

# Visual Tracker Benchmark

## 1. Opis

[Visual Tracker Benchmark](#) - projekt który zawiera wyniki, kod i dane dla ewaluacji i badania różnych metryk dla śledzenia pojedynczego obiektu. Również pozwala na wykreślenie zbadanych metryk oraz porównanie wyników z innymi algorytmami śledzącymi.

## 2. Problemy

Kod benchmarku jest udostępniony dla dwóch języków programowania: Python oraz Matlab. Nie korzystaliśmy z kodu w matlabie, jednak kod pythonowy jest wykonany w wersji Python2, która od tego roku nie jest już wspierana, stąd pojawiają się problemy ze znalezieniem oraz instalacją tej wersji oraz niezbędnych pakietów. W związku z tym, wymagane było przekonwertowanie kodu do wersji Python3 oraz poprawienia kompatybilności z biblioteką urllib. Ponadto kod jest napisany w mało przejrzysty sposób i nie ma dokumentacji.

## 3. Instalacja

- a) Sklonować repozytorium [https://github.com/jwlim/tracker\\_benchmark](https://github.com/jwlim/tracker_benchmark) do katalogu projektu
- b) Przekonwertować kod źródłowy za pomocą programu 2to3
- c) Poprawić absolutne importy
- d) Zaimplementować funkcję `run_{nazwa_modeli}` w folderze `scripts` w pliku o takiej samej nazwie oraz dodać import w pliku `__init__.py` w tym samym katalogu. Przy badaniu metryki TRE uwzględnić `seq.startFrame` oraz `seq.endFrame` bo metryka jest wywoływana na różnych podzbiorach oryginalnej sekwencji 20 razy. Funkcja ma zwrócić słownik z polami "type" = "rect", "fps" - ilość klatek na sekundę ( $\text{len}(\text{seq}) / (\text{end} - \text{start})$ ), "res" - lista Pythonowa z boksem w każdej klatce w formacie tlwh (top, left, width, height).
- e) Poprawić funkcję `download_sequence` w pliku `butil/seq_config.py`
- f) W pliku `data/tb100.txt` zostawić filmy z ludźmi oraz usunąć filmy gdzie są dwa `init_rect` (Jogging-1, Jogging-2..)
- g) Wyszukać wszystkie zapisywania do pliku i dodać `.encode()` do stringu

## 4. Uruchomienie

- a) Generowanie wyników:  
`python -m [ścieżka do pliku run_trackers.py (jako pythonowy import)] -t [nazwa modeli] -s tb100 -e TRE`

tb100 - jest nazwą pliku txt w katalogu data, jeśli chcemy uruchomić benchmark na prywatnych i/lub części dostępnych danych, to możemy albo

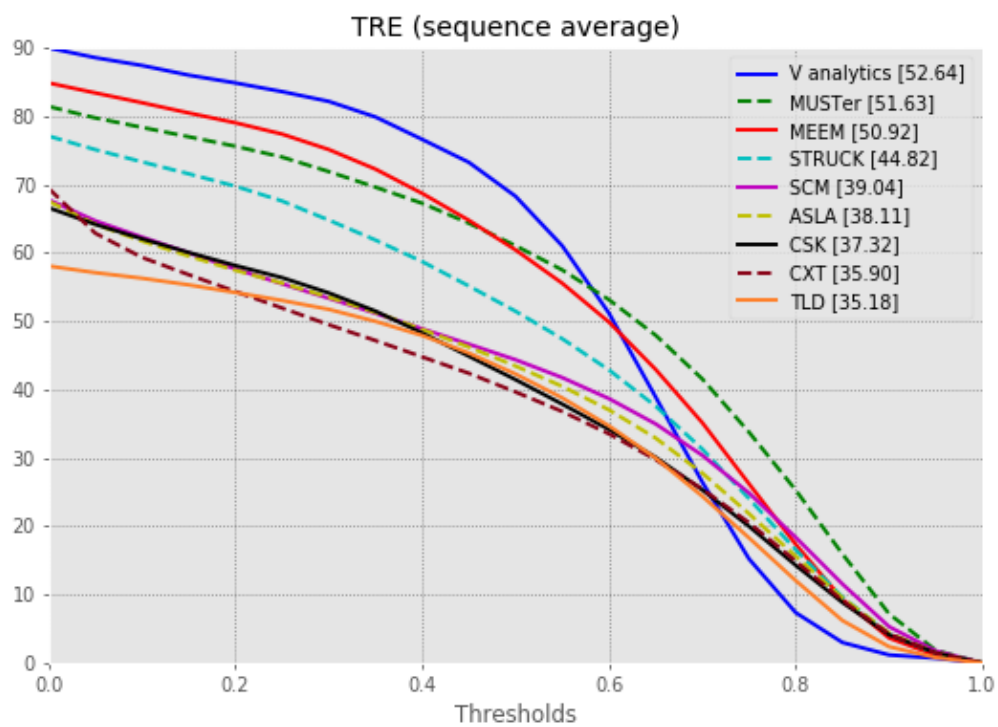
wyedytować ten plik i zostawić tylko potrzebne filmy. Również można stworzyć swoje pliki txt w tym katalogu i zmienić funkcje `get_seq_names` w pliku `butil/seq_config.py`.

b) Generowanie wykresów:

