

Politechnika Warszawska
OŚRODEK KSZTAŁCENIA NA ODLEGŁOŚĆ - OKNO

Raport z realizacji projektu:
„Wizyjna kontrola jakości”

Przedmiot:
Systemy Wizyjne
Rok akademicki: 2020/2021, semestr zimowy

Wykonał:
inż. Adrian Kuligowski
numer albumu 267732

Kierunek: Automatyka, Robotyka i Informatyka Przemysłowa

Data przygotowania raportu: 21.01.2021 r

I. Realizacja wirtualnej kurtyny świetlnej.

Opracowanie i implementacja sposobu wykonywania zdjęć kolejnych produktów pojawiających się na obrazie z wirtualnej kamery - **wirtualnej kurtyny świetlnej** wyzwalaającej migawkę w momencie pojawienia się produktu. Kontrola jakości jest przeprowadzana na obrazach statycznych, pojedynczych klatkach sekwencji, a zatem ważne jest by każdy produkt został zarejestrowany tak by możliwa była dalsza analiza jego obrazu. Oczekiwanym efektem końcowym jest seria pojawiających się kolejno widoków kolejnych produktów na linii produkcyjnej.

- Omówienie celu

Aby zrealizować ten cel zostanie stworzona linia w połowie wysokości klatki wideo, której kolor będzie wartością maksymalną. Linia ta, w połączeniu z kolejnymi klatkami sekwencji, zostanie wykorzystana do operacji rekonstrukcji. W jej wyniku widoczne będą tylko obiekty, które są przecięte przez tę linię. A skoro produkty przemieszczają się pojedynczo, będzie widoczny tylko pojedynczy obiekt.

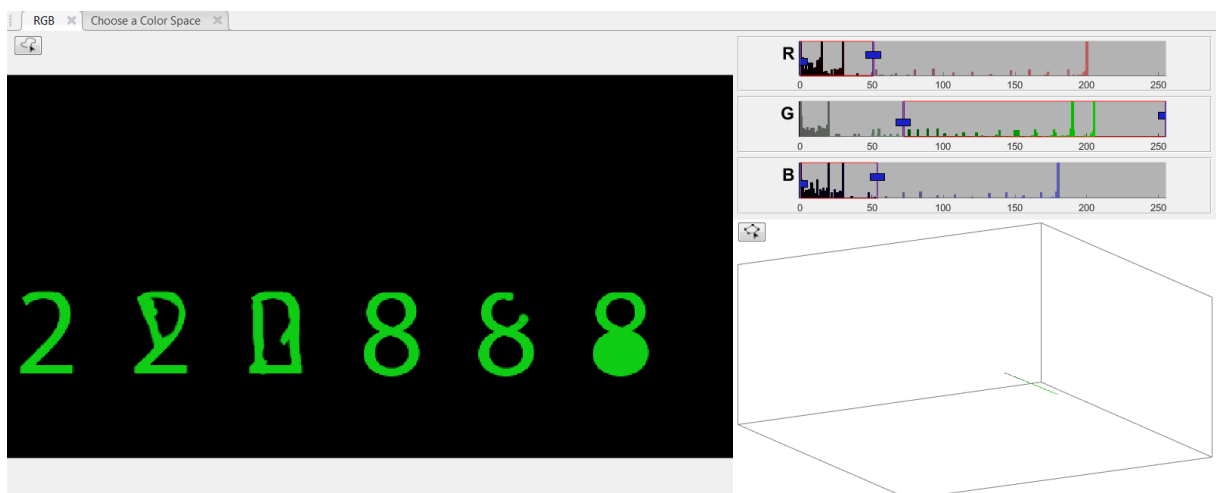
Jednak produkt przemieszczając się, przetnie się z linią detekcji nawet kilkadziesiąt razy, dlatego został postawiony dodatkowy warunek, że obiekt musi zostać wykryty 20 razy z rzędu przez tę linię (pozwoli to na ignorowanie małych obiektów, które nie są produktami). Zostanie to zrealizowane w następujący sposób: po rekonstrukcji następuje segmentacja i filtracja klatki sekwencji. Tak otrzymany obraz zostaje poddany operacji etykietowania. Jeśli 20 razy z rzędu liczba obiektów po etykietowaniu będzie większa niż 0, to nastąpi wydzielenie klatki sekwencji do dalszej obróbki.

Należy też określić parametry do segmentacji i filtracji. Sekwencja wideo składa się z kolorowych obrazów RGB (typ danych: uint8). Przeprowadzenie filtracji po segmentacji pozwoli na zmniejszenie ilości obliczeń (po segmentacji obrazu otrzymuje się jedną macierz, zamiast trzech). Zostanie wykorzystana metoda segmentacji przez progowanie kolorów.

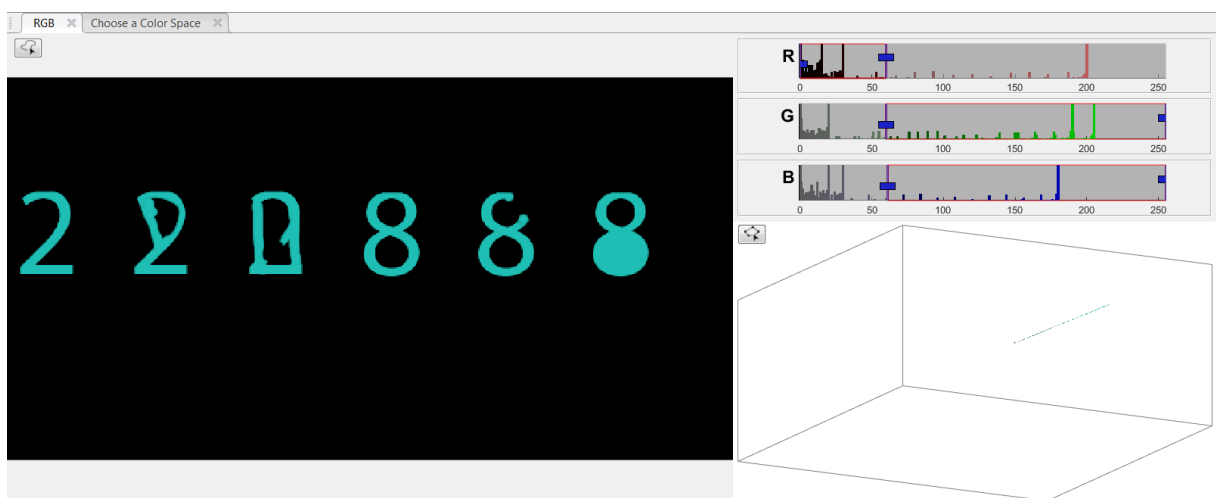
W celu określenia progów zostało wykorzystane narzędzie „Color Thresholder” i obraz „OKNO_1_ref.png” zawierający widok produktów. Na poniższych rysunkach przedstawiono uzyskane progi dla produktów o poszczególnych kolorach.



