6. Sztuczna inteligencja

Leszek Klich

6.1 Wstęp

Sztuczna inteligencja to termin, który coraz częściej pojawia się w kontekście nowoczesnego przemysłu, fabryk oraz wysokich technologii. W praktyce, sztuczną inteligencję - a dokładniej - uczenie maszynowe znajdziemy w urządzeniach powszechnego użytku! Tyle tylko, że większość użytkowników nie zdaje sobie z tego sprawy.

Dla przykładu - wszyscy korzystamy ze smartfonów, tabletów czy komputerów osobistych, które wykorzystujemy w coraz szerszym zakresie. Kiedyś telefony posiadały wyłącznie funkcje rozmowy i pisania krótkich wiadomości SMS. Obecne telefony wyposażone są w wydajne procesory, ogromną pamięć oraz duże i czytelne ekrany. Dzisiejsze smartfony to bardzo wydajne komputery, zamknięte w niewielkiej obudowie. To radykalnie rozszerzyło ich możliwości. Popularne stały się komunikatory, aplikacje sieci społecznościowych, streaming filmów, gry czy inne aplikacje mobilne. Nieodłącznym elementem współczesnego smartfona jest wbudowany aparat fotograficzny, który potrafi zapisywać wysokiej jakości zdjęcia i filmy. Jakość wbudowanego aparatu stanowi, dla niektórych, decydującą zachętę do wyboru konkretnego modelu.

Ale co ma aparat do sztucznej inteligencji? W połączeniu z oprogramowaniem wyposażonym w specjalne algorytmy można retuszować

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

zdjęcia, nakładać filtry zmieniające nasz wygląd, rozpoznawać osoby, zmieniać tło czy identyfikować przedmioty na zdjęciu.

W niniejszym rozdziale pokażę, jak napisać bardzo prosty program w języku Python, który będzie potrafił identyfikować twarze na zdjęciach.

6.2 Co to jest sztuczna inteligencja?

Sztuczna inteligencja to w uproszczeniu algorytmy, które umożliwiają systemom technicznym rozwiązywanie zadań, w których mamy do czynienia z bardzo dużą ilością danych, a dodatkowo dane te są na tyle złożone, że człowiek nie jest w stanie ich poprawnie zinterpretować i wyciągnąć właściwe wnioski.

Każdy komputer odbiera dane z pewnych czujników (np. kamera, mikrofon), przetwarza je i odpowiednio reaguje. Analizując dane pochodzące z kamery, komputer jest w stanie do pewnego stopnia dostosować swoje zachowanie, poprzez analizę skutków wcześniejszych działań i działając autonomicznie.

Definicja 6.1 — Sztuczna inteligencja (ang. artificial intelligence, AI). To możliwość samouczenia się maszyn na podstawie doświadczeń, dostosowywanie się do nowo pobranych danych i wykonywanie zadań podobnych do ludzkich. Nowoczesne algorytmy uczenia maszynowego umożliwiają szkolenie, które pozwala na wykonywanie konkretnych zadań poprzez przetwarzanie olbrzymich ilości danych i identyfikację wzorców w tych danych.

Nauka o sztucznej inteligencji jest postrzegana jako centralny element cyfrowej transformacji społeczeństwa i stała się priorytetem w Unii Europejskiej. Z tego powodu kładzie się szczególny nacisk na kierunki rozwoju tej dziedziny nauki, bowiem przyszłe zastosowania mogą przynieść ogromne zmiany i przyczynić się do szybkiego rozwoju wielu obszarów naszego życia.

6.3 Język Python

Python to darmowy język programowania charakteryzujący się czytelną składnią i łatwością nauki. Jest też na tyle uniwersalny, że przy jego pomocy można tworzyć aplikacje o wielu zastosowaniach. Kolejną istotną cechą nie języka, ale społeczności jest olbrzymia ilość bezpłatnych bibliotek. Dzięki temu wiele zadań, które chcemy wykonać, jest dostępnych w postaci gotowych paczek.

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

Warto także wspomnieć, że kod, który musimy napisać, by uzyskać gotowy efekt, jest czasami wielokrotnie krótszy w porównaniu z innymi językami. W obszarach sztucznej inteligencji to właśnie Python jest najbardziej popularny, ponieważ posiada wiele zalet w tym obszarze. Kilka z nich wymieniłem poniżej:

- posiada bardzo dobrą dokumentację;
- dzięki popularności łatwo uzyskać pomoc od innych i dzielić się doświadczeniami;
- działa w systemie Windows, Mac OS oraz Linux;
- jest darmowy;
- jest bardzo prosty w nauce;
- uniwersalność sprawia, że może być używany do wielu typów aplikacji, takich jak aplikacje naukowe, matematyczne, internetowe itp.;
- dostępnych jest wiele bibliotek, które programista może wykorzystać we własnych aplikacjach.

