

---

# **Taggy**

***Wydanie v1.0***

**Aleksander Okrasa**

**18 sty 2025**



---

## Spis treści:

---

<b>1</b>	<b>Dependencies</b>	<b>1</b>
1.1	CLI - Taggy . . . . .	1
1.1.1	taggy_cli . . . . .	2
1.1.1.1	duplicates . . . . .	2
1.1.1.2	search . . . . .	3
1.1.1.3	tag . . . . .	3
1.2	Tagger obrazów - Taggy . . . . .	4
1.3	Moduły pomocnicze . . . . .	8
1.3.1	utils.file_utils . . . . .	8
1.3.2	utils.logger . . . . .	10
<b>2</b>	<b>Indices and tables</b>	<b>13</b>
	<b>Indeks modułów Pythona</b>	<b>15</b>
	<b>Indeks</b>	<b>17</b>



Your project requires the following Python packages to function:

- **torch**: Used for loading models from *torch.hub*.
- **torchvision**: Provides pre-trained models.
- **click**: Command-line interface support.
- **rich**: For enhanced logging.
- **numpy** and **pillow**: For loading and handling images.
- **opencv-python**: For simple image analysis, such as sharpness or face detection.

Additional tools for documentation and translations:

- **Sphinx** and its extensions: *sphinx-click*, *sphinx-intl*, *myst-parser*.
- **Furo**: Modern documentation theme.
- **py pandoc**: For exporting documentation to DOCX.

Special models and utilities:

- **CLIP**: Main model for image analysis (from OpenAI).
- Optional: *tensorflow* and *toml* for additional model support.

## 1.1 CLI - Taggy

**Moduł, który zapewnia zarówno synchroniczną funkcjonalność dla:**

- Oznaczanie obrazów za pomocą CLIP
- Wyszukiwanie obrazów na podstawie podobieństwa do zapytania tekstowego
- Znajdowanie i grupowanie zduplikowanych obrazów poprzez osadzanie podobieństwa
- **Proponowanie najlepszych obrazów w każdej grupie na podstawie wielu wskaźników:**

- Naiwna „ostrość” dzięki Laplacianowi
- Ostrość oparta na wykrywaniu twarzy
- Opcjonalne umieszczenie nieduplikatów w folderze „non\_duplicates”.
- Zapisywanie wyników grupowania w formacie JSON

### 1.1.1 taggy\_cli

Główny punkt wejścia dla Taggy CLI.

```
taggy_cli [OPTIONS] COMMAND [ARGS]...
```

#### Opcje

**-c, --config-file** <config\_file>

Ścieżka do pliku konfiguracyjnego.

#### 1.1.1.1 duplicates

Znajduje i grupuje zduplikowane obrazy na podstawie ich podobieństwa osadzenia. Następnie organizuje je w podfoldery, wybierając w razie potrzeby „najlepsze obrazy”.

```
taggy_cli duplicates [OPTIONS]
```

#### Opcje

**-i, --images-path** <images\_path>

**Wymagane** Ścieżka do folderu zawierającego obrazy.

**-o, --output-folder** <output\_folder>

Folder do umieszczania zgrupowanych duplikatów.

**-l, --labels** <labels>

Lista etykiet używanych do grupowania nazw duplikatów (jeśli nie podano, używana jest wartość domyślna).

**-op, --operation** <operation>

Operacje na plikach do wykonania podczas grupowania duplikatów.

##### opcje

copy | symlink | move

**-t, --similarity-threshold** <similarity\_threshold>

Próg (0..1) dla uznania obrazów za duplikaty.

**--face-cascade** <face\_cascade>

Ścieżka do pliku XML kaskady Haara do wykrywania twarzy.

**-b, --propose-best**

Jeśli to możliwe, proponuje „najlepsze” obrazy do zachowania w każdej grupie.

**--best-method** <best\_method>

Jak oceniać „najlepsze” obrazy.

