

# **Projekt**

## **Algorytmy przetwarzania obrazów**

**Temat: Klasyfikator owoców**

Autorzy: Bartosz Palmer 260346

Nataniel Jargiło 252888

## 1. Wstęp

Celem projektu jest stworzenie systemu klasyfikacji owoców na podstawie ich cech wizualnych, wykorzystując klasyczne metody przetwarzania obrazów. System ten opiera się na ekstrakcji cech, takich jak kolor, kształt, tekstura oraz krawędzie, i wykorzystuje klasyfikator K-NN do przypisywania obrazów do odpowiednich klas.

Ze względu na dużą liczbę klas w zbiorze danych Fruits 360 (ponad 140 klas), wybrano 50 klas owoców, aby zmniejszyć czas przetwarzania i przyspieszyć działanie algorytmów. Ograniczenie to pozwala również na dokładniejsze skupienie się na kluczowych aspektach projektu, takich jak optymalizacja ekstrakcji cech i ewaluacja modelu.

## 2. Wykonane prace i aktualny stan projektu

W ramach dotychczasowych działań zrealizowano następujące zadania:

### 2.1. Przygotowanie zbioru danych

Dane zebrano ze zbioru Fruits 360, organizując je w strukturze folderów. Foldery te zawierają pod foldery odpowiadające 50 wybranym klasom owoców. Przygotowano zbiór treningowy oraz testowy z odpowiednio podzielonymi danymi, zgodnie z oryginalnym podziałem w zbiorze Fruits 360.

### 2.2. Implementacja ekstrakcji cech

Zaimplementowano algorytmy, które wyodrębniają kluczowe cechy wizualne obrazów. W ramach tego wykonana została konwersja kolorów do przestrzeni HSV, dla których przeprowadzono analizę histogramów. Wykorzystana została również obliczanie cech tekstury owoców za pomocą macierzy współwystępowania poziomów szarości (GLCM) oraz detekcja krawędzi metodą Canny, na podstawie której wyznaczana jest gęstość krawędzi.

### 2.3. Trening klasyfikatora K-NN i wstępna ewaluacja

Na podstawie wyekstrahowanych cech przeprowadzono trenowanie klasyfikatora K-Nearest Neighbors (K-NN) z pięcioma sąsiadami. Klasyfikator został przetestowany na zbiorze walidacyjnym, uzyskując dokładność na poziomie około 59%. Wstępna ewaluacja systemu została przeprowadzona z wykorzystaniem metryk dokładności, F1-score oraz macierzy pomyłek. Wyniki wskazują, że klasyfikator skutecznie identyfikuje większość klas owoców, jednak zauważono trudności w przypadku wizualnie podobnych gatunków.

Aktualnie system działa poprawnie, a kluczowe elementy procesu, takie jak ekstrakcja cech i klasyfikacja, zostały zaimplementowane. W dalszej części projektu planowane są działania mające na celu dalszą optymalizację oraz szczegółową analizę wyników na zbiorze testowym.

## 3. Opis zaimplementowanych algorytmów

Ekstrakcja cech wizualnych rozpoczyna się od zmiany rozmiaru obrazu na 100x100 px, w celu zapewnienia jednolitości danych. Następnie obraz jest konwertowany do przestrzeni barw HSV, a histogram kolorów jest obliczany z podziałem na 8 przedziałów dla każdego z trzech kanałów (H, S, V), co daje histogram o rozmiarze 512 elementów. Zakresy wartości kanałów wynoszą

odpowiednio [0, 180] dla H oraz [0, 256] dla S i V. Histogram jest normalizowany i zapisany jako wektor cech opisujących rozkład barw w obrazie.

Obraz jest również przekształcany do skali szarości w celu analizy tekstury. Za pomocą macierzy współwystępowania poziomów szarości (GLCM) obliczane są cechy tekstury: kontrast, niepodobieństwo, jednorodność, energia, korelacja oraz średnia energia (ASM). Parametry używane w GLCM obejmują odległość między pikselami wynoszącą 1 oraz kąt analizy równy 0 stopni. Wartości w macierzy są znormalizowane, a liczba poziomów szarości wynosi 256.

Dodatkowo przeprowadzana jest detekcja krawędzi za pomocą algorytmu Canny, w którym ustawiono wartości progów na 100 (dolny próg) i 200 (górny próg). Obliczana jest proporcja pikseli należących do krawędzi w całym obrazie, co pozwala na wyznaczenie gęstości krawędzi.