Jak zainstalować język Python?

Aby zainstalować język Python w systemie Windows skorzystaj z Instrukcji instalacji oprogramowania Python, która została szczegółowo opisana w załączniku A.

Środowisko programistyczne

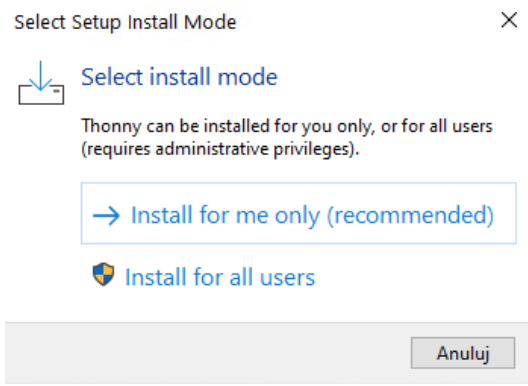
Choć do programowania w języku Python wystarczy dostępny w systemie Notatnik i wiersz polecenia, w praktyce nie jest to jednak efektywne.

Aby wygodnie programować, należy pobrać i zainstalować środowisko programistyczne, które znacznie ułatwi programowanie. W skład środowiska programistycznego (IDE - Integrated Development Enviroment) wchodzi między innymi zaawansowany edytor kodu, który ułatwi programowanie oraz pozwoli jednym przyciskiem uruchamiać napisany program. Dla języka Python istnieje wiele darmowych oraz płatnych środowisk programistycznych, ale wybrałem proste, lecz bardzo funkcjonalne środowisko o nazwie Thonny.

Bezpośrednio po uruchomieniu instalatora, program instalacyjny zapyta o wybór trybu instalacji. W tym miejscu można wybrać proponowany tryb, czyli **Install for me only (recommended)**, co ilustruje obrazek 6.1.

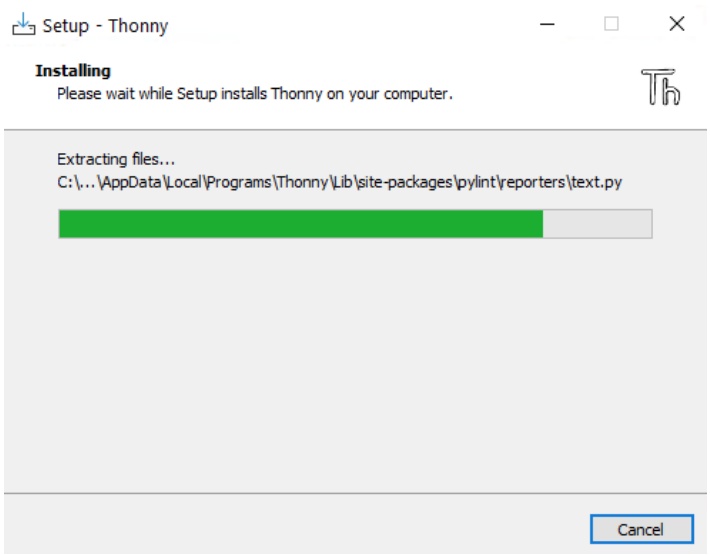
Instalator środowiska Thonny jest intuicyjny, zaś proces instalacji sprowadza się do zaakceptowania warunków licencji i klikania przycisku **Next**. Nie wymaga zatem wyjaśnień. Warto jedynie zaznaczyć -

Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Norway grants
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.



Rysunek 6.1: Instalacja Thonny - wybór instalacji

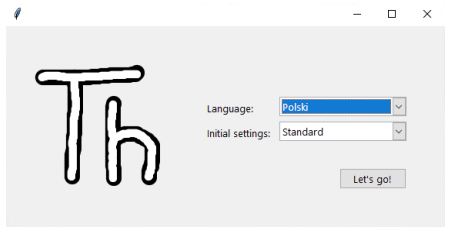
w jednym z etapów instalacji - pole wyboru utworzenia skrótu na pulpicie.