##### opcje

zaawansowany laplacian

### 1.1.1.2 search

Wyszukuje obrazy najbardziej podobne do podanego zapytania tekstowego przy użyciu osadzania CLIP.

```
taggy_cli search [OPTIONS]
```

#### Opcje

**-i, --images-path** <images\_path>

**Wymagane** Ścieżka do folderu zawierającego obrazy.

**-q, --query** <query>

**Wymagane** Zapytanie tekstowe do wyszukiwania podobnych obrazów.

**-o, --output-folder** <output\_folder>

Folder do umieszczenia wyników wyszukiwania. Użyj, jeśli chcesz wykonać jakąś operację na znalezionych plikach.

**-op, --operation** <operation>

Operacja na pliku do wykonania podczas wyszukiwania obrazów. Domyślnie jest to „kopiuj

#### opcje

copy | symlink | move

**-k, --top-k** <top\_k>

Liczba najlepszych wyników do zwrócenia.

### 1.1.1.3 tag

Przypisuje etykiety do każdego obrazu w images\_path przy użyciu modelu CLIP.

```
taggy_cli tag [OPTIONS]
```

#### Opcje

**-i, --images-path** <images\_path>

**Wymagane** Ścieżka do folderu zawierającego obrazy.

**-t, --threshold** <threshold>

Minimalny próg prawdopodobieństwa przypisania etykiety.

**-k, --top-k** <top\_k>

Liczba najlepszych tagów do zwrócenia.

**-l, --labels** <labels>

Lista etykiet używanych do tagowania (jeśli nie podano, używana jest wartość domyślna).

**--one-output-json** <one\_output\_json>

Jeśli podano, zapisuje tagi dla każdego pliku w jednym podanym pliku JSON.

**-op, --operation** <operation>

Operacja na pliku do wykonania. Pogrupowane według wykrytych etykiet, działające z parametrem --output-folder. Domyślnie „symlink”.

#### opcje

copy | symlink

**-o, --images-output-folder** <images\_output\_folder>

Folder do umieszczania oznaczonych obrazów. Gdy chcesz wykonać jakąś operację na plikach, które zostały znalezione, grafika może się powtarzać (jeden obraz może znajdować się w wielu folderach).

**-j, --create-many-output-jsons**

Jeśli podano, zapisuje tagi dla każdego pliku w oddzielnych plikach JSON.

## 1.2 Tagger obrazów - Taggy

**Moduł, który zapewnia zarówno synchroniczną funkcjonalność dla:**

- Oznaczanie obrazów za pomocą CLIP
- Wyszukiwanie obrazów na podstawie podobieństwa do zapytania tekstowego
- Znajdowanie i grupowanie zduplikowanych obrazów poprzez osadzanie podobieństwa
- **Proponowanie najlepszych obrazów w każdej grupie na podstawie wielu wskaźników:**
  - Naiwna „ostrość” dzięki Laplacianowi
  - Ostrość oparta na wykrywaniu twarzy
- Opcjonalne umieszczenie nieduplikatów w folderze „non\_duplicates”.
- Zapisywanie wyników grupowania w formacie JSON

```
class taggy.utils.image_tagger.ImageTagger(model_name: str = 'CLIP', labels: list = None,  
                                           face_cascade_path: str = None)
```

Klasa używana do tagowania obrazów, wyszukiwania obrazów według podobieństwa, znajdowania i grupowania zduplikowanych obrazów oraz proponowania najlepszych obrazów w każdej grupie na podstawie wielu wskaźników.

**model\_name**

Nazwa modelu do załadowania.

**typ**  
*str*

**labels**

Lista możliwych tagów.

**typ**  
*list*

**face\_cascade\_path**

Ścieżka do pliku kaskady Haara do wykrywania twarzy.

**typ**  
*str*

**device**

Urządzenie, na którym ma zostać uruchomiony model (CPU lub GPU).

**typ**  
*str*

**model**

Załadowany model CLIP.



**typ**  
torch.nn.Module

### preprocess

Funkcja przetwarzania wstępnego dla modelu CLIP.

**typ**  
wywoływalny

**\_\_init\_\_**(*model\_name: str = 'CLIP', labels: list = None, face\_cascade\_path: str = None*)

Initializes the ImageTagger instance.

### Parametry

- **model\_name** (*str*) – Name of the model to load. Defaults to „CLIP”.
- **labels** (*list, optional*) – List of possible tags. Defaults to None.
- **face\_cascade\_path** (*str, optional*) – Path to the Haar cascade file for face detection. Defaults to None.