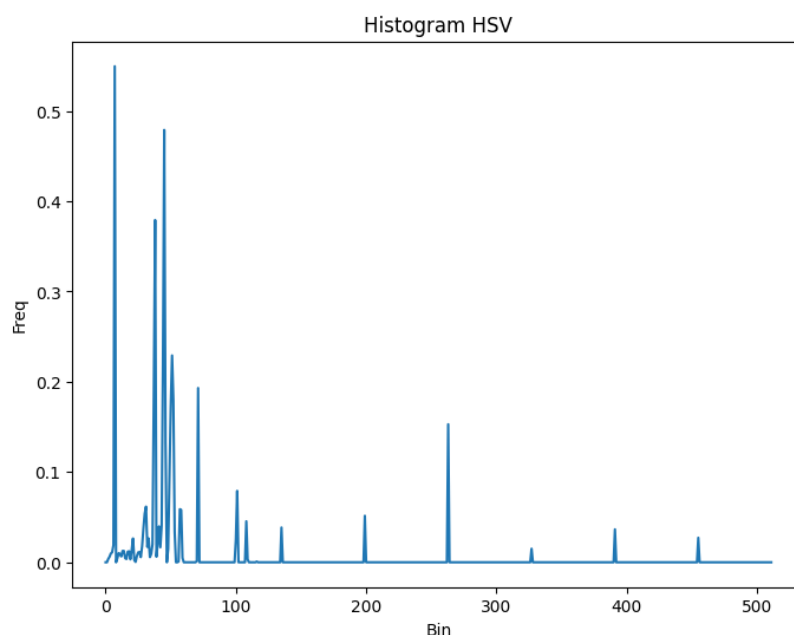
Wszystkie wyekstrahowane cechy – histogram kolorów (512 elementów), cechy tekstury (6 elementów) oraz gęstość krawędzi (1 element) – są łączone w wektor cech o łącznym rozmiarze 519 elementów. Wektor ten jest następnie używany w procesie klasyfikacji.

Do klasyfikacji zastosowano algorytm K-NN, który przypisuje próbkę do klasy na podstawie 5 najbliższych sąsiadów w przestrzeni cech. Algorytm korzysta z odległości euklidesowej do porównywania próbek.

#### 4. Wyniki

Na rysunku 1. Przedstawiono przykładowy histogram kolorów obrazu w przestrzeni barw HSV. Histogram ten został obliczony jako rozkład wartości kolorów w trzech kanałach: odcień (hue), nasycenie (saturation) oraz jasność (value). Przedstawiony histogram składa się z 512 binów, wynikających z podziału każdego kanału na 8 przedziałów.

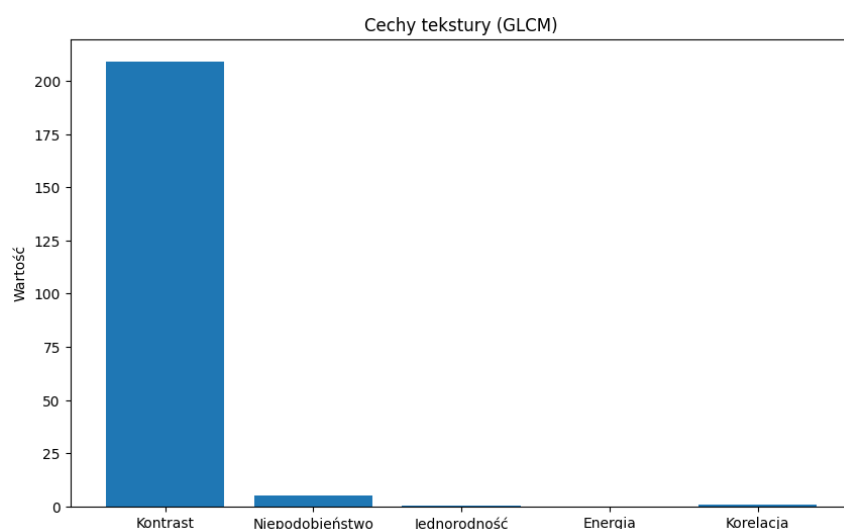
Histogram pokazuje, że największa częstotliwość występuje w niskich wartościach kanału odcienia (Hue), co wskazuje na dominację określonych barw charakterystycznych dla analizowanego obrazu. Dalsze peak'i w wyższych zakresach mogą reprezentować mniejsze składowe barwne obrazu. Niskie wartości nasycenia i jasności są widoczne jako niewielkie składowe w odpowiednich zakresach histogramu.



