

# Sprawozdanie

PROJEKT 4

KRZYSZTOF KOWALSKI 407142

## Cel projektu

Zbadanie działania różnych algorytmów służących do wykrywania podobnych cech na fragmencie zdjęcia do całości zdjęcia, a następnie interpretacja wyników.

## Przebieg ćwiczenia

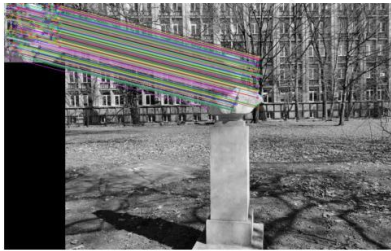
Ponieważ w pythonie nie ma implementacji SURF, zdecydowałem się wybrać 4 algorytmy, które oparte są na różnym działaniu, dzięki czemu pozwolą ciekawie je rozróżnić.



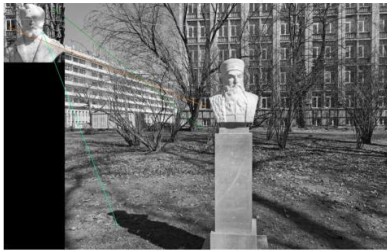
*Rysunek 1 Fragment wykorzystanego zdjęcia pochodzący z image031.jpg*

1. **SIFT (Scale-Invariant Feature Transform)** jest algorytmem służącym do detekcji i opisywania cech obrazów, który jest odporny na zmiany skali, rotację oraz zmiany oświetlenia. Wykorzystuje on różnice gaussowskie w obrazie, aby wykryć punkty kluczowe, które stabilne względem zmiany skali. Po wykryciu punktów kluczowych, SIFT generuje opis tych punktów, które są oparte na lokalnych gradientach intensywności pikseli w otoczeniu punktu kluczowego. Jest jednym z najbardziej zaawansowanych i dokładnych algorytmów detekcji cech obrazów, jednak jego złożoność obliczeniowa może być duża, co może sprawić trudności w zastosowaniach wymagających szybkiej analizy obrazów.

Podobieństwa do 031 (SIFT)



Podobieństwa do 032 (SIFT)



Podobieństwa do 033 (SIFT)



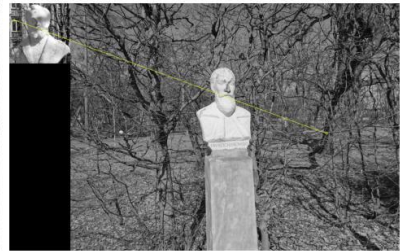
Podobieństwa do 034 (SIFT)



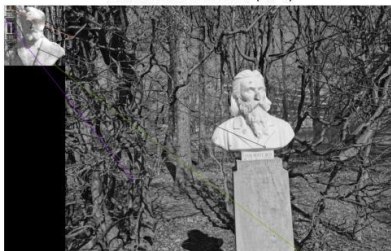
Podobieństwa do 035 (SIFT)



Podobieństwa do 036 (SIFT)



Podobieństwa do 037 (SIFT)