**\_combined\_image\_score**(*gray: ndarray*)

Oblicza łączny wynik jakości obrazu na podstawie ostrości i wykrywania twarzy.

Metoda ta oblicza ostrość całego obrazu przy użyciu wariacji Laplaciana. Jeśli dostępny jest plik kaskady Haara do wykrywania twarzy, oblicza on również ostrość wykrytych twarzy i łączy te wyniki, aby uzyskać ostateczny wynik jakości.

### Parametry

**gray** (*np.ndarray*) – Tablica obrazów w skali szarości.

### Zwraca

Łączny wynik jakości obrazu.

### Typ zwracany

float

**\_generate\_tags**(*image\_path: str, threshold: float = 0.3*)

Generuje listę nazw tagów dla danego obrazu.

### Parametry

- **image\_path** (*str*) – Ścieżka do obrazu.
- **threshold** (*float, optional*) – Próg prawdopodobieństwa. Domyślnie 0,3.

### Zwraca

Lista nazw tagów przypisanych do obrazu.

### Typ zwracany

list[str]

**\_load\_images**(*images\_path: str*)

Ładuje wszystkie obrazy z katalogu i wstępnie przetwarza je w Tensor (synchronicznie).

### Parametry

**images\_path** (*str*) – Ścieżka do katalogu zawierającego obrazy.

### Zwraca

- List of image file paths
- Skonkatelowany torch.Tensor wstępnie przetworzonych obrazów

**Typ zwracany**`tuple(list[str], torch.Tensor)`

**\_process\_duplicate\_groups**(*grouped: dict*, *output\_folder: str*, *operation: str = 'copy'*, *propose\_best: bool = True*, *best\_scoring\_method: str = 'advanced'*)

Przetwarza każdą grupę, aby: - znalezienia najczęściej występującego tagu - opcjonalnego pomiaru jakości w celu wybrania najlepszych obrazów - wykonania operacji na plikach i zebrania danych wyników

**Parametry**

- **grouped** (*dict[int, set]*) – group\_id => zestaw ścieżek obrazów
- **output\_folder** (*str*) – Destination folder
- **operation** (*str, optional*) – File op («copy», «move», etc.). Defaults to „copy”.
- **propose\_best** (*bool, optional*) – If True, measure image quality. Defaults to True.
- **best\_scoring\_method** (*str, optional*) – „advanced” or „laplacian”.

**Zwraca**

Lista rekordów grup z informacjami o obrazach i najlepszych obrazach

**Typ zwracany**`list[dict]`

**\_process\_image**(*image\_path: str*, *threshold: float*, *labels: list = None*)

Przejdzie do przodu (synchroniczne) przez model CLIP dla pojedynczego obrazu, zwracające etykiety powyżej progu prawdopodobieństwa.

**Parametry**

- **image\_path** (*str*) – Ścieżka do pliku obrazu.
- **labels** (*list*) – Lista etykiet tekstowych.
- **threshold** (*float*) – Probability threshold for tag assignment.

**Zwraca**

A list of dictionaries with «tag» and «probability» keys.

**Typ zwracany**`list[dict]`

**find\_duplicates**(*images\_path: str*, *similarity\_threshold: float = 0.9*)

Identyfikuje zduplikowane obrazy na podstawie kosinusowego podobieństwa osadzeń, synchronicznie.

Ta metoda ładuje obrazy z określonego katalogu, oblicza ich osadzenia przy użyciu modelu CLIP, a następnie oblicza podobieństwo kosinusowe między każdą parą obrazów. Jeśli podobieństwo przekracza określony próg, para jest uznawana za duplikat.

**Parametry**

- **images\_path** (*str*) – Ścieżka do folderu zawierającego obrazy.
- **similarity\_threshold** (*float*) – Próg uznawania obrazów za duplikaty, w zakresie od 0 do 1. Domyślnie 0,9.

**Zwraca**

Lista krotek, gdzie każda krotka zawiera dwie ścieżki obrazu i ich wynik podobieństwa.

