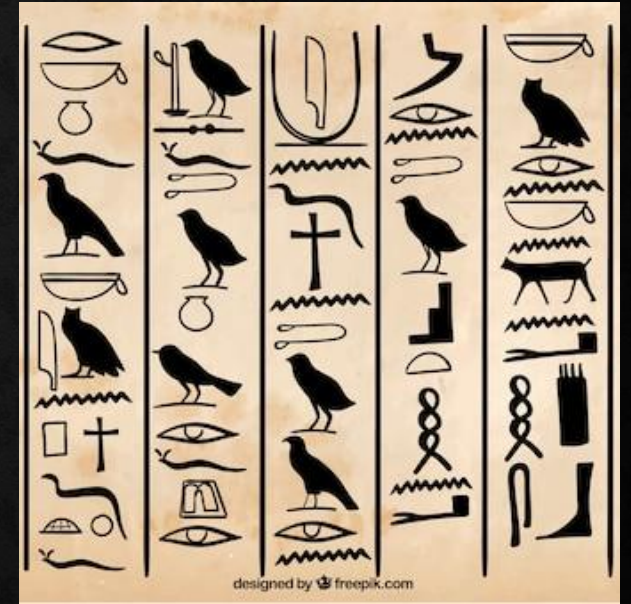


ROZPOZNAWANIE I TŁUMACZENIE HIEROGLIFÓW



Autorzy:

- TR
- MS
- JS

Politechnika Krakowska im. Tadeusza Kościuszki
Wydział Informatyki i Telekomunikacji
Kierunek: Informatyka
Semestr: VIII

CEL PROJEKTU:

Skonstruowanie systemu do odczytywania hieroglifów przy użyciu biblioteki scikit-image.



ZAKRES PROJEKTU

- Wczytywanie i przetwarzanie obrazów
- Model regresji logistycznej ze zbiorem danych do rozpoznawania hieroglifów
- Interfejs użytkownika z wyborem kierunku odczytu znaków
- Odczyt znaków z obrazka, wizualizacja wyników i otrzymanego tłumaczenia danego tekstu



CZYM SĄ HIEROGLIFY

- Hieroglify - „święte znaki”
- Najstarszy rodzaj pisma starożytnego Egiptu
- Każdy znak posiada swoje lustrzane odbicie
- Liczby zapisywane są w systemie dziesiętnym niepozycyjnym



JAK CZYTAMY HIEROGLIFY

Jeżeli znaki skierowane są w lewą stronę,
to czytamy od lewej do prawej.



Jeżeli znaki skierowane są w prawą stronę,
to prawej od prawej do lewej.



Jeżeli znaki są pionowe, to czytamy z góry na dół z
zachowaniem wcześniejszych zasad kierunków.



JAK CZYTAMY HIEROGLIFY

Pomimo że hieroglify mają swoje znaczenie, spora część z nich służy do zapisu wyrazów w taki sposób, w jaki my używamy liter. Różnica polega na tym, iż poszczególne glify mogą odpowiadać za zapis więcej niż jednej głoski.

Hieroglify działają na podobnej zasadzie co pisma abdżadowe, czyli nie oznaczane są samogłoski. Przyjmuje się, że dwie spółgłoski domyślnie rozdziela się głoską 'e'.

TRANSLITERACJA

Nasz program używa transliteracji odpowiadającej polskiej wymowie poszczególnych znaków w oparciu o tabelę z książki [1] oraz opisu listy znaków Gardinera [2].

SIGN	TRANS-LIT.	SAY	SIGN	TRANS-LIT.	SAY	SIGN	TRANS-LIT.	SAY
	<i>a</i>	a		<i>m</i>	m		<i>ś</i>	sh
	<i>i</i>	i/a		<i>n</i>	n		<i>k</i>	k
or	<i>y</i>	y		<i>r</i>	r		<i>k</i>	k
	<i>a</i>	a		<i>h</i>	h		<i>g</i>	g
or	<i>w</i>	w/u		<i>h</i>	h		<i>t</i>	t
	<i>b</i>	b		<i>h</i>	kh		<i>t</i>	tj
	<i>p</i>	p		<i>h</i>	kh		<i>d</i>	d
	<i>f</i>	f	or	<i>s</i>	s		<i>d</i>	j

JAK TŁUMACZYM Y HIEROGLIFY

Lista znaków Gardinera

D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
D11	D12	D13	D14	D15	D16	D17	D18	D19	D20
D21	D22	D23	D24	D25	D26	D27	D28	D29	D30
D31	D32	D33	D34	D35	D36	D37	D38	D39	D40
D41	D42	D43	D44	D45	D46	D47	D48	D49	D50
D51	D52	D53	D54	D55	D56	D57	D58	D59	D60
D61	D62	D63	D64	D65	D66	D67	D68	D69	D70

- Znak Gardinera ==> Glif (UTF-8)

W11 -

- Znak Gardinera ==> Litera (UTF-8)

W11 - G

- Glif (UTF-8) ==> Litera (UTF-8)

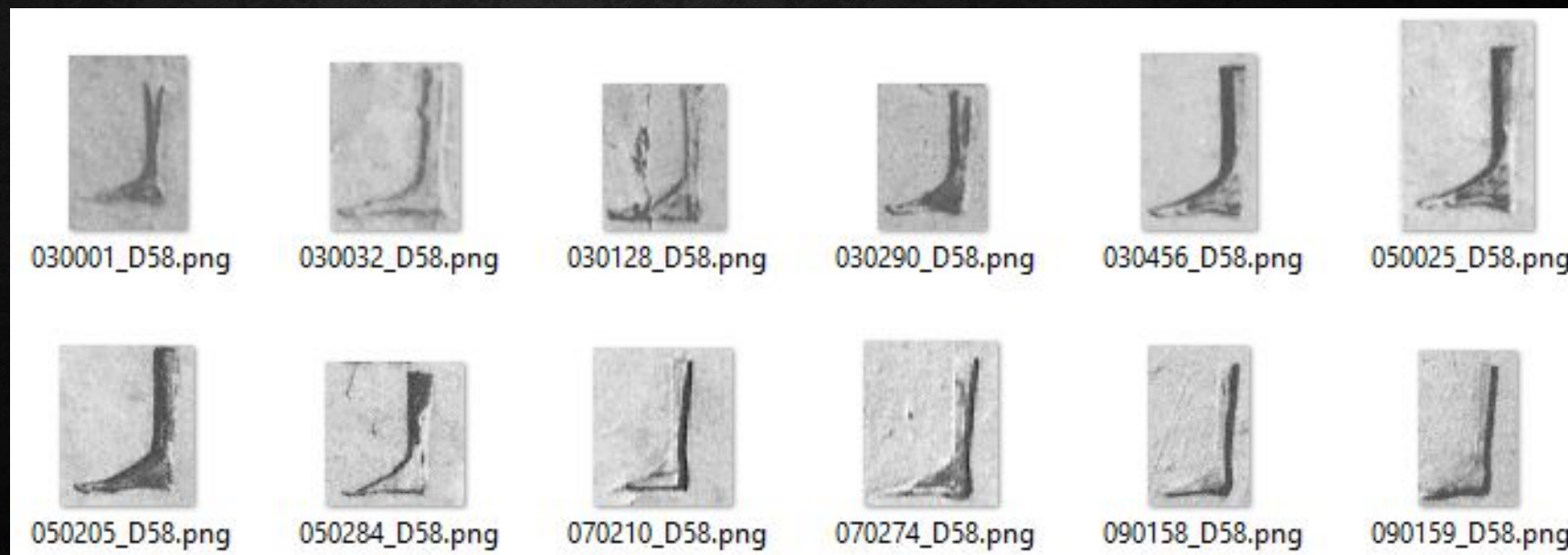
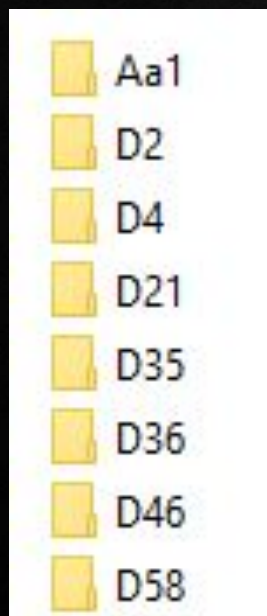
-

G

ZBIÓR UCZĄCY

Znaczącą część zbioru uczącego stanowiły sklasyfikowane już hieroglify z repozytorium [3]. Uzupełniliśmy brakujące znaki o odpowiednio przetworzone glify wycięte ze zdjęć znalezionych na Wikipedii w sekcjach poszczególnych glifów.