Rysunek 6.2: Proces instalacji Thonny

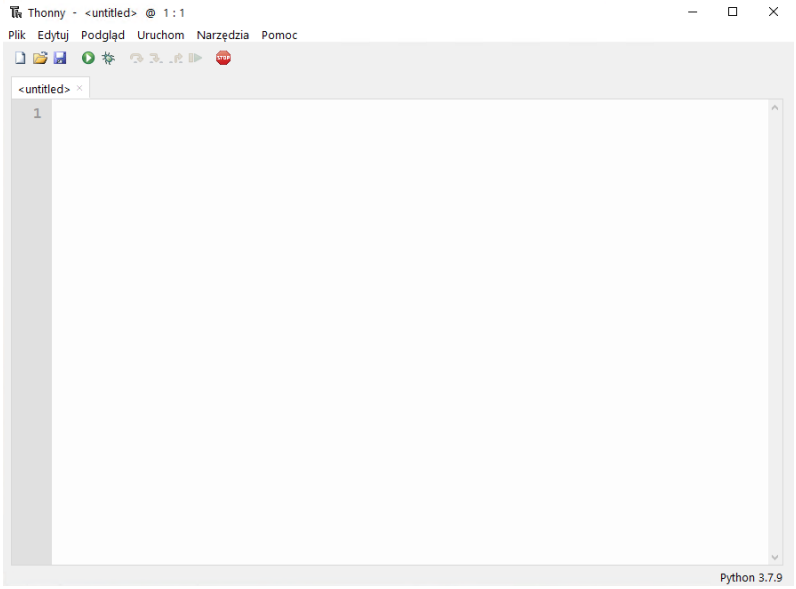
Po zainstalowaniu programu, podczas pierwszego uruchomienia, wyświetli się okno wyboru języka. Warto wybrać z menu język polski i kliknąć przycisk **Let's go!**

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.



Rysunek 6.3: Pierwsze uruchomienie Thonny - ustawienie języka

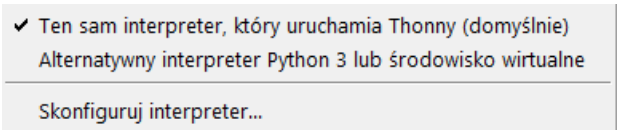
Na rysunku 6.4 widać uruchomione środowisko programistyczne gotowe do pracy. Jeśli przyjrzymy się bliżej (prawy, dolny róg okna), zauważymy wersję języka Python, który jest domyślnie używany. Dzieje się tak, ponieważ Thonny posiada własny interpreter języka Python, jednakże nie jest to najnowsza wersja. Warto jednak trzymać się zasady, by w miarę możliwości używać najnowszej jego wersji.



Rysunek 6.4: Środowisko Thonny gotowe do pracy

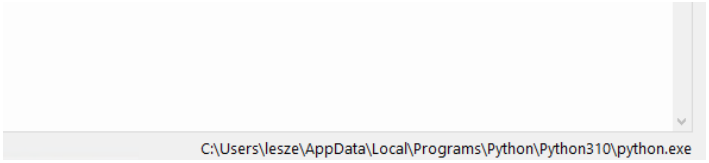
W poprzednim podrozdziale zainstalowaliśmy najnowszą wersję języka Python. Zatem, aby ją wykorzystać wystarczy przełączyć środowisko na nowszą wersję. W tym celu należy kliknąć w prawym dolnym obszarze okna - tam, gdzie znajduje się napis **Python 3.7.9**. Otworzy się dodatkowe menu (rysunek 6.5), z którego należy wybrać opcję **Alternatywny interpreter Python 3 lub środowisko wirtualne**.

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.



Rysunek 6.5: Zmiana wersji języka Python

Po wykonaniu tej czynności możemy być pewni, że pracujemy z najnowszą wersją języka Python (6.6) i nasze środowisko jest w pełni gotowe do programowania.

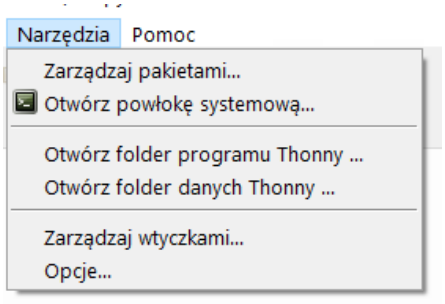


Rysunek 6.6: Poprawna wersja języka Python

6.4 Instalacja niezbędnych bibliotek

W naszych programach będziemy korzystać z biblioteki OpenCV (Open Source Computer Vision Library). Jest to biblioteka zawierająca zbiory procesów i algorytmów wykorzystywanych przy przetwarzaniu obrazów. Charakteryzuje się ona dużą wydajnością oraz prostotą wykorzystania. Biblioteka ta jest wieloplatformowa, co oznacza, że można z niej korzystać w dowolnym systemie operacyjnym.