**Typ zwracany**`list[tuple[str, str, float]]`

**group\_duplicates**(*duplicates: list*, *output\_folder: str*, *operation: str = 'copy'*, *propose\_best: bool = True*, *all\_images: list = None*, *best\_scoring\_method: str = 'advanced'*)

Wysokopoziomowa metoda grupowania duplikatów w podfolderach, obsługująca również nie-duplikaty. - Znajduje grupy z duplikatów - Przetwarza każdą grupę (wspólny tag, najlepsze obrazy) - Umieszcza nie-duplikaty obrazów w podfolderze „non\_duplicates” - Zapisuje podsumowanie JSON wszystkiego

#### Parametry

- **duplicates** (*list[tuple]*) – (img1, img2, podobieństwo).
- **output\_folder** (*str*) – Gdzie będą umieszczane zgrupowane obrazy.
- **operation** (*str, optional*) – Operacja na pliku. Domyślnie „kopiuj”.
- **propose\_best** (*bool, optional*) – Czy mierzyć jakość. Domyślnie True.
- **all\_images** (*list[str], optional*) – Jeśli podano, oddziela nieduplikaty. Domyślnie Brak.
- **best\_scoring\_method** (*str, optional*) – Metoda oceny najlepszych obrazów. Domyślnie „zaawansowane”.

**search\_images**(*query: str*, *images\_path: str*, *top\_k: int = 5*, *output\_path: str = None*, *operation: str = None*)

Wyszukuje obrazy podobne do zapytania tekstowego przy użyciu modelu CLIP.

#### Parametry

- **query** (*str*) – Zapytanie tekstowe do wyszukania.
- **images\_path** (*str*) – Ścieżka do katalogu zawierającego obrazy.
- **top\_k** (*int, optional*) – Liczba najbardziej podobnych obrazów do zwrócenia. Domyślnie 5.
- **output\_path** (*str, optional*) – Ścieżka do zapisania znalezionych plików. Domyślnie Brak.
- **operation** (*str, optional*) – Operacja na pliku. Domyślnie „kopiuj”.

#### Zwraca

Lista krotek zawierająca ścieżki do plików graficznych i ich wyniki podobieństwa.

#### Typ zwracany

*list[tuple[str, float]]*

**tag\_image**(*image\_path: str*, *output\_path: str = None*, *top\_k: int = 5*, *labels=None*, *threshold: float = 0.3*, *operation: str = None*, *output\_folder: str = None*)

Generuje tagi dla danego obrazu za pomocą CLIP, opcjonalnie zapisuje metadane.

#### Parametry

- **image\_path** (*str*) – Ścieżka do obrazu.
- **output\_path** (*str, optional*) – Ścieżka do zapisania metadanych. Domyślnie Brak.
- **top\_k** (*int, optional*) – Liczba najlepszych tagów do zwrócenia. Domyślnie 5.
- **labels** (*list*)
- **threshold** (*float, optional*) – Próg prawdopodobieństwa. Domyślnie 0,3.

#### Zwraca

Lista przypisanych tagów (tag + prawdopodobieństwo).

#### Typ zwracany

*list[dict]*

`taggy.utils.image_tagger._calculate_sharpness(img_gray: ndarray)`

Oblicza ostrość obrazu w skali szarości przy użyciu wariancji Laplaciana.

Ta funkcja oblicza laplasjan obrazu i zwraca wariancję, która jest miarą ostrości. Wyższa wariancja oznacza ostrzejszy obraz.

**Parametry**

**img\_gray** (*np.ndarray*) – Tablica obrazów w skali szarości.

**Zwraca**

Wariancja Laplaciana, reprezentująca ostrość obrazu.

**Typ zwracany**

*float*

`taggy.utils.image_tagger._find_duplicate_groups(duplicates: list)`

Tworzy słownik grupujący zestawy duplikatów. Każda grupa ma numeryczny identyfikator.

**Parametry**

**duplicates** (*list[tuple]*) – e.g. list of (img1, img2, similarity).

**Zwraca**

Mapowanie z group\_id => zestaw ścieżek obrazów

**Typ zwracany**

*dict[int, set]*

`taggy.utils.image_tagger._load_image(image_path: str)`

Wczytuje obraz z podanej ścieżki i konwertuje go na obraz w skali szarości.

