

Języki skryptowe – projekt

Streszczenie projektu

Celem projektu jest implementacja ‘bezdotykowej’ obsługi myszy. Przeprowadzane jest to w następujący sposób:

- Na obrazie z kamery znajdowana jest dłoń
- Na wycinku obrazu, w którym znajduje się dłoń dokonywana jest predykcja gestu
- Dane na temat wykrycia dłoni i gestu zostają wyświetlone w oknie informacyjnym oraz wykonana zostaje akcja przypisana do gestu

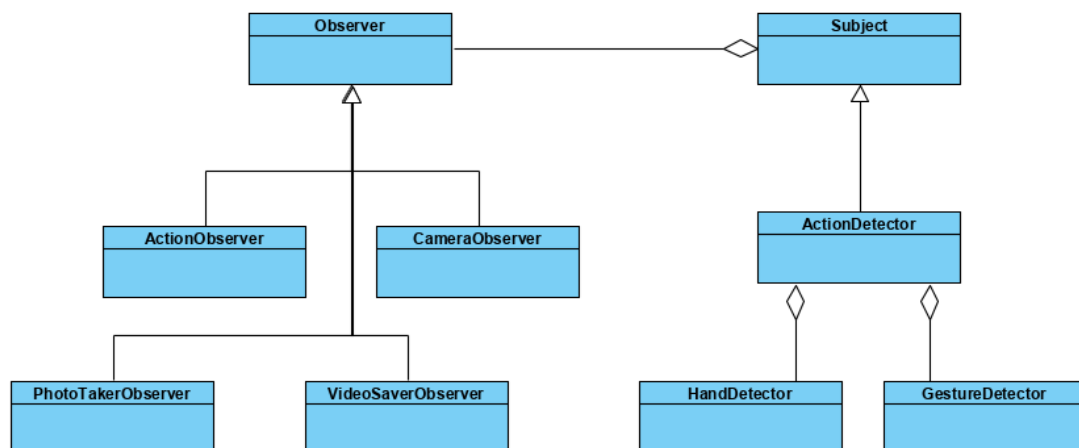
Wykorzystanie technologie / biblioteki

Nazwa biblioteki Python	Opis wykorzystanych metod z biblioteki
OpenCV	Przechwycenie obrazu z kamery, wykonywanie operacji na obrazie, zapisywanie obrazu, wykrycie dłoni za pomocą <i>kaskad Haara</i>
Tkinter	Wyświetlenie okna informacyjnego, pozostającego zawsze nad innymi programami, zawierającego dane algorytmu
Keras	Wyuczenie i obsługa modelu gestów, stworzonego przy pomocy sieci neuronowej <i>SqueezeNet</i>
NumPy	Dopasowanie wielomianu do danych, operacje na macierzach

1 Wykorzystane biblioteki

Architektura

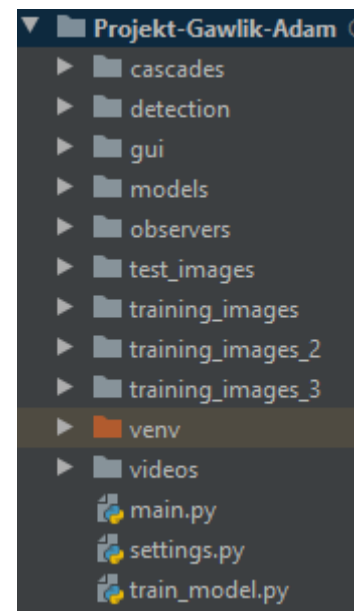
Projekt wykorzystuje wzorec projektowy *Obserwator*.



2 Diagram klas

Struktura projektu:

Projekt został podzielony na pliki, z których te niepowiązane ze sobą znajdują się w osobnych katalogach. W osobnych katalogach znalazły się klasy odpowiedzialne za interfejs graficzny, wykrycie gestów, czy klasy obserwatorów. W osobnych katalogach znajdują się także pliki video nagrane przez program, czy przechwycone obrazy. Osobne katalogi posiadają także wyuczone modele do detekcji gestów i kaskady Haara.