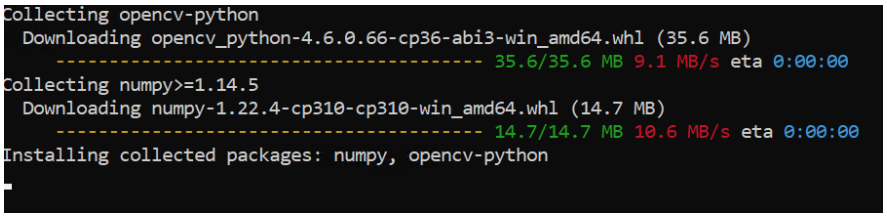
Aby zainstalować bibliotekę w systemie, należy uruchomić środowisko Thonny i z menu **Narzędzia** wybrać polecenie **Otwórz powłokę systemową**, co widać na rysunku 8.1.



Rysunek 6.7: Uruchamianie powłoki systemowej w środowisku Thonny.

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

Następnie w oknie powłoki systemowej należy zainstalować bibliotekę OpenCV poleceniem: **pip install opencv-python** <Enter>. Nastąpi proces pobierania i instalacji, który może potrwać w zależności od prędkości łącza internetowego (rysunek 6.8).



Rysunek 6.8: Instalacja bibliotek OpenCV.

Zainstalujemy także bibliotekę o nazwie Matplotlib, która jest przydatną biblioteką do wizualizacji danych. Aby zainstalować bibliotekę, w oknie powłoki systemowej należy wydać polecenie: **pip install matplotlib** <Enter>.

Podczas każdej instalacji instalowane są tzw. zależności, czyli inne biblioteki, które są niezbędne do działania OpenCV. Przykładem zależności jest w tym wypadku biblioteka Numpy, która implementuje szybkie tablice dla języka Python, ponieważ język ten w standardzie ich nie posiada. Na szczęście narzędzie do instalacji bibliotek o nazwie **pip** (Package Installer for Python) potrafi automatycznie instalować wszystkie zależności. Gdy instalacja zakończy się powodzeniem, proces instalacji środowiska i bibliotek jest zakończony.

6.5 Wykorzystanie biblioteki OpenCV

Każdy obraz zapisany w pamięci, który jest zgodny z OpenCV to wielka macierz, w której każdy piksel widoczny w obrazie reprezentowany jest trzema składowymi kolorów R-red (czerwony), G-green (zielony), B-blue (niebieski). Domyślnie format składowych koloru obrazu (channels) jest przechowywany w odwrotnej kolejności (BGR), co warto zapamiętać podczas pracy z biblioteką. Przyglądając się własności macierzy można zauważyć, że jej format to (H,W,C) [H-height, wysokość; W-width, szerokość; C-channels, kanały]. Każda składowa koloru (R,G,B) przyjmuje wartości w przedziale [0,255] i jest reprezentowana 8 bitową zmienną całkowitą.