Ta funkcja używa OpenCV do odczytania danych obrazu z pliku, zdekodowania go i przekonwertowania na skalę szarości. Jeśli OpenCV nie jest dostępny lub nie można załadować obrazu, funkcja zwraca None.

**Parametry**

**image\_path** (*str*) – Ścieżka do pliku obrazu.

**Zwraca**

Tablica obrazów w skali szarości w przypadku powodzenia, brak w przeciwnym razie.

**Typ zwracany**

*np.ndarray*

`taggy.utils.image_tagger._measure_image_quality(image_path: str)`

Naiwna miara „ostrości” wykorzystująca wariancję Laplaciana na całym obrazie. Wyższy => ostrzejszy obraz.

**Parametry**

**image\_path** (*str*) – Ścieżka do pliku obrazu.

**Zwraca**

Wariancja laplasjanu (0 w przypadku błędu lub braku cv2).

**Typ zwracany**

*float*

## 1.3 Moduły pomocnicze

### 1.3.1 utils.file\_utils

Funkcje narzędziowe do operacji na plikach, tworzenia skrótów (Windows) i zapisywania metadanych.

`taggy.utils.file_utils.copy_file(src, dest)`

Kopiuje plik z src do dest, zachowując metadane (znaczniki czasu itp.).

**Parametry**

- **src** (*str*) – Ścieżka pliku źródłowego.
- **dest** (*str*) – Ścieżka pliku docelowego.

`taggy.utils.file_utils.create_directory(path)`

Tworzy katalog, jeśli nie istnieje.

**Parametry**

**path** (*str*) – Ścieżka do katalogu.

`taggy.utils.file_utils.create_shortcut(target, shortcut_path, description=None, icon_path=None)`

Tworzy skrót Windows (.lnk) do pliku docelowego.

**Parametry**

- **target** (*str*) – Ścieżka bezwzględna do pliku docelowego.
- **shortcut\_path** (*str*) – Ścieżka bezwzględna, w której zostanie utworzony skrót.
- **description** (*str*, *optional*) – Opis skrótu. Domyślnie Brak.
- **icon\_path** (*str*, *optional*) – Ścieżka do pliku ikony. Domyślnie Brak.

`taggy.utils.file_utils.list_supported_image_files(images_path)`

Zwraca listę obsługiwanych ścieżek plików graficznych w podanym katalogu. Obsługiwane rozszerzenia: png, jpg, jpeg, bmp, webp

**Parametry**

**images\_path** (*str*) – Ścieżka do folderu zawierającego obrazy.

**Zwraca**

Lista znalezionych ścieżek plików z obsługiwanymi rozszerzeniami.

**Typ zwracany**

List[str]

`taggy.utils.file_utils.load_config(config_file: str = 'config.ini')`

Wczytuje wartości domyślne z pliku konfiguracyjnego INI i zwraca je jako obiekt podobny do słownika.

**Parametry**

**config\_file** (*str*, *optional*) – Ścieżka do pliku konfiguracyjnego. Domyślnie „config.ini”.

**Zwraca**

Sekcja konfiguracji (podobna do dict) z wartościami domyślnymi.

**Typ zwracany**

configparser.SectionProxy

`taggy.utils.file_utils.perform_file_operation(src, dest_dir, operation, description=None, icon_path=None)`

Wykonuje operację na pliku (kopiowanie lub dowiązanie symboliczne/skrót) z src do dest\_dir.

**Parametry**

- **src** (*str*) – Ścieżka pliku źródłowego.
- **dest\_dir** (*str*) – Ścieżka katalogu docelowego.
- **operation** (*str*) – Typ operacji („kopia”, „dowiązanie symboliczne” itp.).

- **description** (*str*, *optional*) – Używane podczas tworzenia skrótu. Domyślnie ustawiona na Brak.
- **icon\_path** (*str*, *optional*) – Używane podczas tworzenia skrótu. Domyślnie ustawiona na Brak.

taggy.utils.file\_utils.**preprocess\_image**(*img\_path*, *preprocess*, *device*) → **Tensor**

Otwiera i wstępnie przetwarza pojedynczy obraz przy użyciu dostarczonego potoku wstępnego przetwarzania CLIP.