Rysunek 1 – Progi produktów o kolorze czerwonym



Rysunek 2 – Progi produktów o kolorze zielonym



Rysunek 3 – Progi produktów o kolorze cyjan

Na podstawie analizy kolorów, określono progi o wartości 55 dla każdej składowej RGB. Jest tak ponieważ, na tym etapie nie ma potrzeby wyodrębniać obiektów o jednej konkretnej barwie (wręcz przeciwnie, trzeba wyodrębnić wszystkie kolory jednocześnie znajdujące się w wzorcu).

Do filtracji obrazu zostanie wykorzystany filtr medianowy. Należy pamiętać, że może on zniekształcać krawędzie (przy małej masce efekt może być pomijalny). Oglądając sekwencję wideo nie widać dużych zakłóceń, dlatego możliwe byłoby pominięcie filtracji. Jednak w opisie drugiego celu znajduje się polecenie zastosowania filtracji, dlatego zostanie ona przeprowadzona.

- Opis opracowanego algorytmu

Na początku zostaje wczytana sekwencja wideo do zmiennej „v” i utworzona pusta macierz 3-wymiarowa o rozmiarze klatki wideo. Na tej macierzy zostaje stworzona linia w połowie wysokości (we wszystkich 3 wymiarach).

Stworzenie figury o nazwie „f” do rysowania obrazu. Rozpoczęcie pętli while, która wczytuje kolejne klatki z „v” (podobnie jak w przykładzie „czytajwideo.m”). Po wybraniu figury, zostaje wyświetlona obecnie wczytana klatka wraz z widokiem linii detekcji.

Obecnie wczytana klatka zostaje poddana rekonstrukcji, gdzie znacznikiem jest linia detekcji. Na uzyskanym obrazie zostaje przeprowadzona segmentacja przez kolor: każdy wymiar obrazu RGB zostaje porównany do progu o wartości 55, a następnie dodaje się uzyskane obrazy binarne do siebie, co daje końcowy wynik segmentacji.

Obraz posegmentowany zostaje poddany filtracji z użyciem filtra medianowego 3x3. Wynik tej operacji zostaje poddany etykietowaniu w celu określenia, czy został wykryty jakiś obiekt. Jeśli 20 razy z rzędu zostanie wykryty jakiś obiekt, to następuje wybranie klatki z sekwencji wideo do dalszego przetwarzania.

- Opis implementacji

Program realizujący wirtualną kurtynę świetlną został napisany w środowisku MATLAB 2019b. Zawiera się on w programie, który realizuje wszystkie cele z projektu. Program nie stanowi odrębnej aplikacji i uruchamia się go z poziomu oprogramowania MATLAB. Większość działań jest realizowanych podczas pętli wczytującej kolejne klatki sekwencji.

Funkcjonalność dotycząca wirtualnej kurtyny świetlnej korzysta z następujących zmiennych:

- v* – obiekt służący do wczytywania plików wideo, w tym przypadku pliku 'OKNO_1.avi'
- line* – wirtualna linia detekcji: macierz 3-wymiarowa o rozmiarze klatki wideo, zawierająca w połowie poziomą linię
- f* – figura do wyświetlania obrazów
- vidFrame* – zmienna przechowująca aktualnie wczytaną klatkę wideo
- im* – klatka wideo po rekonstrukcji, pokazuje pojedynczy obiekt przecinany przez linię detekcji

s_R, s_G, s_B – obrazy binarne, będące wynikiem segmentacji obrazu „im” w poszczególnych składowych RGB

seg – wynik końcowy segmentacji, suma obrazów binarnych w poszczególnych składowych

seg_f – obraz posegmentowany i poddany filtracji

e – obraz z etykietami, uzyskany z etykietowania obrazu „seg_f”

num – liczba obiektów uzyskanych po segmentacji; jeśli nie ma żadnych obiektów, nie są wykonywane dalsze operacje

i – zmienna do zliczania ilości wykrytych obiektów pod rząd w wyniku etykietowania, jeśli nie jest wykryty żaden obiekt, to wartość „i” jest zerowana