Dane zostały posortowane do folderów odpowiadających nazwom znaków, przykładowe fragmenty:



PRZYKŁADOWE PRZETWARZANIE DANYCH

```
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state=0, k_neighbors=smallest_class_size - 1)
sm_x, sm_y = sm.fit_resample(x_train, y_train)
```

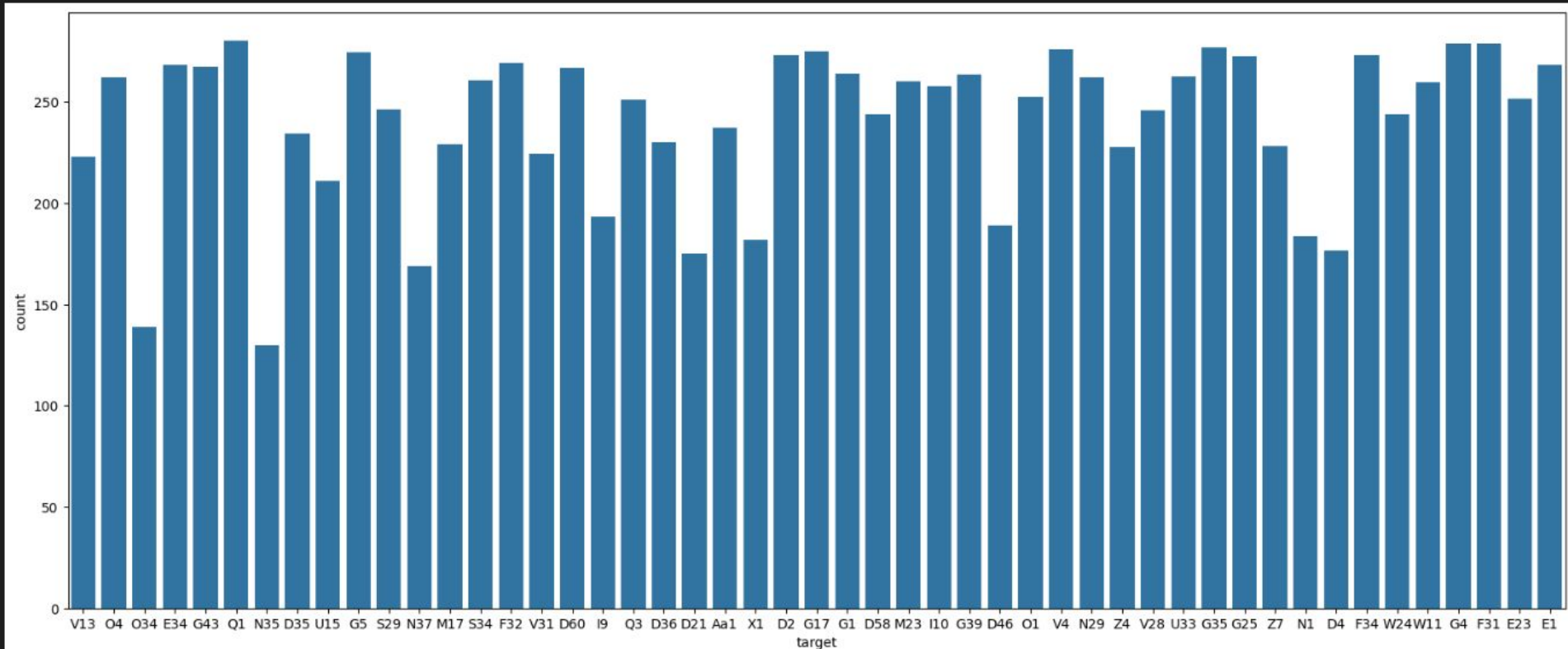
[29] ✓ 1.7s

```
bal_df = pd.DataFrame(sm_x)
bal_df['target'] = pd.DataFrame(sm_y)
plt.figure(figsize=(20, 8))
sns.countplot(x='target', data=bal_df)
```

[30] ✓ 0.9s

... <Axes: xlabel='target', ylabel='count'>

<>



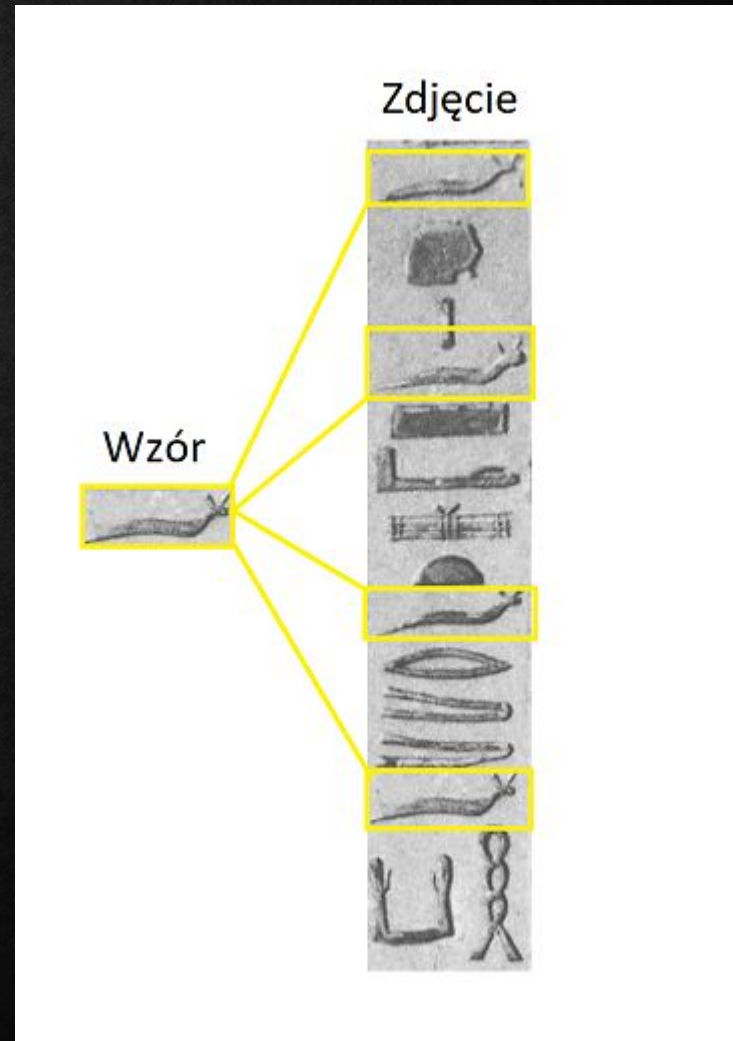
ROZPOZNAWANIE HIEROGLIFÓW – BIBLIOTEKA SCIKIT-IMAGE

- Biblioteka zawiera wiele funkcji umożliwiających przetwarzanie obrazów.
- W projekcie zostały wykorzystane głównie funkcje pozwalające na redukcję szumów, zwiększenie kontrastów i wykrywanie krawędzi.
- Za rozpoznawanie hieroglifów odpowiadają funkcje `match_template` oraz `peak_local_max`.



FUNKCJA MATCH_TEMPLATE

- Funkcja na głównym zdjęciu poszukuje miejsc, które są podobne do wprowadzonego wzoru.
- Funkcja korzysta z normalizacji korelacji krzyżowej.
- Zakres wyników zwracanych przez funkcję zawiera się w przedziale od -1 do 1.
- Do filtrowania najlepszych wyników wykorzystujemy funkcję `peak_local_max`.
- Program porównuje po kolei wszystkie zdjęcia ze zbioru znaków i wybiera te z najlepszymi współczynnikami korelacji.