#### Parametry

- **img\_path** (*str*) – Ścieżka do pliku obrazu.
- **preprocess** (*callable*) – Przekształcenie wstępne CLIP.
- **device** (*str* or *torch.device*) – «cpu» lub «cuda».

#### Zwraca

Wstępnie przetworzony obraz na odpowiednim urządzeniu.

#### Typ zwracany

*torch.Tensor*

taggy.utils.file\_utils.**save\_metadata\_to\_json**(*metadata*, *output\_file*)

Zapisuje metadane do pliku JSON.

#### Parametry

- **metadata** (*dict* or *list*) – Metadane do zapisania.
- **output\_file** (*str*) – Ścieżka do wyjściowego pliku JSON.

## 1.3.2 utils.logger

**class** taggy.utils.logger.**CustomFormatter**(*fmt=None*, *datefmt=None*, *style='%*, *validate=True*, \*, *defaults=None*)

Klasy bazowe: *Formatter*

Niestandardowy formatyzator logowania dodający kolory do komunikatów dziennika w oparciu o ich poziom ważności.

**COLORS** = {10: '\x1b[94m', 20: '\x1b[92m', 30: '\x1b[93m', 40: '\x1b[91m', 50: '\x1b[95m'}

**RESET** = '\x1b[0m'

**format**(*record*)

Formatuje określony rekord jako tekst.

#### Parametry

**record** (*logging.LogRecord*) – Rekord dziennika do sformatowania.

#### Zwraca

Sformatowany komunikat dziennika z kolorem.

#### Typ zwracany

*str*

taggy.utils.logger.**get\_logger**(*name*)

Pobiera rejestrator o podanej nazwie.

**Parametry**

**name** (*str*) – Nazwa rejestratora.

**Zwraca**

Instancja rejestratora.

**Typ zwracany**

*logging.Logger*

`taggy.utils.logger.show_example_logs()`

Generowanie przykładowych komunikatów dziennika o różnych poziomach ważności.





---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### t

`taggy.utils.file_utils`, 8  
`taggy.utils.image_tagger`, 4  
`taggy.utils.logger`, 10



## Symbole

```

__init__() (taggy.utils.image_tagger.ImageTagger me-
toda), 5
_calculate_sharpness() (w module
taggy.utils.image_tagger), 8
_combined_image_score()
(taggy.utils.image_tagger.ImageTagger me-
toda), 5
_find_duplicate_groups() (w module
taggy.utils.image_tagger), 8
_generate_tags() (taggy.utils.image_tagger.ImageTagger
metoda), 5
_load_image() (w module taggy.utils.image_tagger), 8
_load_images() (taggy.utils.image_tagger.ImageTagger
metoda), 5
_measure_image_quality() (w module
taggy.utils.image_tagger), 8
_process_duplicate_groups()
(taggy.utils.image_tagger.ImageTagger me-
toda), 6
_process_image() (taggy.utils.image_tagger.ImageTagger
metoda), 6
--best-method
taggy_cli-duplicates command line
option, 2
--config-file
taggy_cli command line option, 2
--create-many-output-jsons
taggy_cli-tag command line option, 4
--face-cascade
taggy_cli-duplicates command line
option, 2
--images-output-folder
taggy_cli-tag command line option, 3
--images-path
taggy_cli-duplicates command line
option, 2
taggy_cli-search command line option, 3
taggy_cli-tag command line option, 3
--labels
taggy_cli-duplicates command line
option, 2
taggy_cli-tag command line option, 3
--one-output-json
taggy_cli-tag command line option, 3
--operation
taggy_cli-duplicates command line
option, 2
taggy_cli-search command line option, 3
taggy_cli-tag command line option, 3
--output-folder
taggy_cli-duplicates command line
option, 2
taggy_cli-search command line option, 3
--propose-best
taggy_cli-duplicates command line
option, 2
--query
taggy_cli-search command line option, 3
--similarity-threshold
taggy_cli-duplicates command line
option, 2
--threshold
taggy_cli-tag command line option, 3
--top-k
taggy_cli-search command line option, 3
taggy_cli-tag command line option, 3
-b
taggy_cli-duplicates command line
option, 2
-c
taggy_cli command line option, 2
-i
taggy_cli-duplicates command line
option, 2
taggy_cli-search command line option, 3
taggy_cli-tag command line option, 3
-j
taggy_cli-tag command line option, 4