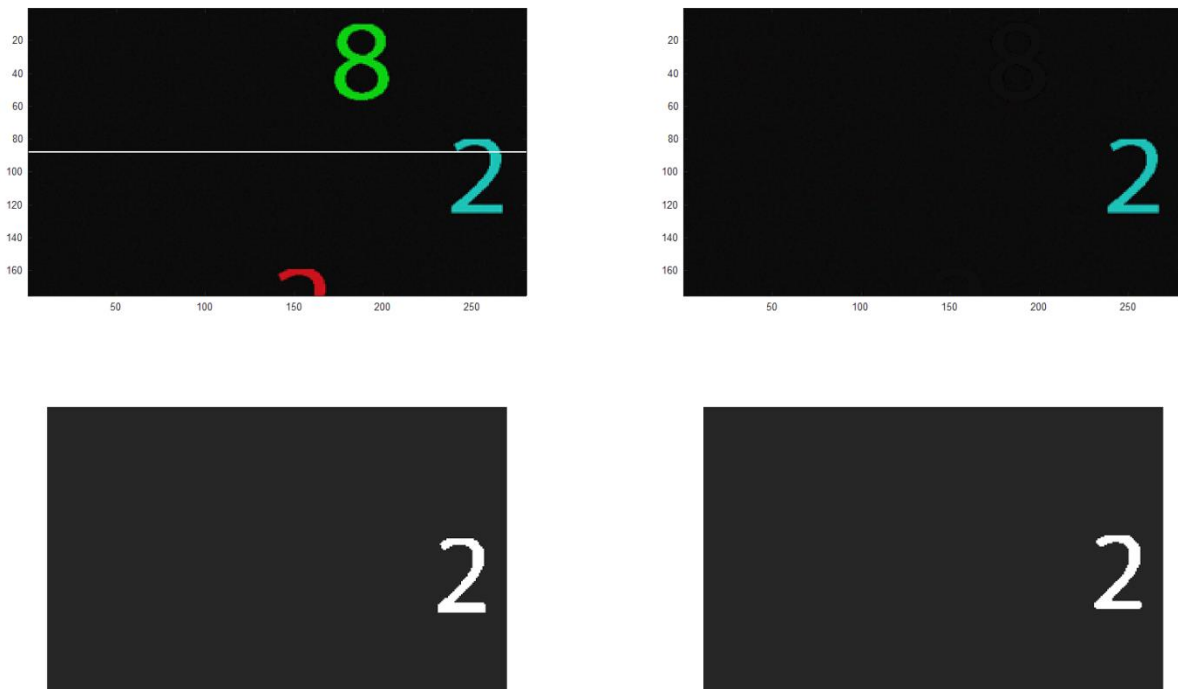
Dodatkowo wyświetlana jest obecnie wczytana klatka sekwencji, aby móc obserwować na bieżąco na jakim etapie jest przetwarzanie sekwencji wideo.

Należy pamiętać, aby program uruchomić w folderze, gdzie znajduje się plik „OKNO_1.avi”.

- Przedstawienie wyników

Końcowym efektem wirtualnej kurtyny świetlnej jest dalsze działanie na pojedynczej klatce sekwencji. Dzięki etykietowaniu, można w dalszej kolejności w prosty sposób wyznaczyć cechy obrazu.

Na poniższym rysunku przedstawiono uzyskane wyniki w kolejnych operacjach.



Rysunek 4 – Wyniki uzyskane w kolejnych operacjach: obraz wczytany z sekwencji z linią detekcji, obraz po rekonstrukcji, obraz po segmentacji, obraz po filtracji.

Napisany kod jest fragmentem całego projektu i zostanie dołączony jako całość.

- Wnioski

Realizując ten zostały wykonane operacje z celów następnych. Dlatego w dalszych pracach, będzie można skorzystać z wcześniej uzyskanych wyników.

Obróbka każdej klatki sekwencji po kolei (rekonstrukcja, segmentacja, filtracja, etykietowanie) może wymagać większych nakładów obliczeniowych. Można zastanowić się nad innym sposobem pobierania pojedynczej klatki z sekwencji. Możliwym rozwiązaniem jest pobieranie klatki co równy odstęp czasu. Jednak wymagałoby to jednostajnej prędkości przemieszczania się obiektów, a także jednakowych odstępów między produktami.

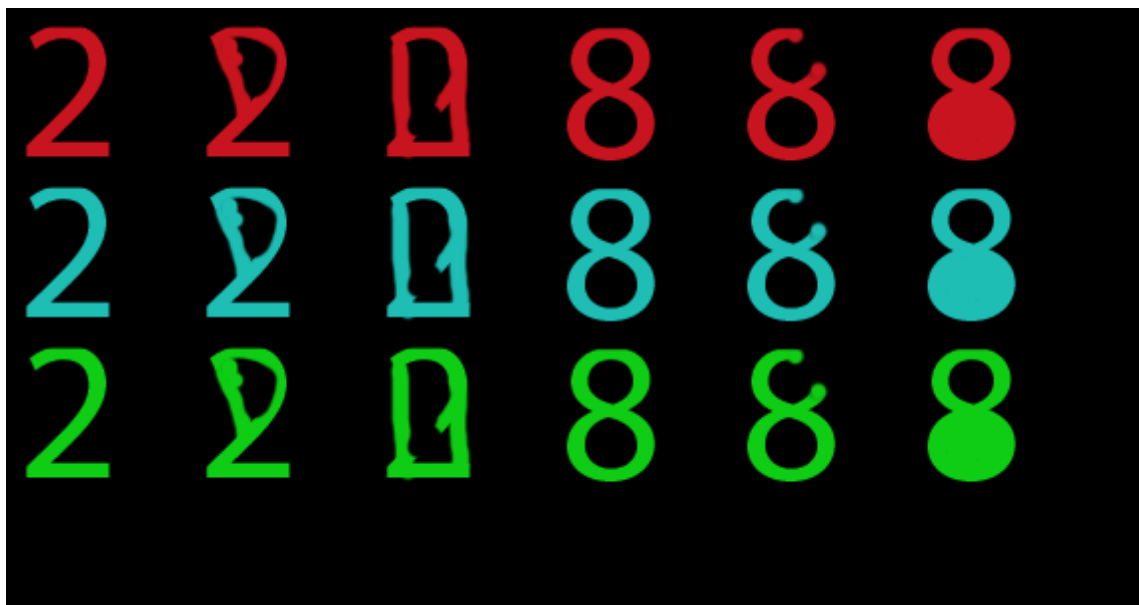
Użycie pojedynczej klatki sekwencji do dalszej obróbki może być ryzykowne. Nie ma gwarancji, że akurat tak wybrany obraz nie będzie akurat zniekształcony lub źle wykonany.

II. Realizacja metody kontroli jakości.

Opracowanie i implementacja **metody kontroli jakości** obrazu statycznego zawierającej produkt(y) składającej się z etapów filtracji, segmentacji oraz ekstrakcji cech obiektu oraz – na ich podstawie – jego klasyfikacji do odpowiedniej kategorii. Kategoriami są klasy odpowiadające poszczególnym produktom prawidłowym w poszczególnych kolorach (6 klas) oraz produktom wadliwym (bez rozróżniania ani koloru ani rodzaju wadliwości – 1 klasa).

