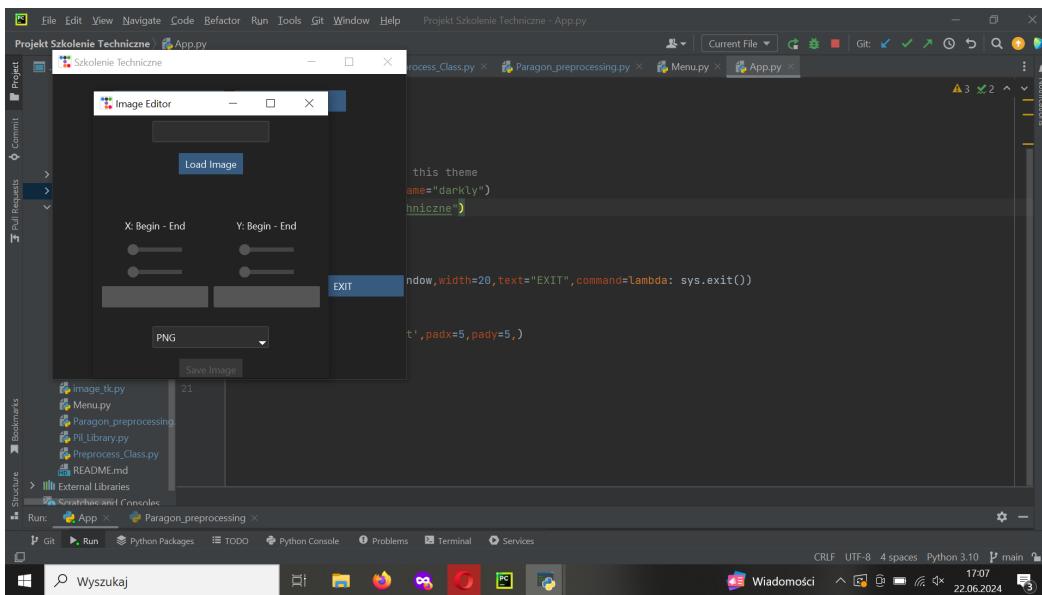
```
import tkinter as tk
from tkinter import ttk
import ttkbootstrap as ttk
from Preprocess_Class import SecondView, Preprocessing, Ocr_Window
import re
import json
import smtplib
import ssl
from email.message import EmailMessage
```

```
# Import modułów niezbędnych do działania aplikacji
import tkinter as tk # Moduł do tworzenia interfejsu użytkownika
from tkinter import ttk # Zaawansowane widgety interfejsu użytkownika
import ttkbootstrap as ttk # Gotowe motywów wyglądu
from Preprocess_Class import SecondView, Preprocessing, Ocr_Window # Klasy do przetwarzania
# wstępnego obrazu
import re # Moduł do wyrażeń regularnych
import json # Moduł do operacji na danych w formacie JSON
import smtplib # Moduł do wysyłania wiadomości e-mail
import ssl # Moduł zapewniający funkcje bezpieczeństwa dla protokołu SSL/TLS
from email.message import EmailMessage # Klasa do tworzenia wiadomości e-mail
```