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

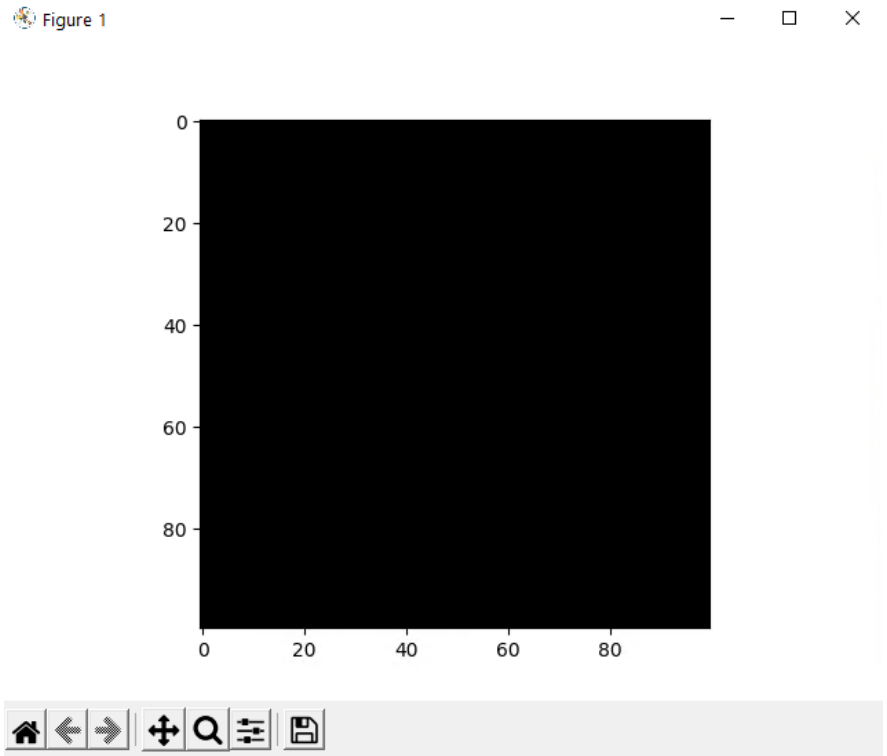
Praca z obrazkami

Obraz jest macierzą, zatem można ją wygenerować przy pomocy biblioteki Numpy przy wykorzystaniu funkcji **zeros**. Poniższy kod tworzy macierz o wysokości 100 i szerokości 100 punktów i wypełnia ją zerami.

```
import numpy as np
import matplotlib.pyplot as plt

img = np.zeros((100,100,3), dtype='uint8')

plt.imshow(img)
plt.show()
```

Listing 6.1: Tworzenie pustego obrazu

Powyższy kod wykorzystuje także bibliotekę Matplotlib do wyświetlenia utworzonej macierzy, która jest w rzeczywistości obrazkiem. Wyniku działania programu jest wyświetlenie czarnego prostokąta (100×100), co zostało zaprezentowane na rysunku 6.9.



Rysunek 6.9: Wynik działania programu

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

Teraz zajmiemy się otwarciem obrazu z dysku przy pomocy wbudowanej w bibliotekę funkcji **imread**. Jednakże najpierw przyjrzyjmy się rysunkowi 6.10. Obraz ten będzie stanowić wzorzec dla kolejnej operacji.

Rysunek 6.10: Obraz wzorcowy

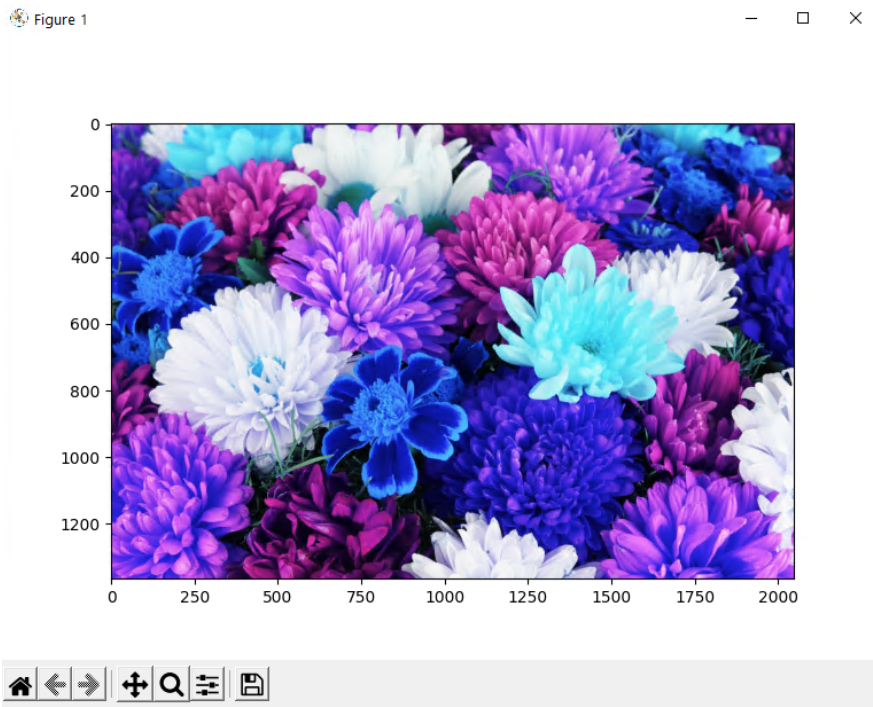
Poniższy kod odczytuje plik wzorcowy przy pomocy funkcji **imread**, a następnie wyświetla go przy pomocy biblioteki Matplotlib.

```
import numpy as np
import matplotlib.pyplot as plt

img = cv2.imread('flower01.jpeg')

plt.imshow(img)
plt.show()
```

Listing 6.2: Otwarcie obrazka z pliku

Wynik działania przykładu jest widoczny na rysunku 6.11. Jak widać, dalece odbiega on od obrazu wzorcowego, widocznego na rysunku 6.10. Wynika to właśnie z niekompatybilności kolejności kanałów, stąd należy je odwrócić.