- Omówienie celu

W pierwszej kolejności poddany zostanie analizie obraz przedstawiający wszystkie możliwe produkty – „OKNO_1_ref.png”.



Rysunek 5 – Obraz przedstawiający wszystkie rodzaje produktów.

Warto zauważyć, że wszystkie produkty wadliwe posiadają tylko jeden otwór. Natomiast produkty prawidłowe (kolumny 1 i 4) mają dwa otwory lub są pozbawione otworów. Cechą która można zostać wyznaczona za pomocą funkcji „regionprops”, uwzględniającą liczbę otworów w obiekcie jest liczba Eulera. Za pomocą tej cechy będzie można rozróżnić produkty wadliwe od prawidłowych, a także czy produkt jest dwójką bądź ósemką.

Należy też określić kolor produktu. Zostanie to zrealizowane za pomocą funkcji „max” i widoku wykrytego obiektu. Pozwoli to na określenie wartości koloru w formacie RGB. Wartość ta zostanie porównana do wartości progów kolorów uzyskanych podczas realizacji poprzedniego celu.

Aby analizować jak najmniejszy fragment klatki sekwencji, zawierający cały obiekt, zostanie wyznaczona cecha „BoundingBox” obiektu. Zawiera ona informację o najmniejszym prostokącie w którym znajduje się obiekt uzyskany w procesie etykietowania.

Skoro filtracja, segmentacja i etykietowanie zostają wykonane w czasie wcześniejszego zadania, nie ma potrzeby realizacji ich na tym etapie – zostaną wykorzystane wyniki uzyskane wcześniej.

Biorąc pod uwagę wcześniejsze informacje została opracowana tabela opisująca produkty w poszczególnych kategoriach:

Tabela 1 – Cechy produktów w poszczególnych kategoriach

Kategoria \ Cecha	L. eulera	Nazwa koloru	Wartość koloru w przestrzeni RGB		
			R	G	B
1	1	czerwony	> 60	< 55	< 55
2	1	cyjan	< 55	> 55	> 55
3	1	zielony	< 55	> 70	< 55
4	-1	czerwony	> 60	< 55	< 55
5	-1	cyjan	< 55	> 55	> 55
6	-1	zielony	< 55	> 70	< 55
7	0	-	-	-	-

- Opis opracowanego algorytmu

Po wykonaniu wcześniejszych działań, został określony obraz statyczny zawierający wykryty produkt. Operacje segmentacji, filtrowania i etykietowania zostały już wykonane wcześniej i dlatego wykorzystane zostaną ich rezultaty.

Wynik etykietowania zostaje wykorzystany do ekstrakcji następujących cech obiektu: liczba Eulera, „BoundingBox” i pole powierzchni (pole będzie potrzebne do realizacji następnych celów. Za pomocą danych z cechy „BoundingBox” wykryty obiekt zostanie wycięty z klatki wideo, a następnie wyświetlony obok klatki wideo. Widok wyciętego produktu posłuży do określenia jego wartości koloru. Aby to zrobić zostaną określone wartości maksymalne każdym wymiarze macierzy obrazu, co daje wartość koloru w formacie RGB. Na podstawie uzyskanych wartości koloru zostanie określona nazwa koloru produktu.

W dalszej kolejności produkt zostanie przypisany do odpowiedniej dla niego kategorii. Odbędzie się to porównując najpierw jego liczbę Eulera, a następnie kolor. Dla produktów kategorii 7 wystarczy porównać tylko jego liczbę Eulera. Dodatkowo jeśli l. Eulera będzie różna od 1, -1 lub 0 obiekt także zostanie zaliczony do kategorii 7.

- Opis implementacji

W celu łatwiejszego opisu wykrytego produktu zostanie stworzona struktura o nazwie „produkt” zawierająca następujące pola danych: pole powierzchni, liczba Eulera, wartość koloru, nazwa koloru i kategoria produktu. Pole powierzchni, l. Eulera i nr kategorii będą liczbami całkowitymi. Wartość koloru będzie miała postać tablicy trzy wymiarowej (w każdym wymiarze będzie jeden element), co odpowiada składowym RGB. Nazwa koloru będzie reprezentowana jako tablica znaków.

Po tym jak obiekt zostanie wykryty 20 razy z rzędu, wynik etykietowania zostaje użyty do wyznaczenia cech za pomocą funkcji „regionprops” (cechy: 'Area', 'BoundingBox' i 'EulerNumber'). Wartości cech 'Area' i 'EulerNumber' zostają zapisane w strukturze „produkt”. W czasie wybrania pojedynczego obrazu z sekwencji zostaje zwiększony licznik wykrytych obiektów. Będzie on służył do indeksowania struktury „produkt”.

Z cechy 'BoundingBox' zostają zapisane do odrębnych zmiennych współrzędne x i y ramki obiektu (ma to na celu głównie uproszczenie zapisu kodu). Następnie zostaje wyodrębniony produkt z klatki wideo jako oddzielny obraz. Na jego podstawie i za pomocą funkcji `max` zostaje określona wartość koloru w formacie tablicy o rozmiarze 1x1x3. Funkcja `max` jako argumenty wejściowe przyjmuje obraz wykrytego obiektu i numery wymiarów 1 i 2.

Po określeniu wartości koloru, zostaje wyznaczona nazwa koloru. Odbyna się to za pomocą spełnienia warunku pętli `if`. Warunkiem jest spełniony, jeśli wartości poszczególnych składowych RGB są większe lub równe od tych określonych w tabeli 1.