Rys. 1. Przykładowy histogram HSV

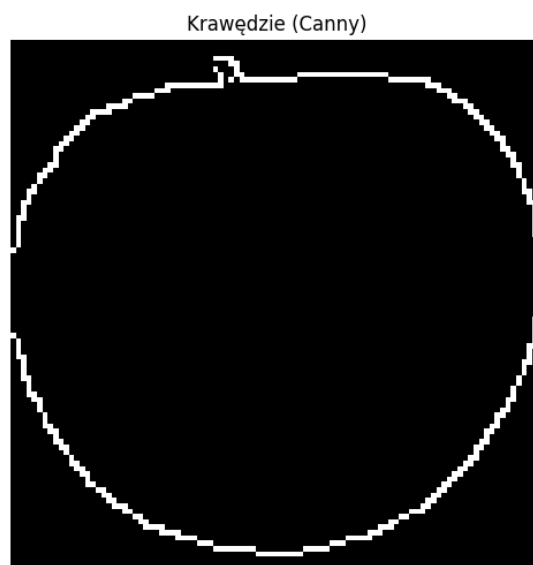
Na rysunku 2. przedstawiono na przykładzie wartości wybranych cech tekstury obliczonych na podstawie macierzy współwystępowania poziomów szarości (GLCM). Analiza obejmuje pięć głównych parametrów: kontrast, niepodobieństwo, jednorodność, energię oraz korelację.

Kontrast osiągnął najwyższą wartość, co wskazuje na znaczną różnicę intensywności między sąsiednimi pikselami i wyraźne zmiany jasności obiektu. Niepodobieństwo, jako miara różnic poziomów szarości, jest niższe, ale zauważalne. Jednorodność i energia przyjęły bardzo niskie wartości, co świadczy o zróżnicowanej strukturze obiektu, a niska korelacja wskazuje na ograniczoną zależność między pikselami.



Rys. 2. Przykładowy wykres wartości cech tekstury

Na poniższym rysunku 3. przedstawiono wynik detekcji krawędzi w na przykładowym obiekcie (jabłku) przy użyciu algorytmu Canny. Metoda ta pozwala na wykrycie konturów, które wskazują na obszary obrazu o największych zmianach jasności. Parametry użyte w detekcji obejmują dolny próg 100 oraz górny próg 200. Uzyskany wynik pokazuje, że kontur obiektu został dokładnie wykryty, co pozwala na jego odseparowanie od tła. Krawędzie są wyraźnie zdefiniowane, i zgadzają się z krawędziami obiektu wejściowego, czyli jabłka.



