Przetwarzanie języka naturalnego

System do odczytywania hieroglifów przy pomocy biblioteki scikit-image

RS

1. Cel pracy:

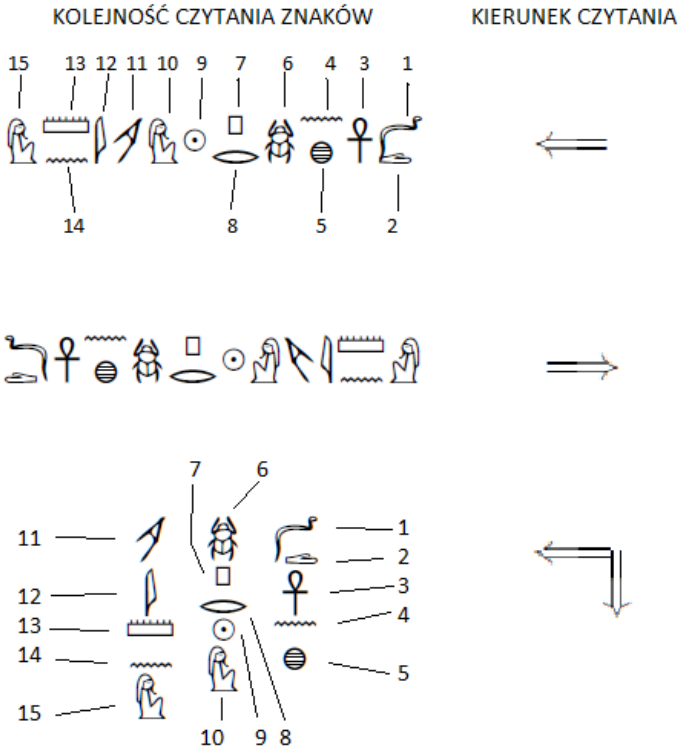
Celem pracy jest projekt i implementacja programu umożliwiającego odczyt i tłumaczenie egipskich hieroglifów z obrazów, przy pomocy biblioteki sckit-image z języka Python.

2. Opis problemu:

Odczyt egipskich hieroglifów nie jest łatwą sprawą z powodu wielu czynników, wśród których można wymienić:

- Kolejność czytania
- Wizja artystyczna
- Kontekst: fonogramy i ideogramy
- Słaba czytelność hieroglifów w świecie rzeczywistym

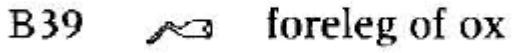
Nie można łatwo zaprogramować kolejności sczytywania hieroglifów z obrazu, ponieważ zdania w tym systemie zapisu można czytać w różnych kierunkach, tak jak to pokazano na Rysunek 1.



Rysunek 1.Kierunki czytania egipskich hieroglifów (Wikipedia, 2023)

Dodatkowo autor hieroglifu mógł przedstawić dany znak w różnej wielkości, lub wkomponować go w inny by zrealizować pewną wizję artystyczną. Widać to na Rysunek 2, gdzie człowiek i noga zwierzęcia którą trzyma, są większe niż pozostałe hieroglify. Ponadto sama noga zwierzęcia jest hieroglifem, skatalogowanym jako B39 wg. (Collier i Manley, 1998), Rysunek 3.

Rysunek 2.Scena ofiarowania z grobowca Senbi z Meir (Collier i Manley, 1998)



Rysunek 3.Hieroglif oznaczające nogę woła

Hieroglify mogą być ideogramami, jak wspomniana wcześniej noga woła, lub fonogramami, których złożenie może przedstawiać np. imię. Na Rysunek 2 można zauważyć blok składający się z czterech hieroglifów, które odpowiadają imieniu Senbi.

Rysunek 4.Imię Senbi zapisane hieroglifami jako fonogramami

Problemem przy odczycie hieroglifów może być także ich czytelność. Na Rysunek 5 można zauważyć inskrypcję, która została pokazana w przybliżeniu na Rysunek 6. Pojawiają się na niej hieroglify oznaczające wspominane wcześniej imię Senbi. Jednak znaki w tym przypadku są bardzo uproszczone względem tych z grobowca a także podniszczone przez upływ czasu.