Dla przykładu pokazane zostanie określenie nazwy koloru czerwonego.






```
if ((produkt(nr).kolor_wartosc(1,1,1) > 60) &&...
    (produkt(nr).kolor_wartosc(1,1,2) < 55) &&...
    (produkt(nr).kolor_wartosc(1,1,3) < 55))
    produkt(nr).kolor = 'czerwony';
end
```

W celu określenia kategorii produkty, najpierw zostaje porównanie wartości liczby Eulera wykrytego produktu, co odbywa się za pomocą trzech pętli `if` (porównanie do wartości: -1, 1 oraz 0 i pozostałych, co daje trzy grupy). Jeśli wartość będzie równa -1, to następnie zostaje porównana nazwa koloru (zgodnie z tabelą 1) i w zależności od spełnionego warunku zostaje określona kategoria produktu. Tak samo dzieje się jeśli liczba Eulera będzie równa 1.

- Przedstawienie wyników

W wyniku realizacji tego celu otrzymuje się strukturę, w której znajdują się kolejne produkty. Poniżej przedstawiono strukturę dla kilku pierwszych produktów.

 1x3 struct with 5 fields

...	 pole	 euler	 kolor	 kolor_wartosc	 kategoria
1	510	1	'czerwony'	1x1x3 uint8	1
2	502	1	'cyjan'	1x1x3 uint8	2
3	739	-1	'zielony'	1x1x3 uint8	6
4					
5					
6					
7					

Rysunek 6 – Widok wypełnionej struktury „produkt” dla kilku kolejnych obiektów.

Otrzymany zostaje też widok ostatnio wykrytego produktu. Na poniższym rysunku pokazano przykładowy produkt.



Rysunek 7 – Widok ostatnio wykrytego obiektu.

- Wnioski

Uzyskany widok obiektu wyciętego z całej klatki sekwencji dopełnia realizację celu poprzedniego. Wcześniej uzyskano pojedynczą klatkę sekwencji z całym tłem, które nie miało znaczenia, a rozmiar obrazu był większy niż to konieczne. Teraz uzyskano mniejszy obraz, ale pokazujący cały obiekt.

Do struktury są zapisywane wszystkie dane, niezależnie od kategorii produktu. W przypadku produktów wadliwych dane dotyczące koloru i pola powierzchni nie są wymagane.

Sposób implementacji może nie być wydajny pod względem użycia pamięci, ale w poleceniu nie było konieczności optymalizacji pod tym względem.

Istniała też możliwość wykorzystania niezmienników momentowych lub transformacji Fouriera do porównania wykrytego obiektu z wzorcem (jako wzorzec posłużyłby plik „OKNO_1_ref.png”). Jeśli produkt zgadzałby się z którymś kształtem z pliku z wzorcami, można by określić czy jest to dwójka, ósemka lub produkt wadliwy.

III. Realizacja metody zbierania danych.

Opracowanie i implementacja metody **zbierania danych** na temat całej partii produktów (kompletnej sekwencji wideo) – kategorii kolejnych produktów wraz ze zliczaniem produktów w poszczególnych kategoriach. Dla produktów prawidłowych w każdej kategorii powinny być także zliczane ich pola powierzchni.

- Omówienie celu

Określenie i zapis kategorii kolejnych produktów zostało zrealizowane podczas wcześniejszego celu – kategoria produktu znajduje się w strukturze „produkt”.

W celu realizacji zliczania produktów w poszczególnych kategoriach wystarczy zastosować zmienne, których wartość będzie zwiększała się o 1, gdy zostanie wykryty produkt o danej kategorii.

Pola powierzchni zostały zliczone wcześniej i znajdują się w strukturze produkt. Z tą różnicą, że w strukturze pola są liczone dla wszystkich kategorii. W razie potrzeby można odfiltrować niepotrzebne pola powierzchni.

Innym rozwiązaniem jest analiza struktury „produkt” po przetworzeniu wszystkich klatek sekwencji wideo. Jednak liczenie produktów na bieżąco daje większe możliwości i jest łatwiejsze do testowania w czasie opracowywania programu.

- Opis opracowanego algorytmu

W celu zliczania produktów w poszczególnych kategoriach na początku zostanie stworzony wektor 7 elementowy o nazwie „produkty_ilosc”. Numer elementu odpowiada numerowi kategorii. Jego początkowa wartość jest równa zero.

Po wykryciu obiektu i określeniu jego kategorii, zostaje zwiększona ilość produktów w danej kategorii. Odbyna się to przez wybranie danego elementu wektora „produkty_ilosc” (numer kategorii aktualnie wykrytego produktu zostaje użyty jako index wektora) i zwiększenie jego wartości o 1.

Aby odfiltrować pola powierzchni produktów prawidłowych można stworzyć tablicę dwuwymiarową, gdzie będą zapisywane wartości pól powierzchni obiektów z kategorii 1-6. Jeden wymiar określałby numer kategorii, a drugi numer produktu w danej kategorii.

- Opis implementacji

Przed pętlą wczytującą klatki sekwencji stworzono zmienną „produkty_ilosc” o rozmiarze 1x7 zawierającą same zera. Po określeniu kategorii, następuje zwiększenie wartości elementu (o indexie równym numerowi kategorii) zmiennej „produkty_ilosc” o 1.

Uznano, że nie ma potrzeby filtracji pól powierzchni wykrytych produktów i dlatego nie została ona zrealizowana. Zdecydowano tak, ponieważ nie ma potrzeby tworzenia dodatkowej zmiennej, a wartości pól powierzchni z kategorii 1-6 zostaną pobrane z struktury „produkt” w razie konieczności.

- Przedstawienie wyników

Wynikiem jest wektor „produkty_ilosc” zawierający informacje o ilości produktów w danej kategorii.

- Wnioski

Możliwe jest, że lepszym rozwiązaniem mogłoby być zliczenie produktów w danej kategorii po przetworzeniu sekwencji wideo. Problemem może być duża ilość działań, które zostają wykonane w momencie wykrycia produktu i określenia jego cech. Proponowane rozwiązanie może pomóc z problemem.

