

Laboratorium: Python i wizualizacja

Wojciech Jaśkowski

termin: Lab 9

1 Projekt

Napisz program realizujący wybrane rozpoznawanie obrazów w kontekście komunikacji człowiek-komputer. Zakres pracy i jej efekty skonsultuj z prowadzącym.

Projekt można realizować w grupach dwuosobowych wykorzystując dowolną technologię (np. skimage, openCV, python, Java, etc.).

Uwaga: w wszystkich projektach główne zadanie polega na detekcji wybranych elementów w obrazie. Innymi słowy stan “rzeczywisty” – czyli to co człowiek sam by odczytał – powinien zostać przeniesiony do stanu “cyfrowego”. Np. w projekcie “nuty” program powinien prawidłowo (1) rozpoznać rodzaj nuty, (2) wysokość i (3) i ich kolejność. Dzięki znajomości tych 3 elementów program mógłby odtworzyć melodię (ale to nie jest wymagane; nie podlega to ocenie ale można to zrobić dla własnej satysfakcji).

Uwaga: jako wynik można albo wypisać komplet informacji w konsoli, albo zaznaczać w czytelny sposób rozpoznane elementy na oryginalnym obrazie. Ten drugi sposób jest preferowany, ponieważ pozwala on natychmiast zweryfikować skuteczność programu. Sugerowane tutaj jest naniesienie na oryginalny obraz odpowiednich konturów oraz etykiet (np. "2zł", "5 oczek", itp). Można także wykorzystać odpowiednie kolorowanie konturów poza samym naniesieniem etykiet.

2 Przykładowe tematy

- Liczenie sum pieniędzy – np. monet, banknotów – położonych np. na stole:
 - sugerowane jest skupienie się na detekcji takiego zestawu monet: 1gr, 10gr, 1zł, 2zł, 5zł
 - na zdjęciu powinno się znajdować przynajmniej kilka rodzajów monet/banknotów jednocześnie
 - jako wynik można na obrazie przedstawić kontury rozpoznanych monet oraz nadać im odpowiednie etykiety (np. "1zł").
- Wykrywanie liczby oczek na kostce:
 - program musi rozpoznawać konkretne ścianki, nie same pozycje kostek
 - program powinien sobie dobrze radzić w sytuacji, gdy na zdjęciu jest stosunkowo dużo kostek

- kostki mogą się stykać krawędziami
- jako wynik można na obrazie przedstawić kontury rozpoznanych kostek oraz nadać im odpowiednie etykiety (np. "2 oczka", albo skrótowo "2").
- Rozpoznawanie stanu rozgrywki w grze planszowej (np. Catan, Wsiąść do pociągu, Takenoko, itp.). Tutaj wybrane elementy stanu gry powinny zostać rozpoznane przez program. W razie wątpliwości odnośnie zakresu pracy, proszę o kontakt.
- Rozpoznawanie kart:
 - można się skupić na kartach 9, 10, walet, dama, król, as.
 - wymagane jest rozpoznanie 4 figur
 - jako wynik można na obrazie przedstawić kontury rozpoznanych kart oraz nadać im odpowiednie etykiety (np. "9 pik").
- Rozpoznawanie nut (kilka pięciolini na kartce, kilka rodzajów nut, klucz wiolinowy i basowy)
 - na kartce może się znajdować kilka pięciolinii,
 - nuty winny być narysowane odręcznie (ale ładnie)
 - wymagana jest detekcja tylko następujących elementów: nuty od całej do ósemki; klucz wiolinowy i basowy.
 - jako wynik można na obrazie przedstawić kontury rozpoznanych nut oraz informacje o ich kolejności, wysokości, oraz rodzaju (jako etykiety przy konturach).
- Wykrywanie boiska, piłki itp., w grze w siatkówkę, piłkę nożną itp.
- Rozpoznawanie planszy w grze w kółko i krzyżyk
 - proszę uwzględnić sytuację, gdy jest kilka plansz na jednym zdjęciu.
 - plansze winny być narysowane odręcznie
 - jako wynik można odrysować na obrazie linie plansz oraz nadać etykiety ich polom (np. 'x', 'o', '-' - to ostatnie to brak wypełnienia).