Rysunek 5. Statuetka zarządcy Senbi (metmuseum.org, 2023)

Rysunek 6.Inskrypca na statuetce zarządcy Senbi (metmuseum.org, 2023)

3. Opis działania i implementacja

Z powodu problemów z odczytem hieroglifów przedstawionych w podpunkcie 2, zdecydowano się na projekt programu, odczytującego pojedyncze hieroglify. W niektórych przypadkach, odczytywane obrazy zostaną poddane prostej obróbce w programie graficznym, by polepszyć działanie systemu. Do takich działań można zaliczyć na przykład obrót obrazu w poziomie, by uwzględnić kierunek czytania hieroglifów. Program będzie napisany w języku Python, do przetwarzania obrazu wykorzystano bibliotekę sckit-image, natomiast od rozpoznawania hieroglifów wykorzystano sieć neuronową keras z biblioteki tensorflow.

Obrazy to trenowania sieci neuronowej zostały skopiowane z książki (Collier i Manley, 1998) oraz ze strony internetowej (Wikipedia, 2023). Udało się zebrać około 50 obrazów, które zostały nazwane w konwencji fonem_znaczenie.jpg. Przykład obrazu użytego do trenowania sieci neuronowej został pokazany na Rysunek 7.

Rysunek 7.Przykład obrazu uzytego do trenowania sieci neruonowej o nazwie šnj_hari.jpg

Obrazy poddano standaryzacji przy pomocy biblioteki sckit-image poprzez następujące kroki:

- Konwersja kolorów do odcieni szarości metodą „rgb2gray”
- Progowanie metodą „threshold_otsu”
- Usuwanie szumów metodą „remove_small_objects”

Następnie przy pomocy biblioteki NumPy skonwertowano obrazy do postaci tablicy o stałym rozmiarze.

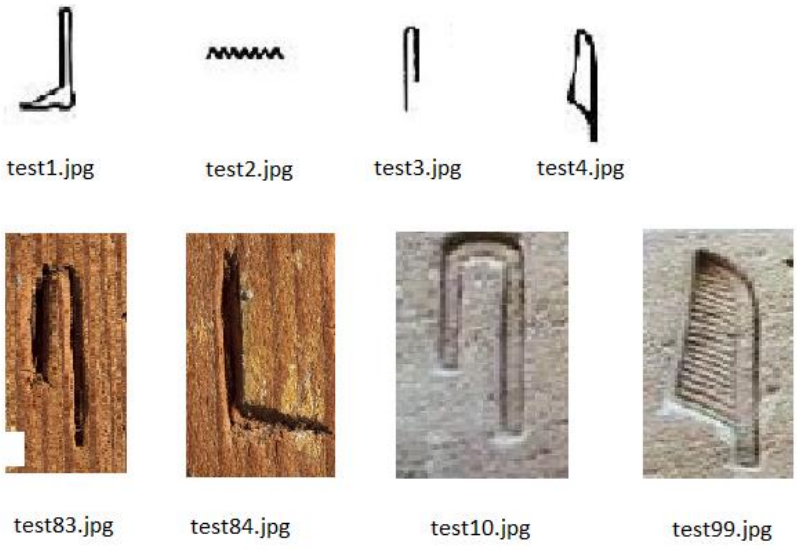
Gotowe obrazy posłużyły do trenowania sieci neuronowej, do którego modelowania wykorzystano metody „Conv2D”, „Dense”, „MaxPooling2D”, „DropOut” oraz „Flatten”. Wynik trenowania sieci został przedstawiony na Rysunek 8.

```
Epoch 55/60
10/10 [==============================] - 0s 10ms/step - loss: 0.0863 - accuracy: 0.9750
Epoch 56/60
10/10 [==============================] - 0s 10ms/step - loss: 0.0557 - accuracy: 0.9750
Epoch 57/60
10/10 [==============================] - 0s 10ms/step - loss: 0.1395 - accuracy: 0.9000
Epoch 58/60
10/10 [==============================] - 0s 11ms/step - loss: 0.1120 - accuracy: 0.9750
Epoch 59/60
10/10 [==============================] - 0s 10ms/step - loss: 0.2420 - accuracy: 0.8750
Epoch 60/60
10/10 [==============================] - 0s 10ms/step - loss: 0.0825 - accuracy: 0.9500
```