IV. Realizacja metody prezentacji wyników.

Opracowanie i implementacje metod **prezentacji wyników** w formie prezentacji na panelu operatorskim oraz danych zbiorczych dla całej partii produkcyjnej (pełnej sekwencji). Obraz na **wirtualnym panelu operatorskim** powinien zawierać widok ostatniego produktu na linii wraz z informacją o jego kategorii, kolejnym numerze w całej sekwencji oraz w swojej klasie. **Raport zbiorczy** powinien zawierać łączną liczbę przebadanych obiektów, liczbę obiektów w każdej z klas (6 „prawidłowych” i 1 „wadliwa”). Dla każdej klasy prawidłowej powinien zawierać także średnią powierzchnię obiektu oraz odchylenie standardowe pola powierzchni.

- Omówienie celu

Na wirtualnym panelu operatorskim znajdować będą się: widok ostatniego produktu na linii (uzyskany za pomocą cechy 'BoundingBox'), informacje o jego kategorii, kolejnym numerze w całej sekwencji oraz w swojej klasie (informacje te będą wyświetlone w postaci tekstu pod widokiem wykrytego obiektu). Dodatkowo będzie widoczna aktualnie przetwarzana klatka sekwencji wideo wraz z linią detekcji.

Informacja o kategorii obiektu znajduje się w strukturze „produkt” i zostanie wykorzystana do opisu ostatnio wykrytego obiektu. Numer obiektu w całej sekwencji zostanie przechowywany w postaci zmiennej, której wartość będzie zwiększała się z każdym wykryciem obiektu (czyli wybraniu pojedynczej klatki z sekwencji). Numer obiektu w swojej klasie jest wyznaczany podczas wcześniejszych działań, więc nie ma potrzeby jego ponownego liczenia.

Informacje potrzebne do raportu zbiorczego dotyczące liczby wszystkich przebadanych produktów i liczby obiektów w każdej z klas zostały zebrane podczas poprzednich działań. W celu obliczenia średniej powierzchni obiektu każdej klasy prawidłowej należy podzielić sumę pól powierzchni obiektów w danej kategorii przez ilość produktów w tej kategorii. Do obliczenia odchylenia standardowego pola powierzchni zostanie wykorzystana gotowa funkcja z MATLABa: `std`. Potrzebuje ona jako argumentu wektora zawierającego pola powierzchni obiektów w danej kategorii.

Raport zbiorczy może mieć postać nowego okna, gdzie zostanie wyświetlony tekst z odpowiednimi informacjami.

- Opis opracowanego algorytmu

Widok aktualnie wczytanej klatki sekwencji i ostatnio wykrytego obiektu został zrealizowany już wcześniej. Należy tylko dołączyć do wyświetlonego obrazu tekst z odpowiednimi informacjami, co będzie się odbywało w momencie detekcji obiektu, a dokładniej po określeniu wymaganych cech obiektu. Do zamieszczenia informacji posłuży funkcja `text`. Potrzebuje ona jako argumentu współrzędnych położenia tekstu na obrazie, łańcucha znaków i cech tekstu.

Położenie tekstu zostaje określone na podstawie rozmiaru wykrytego obiektu. Wysokość położenia tekstu jest równa wysokości widoku obiektu powiększona o 2, aby tekst nie stykał się z obrazem. Druga współrzędna określa położenie tekstu w poziomie, a skoro tekst powinien zaczynać się od lewej strony, jego wartość powinna być równa 1. Wartość tekstu do wyświetlenia zostaje przygotowana w zmiennej „str”. Zawiera ona odpowiednią treść do wyświetlenia, a informacje są pobierane z odpowiednich zmiennych i konwertowane na zmienną typu string. Tekst powinien być wyrównany do góry, aby jego kolejne wiersze pojawiały się poniżej określonych wcześniej współrzędnych.

Prawie w tym samym momencie zostaje obliczone średnie pole powierzchni dla obiektów w aktualnie wykrytej kategorii (tylko jeśli numer kategorii jest mniejszy niż 7). W tym celu na początku programu zostały przygotowane dwa wektory 6-elementowe: „suma_pole” i „sr_pole”. Jeden zawiera wartości sum pól powierzchni w poszczególnych kategoriach, a drugi wartości średniego pola w kolejnych kategoriach. Suma pola w aktualnie wykrytej kategorii jest zwiększana o wartość pola aktualnie wykrytego obiektu. Na podstawie sum pól i liczby produktów w kategoriach można obliczyć średnią wartość pola.

Obliczenie odchylenia standardowego pola powierzchni nastąpi po przetworzeniu całej sekwencji wideo. Wartości będą zapisywane w 6-elementowej zmiennej „S”. W celu obliczeń najpierw trzeba stworzyć wektor zawierający pola obiektów w jednej kategorii. Zostanie to zrealizowane za pomocą pętli for, w której będzie następował zapis wartości pola z struktury „produkt” do wektora, jeśli kategoria struktury będzie się zgadzała się z wybranym numerem kategorii. Po stworzeniu takiego wektora pól zostaje obliczone odchylenie standardowe i zapisane do zmiennej S. Procedura jest powtarzana dla kategorii od 1 do 6.

W dalszej kolejności zostaje przygotowany tekst do wyświetlenia. Odbywa się to za pomocą dwóch zmiennych tekstowych, w których są odpowiednie wyrażenia i wartości pobrane z zmiennych. Wartości liczbowe są konwertowane na zmienną typu string z istotnością trzech pierwszych cyfr.

W nowym oknie zostają wyświetlone wartości zmiennych tekstowych za pomocą funkcji `annotation` jako pola tekstowe. Należy też określić położenie tych pól tekstowych i ich rozmiary. Jedno z nich będzie miało rozmiar całego okna i zostanie umiejscowione w punkcie (0;0). Spowoduje to rozpoczęcie tekstu w górnym lewym rogu. Położenie drugiego okna tekstowego należy określić metodą prób i błędów, aby wyświetlony tekst było poprawnie położony w stosunku do poprzedniego. Końcowa postać tego okna będzie stanowiła raport zbiorczy, który należało uzyskać w czasie realizacji tego celu.