3 Struktura katalogów

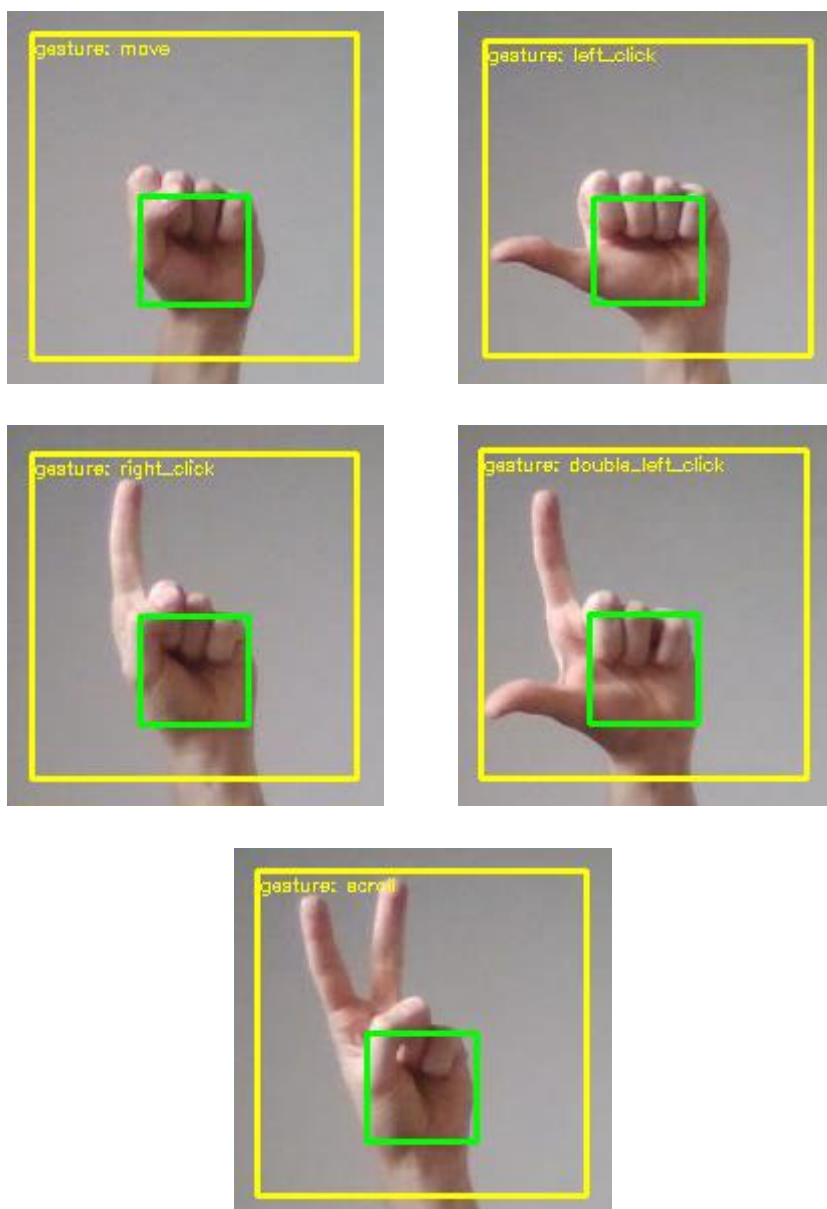
Zrzuty ekranów z projektu

Wykrycie gestów

Wykrywanie ręki i gestów odbywa się w obiekcie *ActionDetector*. Zawiera on instancję klasy *HandDetector* służącej do wykrywania położenia dłoni przy pomocy *kaskad Haara* (za pomocą dostępnego klasyfikatora), a także klasy *GestureDetector* wykrywającej gest na podstawie specjalnie wyuczonego modelu sieci neuronowej *SqueezeNet*.

Model wykrywania gestów został wyuczony do wykrycia 5 gestów:

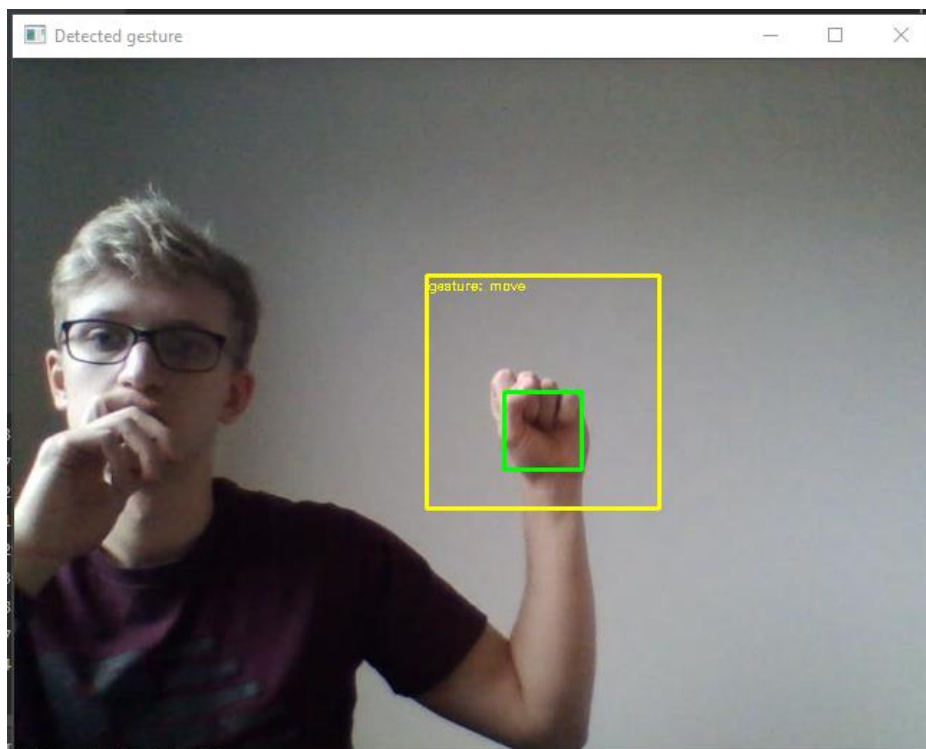
- move – odpowiada poruszaniu kursorem lub brakowi wykrycia gestu
- left_click – odpowiada kliknięciu lewego przycisku myszy
- right_click – odpowiada prawemu przyciskowi myszy
- double_left_click – odpowiada podwójnemu kliknięciu lewego przycisku myszy
- scroll – odpowiada scrollowaniu, kierunek scrollowania jest ustalany na podstawie ruchu dłoni



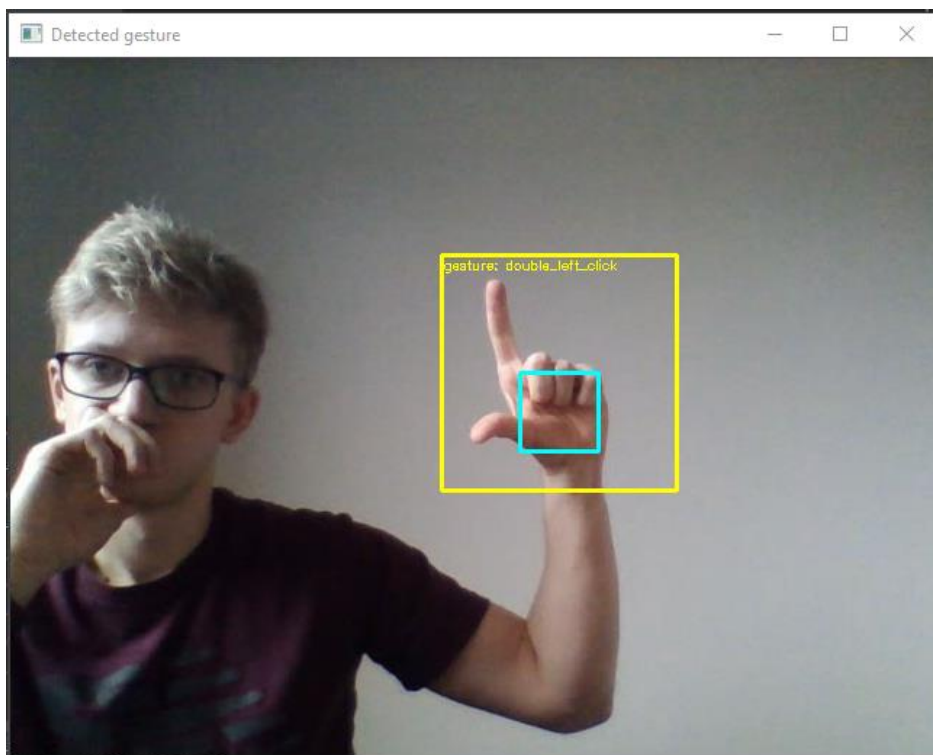
4 Wykrywane gesty

CameraObserver

Zadaniem tego obserwatora jest wyświetlenie obrazu przechwyconego z kamery. Dodatkowo, jeżeli dłoń została znaleziona, zaznaczany zostaje obszar, w którym została ona wykryta (kolor zielony w przypadku wykrycia, kolor niebieski, gdy wyświetlane jest ostatnie wykryte położenie). Zaznaczany jest także obszar, w którym wykrywany jest gest (kolor żółty).