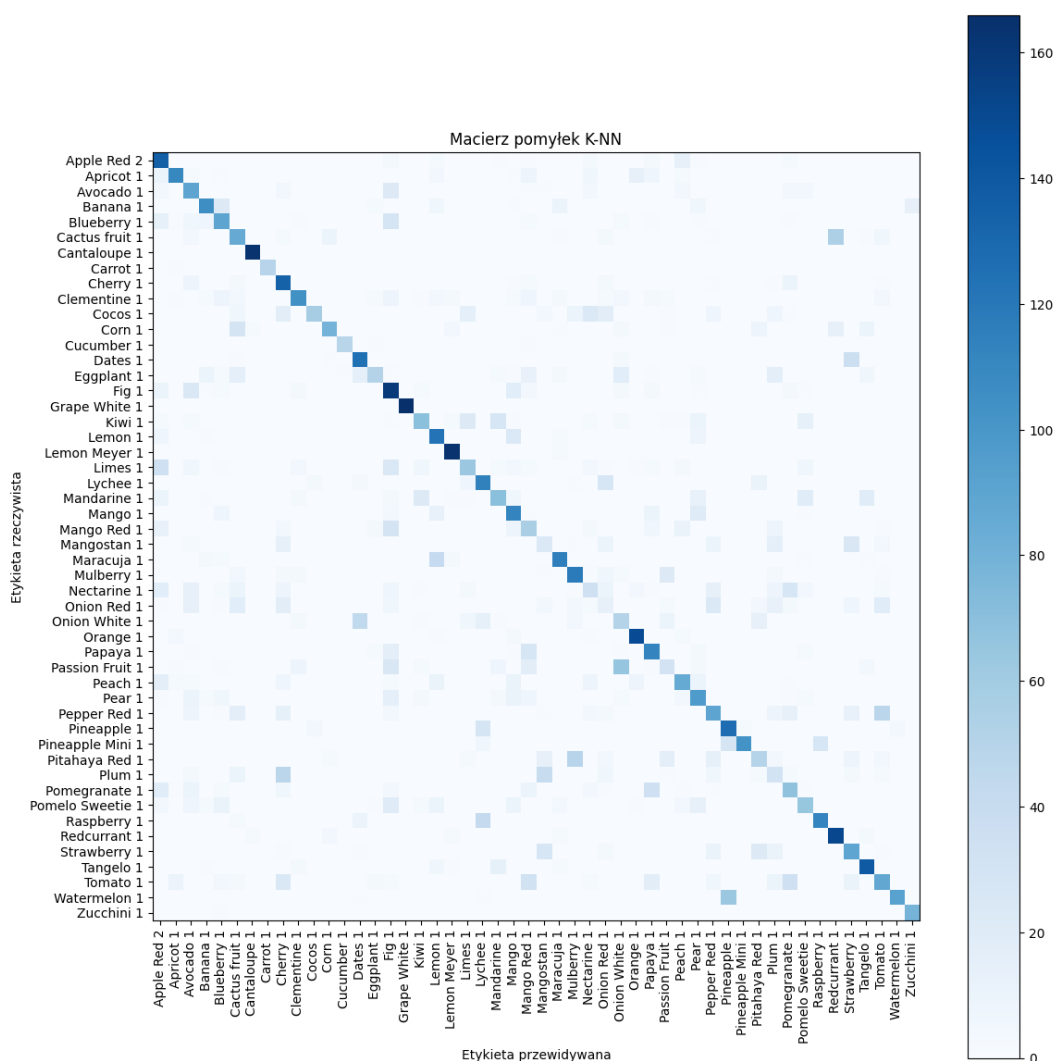
Rys. 3. Przykładowy wynik algorytmu Canny

Klasyfikator K-Nearest Neighbors (K-NN) został przetestowany na zbiorze testowym zawierającym 50 klas owoców, a jego skuteczność oceniono za pomocą miar takich jak dokładność, precision, recall, F1-score oraz analizy macierzy pomyłek.

- Dokładność modelu, definiowana jako odsetek poprawnie sklasyfikowanych próbek, wyniosła 59%. Wskazuje to, że model w dużej mierze prawidłowo przypisuje próbki do odpowiednich klas.
- Szczegółowy raport klasyfikacji wykazał, że klasyfikator osiąga zadowalającą precyzję i czułość dla większości klas, co oznacza skuteczność w rozróżnianiu różnych kategorii owoców. Jednak dla niektórych klas zaobserwowano niższe wartości F1-score, co może wynikać z podobieństwa cech wizualnych między owocami lub niewystarczającej liczby próbek w zbiorze treningowym.

Na poniższym rysunku 4. przedstawiono macierz pomyłek, która ilustruje, jak często etykiety przewidywane przez model zgadzały się z rzeczywistymi etykietami oraz gdzie wystąpiły błędy klasyfikacji.

- Zauważamy, że wartości na przekątnej macierzy są dominujące, co wskazuje, że model poprawnie klasyfikuje większość próbek do odpowiednich klas.
- Niektóre klasy (np. Apple Red 2) mają wyraźnie więcej próbek niż inne, co może wpływać na równowagę w wynikach klasyfikatora.



Rys. 4. Macierz pomyłek K-NN

## 5. Planowane prace

W dalszej części projektu planowane są następujące prace, mające na celu poprawę wyników klasyfikacji:

- Wyrównanie ilości próbek dla klas
- Eksperymentowanie z różnymi wartościami parametru  $K$  w klasyfikatorze  $K$ -NN
- Eksperymentowanie z alternatywnymi miarami odległości dla  $K$ -NN, np. Manhattan
- Porównanie wyników z innym klasyfikatorem – w celu porównania z  $K$ -NN zostanie zaimplementowany klasyfikator SVM
- Stworzenie dodatkowych wizualizacji wyników