- Opis implementacji

Wartości średnich pól są obliczane na bieżąco, po każdym wykryciu obiektu. Ma to taką zaletę, że te wartości można łatwiej analizować w czasie testowania programu. Aby wyznaczyć pola tylko dla kategorii poprawnych, został zastosowany warunek, że obecnie wykryty obiekt musi mieć numer kategorii mniejszy od 7.

Tekst zawierający informacje o aktualnie wykrytym obiekcie został przygotowany w następujący sposób:

```
str = [{'Kategoria: ', num2str(produkt(nr).kategoria)], ...  
      ['Ilość produktów w sekwencji: ', num2str(nr)], ...  
      ['Ilość produktów w tej kategorii: ', num2str(produkty_ilosc(produkt(nr).  
                                                                kategoria))]];
```

Składa się on z 3 komórek tekstowych. Każda komórka oznacza nowy wiersz tekstu. Informacje o numerze kategorii, kolejności obiektu w całej sekwencji i ilości produktów w kategorii są pobrane z odpowiednich zmiennych i konwertowane na zmienną typu string.

W tym momencie nie ma więcej operacji znajdujących się w pętli przetwarzającej sekwencję wideo. Następne działania zostaną wykonane po zakończeniu pętli.

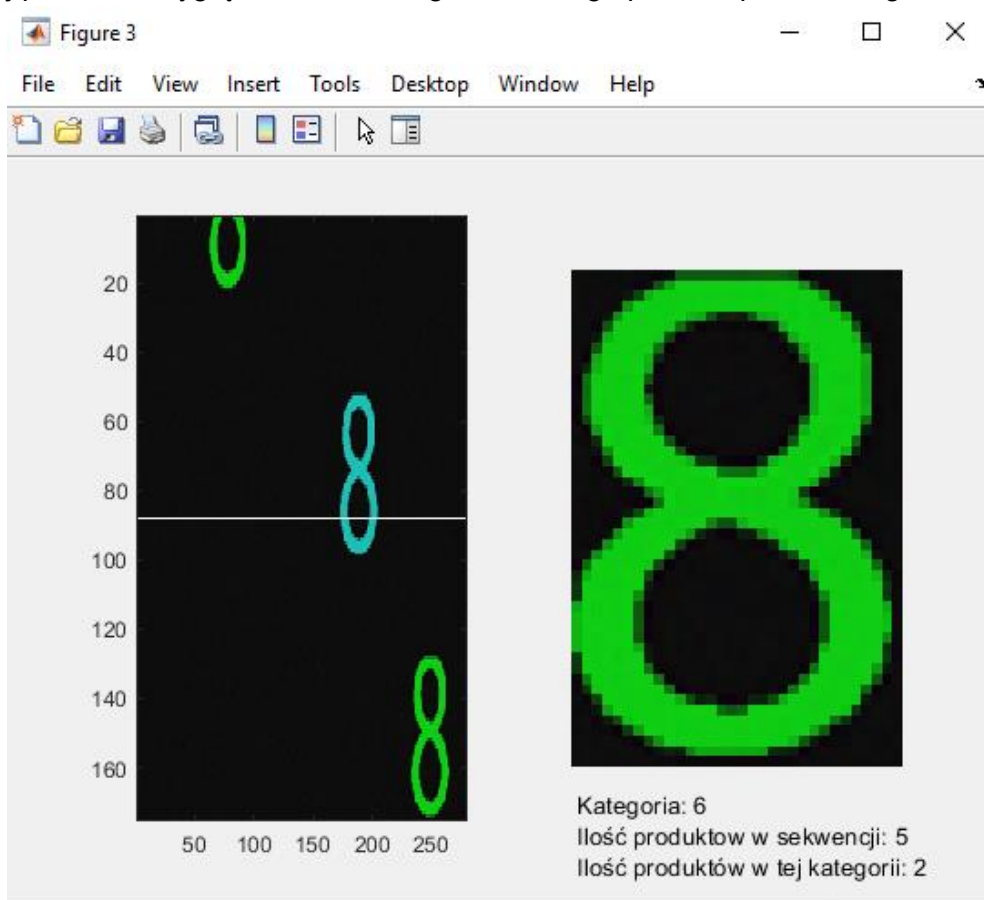
Aby policzyć odchylenie standardowe dla każdej kategorii trzeba odfiltrować pola powierzchni z struktury „produkt” dla produktów należących do jednakowych kategorii (ale tylko tych poprawnych – kategorie 1:6). Pola te są potrzebne tylko do wyznaczenia odchylenia standardowego, więc nie ma potrzeby ich zapisu na stałe. Dla aktualnie filtrowanej kategorii wartości pól przechowywane będą w wektorze „pola”. Na podstawie tego wektora i funkcji `std` zostanie obliczona wartość odchylenia standardowego i zapisana do zmiennej „S” (w której index oznacza numer kategorii).

Następnie zostaje przygotowany tekst do wyświetlenia w raporcie zbiorczym. Część tekstu zostaje przygotowana w zmiennej „tekst”, a druga w zmiennej „dane”. Ma to na celu uprościć przygotowanie tekstu. Zmienne te są tablicami komórek. Każda komórka zostanie wyświetlona jako nowa linijka tekstu.

Do wyświetlenia tekstu wykorzystana zostanie funkcja `annotation`, z obiektem `'textbox'`. Domyślnie położenie pola tekstowego określa się za pomocą jednostek znormalizowanych (czyli określających stosunek do całej figury - wartości od 0 do 1). Współrzędne pierwszego pola tekstowego zostały określone w opisie algorytmu. Współrzędne drugiego pola zostały określone na (0,13 ; 0,3), przy rozmiarze pola [1 ; 0,5]. W ten sposób tekst jest położony odpowiednio do wcześniejszego. Dodatkowo ustawiono brak linii obramowania pola tekstowego (domyślnie jest wyświetlana; zmiana właściwości `'LineStyle'` na wartość `'none'`).