5 Dłoń wykryta



6 Dłoń niewykryta

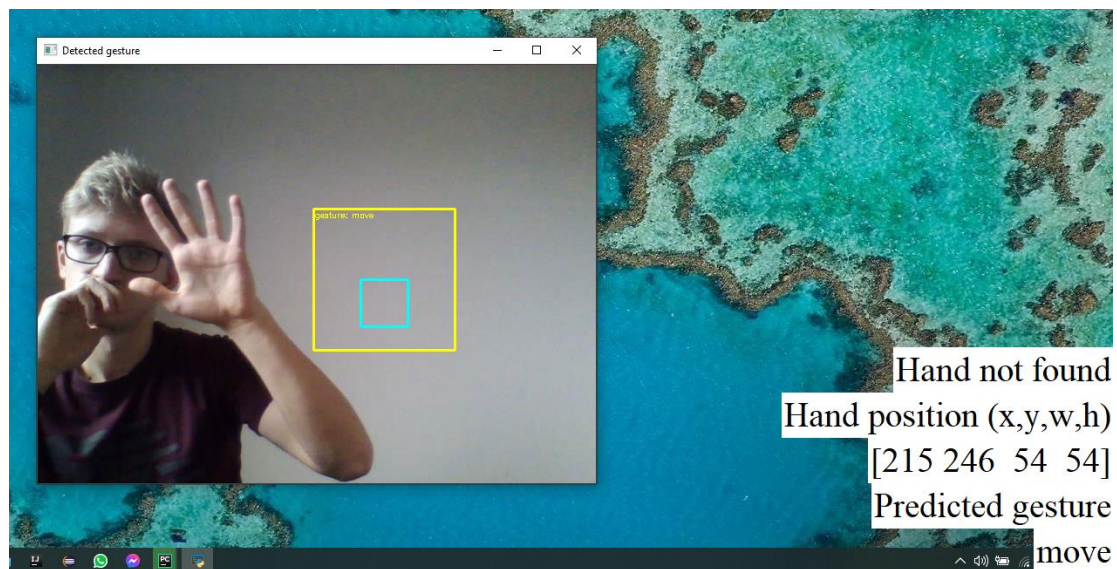
ActionObserver

Zadaniem tego obserwatora jest podjęcie decyzji na temat wykonywanej akcji. Zawiera on tablice przechowujące określoną liczbę poprzednich położeń dłoni, a także predykowanych gestów.

Akcja przypisana gestowi zostaje wykonana tylko w momencie, gdy jego dokładność jest na poziomie $\geq 70\%$.

Ruch kursora oraz scrollowanie wykonywane są na podstawie ostatnich położeń dłoni. W celu uniknięcia dużych wahań stosowane jest dopasowanie wielomianu pierwszego stopnia do danych pozycji. Na podstawie współczynnika a_1 wielomianu dobierana jest szybkość poruszania się kursora czy scrollowania.

Obserwator ten przekazuje także dane do okna informacyjnego.