Rysunek 8.Wynik trenowania sieci neuronowej

4. Testowanie programu

Program następnie poddano testom, by ocenić w jakim stopniu udało się zrealizować cele projektu. Do testów wybrano osiem obrazów: 4 które występowały w zbiorze uczącym, oraz 4 nowe reprezentujące możliwość odczytu hieroglifów z rzeczywistych zdjęć. Zbiór testowych obrazów przedstawiono na Rysunek 9.



Rysunek 9.Obrazy wykorzystane do sprawdzenia możliwości zaimplementowanego systemu

Wywołanie testów oraz wyniki przedstawiono na Rysunek 10,Rysunek 11.

```
print("should be b")
readHiero("test1.jpg")
print("should be n")
readHiero("test2.jpg")
print("should be s")
readHiero("test3.jpg")
print("should be y")
readHiero("test4.jpg")
print("should be s")
readHiero("test83.jpg")
print("should be y")
readHiero("test99.jpg")
print("should be b")
readHiero("test84.jpg")
print("should be s")
readHiero("test10.jpg")
```

Rysunek 10.Wywołanie funkcji testujących

```
should be b
1/1 [==============================] - 0s 87ms/step
Predicted hieroglyph:  b_x
should be n
1/1 [==============================] - 0s 21ms/step
Predicted hieroglyph:  n_x
should be s
1/1 [==============================] - 0s 22ms/step
Predicted hieroglyph:  s_x
should be y
1/1 [==============================] - 0s 19ms/step
Predicted hieroglyph:  y_x
should be s
1/1 [==============================] - 0s 90ms/step
Predicted hieroglyph:  s_x
should be y
1/1 [==============================] - 0s 63ms/step
Predicted hieroglyph:  y_x
should be b
1/1 [==============================] - 0s 21ms/step
Predicted hieroglyph:  b_x
should be s
1/1 [==============================] - 0s 32ms/step
Predicted hieroglyph:  s_x

Process finished with exit code 0
```

Rysunek 11.Wynik odczytu hieroglifów dla danych testowych

Jak można zauważyć po powyższych rysunkach, program jest w stanie odczytać hieroglify ze zdjęć przedstawiających prawdziwe hieroglify. Zazwyczaj jednak dokładność nie jest stuprocentowa. Obrazy użyte do trenowania, odczytywane są poprawnie zawsze, te o nazwach test83.jpg i test84.jpg są odczytywane bardzo często poprawnie. Gorzej ma się sprawa z obrazami test10.jpg i test99.jpg. Może to być spowodowane złą jakością zdjęć, i niewystarczającym kontrastem pomiędzy samym znakiem a tłem.

5. Podsumowanie

W ramach projektu wykonano program tłumaczący pojedyncze hieroglify ze zdjęć. Skuteczność programu jest dobra jednak nie doskonała, co skłania nad zastanowieniem się nad możliwymi modyfikacjami polepszającymi jego działanie. Wyższa jakość obrazów, służących jako dane wejściowe na pewno usprawniłoby działanie programu. Można by w tym celu wykorzystać hieroglify dostępne w systemach komputerowych pod postacią znaków Unicode. Dodatkowo, dalsza obróbka zdjęć przy pomocy biblioteki sckit-image lub inny model sieci neuronowej mogłaby wpłynąć pozytywnie, na większą dokładność odczytywanych hieroglifów.

Bibliografia

Collier, M. i Manley, B. (1998). *How to Read Egyptian Hieroglyphs: A Step by Step Guide to Teach Yourself.* London: British Museum Press.

metmuseum.org. (2023). Pobrano z lokalizacji https://www.metmuseum.org/art/collection/search/545477

Wikipedia. (2023). Pobrano z lokalizacji https://pl.wikipedia.org/wiki/Pismo_hieroglificzne

Wikipedia. (2023). Pobrano z lokalizacji https://en.wikipedia.org/wiki/List_of_Egyptian_hieroglyphs