- Przedstawienie wyników

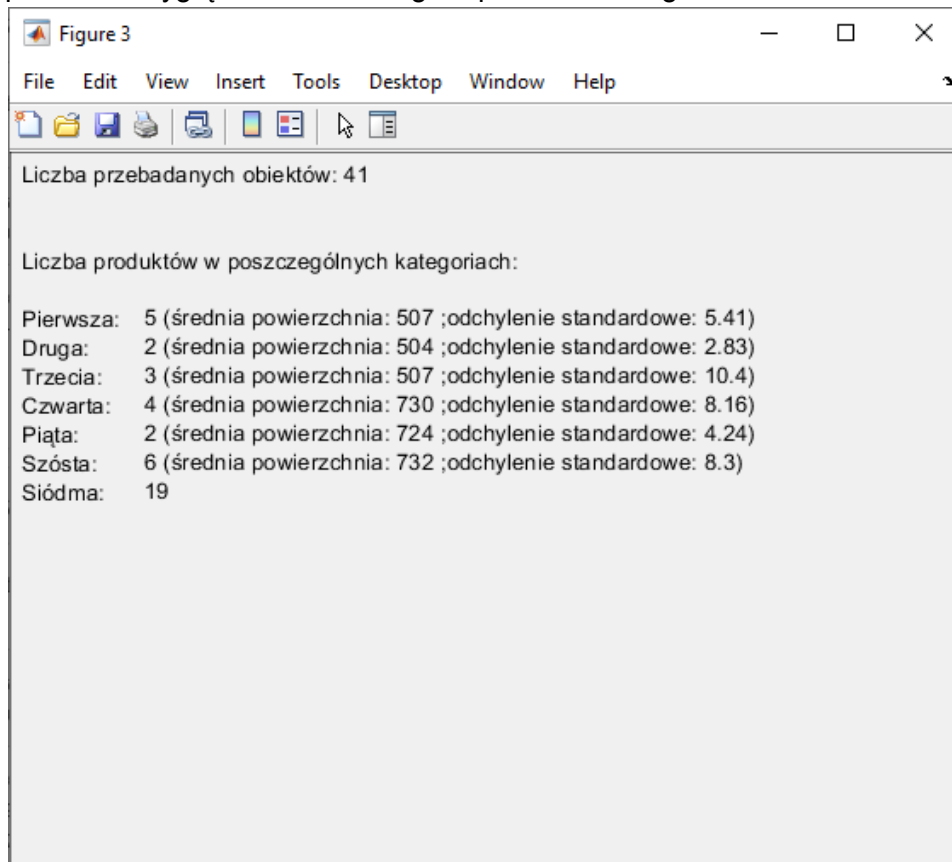
Poniżej pokazano wygląd zrealizowanego wirtualnego panelu operatorskiego.



Rysunek 8 – Podgląd wirtualnego panelu operatorskiego.

Poza wymaganymi informacjami (czyli widokiem wykrytego obiektu, numerze kategorii produktu, liczbie produktów w sekwencji i w danej kategorii) przedstawiony został też widok aktualnie przetwarzanej klatki sekwencji, z narysowaną wirtualną linią detekcji.

Poniżej pokazano wygląd zrealizowanego raportu zbiorczego.



Rysunek 8 – Podgląd raportu zbiorczego.

Na raporcie widać liczbę przebadanych obiektów, liczbę produktów w poszczególnych kategoriach, a dla kategorii prawidłowych (1:6) podane są średnie powierzchnie i odchylenie standardowe. Więc cel został zrealizowany.

- Wnioski

Wyświetlanie każdej klatki z sekwencji wideo w pewnym stopniu spowalnia proces przetwarzania. Wyświetlone klatki sekwencji są zdeformowane. Obliczenie średnich pól powierzchni może zostać przeniesione na etap po wykonaniu pętli głównej (zmniejszyłoby to ilość operacji w trakcie wykrycia obiektu). Do panelu operatorskiego można dodać elementy sterujące (np. zatrzymujące sekwencję lub wyłączające wykrywanie obiektów).

W przypadku raportu zbiorczego pojawia się problem w momencie zmiany wyświetlonego okna: tekst ulega przesunięciu. Jest tak ponieważ zostały użyte zmienne znormalizowane. Można to rozwiązać używając jako współrzędnych jednostek nie zależnych od rozmiaru okna.

Dane w raporcie zbiorczym mogły zostać wyświetlone jako tabela. Jednak problemem jest inna ilość danych dla kategorii siódmej (brak średniej powierzchni i odchylenia standardowego). Teoretycznie można puste miejsca wypełnić zerami, ale to nie jest jednoznaczne z brakiem elementu

Różnice w wartości średnich powierzchni obiektów o jednakowych kształtach (np. ósemki, czyli kategorie 4:6) wynikają prawdopodobnie z przyjętych progów do segmentacji.

V. Podsumowanie.

Wszystkie cele opisane w treści projektu zostały zrealizowane. Pojawiły się też dodatkowe elementy, jak np. widok aktualnie przetwarzanej klatki wideo.

Były też podjęte próby realizacji elementów sterujących przetwarzania sekwencji. Jednak z powodu licznych trudności pomysł ten został nie zrealizowany.

Sposoby realizacji niektórych celów zostały zmieniane w trakcie prac, na rzecz prostszych dla autora do realizacji. Np. początkowo chciano analizować podobieństwo wykrytego obiektu do wzorców produktów, na podstawie niezmienników momentowych. Lub raport zbiorczy przedstawić w formie tabeli lub pliku tekstowego.

Należy pamiętać, aby w folderze zawierającym program „Projekt_Kuligowski.m” znajdował się też plik „OKNO_1.avi”.

W trakcie realizacji projektu rozważano na temat przetwarzania obrazu w czasie rzeczywistym. Proces przetwarzania obrazu powinien się różnić w zależności od warunków w jakich zostaje przeprowadzony. Jeśli przetwarzany jest obraz w czasie rzeczywistym, to priorytetem będzie czas w jakim będzie zachodziło przetwarzanie. Nie może ono powodować opóźnień w rejestracji obrazów. Jeśli przetwarzamy wcześniej zarejestrowaną sekwencję obrazów, jest możliwość realizacji większej ilości operacji, ponieważ czas przetwarzania nie jest już tak istotny.