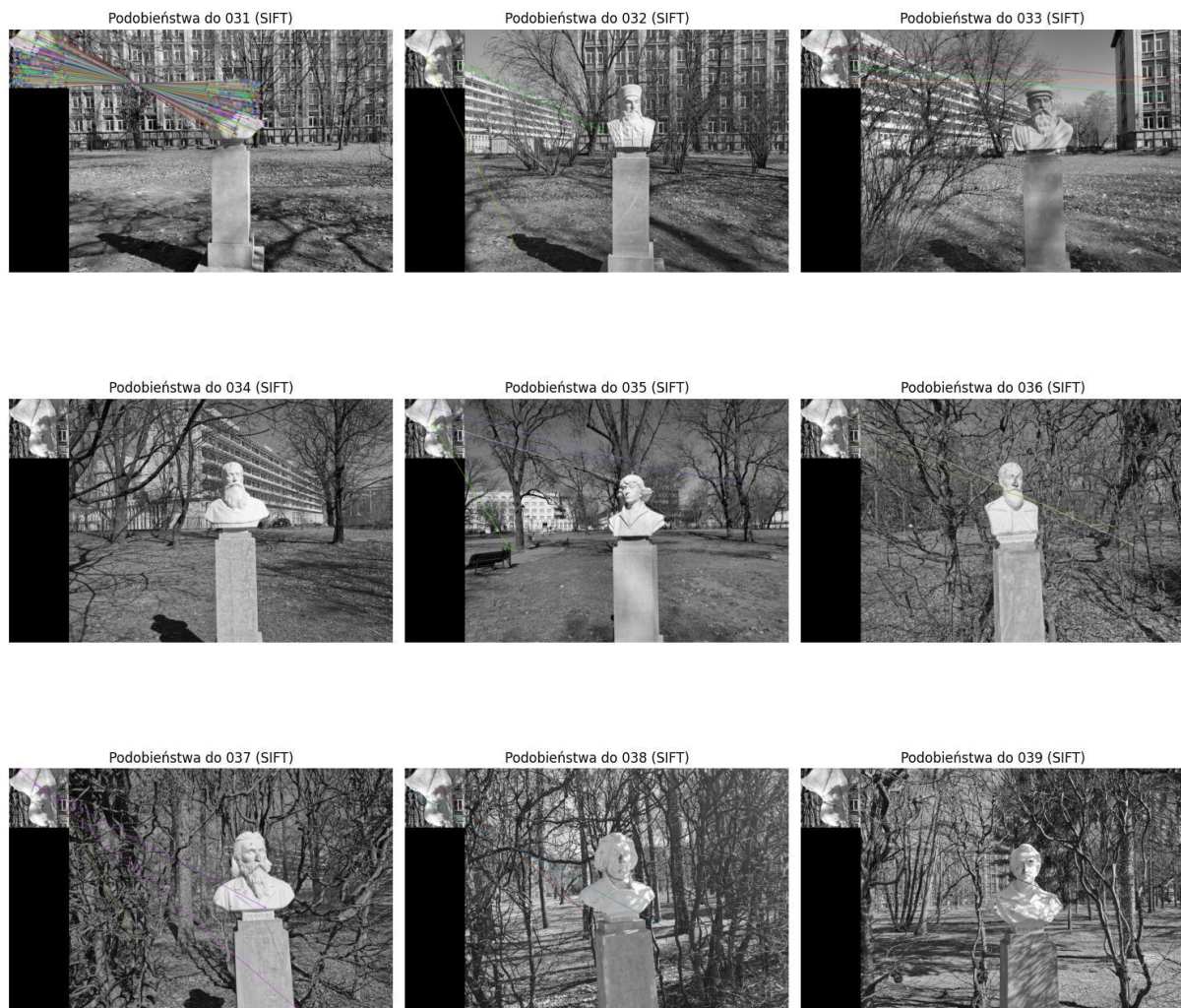
Podobieństwa do 038 (SIFT)



Podobieństwa do 039 (SIFT)



*Rysunek 2 Podobieństwa - SIFT*



*Rysunek 3 Odwrócone SIFT*

Algorytm SIFT znajduje wiele podobieństw zarówno do oryginalnego obrazu jak i obrazu odwróconego. Dobrze rozpoznaje obraz z którego pochodzi fragment. W innych obrazach znajduje bardzo mało cech podobieństwa.

2. **ORB (Oriented FAST and Rotated BRIEF)** jest algorytmem detekcji cech oraz deskryptorów stosowanym do wykrywania i opisywania punktów charakterystycznych na obrazie.

Algorytm ORB składa się z dwóch głównych części:

- Detekcji punktów charakterystycznych za pomocą algorytmu FAST (Features from Accelerated Segment Test).
- Opisywania tych punktów za pomocą algorytmu BRIEF (Binary Robust Independent Elementary Features).