PRZETWARZANIE ZDJĘĆ

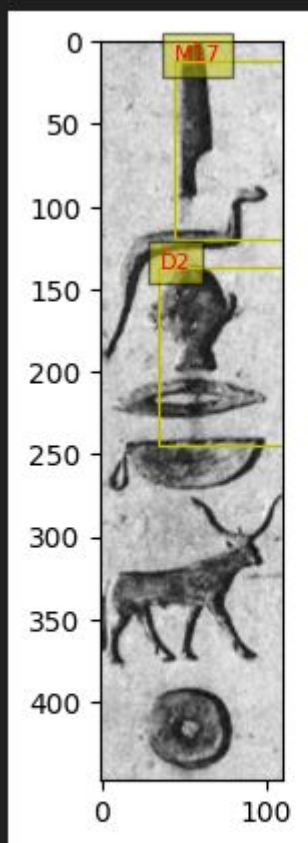
W ramach projektu zostały podjęte próby optymalizacji otrzymywanych wyników poprzez odpowiednie przetwarzanie zdjęcia wejściowego i zdjęć wzorów.

Przeprowadzane operacje:

- Wyrównywanie histogramu - poprawa kontrastu obrazu
- Wykrywanie krawędzi
- Progowanie obrazu
- Usuwanie szumów

Najlepsze wyniki otrzymano na nieprzetworzonym zdjęciu.

WYNIKI DZIAŁAJĄCEGO PROGRAMU

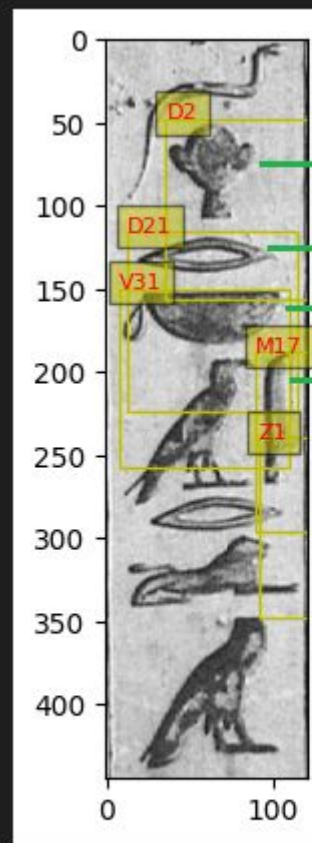


M17 - i

D2- face

['D2', 'M17']

2/7



D2 - Face

D21 - r

V31 - k

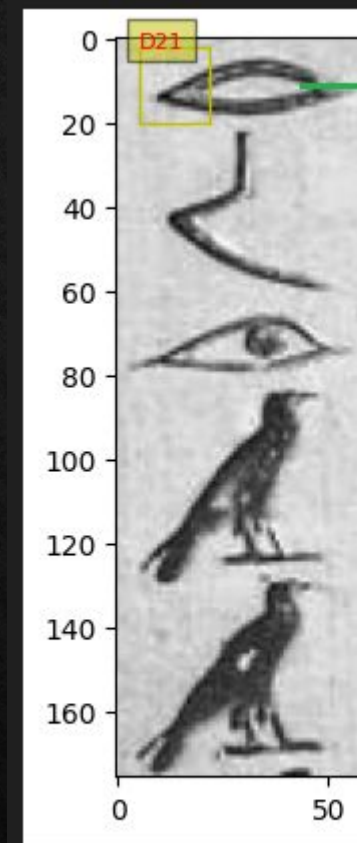
M17 - i

4/9

Z1 - Wskazuje, że znak poprzedzający jest ideogramem

1/9

['D2', 'D21', 'M17', 'V31', 'Z1']

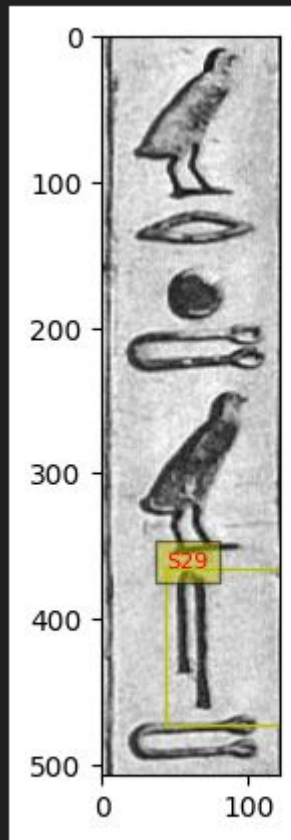


D21- r

1/5

['D21']

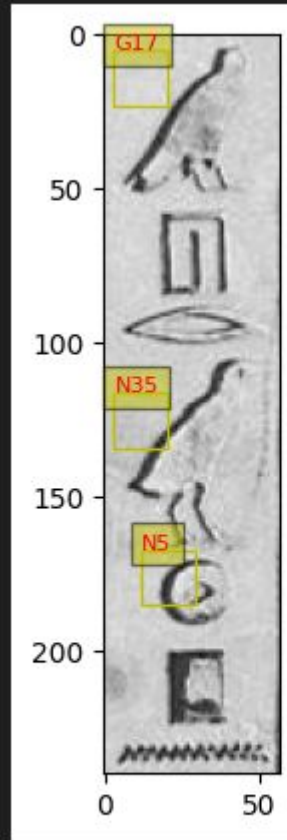
PRAWDOPODOBIEŃSTWO ROZPOZNANIA ZNAKU TO 20-30%*



S29 - s

1/7

['S29']



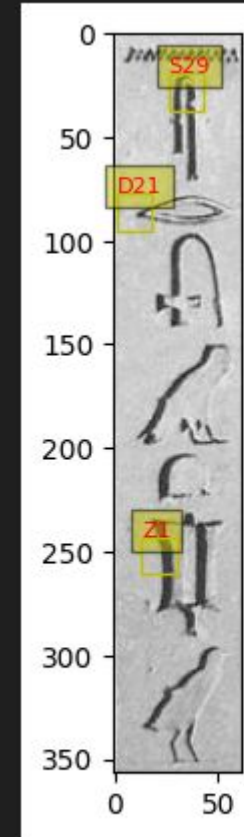
G17 - m

N35 - n

N5 - sun

3/7

['G17', 'N35', 'N5']



S29 - s

D21 - r

2/8

Z1 - Wskazuje, że
znak poprzedzający
jest ideogramem

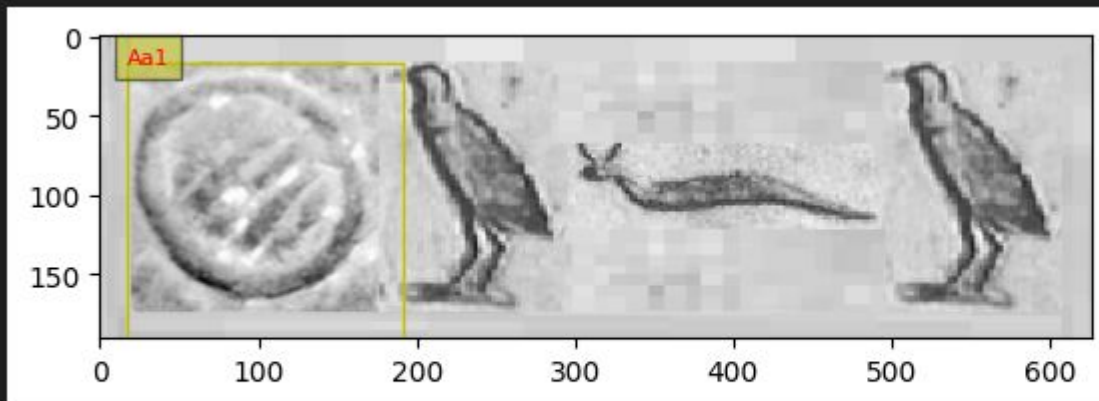
1/8

Z1



['D21', 'S29', 'Z1']