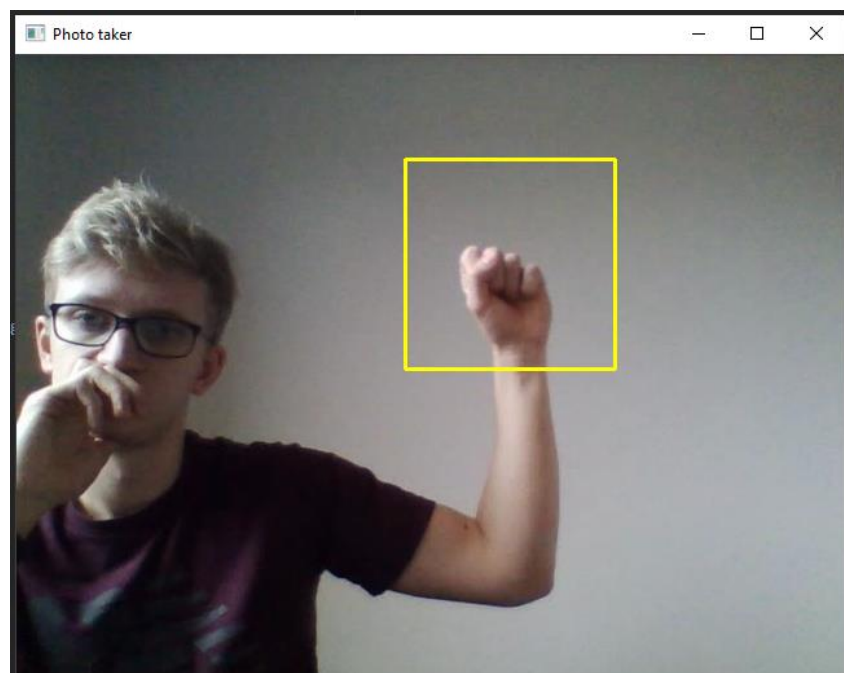


7 Widok z CameraObserver i okno informacyjne

PhotoTakerObserver

Jest to obserwator służący do zapisywania wycinków ekranu w miejscu przewidzianym na gest dłoni. Zapisane zdjęcia mogą następnie służyć do wyuczenia modelu.

W trakcie działania obserwatora wyświetlane jest okno z obrazem przechwyconym z kamery, a także zaznaczony jest na nim obszar przewidziany dla wykrycia gestu (aktualizowany w czasie rzeczywistym na podstawie wykrytego położenia dłoni), który zostanie zapisany, gdy naciśnięty zostanie przycisk 's'.

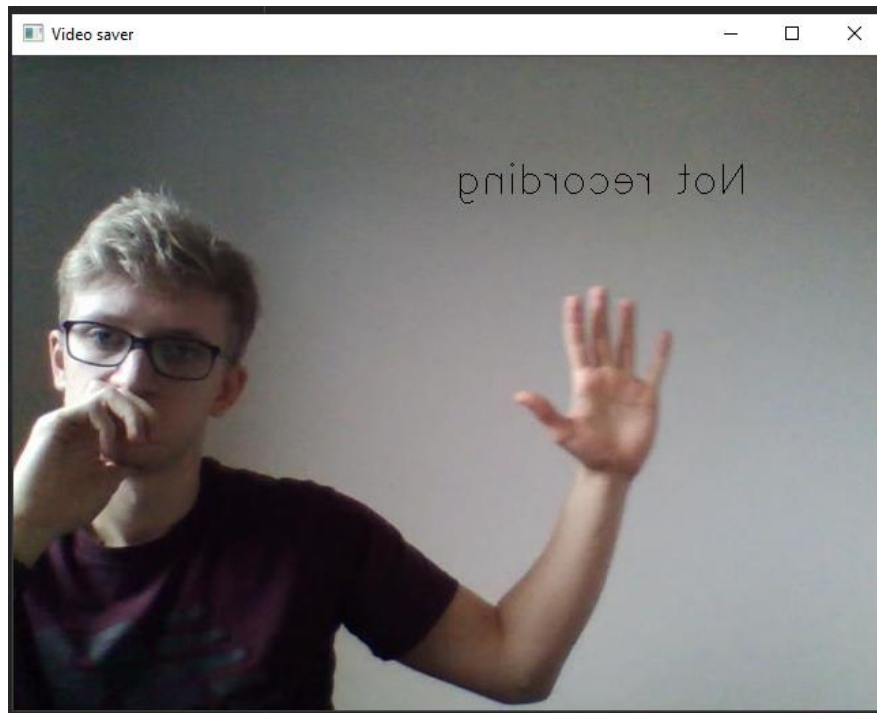


8 Widok z PhotoTakerObserver

VideoSaverObserver

Zadaniem tego obserwatora jest zapisanie obrazu przechwytywanego przez kamerę jako plik video. Zapisany plik może następnie służyć do przeprowadzania testów działania algorytmu na tym samym pliku wsadowym.

W czasie działania wyświetlane jest okno z podglądem obrazu z kamery, a także informacją czy w danym momencie obraz jest nagrywany.



9 Widok z VideoSaverObserver