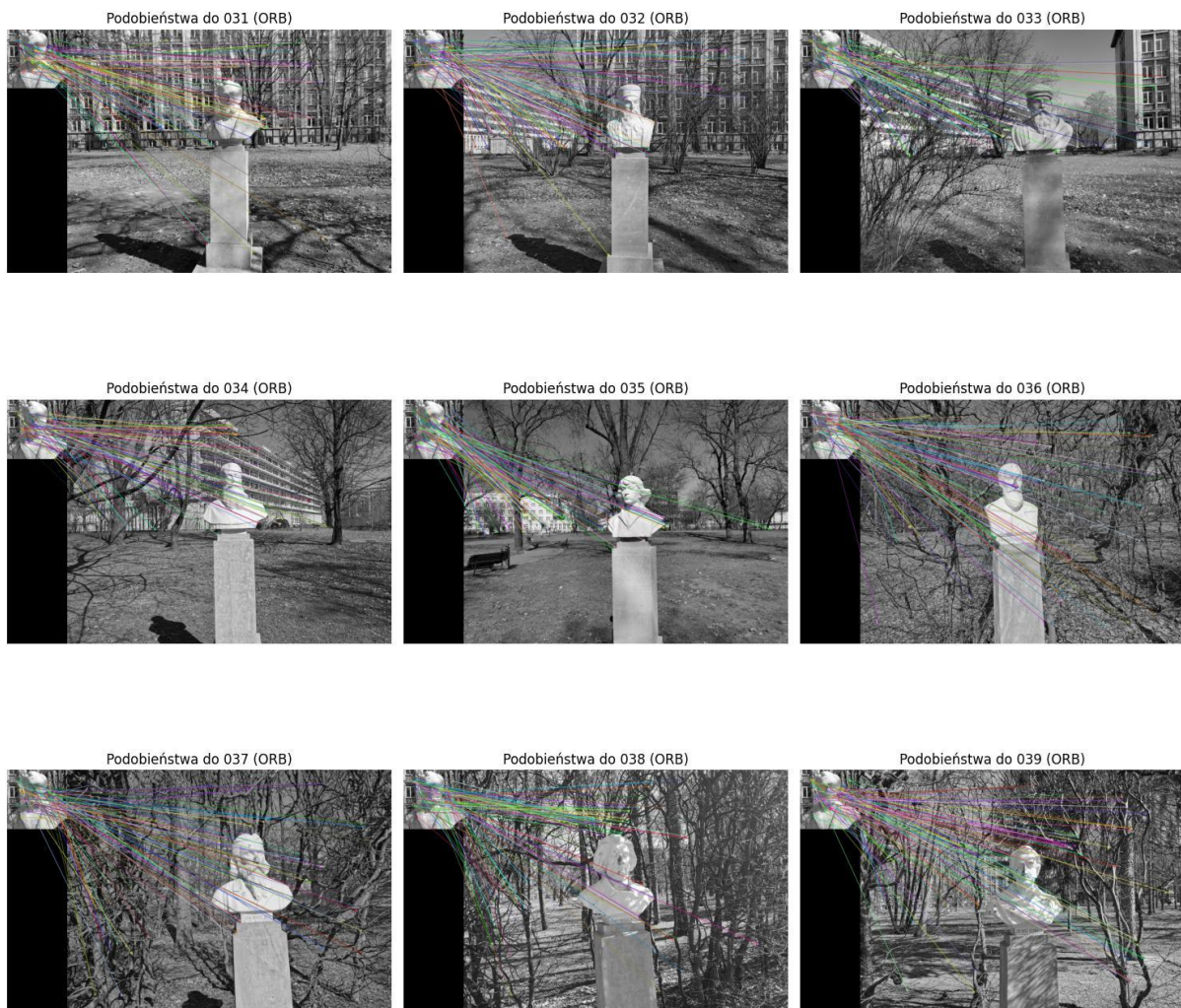
FAST jest szybkim algorytmem detekcji punktów charakterystycznych, który wykorzystuje test segmentów do przyspieszenia procesu.

Wyznacza punkty, które mają różne intensywności w porównaniu z ich otoczeniem.