ZDJĘCIE ZŁOŻONE Z POJEDYNCZYCH ELEMENTÓW ZBIORU



Aa1

1/4

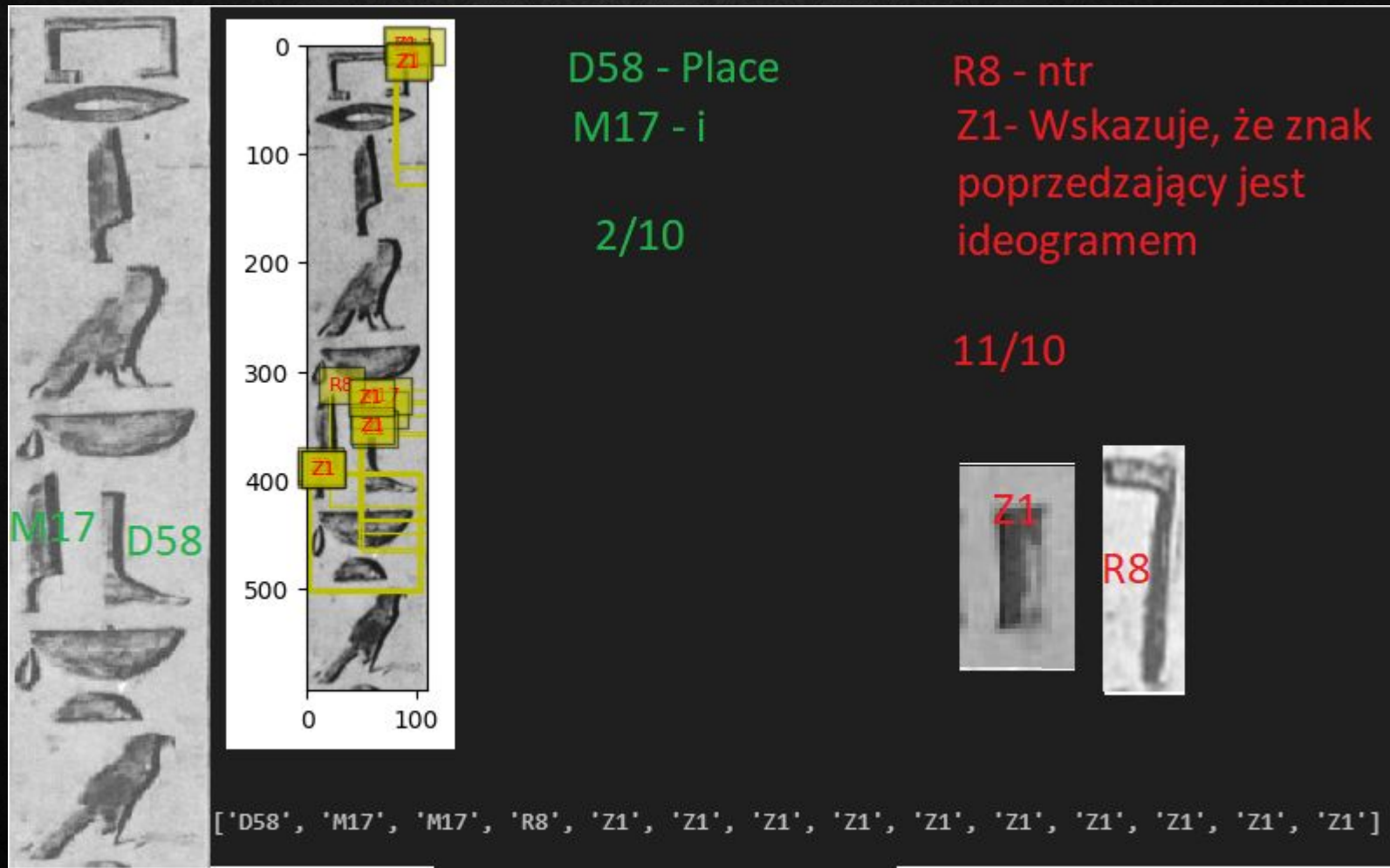
['Aa1']

PROBLEMY Z WYSZUKIWANIEM

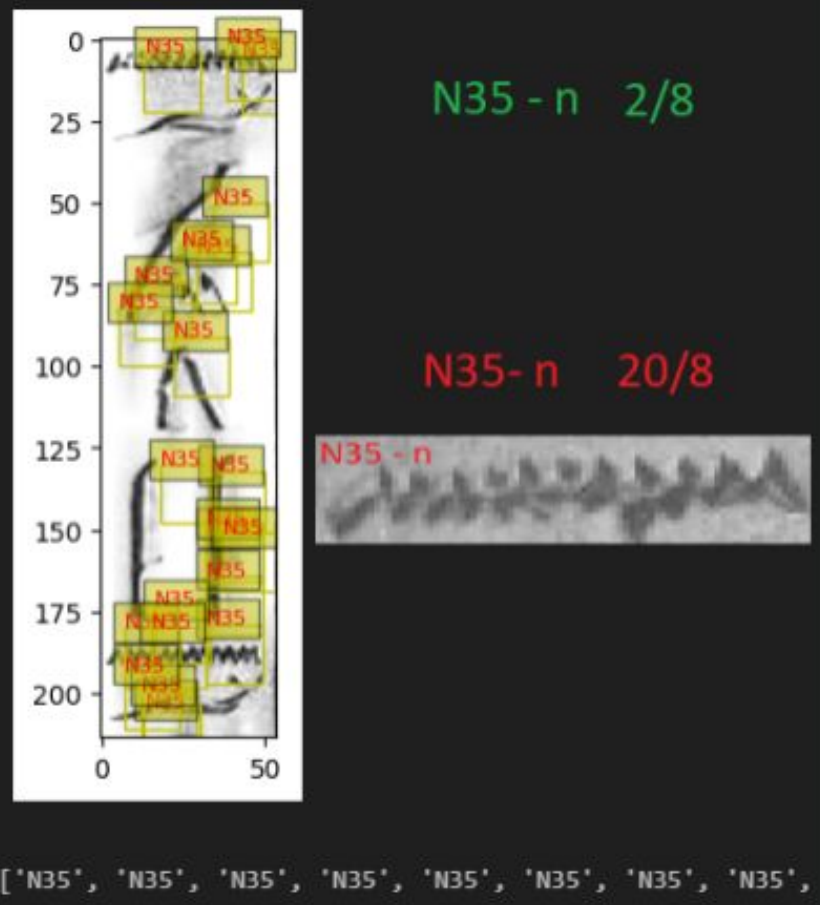
- Nieprawidłowa rozdzielczość zdjęcia głównego – za duża lub za mała.
- Wielokrotne dopasowanie tego samego znaku.
- Dopasowanie znaku wewnątrz innego znaku.
- Wyniki fałszywie pozytywne, w szczególności dla znaków : Z1, N35



WYNIKI FAŁSZYWIE POZYTYWNE



NIEPRAWIDŁOWA ROZDZIELCZOŚĆ ZDJĘCIA GŁÓWNEGO



PODSUMOWANIE

- Przeciętny wynik 20-30%.
- Najlepszy wynik 80%.
- Najgorszy wynik 0%.
- Niewłaściwa rozdzielczość prowadzi do dużej ilości fałszywie pozytywnych wyników.
- Najlepsze wyniki uzyskano na zdjęciach bez dodatkowych filtrów.

WNIOSKI

- Biblioteka scikit-image i zawarta w niej funkcja match_template nie nadaje się do rozpoznawania hieroglifów ze względu na dużą ilość błędów przy rozpoznawaniu.
- Wyszukiwanie za pomocą tej funkcji zużywa sporo zasobów.
- Stosowanie obróbki zdjęć nie zwiększa znacząco prawdopodobieństwa znalezienia wzoru.
- System można ulepszyć poprzez odpowiednie skalowanie wzorów lecz wiąże się to z wielokrotnym wydłużeniem czasu wyszukiwania.

ROZPOZNAWANIE HIEROGLIFÓW – BIBLIOTEKA OPEN-CV

- Biblioteka zawiera bardzo wiele funkcji umożliwiających przetwarzanie obrazów.
- W projekcie zostały wykorzystane różne przetworzenia obrazu, skala szarości, poprawa kontrastu, filtr Gaussowski, wykrywanie krawędzi, operacje morfologiczne, obliczenie parametrów wydzielonych histogramów.