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.



Rysunek 6.11: Obraz wzorcowy odczytany przy użyciu OpenCV

Odwrócenie kolejności kanałów można przeprowadzić na dwa sposoby. Pierwszym z nich jest odwrócenie kolejności w macierzy na ostatniej osi (osi kanałów) img = img[...,::-1], zaś drugim sposobem jest wykorzystanie wbudowanej funkcji konwertującej w OpenCV: cvtColor(obraz, cv2.COLOR_BGR2RGB). Rozbudowany kod programu wygląda teraz następująco:

```
import numpy as np
import matplotlib.pyplot as plt

img = cv2.imread('flower01.jpeg')

# pierwszy sposób
# img = img[...,::-1]

# drugi sposób
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

plt.imshow(img)
plt.show()
```

Listing 6.3: Otwarcie i konwersja obrazu

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

Tym razem obraz został poprawnie wyświetlony, co widać na rysunku 6.12.

Rysunek 6.12: Obraz skonwertowany prawidłowo

Wykrywanie krawędzi

Kontury to linie łączące wszystkie punkty wzdłuż granicy obrazu, które mają tę samą intensywność. Wykrywanie konturów przydaje się przy analizie kształtów, identyfikacji rozmiaru obiektów na obrazku czy wykrywania obiektów. Bibliotek OpenCV oferuje wiele funkcji wyodrębniania konturów. Popularnym algorytmem wykrywania krawędzi jest Canny, który został opracowany przez Johna F. Canny'ego. Jest to złożony algorytm wieloetapowy, lecz biblioteka OpenCV umożliwia bardzo jego bardzo proste wykorzystanie.

```
import cv2 as cv
from matplotlib import pyplot as plt

img = cv.imread('flower01.jpeg', 0)

edges = cv.Canny(img, 100, 200)

plt.subplot(121), plt.imshow(img, cmap = 'gray')
```

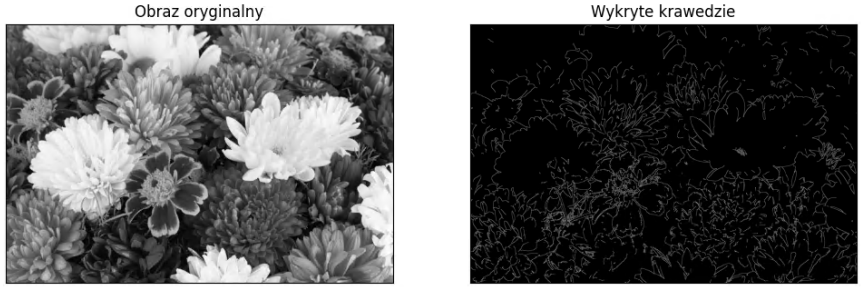
Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

```
plt.title('Obraz oryginalny'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(edges, cmap = 'gray')
plt.title('Wykryte krawędzie'), plt.xticks([]), plt.yticks
    ([])
plt.show()
```

Listing 6.4: Wykrywanie konturów

Program (z listingu powyżej) odczytuje obrazek przy pomocy funkcji imread. Warto zauważyć, że drugim argumentem funkcji jest tzw. flaga, która w tym przypadku jest ustawiona na wartość 0. Alternatywnie można także podać flagę cv2.IMREAD_GRAYSCALE. Flaga w tym wypadku oznacza, że odczytany obrazek od razu zostanie skonwertowany do skali szarości. Po przetworzeniu obrazu program wyświetli w jednym oknie obok siebie dwa obrazki - dzięki wykorzystaniu metody subplot(). Dodatkowo przy pomocy title wyświetlane są tytuły dla każdego obrazka z osobna.