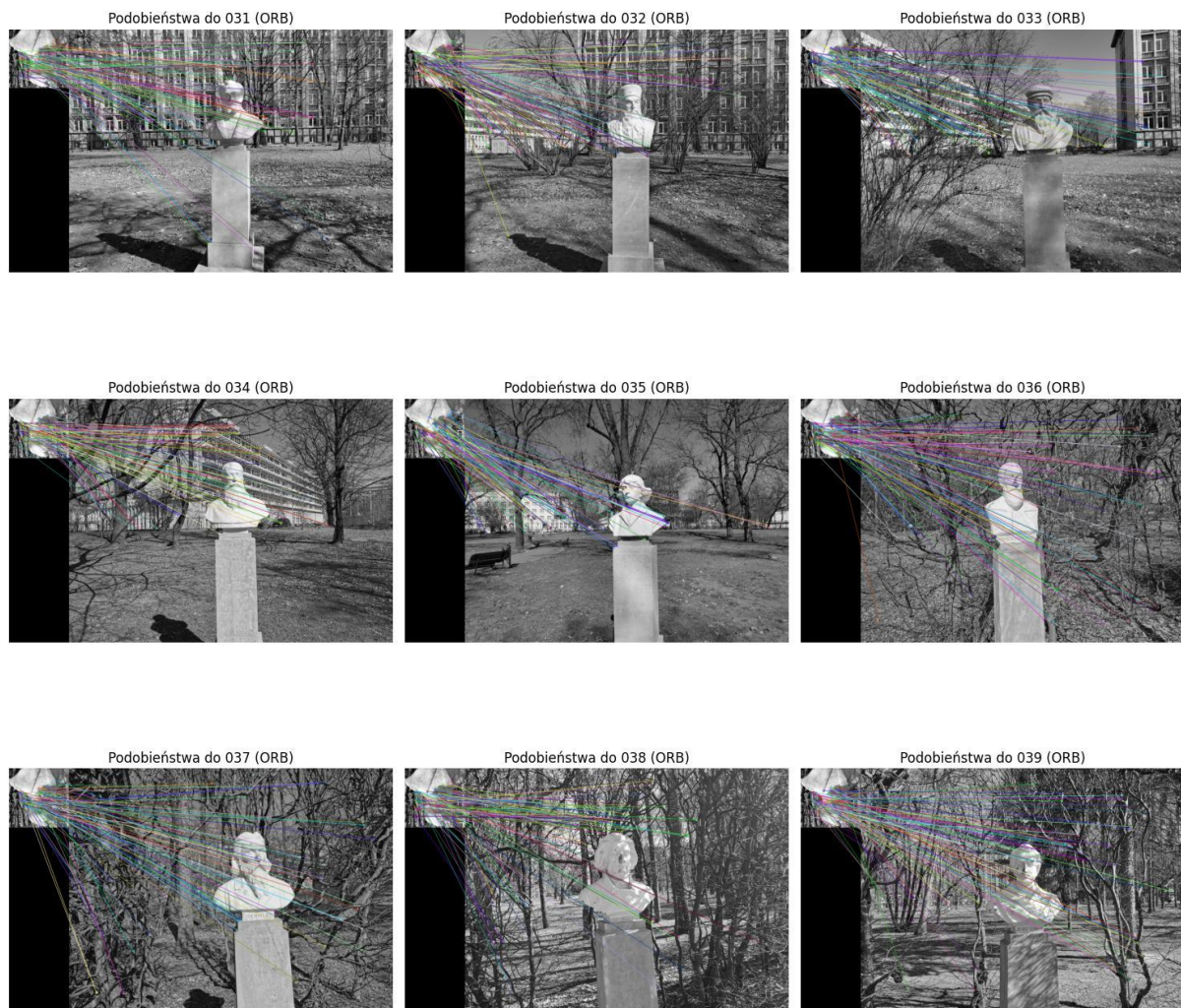
BRIEF jest algorytmem opisującym punkty charakterystyczne za pomocą krótkich binarnych wektorów cech, które są niezależne od rotacji obrazu i zmiany skali.



ORB jest stosunkowo szybki i efektywnym algorytmem. Szybkość działania sprawia, że jest on często stosowany w aplikacjach wymagających szybkiej detekcji i opisywania cech obrazów, takich jak systemy wizyjne w robotyce, śledzenie obiektów czy dopasowanie obrazów.