PRZETWARZANIE ZDJĘĆ

W ramach projektu zostały podjęte próby optymalizacji otrzymywanych wyników poprzez odpowiednie przetwarzanie zdjęcia wejściowego i zdjęć wzorów. Przeprowadzane operacje:

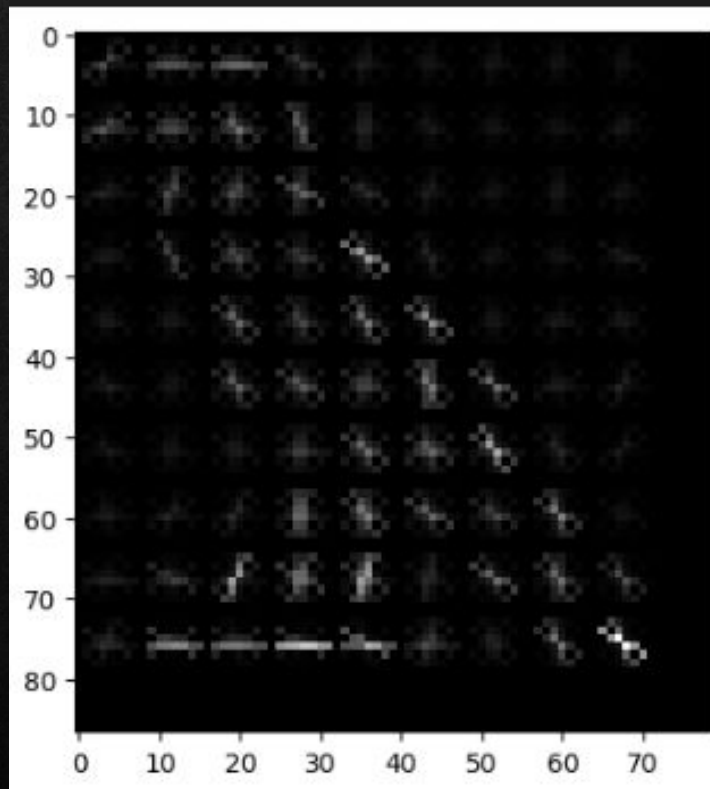
- Konwersja na skalę szarości
- Równanie histogramu: Poprawia kontrast obrazu w skali szarości.
- Filtr Gaussa: Redukuje szумы o wysokiej częstotliwości.
- Wykrywanie krawędzi metodą Canny'ego
- Operacje morfologiczne: Zamyka luki w wykrytych krawędziach za pomocą prostokątnego elementu strukturalnego.
- Obliczenie parametrów Histogramu Gradientów dla każdego wyznaczonego prostokąta.

PRZYKŁADOWY HISTOGRAM GRADIENTÓW DLA LITERY A - ZNAK G1

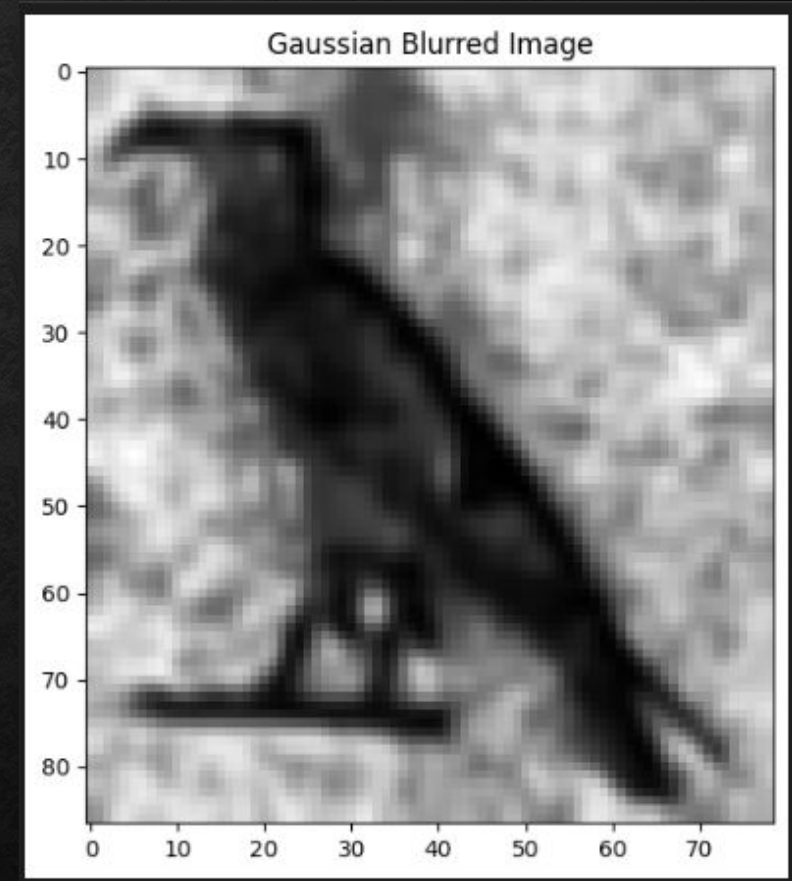
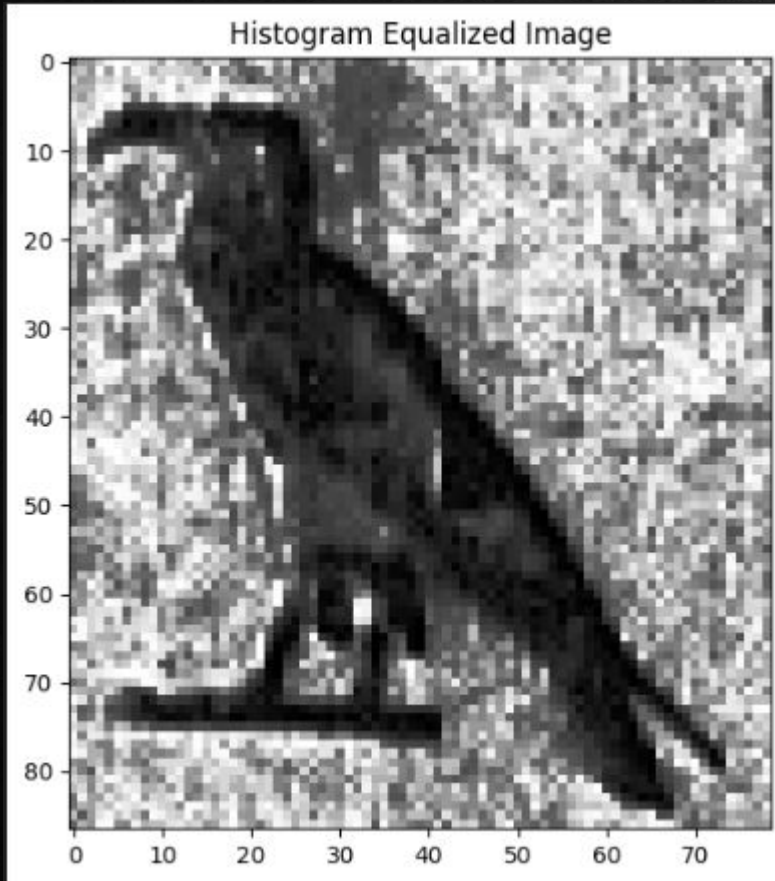
```
im_test = cv2.imread('data/train/G1/070410_G1.png',0)
_,hog_img= hog(im_test,orientations=9,pixels_per_cell=(8,8), cells_per_block=(1, 1),visualize=True)
plt.imshow(hog_img,cmap='gray')
```