Algorytm wykrywania konturów OpenCV, przedstawiony na powyższym przykładzie, realizuje jedna funkcja Canny(). Jako argumenty przyjmuje ona obraz wejściowy, zaś drugim i trzecim argumentem są odpowiednio wartość minimalna i maksymalna, czyli argumenty progowe. Wszelkie krawędzie z gradientem intensywności większym niż wartość maksymalna będą krawędziami, zaś te poniżej wartości minimalnej nie będą krawędziami, zatem algorytm je odrzuci. Efekt działania programu ilustruje rysunek 6.13.



Rysunek 6.13: Wykrywanie krawędzi

Odpowiednio, eksperymentując z argumentami funkcji Canny(), można dopasować dokładność wykrywania krawędzi w zależności od analizowanego obrazu i jego zawartości, doświetlenia oraz treści.

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

Detekcja twarzy

W tym podrozdziale zostanie zaprezentowana prosta metoda do detekcji twarzy na obrazku. Wykrywanie twarzy jest procesem skomplikowanym i wymagającym dużej mocy obliczeniowej. Na szczęście biblioteka OpenCV posiada wbudowane algorytmy uczenia maszynowego, które umożliwiają ich bardzo proste użycie. W konsekwencji napisanie programu rozpoznającego twarz, wymaga jedynie umiejętności ich użycia w praktyce.

Do rozpoznawania twarzy wykorzystany zostanie algorytm uczenia maszynowego - tzw. klasyfikator Haara. Jest to, oparta na funkcjach Haar, metoda wykrywania obiektów. Została ona zaproponowana przez Paula Violę i Michaela Jonesa w artykule „Rapid Object Detection using a Boosted Cascade of Simple Features" w 2001 roku. W dużym uproszczeniu jej działanie jest oparte na funkcji kaskadowej, która musi zostać wytrenowana przy pomocy wielu pozytywnych (zawierających twarze) i negatywnych (nie zawierających twarzy) obrazów. Dodatkowo należy wydobyć z nich pewne cechy. Proces trenowania nie jest łatwy dla początkującego programisty, dlatego biblioteka OpenCV zawiera wiele gotowych do wykorzystania klasyfikatorów. Można je pobrać z witryny https://github.com/opencv/ opencv/tree/master/data/haarcascades.

Jak widać, na liście plików znaleźć można wiele klasyfikatorów, zaś ich przeznaczenie jest ściśle określone. Do detekcji twarzy wykorzystamy klasyfikator o nazwie haarcascade_frontalface _default.xml. Będzie także potrzebne dowolne zdjęcie zawierające twarz. Przykładowe zdjęcie znajduje się na rysunku 6.14.

Rysunek 6.14: Obrazek przedstawiający przykładowe twarze

Program realizujący rozpoznawanie twarzy znajduje się na listingu poniżej. Po wczytaniu klasyfikatora za pomocą funkcji CascadeClas-

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

sifier, rozpoznanie następuje przy pomocy funkcji detectMultiScale, która przyjmuje kilka argumentów, z których pierwszym jest obrazek. W tym ćwiczeniu nie będą omówione dodatkowe argumenty, ponieważ ich omówienie musi zostać poprzedzone teorią dotyczącą klasyfikatorów Haar. Jeśli zostanie wykryta przynajmniej jedna twarz, funkcja zwróci ich pozycje jako Rect(x,y,w,h). Dzięki temu można wykorzystać je w funkcji **rectangle** do narysowania ramki wokół zidentyfikowanego obszaru obrazka.

```
import cv2

# Załadowanie klasyfikatora
face_cascade = cv2.CascadeClassifier('
    haarcascade_frontalface_default.xml')

# Odczyt obrazka z pliku
img = cv2.imread('faces.jpg')

# Konwersja do skali szarości
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

# Detekcja twarzy
faces = face_cascade.detectMultiScale(gray, 1.1, 4)

# Rysowanie obwódki wokół rozpoznanej twarzy
for (x, y, w, h) in faces:
    cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
    # Wyświetlenie obrazka
    cv2.imshow('img', img)

    # Oczekiwanie na wciśnięcie dowolnego klawisza
    cv2.waitKey()

# Usunięcie wszystkich okien z pamięci
cv2.destroyAllWindows()
```