```

-k taggy\_cli-search command line option, 3  
taggy\_cli-tag command line option, 3

-l taggy\_cli-duplicates command line option, 2  
taggy\_cli-tag command line option, 3

-o taggy\_cli-duplicates command line option, 2  
taggy\_cli-search command line option, 3  
taggy\_cli-tag command line option, 3

-op taggy\_cli-duplicates command line option, 2  
taggy\_cli-search command line option, 3  
taggy\_cli-tag command line option, 3

-q taggy\_cli-search command line option, 3

-t taggy\_cli-duplicates command line option, 2  
taggy\_cli-tag command line option, 3

## C

COLORS (*taggy.utils.logger.CustomFormatter* atrybut), 10  
copy\_file() (w module *taggy.utils.file\_utils*), 8  
create\_directory() (w module *taggy.utils.file\_utils*), 9  
create\_shortcut() (w module *taggy.utils.file\_utils*), 9  
CustomFormatter (klasa w module *taggy.utils.logger*), 10

## D

device (*taggy.utils.image\_tagger.ImageTagger* atrybut), 4

## F

face\_cascade\_path(*taggy.utils.image\_tagger.ImageTagger* atrybut), 4  
find\_duplicates() (*taggy.utils.image\_tagger.ImageTagger* metoda), 6  
format() (*taggy.utils.logger.CustomFormatter* metoda), 10

## G

get\_logger() (w module *taggy.utils.logger*), 10  
group\_duplicates() (*taggy.utils.image\_tagger.ImageTagger* metoda), 6

## I

ImageTagger (klasa w module *taggy.utils.image\_tagger*), 4

## L

labels (*taggy.utils.image\_tagger.ImageTagger* atrybut), 4  
list\_supported\_image\_files() (w module *taggy.utils.file\_utils*), 9  
load\_config() (w module *taggy.utils.file\_utils*), 9

## M

model (*taggy.utils.image\_tagger.ImageTagger* atrybut), 4  
model\_name (*taggy.utils.image\_tagger.ImageTagger* atrybut), 4  
module  
taggy.utils.file\_utils, 8  
taggy.utils.image\_tagger, 4  
taggy.utils.logger, 10

## P

perform\_file\_operation() (w module *taggy.utils.file\_utils*), 9  
preprocess (*taggy.utils.image\_tagger.ImageTagger* atrybut), 5  
preprocess\_image() (w module *taggy.utils.file\_utils*), 10

## R

RESET (*taggy.utils.logger.CustomFormatter* atrybut), 10

## S

save\_metadata\_to\_json() (w module *taggy.utils.file\_utils*), 10  
search\_images() (*taggy.utils.image\_tagger.ImageTagger* metoda), 7  
show\_example\_logs() (w module *taggy.utils.logger*), 11

## T

tag\_image() (*taggy.utils.image\_tagger.ImageTagger* metoda), 7  
taggy.utils.file\_utils module, 8  
taggy.utils.image\_tagger module, 4  
taggy.utils.logger module, 10  
taggy\_cli command line option  
--config-file, 2  
-c, 2  
taggy\_cli-duplicates command line option  
--best-method, 2  
--face-cascade, 2  
--images-path, 2  
--labels, 2  
--operation, 2

```
--output-folder, 2
--propose-best, 2
--similarity-threshold, 2
-b, 2
-i, 2
-l, 2
-o, 2
-op, 2
-t, 2
taggy_cli-search command line option
--images-path, 3
--operation, 3
--output-folder, 3
--query, 3
--top-k, 3
-i, 3
-k, 3
-o, 3
-op, 3
-q, 3
taggy_cli-tag command line option
--create-many-output-jsons, 4
--images-output-folder, 3
--images-path, 3
--labels, 3
--one-output-json, 3
--operation, 3
--threshold, 3
--top-k, 3
-i, 3
-j, 4
-k, 3
-l, 3
-o, 3
-op, 3
-t, 3
```