*Rysunek 4 Podobieństw ORB*

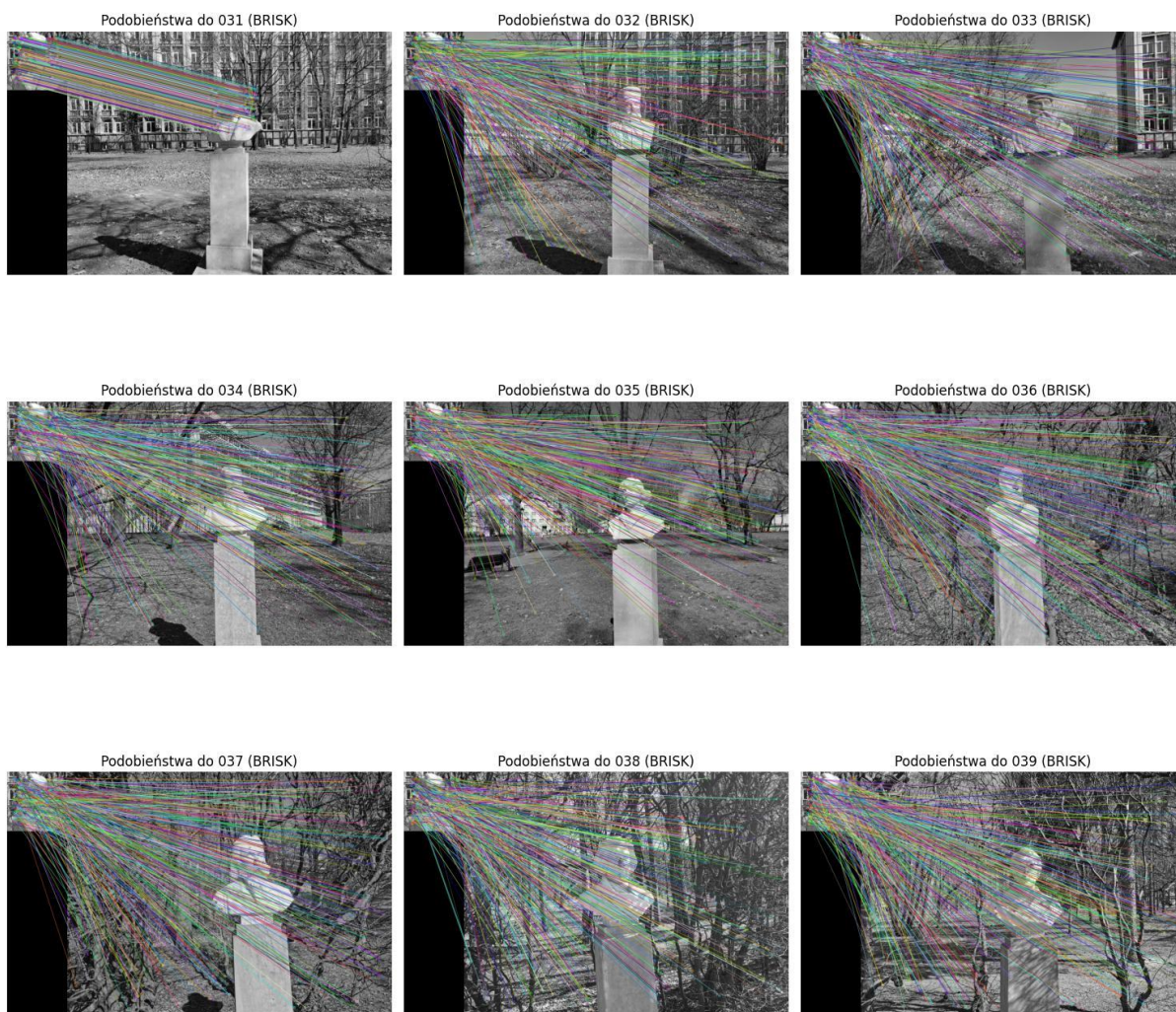


*Rysunek 5 Odwrócone ORB*

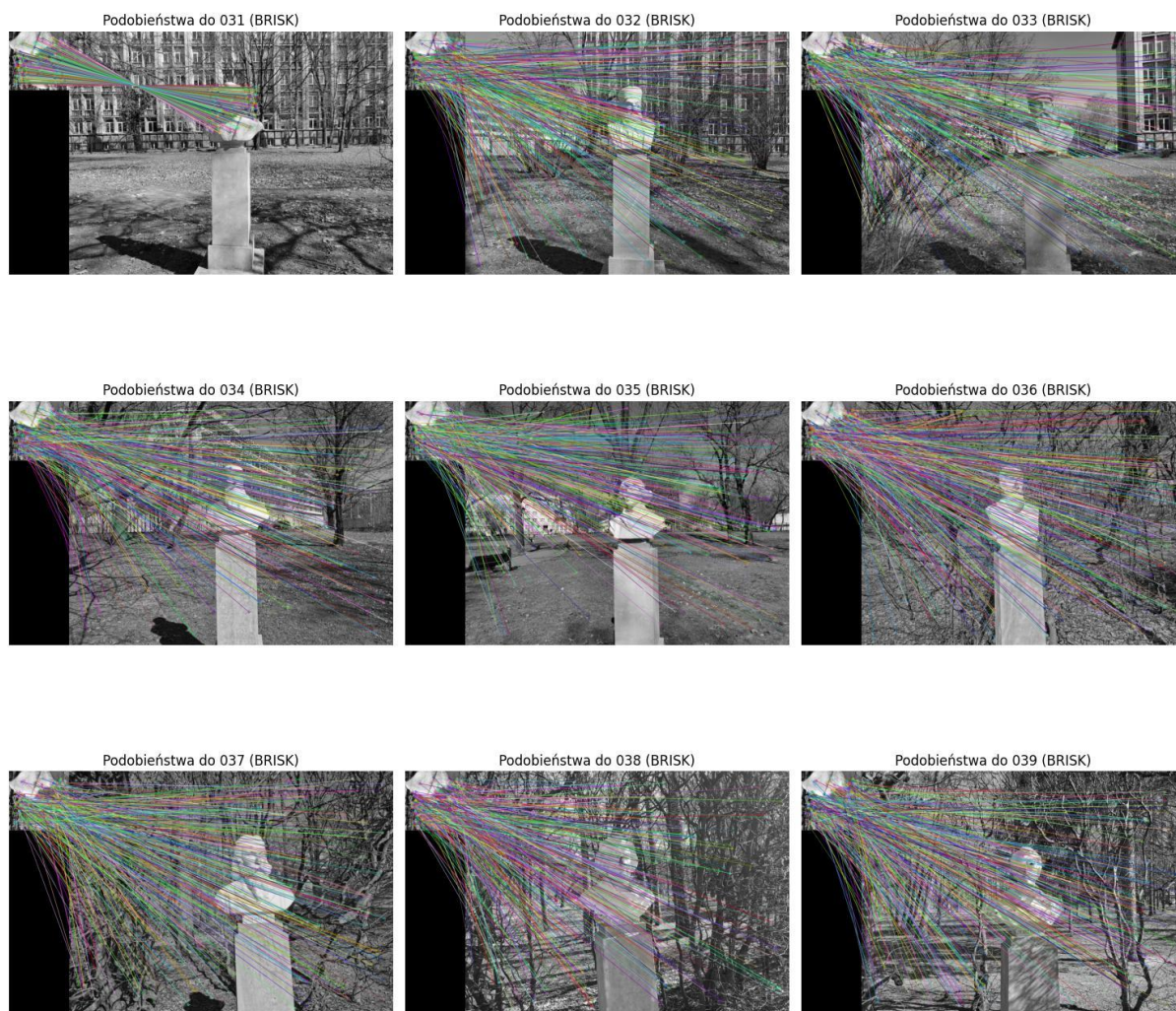
Algorytm ORB nie poradził sobie najlepiej z omawianym przykładem. Znajduje wiele cech podobieństwa pomiędzy obrazami, ale wszystkimi więc nie radzi sobie ze znalezieniem obrazu z którego pochodził fragment.