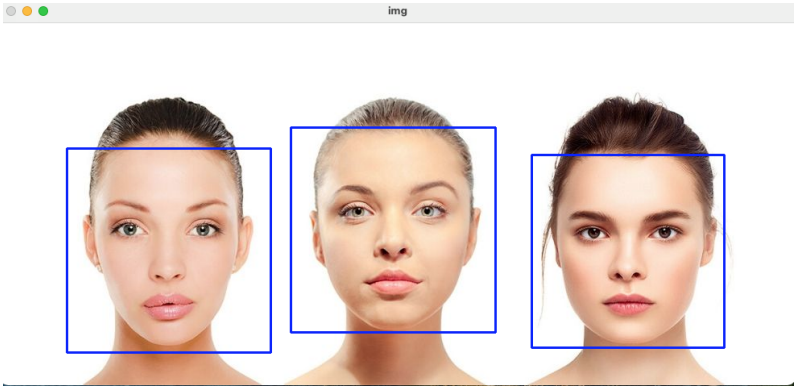
Listing 6.5: Detekcja twarzy

Jak widać na rysunku 6.15, program nakreślił ramki wokół zidentyfikowanych twarzy. Algorytm potrafi zidentyfikować wiele twarzy na obrazku.

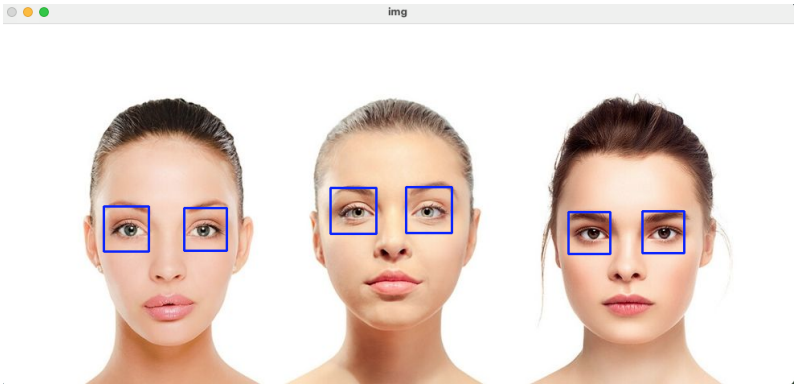
Jeśli zajdzie potrzeba identyfikacji wyłącznie oczu, można wykorzystać inny klasyfikator, np. o nazwie haarcascade_eye.xml. Należy zatem zmienić linię programu odpowiadającą za wczytanie klasyfikatora: face_cascade = cv2.CascadeClassifier('haarcascade_eye.xml') i ponownie uruchomić program. Może się okazać, że oprócz oczu, algorytm błędnie rozpozna dodatkowe obiekty, takie jak usta czy nozdrza. Wówczas należy zmodyfikować parametr detectMultiScale(gray, 1.1, **7**) i ponownie uruchomić program.

Jak widać na rysunku 6.16, detekcja oczu przebiegła prawidłowo. Warto podkreślić, że dysponując komputerem o odpowiednio wydajnym

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.



Rysunek 6.15: Obrazek przedstawiający rozpoznane twarze



Rysunek 6.16: Obrazek przedstawiający rozpoznane oczy

procesorze, w pojedynczym programie można użyć kilka klasyfikatorów. Dla przykładu można jednocześnie poszukiwać na obrazkach zarówno twarzy, jak i oczu czy innych części ciała.

6.6 Podsumowanie

Celem rozdziału było zapoznanie czytelnika z podstawowymi funkcjami biblioteki OpenCV do tworzenia systemów wizyjnych. Choć omówiono wyłącznie kilka algorytmów dostępnych w bibliotece, a zaprezentowane przykłady są krótkie i łatwe w zrozumieniu, wyniki ich działania można uznać jako imponujące. Warto zapoznać się bliżej z biblioteką i poznać inne algorytmy do różnych zastosowań (np. identyfikowania obiektów, klasyfikowania ludzkich działań, śledzenia ruchów kamery, śledzenia poruszających się obiektów itp.). Biblioteka OpenCV znajduje zastoso-

Norway grants
Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).
Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.

wanie w wielu urządzeniach i jest używana przez firmy, grupy badawcze, ale także przez organy rządowe.

Norway grants

Projekt pt.: „MODELOWE ROZWIĄZANIA NA TRUDNE WYZWANIA – Plan Rozwoju Lokalnego i Instytucjonalnego Stalowej Woli" współfinansowany jest ze środków Norweskiego Mechanizmu Finansowego 2014-2021 (85%) oraz Budżetu Państwa (15%).

Wspólnie działamy na rzecz Europy zielonej, konkurencyjnej i sprzyjającej integracji społecznej.