# SZCZEGÓŁY IMPLEMENTACJI

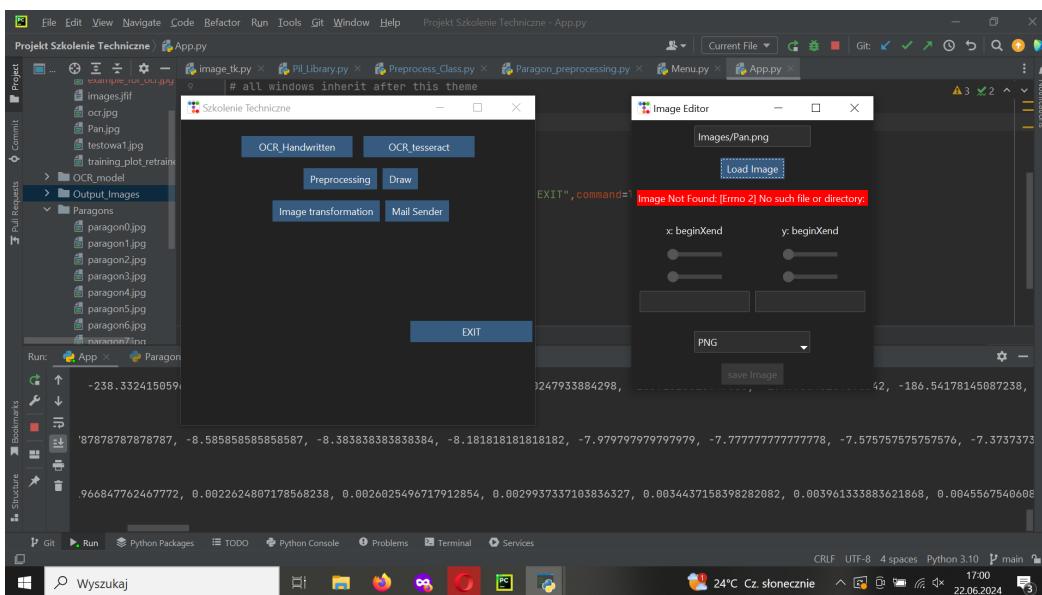
## Klasa ImageEditor

Klasa ImageEditor jest główną klasą modułu, która zarządza interfejsem użytkownika oraz implementuje funkcje do wczytywania, edycji i zapisywania obrazów.



## Wczytywanie obrazu

Metoda `load_image` otwiera okno dialogowe do wyboru obrazu, wczytuje wybrany obraz za pomocą PIL, a następnie wyświetla go na kanwie (canvas).

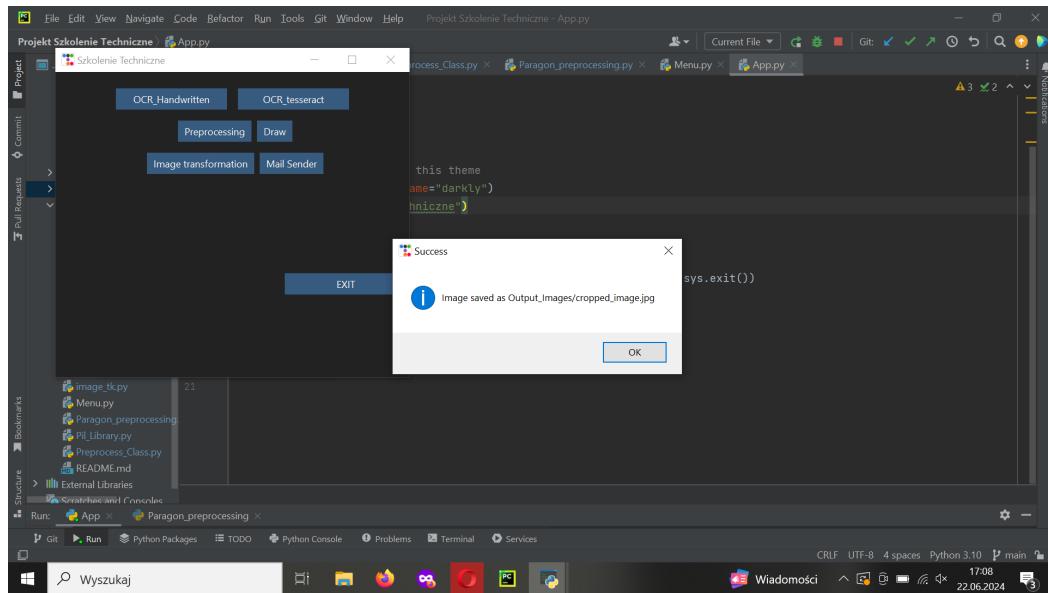


## Zapis obrazu

metoda save\_image Umożliwienie zapisu stworzonych rysunków oraz ich późniejsze wczytywanie jest kluczowe dla funkcjonalności modułu. Użytkownicy mogą zapisać swoje prace do plików graficznych (np. PNG, JPG).

## Edycja obrazów

Moduł może być rozszerzony o dodatkowe funkcje edycji obrazów, takie jak przycinanie, zmiana rozmiaru, obracanie czy zastosowanie filtrów. Poniżej przykładowa funkcja do przycinania obrazu:



## Integracja z główną aplikacją

Klasa ImageEditor jest zintegrowany z główną aplikacją, co umożliwia użytkownikom dostęp do funkcji edycji obrazów bezpośrednio z interfejsu aplikacji.

## Obsługa błędów i komunikaty

Moduł zawiera mechanizmy obsługi błędów oraz wyświetlanie komunikatów informacyjnych, co zapewnia użytkownikom informacje na temat ewentualnych problemów z wczytywaniem, edycją lub zapisywaniem obrazów.