3. **BRISK (Binary Robust Invariant Scalable Keypoints)** jest to rozwinięcie algorytmu FAST, które dodaje do niego deskryptor oparty na operacjach binarnych, co sprawia, że jest bardziej odporny na zmiany w skali i rotacji obrazu. Algorytm BRISK wykorzystuje informacje o orientacji punktów charakterystycznych uzyskane za pomocą algorytmu FAST oraz wykorzystuje informacje o intensywnościach pikseli w różnych kierunkach do konstrukcji binarnego deskryptora punktu charakterystycznego. Jedną z zalet algorytmu BRISK jest jego odporność na zmiany w skali i rotacji obrazu, co czyni go bardziej elastycznym niż algorytmy takie jak FAST czy SIFT. BRISK jest również bardziej wydajny obliczeniowo niż niektóre inne algorytmy, co czyni go atrakcyjnym do zastosowań w czasie rzeczywistym. Obydwa algorytmy, FAST i BRISK, są używane do detekcji punktów charakterystycznych w obrazach i znajdują zastosowanie w różnych aplikacjach, takich jak śledzenie obiektów, rekonstrukcja sceny 3D, czy rozpoznawanie obrazów.





*Rysunek 6 Podobieństwa BRISK*



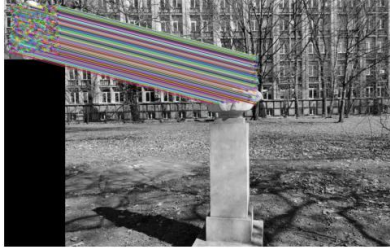
*Rysunek 7 Obrócone BRISK*

Algorytm BRISK znajduje ogromną ilość cech pomiędzy każdym z obrazów, natomiast obraz z którego pochodził fragment (zarówno oryginalny jak i obrócony) posiada cechy silnie skupione na popiersiu dzięki czemu można zaklasyfikować odpowiedni obraz – dokonać poprawnego rozpoznania.

4. **FAST (Features from Accelerated Segment Test)** jest to szybki algorytm, który znajduje punkty charakterystyczne poprzez porównywanie jasności pikseli w danym okręgu wokół badanego piksela. Algorytm ten polega na przeprowadzeniu testu na każdym pikselu obrazu w celu sprawdzenia, czy jest on punktem charakterystycznym. Test ten sprawdza, czy badany piksel jest jaśniejszy lub ciemniejszy od sąsiadujących pikseli w danym okręgu. Jeśli większość z  $n$  pikseli sąsiadujących z badanym pikselem ma wartość jasności większą lub mniejszą od wartości jasności badanego piksela plus lub minus zdefiniowaną wartość progową, to piksel ten jest uznawany za punkt charakterystyczny. Zaletą algorytmu FAST jest jego szybkość, dzięki czemu nadaje się do detekcji punktów charakterystycznych w czasie rzeczywistym. Jednakże, algorytm ten jest wrażliwy na szum i nie jest skalowalny.



Podobieństwa do 031 (FAST)



Podobieństwa do 032 (FAST)



Podobieństwa do 033 (FAST)



Podobieństwa do 034 (FAST)



Podobieństwa do 035 (FAST)



Podobieństwa do 036 (FAST)



Podobieństwa do 037 (FAST)