3 Kryteria oceny

Projekt będzie oceniany za:

- wkład pracy (trudność zadania)
- jakość wykonania
- uzyskane wyniki
- jakość wykonanych eksperymentów (ocena jakości rozwiązania, efektywność) przy sprawozdaniu
- pokaz wyników częściowych w ramach lab 7
- jakość sprawozdania

Uwaga: Projekt dotyczy technik przetwarzania obrazu, nie uczenia maszynowego. Użycie algorytmów uczenia maszynowego jest więc zabronione, podobnie jak użycie procedur, które – w uproszczeniu – robią zbyt wiele za programistę :). **Można oczywiście samemu implementować proste techniki klasyfikacji – wyznaczenie cech obiektów + proste porównanie z wzorcem.**

Uproszczony schemat oceniania:

- **Na 3.0:** Program “po prostu działa”, jednak jego efektywność jest przeciętna + sprawozdanie jest umiarkowanej jakości.
- **Na 4.0:** Program wykazuje się wysoką efektywnością + sprawozdanie jest dobrej jakości.
- **Na 5.0:** Program wykazuje się bardzo wysoką efektywnością. Samo zadanie musi być nietrywialne a rozwiązanie bardzo dobre (efekt “wow”) + sprawozdanie jest b. dobrej jakości.

Uwaga: Sugerowane jest przygotowanie 3 zbiorów z zdjęciami (tak po minimum 10 zdjęć w każdym zbiorze): łatwe, średnie, i trudne. Zdjęcia powinny się różnić ze względu na takie elementy jak:

- kompozycja (np. dodatkowe elementy na stole, takie jak ołówek itp.)
- oświetlenie (słabe, intensywne, punktowe, rozproszone).
- perspektywa (widok idealnie z góry", widok pod lekkim kątem, przybliżenie/oddalenie, itp.)

Łatwe zdjęcia powinny być traktowane jako benchmark na zaliczenie. Program nie musi poprawnie przetworzyć wszystkich zdjęć trudnych, by dostać 5.0. Raczej chodzi o pokazanie pewnej górnej granicy skuteczności działania programu poprzez weryfikację na bardzo trudnych przypadkach.

Wskazówki/wymagania do sprawozdania:

- Sprawozdanie powinno zostać wykonane w LaTeX-u.
- Układ tekstu i grafik powinien zostać odpowiednio sformatowany. Kiepsko wyglądające prace oraz prace w których jest dużo błędów językowych nie będą oceniane.
- Sprawozdanie nie powinno być zbyt długie, lecz konkretne. Sekcje takie jak “motywacja” lub “plan dalszego rozwoju aplikacji” można pominąć. Tekst powinien zawierać:
 - Temat + krótki opis rozwiązania problemu. Tekst ten powinien dosyć dobrze streścić jak działa przygotowana aplikacja.
 - Bardziej szczegółowe pokazanie w kolejnych krokach jak działa program. Zalecam przedstawienie efektów nakładania kolejnych technik przekształcania obrazu, bądź innych pomocnych metod (krótki opis + wyjaśnienie dlaczego takie przekształcenie + ilustracje przedstawiające wynik).
 - Pokazanie kompletnych wyników dla przygotowanego zbioru zdjęć/filmów. Można pokazać w kolejnych podsekcjach wyniki dla zdjęć łatwych, średnich, trudnych (pary zdjęć input i output).

- Podsumowanie wyników. W szczególności omówienie przypadków, kiedy program nie zadziałał dobrze (wyjaśnić dlaczego nie zadziałał). Można zasugerować sposób na poprawę (chętni mogą poprzeć tę sugestię szybką implementacją algorytmu i wynikiem). Chętni: jeżeli zbiór zdjęć/filmów zawiera dużo obiektów, które należy wykryć i zaklasyfikować, można policzyć skuteczność klasyfikacji poszczególnych obiektów (np. poprawnie wykryto 18 z 20 monet 5 złotych). Można pójść dalej i pokazać w macierzy odpowiedzi programu, t.j. ile jakich odpowiedzi dawał program dla poszczególnych klas obiektów.