✓ 0.4s

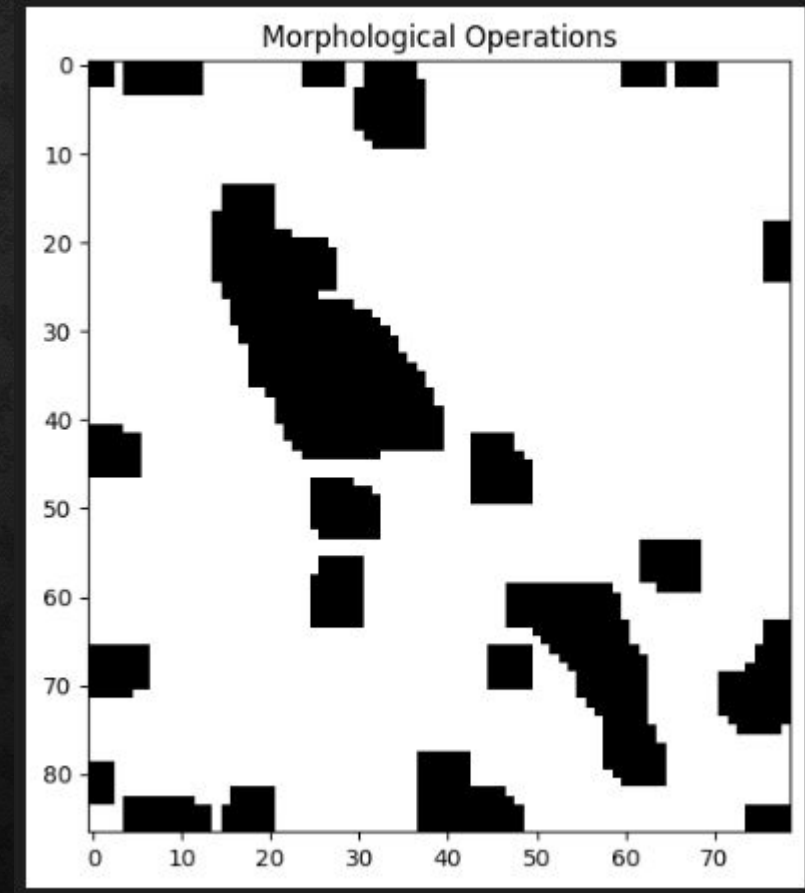
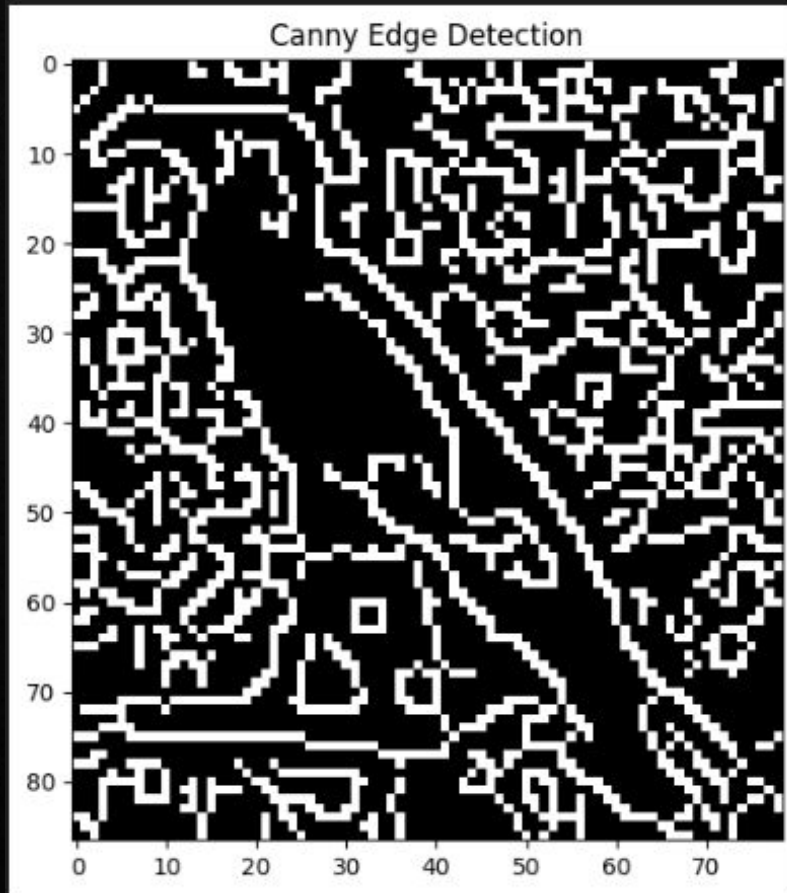
<matplotlib.image.AxesImage at 0x2a9a3ea6450>



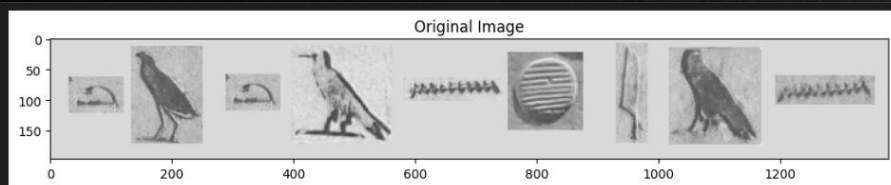
SKALA SZAROŚCI, RÓWNANIE HISTOGRAMU I ROZMYCIE GAUSSOWSKIE



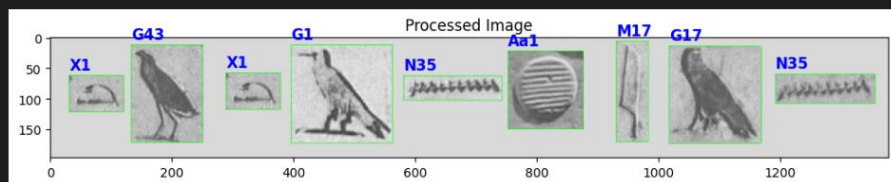
WYKRYWANIE KRAWĘDZI METODĄ CANNYEGO I OPERACJE MORFOLOGICZNE



WYNIKI DZIAŁAJĄCEGO PROGRAMU



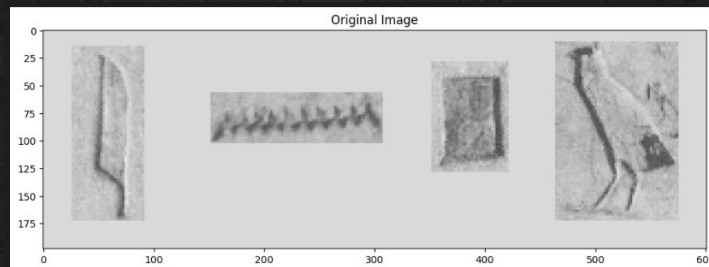
Results:



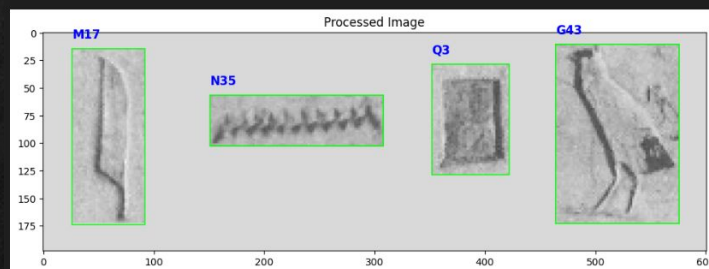
Word:

tutanchimn

Hieroglyphic:



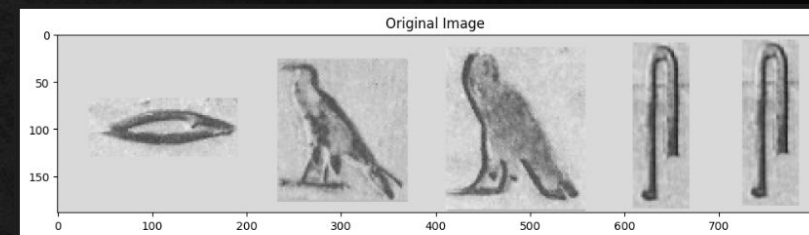
Results:



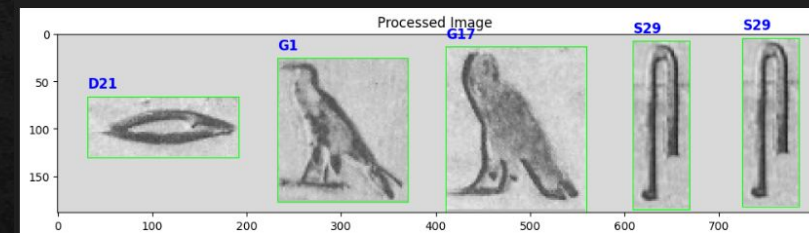
Word:

inpu

Hieroglyphic:



Results:



Word:

ramss

Hieroglyphic:



PRAWDOPODOBIENSTWO ROZPOZNANIA ZNAKU TO 90-100%

Hieroglyph Reader

h min: 16 w min: 16
ratio min: 0.2 ratio max: 5
Direction from left
Change

Image Path: D:/Hieroglyph-Reader-23.01/data/words/ramses.png Open

Recognize

D21 G1 G17 S29 S29

ramss

Hieroglyph Reader

h min: 16 w min: 16
ratio min: 0.2 ratio max: 5
Direction from right
Change

Image Path: D:/Hieroglyph-Reader-23.01/data/words/ramses.png Open

Recognize

S29 S29 G17 G1 D21

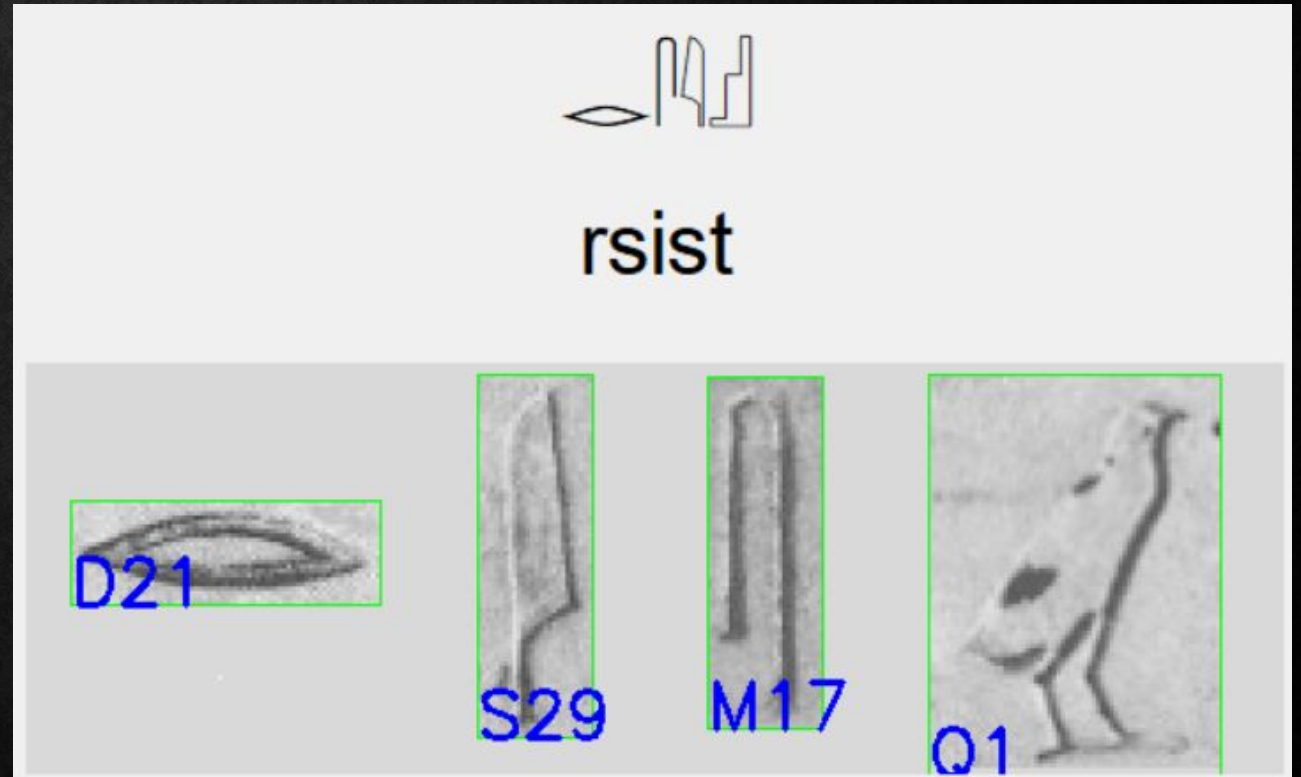
ramss

PROBLEMY Z WYSZUKIWANIEM

- Nieprawidłowa rozdzielczość zdjęcia głównego – za mała, źle wykrywa rozmiary poniżej 100 px
- Czasami fałszywe wykrycia podobnych do siebie znaków

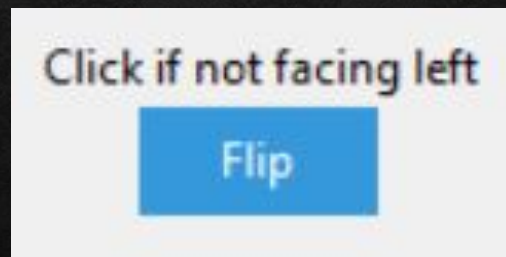
KIERUNKI CZYTANIA

Za podstawowy kierunek czytania uznaliśmy od lewej do prawej, zgodnie ze znakami unicode. Problem pojawił się przy próbach odczytania hieroglifów skierowanych w prawo. Zdecydowaliśmy się sprawdzić dwa sposoby.

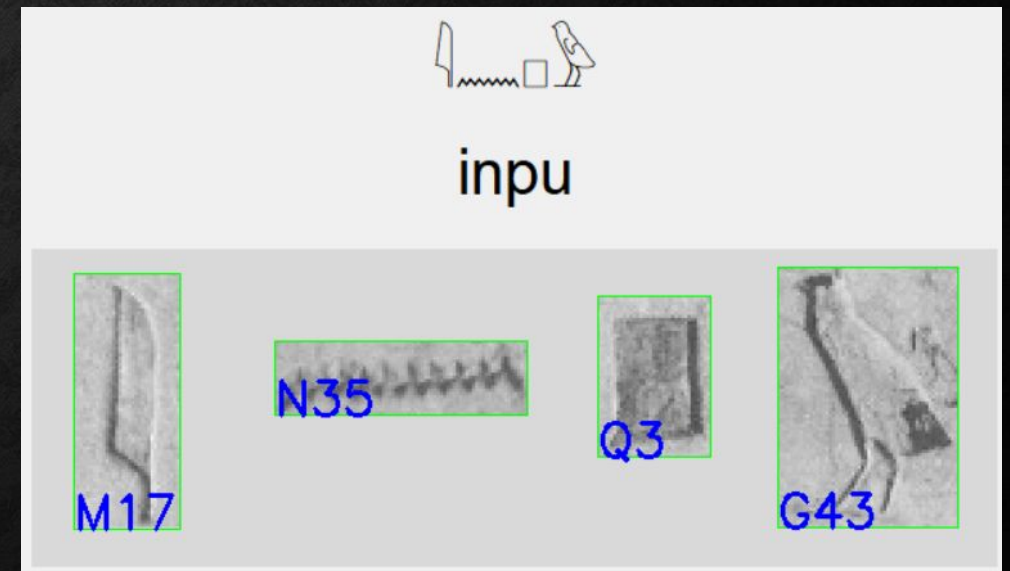
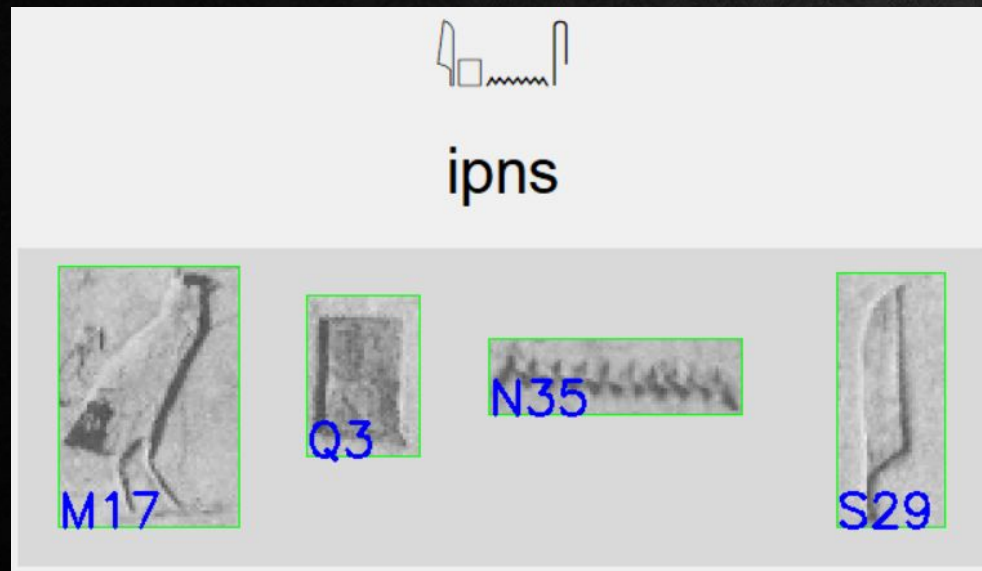


SPOSÓB 1: ODBICIE OBRAZU

Jednym z rozwiązań było odbicie obrazu wejściowego względem osi Y. Program nie rozpoznaje automatycznie kierunku, w którym zwrócone są hieroglify, dlatego użytkownik musi kliknąć odpowiedni przycisk zawarty w GUI. Dzięki temu rozpoznawanie przebiega tak samo jak dla hieroglifów zwróconych w lewo.