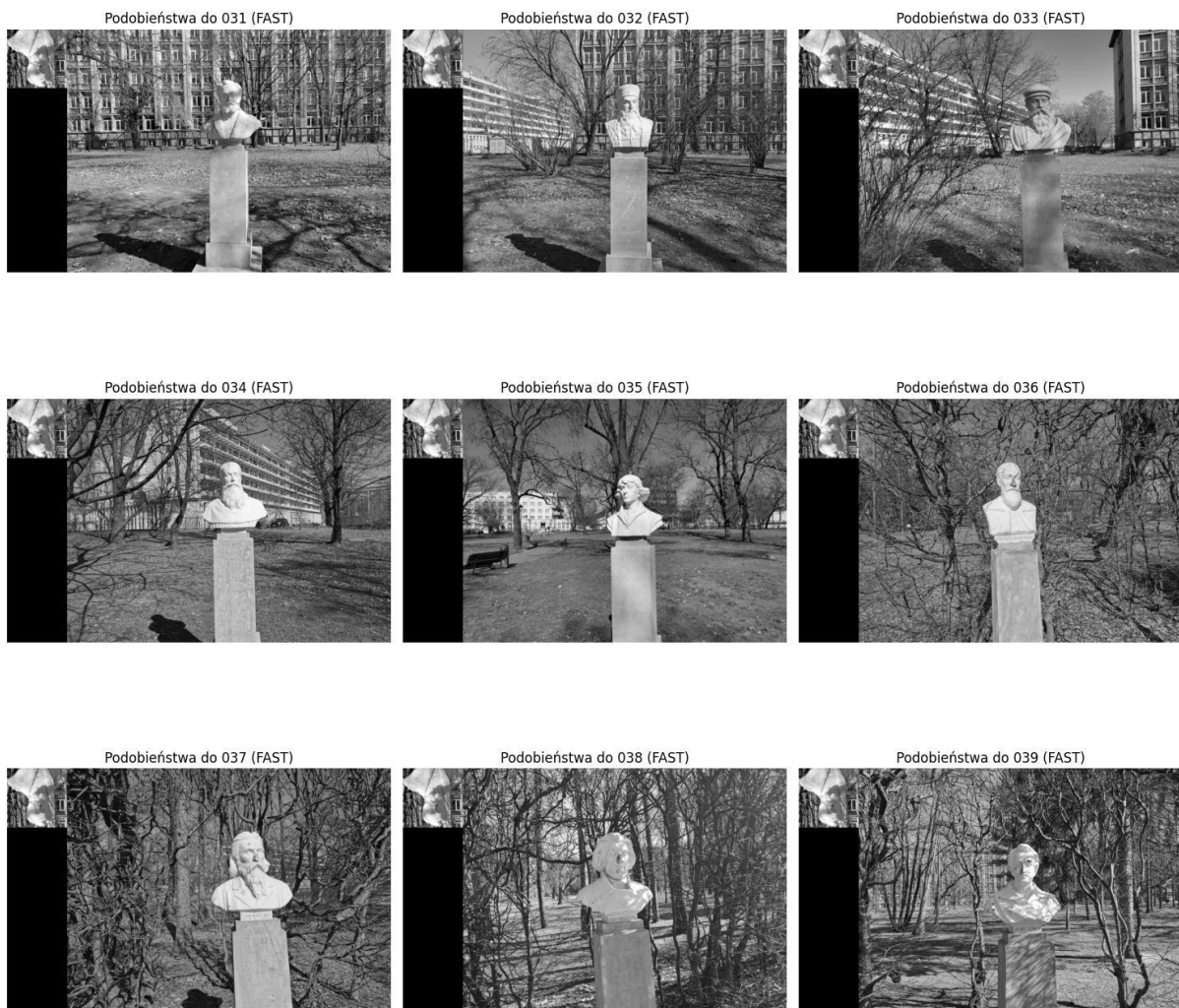
Podobieństwa do 038 (FAST)



Podobieństwa do 039 (FAST)



*Rysunek 8 Podobieństwa FAST*



*Rysunek 9 Obrócony FAST*

Algorytm FAST bardzo dobrze poradził sobie w znalezieniu odpowiedniego obrazu dla oryginalnego obrazu, w przypadku innych obrazów nie widzi żadnych cech podobieństwa. Ciekawe wyniki daje on dla obrazu obróconego – żadne cechy podobieństwa nie są widoczne.

## **Wnioski i przemyślenia**

- Istnieje wiele algorytmów służących do znalezienia cech podobnych do obrazu, które następnie można wykorzystać do odnalezienia zdjęcia z którego pochodził fragment.
- Każdy algorytm jest inny i może zostać wykorzystany do innej specyfiki problemu.
- Jedynie algorytm ORB nie poradził sobie z zagadnieniem. Pozostałe 3 potrafią odnaleźć obraz jednak każdy z nich ma inne możliwości.
- Gdy nie zmieniamy obrócenia pomiędzy fragmentem, a oryginałem najlepszy wydają się być algorytm FAST – jest on również najbardziej czuły, co powoduje brak podobieństwa dla obrazu obróconego.
- Gdy istnieje możliwość zmiany kąta pomiędzy fragmentem, a oryginałem najlepiej radzi sobie SIFT.
- Algorytm BRISK jest bardziej czuły od jego pierwowzoru FAST na przesunięcia i kąty. Znajduje wiele podobieństwa na każdym ze zdjęć, jednak pozwala łatwo wyodrębnić oryginalny obraz (zarówno obrócony jak i nie).
- Przeprowadzone rozważania świadczą o pomyślności wykonanego projektu.