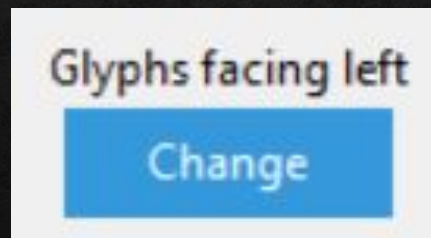


ODBICIE OBRAZU

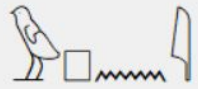


SPOSÓB 2: ZWIĘKSZENIE ZBIORU UCZĄCEGO

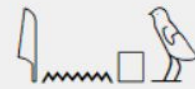
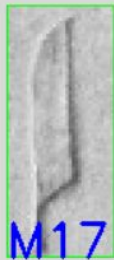
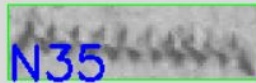
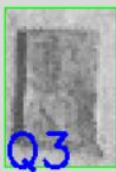
Innym rozwiązaniem jest zwiększenie zbioru uczącego o znaki zwrócone w prawo. Dzięki temu program jest w stanie odpowiednio rozpoznawać glify bez względu na ich zwrot. Kolejnością odczytywania użytkownik steruje za pomocą odpowiedniego przycisku w GUI. Użyliśmy tutaj wbudowanego parametru reverse funkcji sortowania znalezionych wzorów.



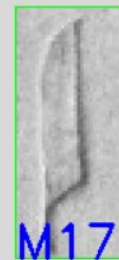
ZMIANA KOLEJNOŚCI ODCZYTU



upni



inpu



PORÓWNIANIE OBU SPOSOBÓW

Odbicie obrazu

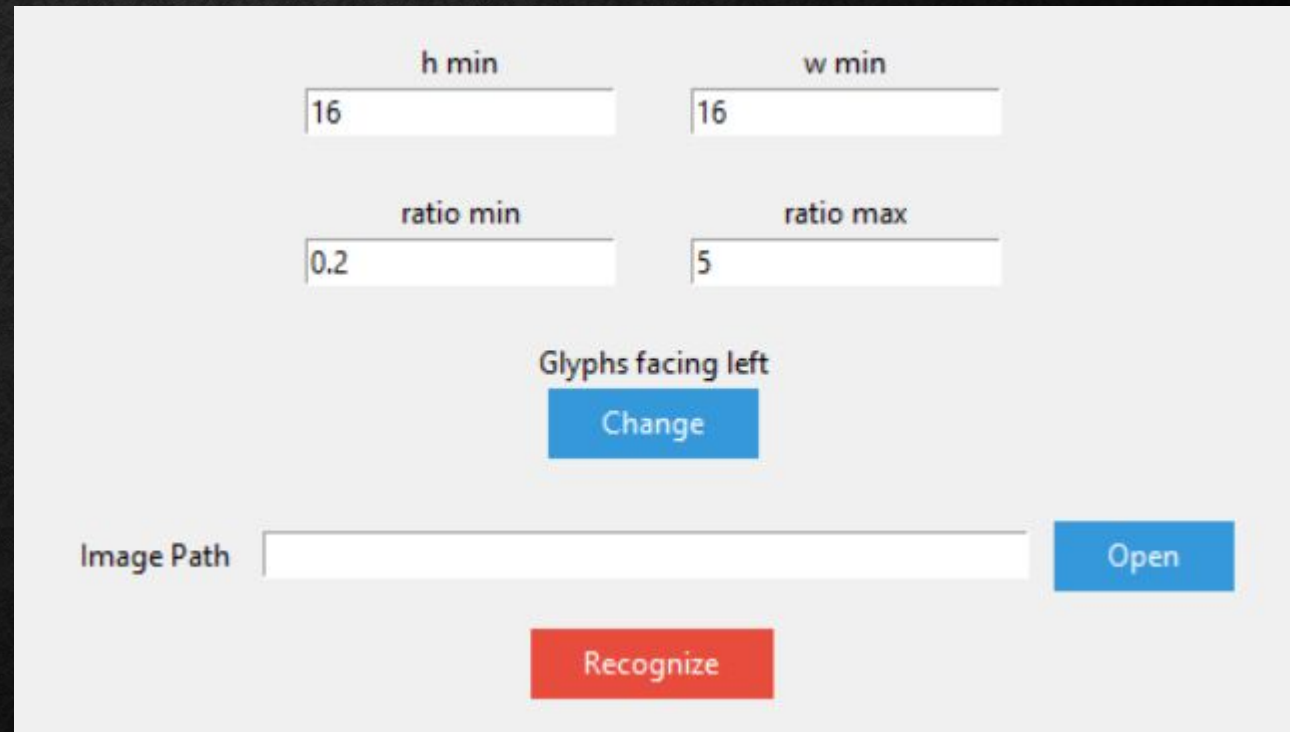
- Mniejszy zbiór uczący
- Przetwarzany obraz docelowo wygląda tak jak transkrypcja unicode
- Niepoprawne rozpoznawanie hieroglifów zwróconych w prawo przed odbiciem
- Ręczne sterowanie kierunkiem odczytu

Zmiana kolejności odczytu

- Poprawne rozpoznawanie dowolnych hieroglifów
- Brak konieczności odbijania obrazu
- Większy zbiór uczący
- Ręczne sterowanie kierunkiem odczytu

INTERFEJS GRAFICZNY

Zawarliśmy w GUI nie tylko wspomniane wyżej przyciski sterowania kierunkiem odczytu, ale także wszystkie ważne parametry rozpoznawania oraz oczywiście pole wyboru pliku wejściowego.



The image shows a graphical user interface (GUI) for image recognition. It features several input fields and buttons:

- h min**: Input field with the value 16.
- w min**: Input field with the value 16.
- ratio min**: Input field with the value 0.2.
- ratio max**: Input field with the value 5.
- Glyphs facing left**: A label above a blue **Change** button.
- Image Path**: A label next to a long white input field.
- Open**: A blue button next to the Image Path input field.
- Recognize**: A red button at the bottom center.

PARAMETRY ROZPOZNAWANIA

Pola są wypełnione wartościami domyślnymi, jednak nie sprawdzają się one w każdym przypadku. Aby użytkownik miał możliwość odczytania pliku o dowolnym rozmiarze, musieliśmy pozwolić mu na sterowanie progami rozpoznawania, tzn. minimalnym rozmiarem glifu oraz granicami stosunku jego długości do szerokości.

Wartości domyślne należy zmienić, jeśli na przetworzonym obrazie nie rozpoznano odpowiedniej ilości glifów (tzn. część znaków jest pominięta lub pojawiają się znaki nadmiarowe).

h min	w min
<input type="text" value="16"/>	<input type="text" value="16"/>
ratio min	ratio max
<input type="text" value="0.2"/>	<input type="text" value="5"/>

PRZYKŁAD ŹŁE DOBRANYCH PARAMETRÓW

h min 100 w min 100

ratio min 0.2 ratio max 5

am

G1 G17

PODSUMOWANIE

- Przeciętny wynik 90-100%.
- Najlepszy wynik 100%.
- Najgorszy wynik 80%.
- Niewłaściwa rozdzielczość prowadzi do błędnych wykryć lub braku wyników.
- Najlepsze wyniki uzyskano na zdjęciach bez dodatkowych filtrów.

ŹRÓDŁA

- [1] M. Collier, B. Manley, *How to read Egyptian hieroglyphs*, British Museum Press
- [2] https://en.wikipedia.org/wiki/List_of_Egyptian_hieroglyphs
- [3] <https://github.com/piotrjasiu95/hieroglyphsDetection>
- [4] <https://medium.com/analytics-vidhya/ocr-with-machine-learning-55c7d082fe78>



PYTANIA ?

DZIĘKUJEMY ZA UWAGĘ

Autorzy:

- TR
- MS
- JS