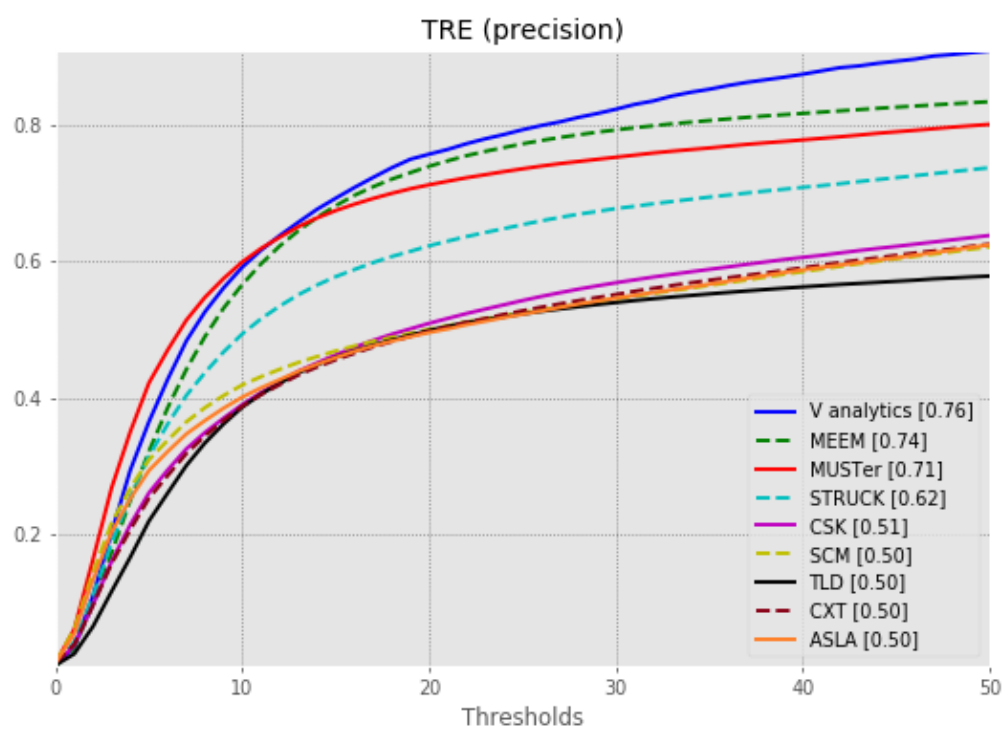
`python -m [ścieżka do pliku draw_graph.py (jako pythonowy import)]`

## 5. Wyniki

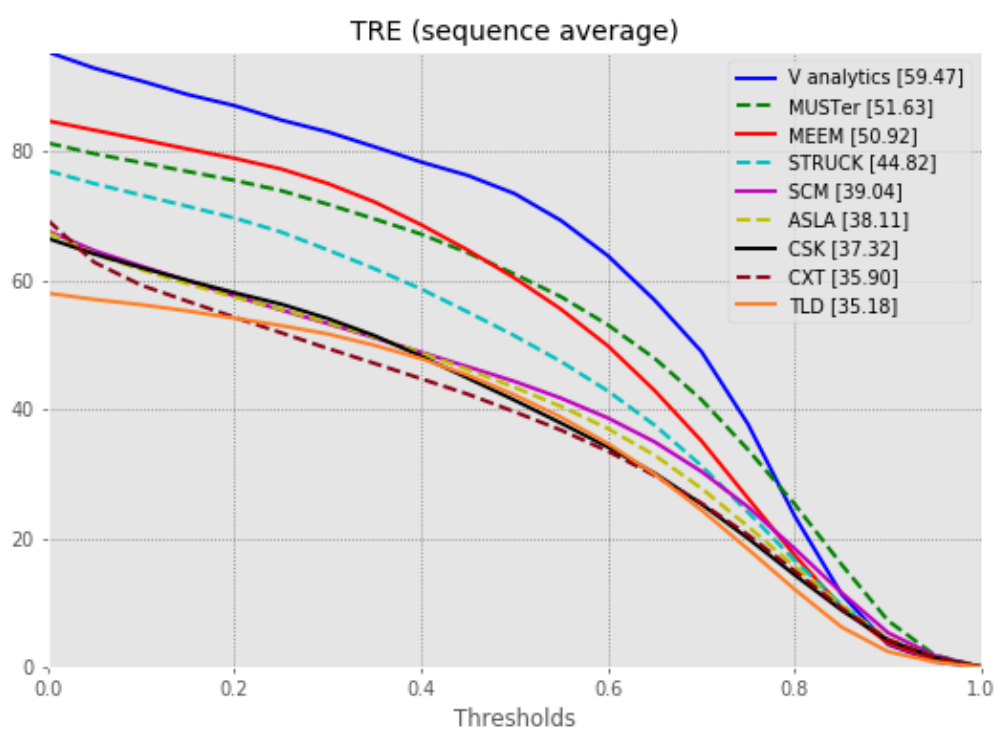
Stworzyliśmy dwa pliki: `cars.txt` (filmiki z samochodami z danych TB100) oraz `humans.txt` (wybrane przez nas filmy z ludźmi) i uruchomiliśmy model na tych plikach i porównaliśmy wyniki z ośmiu najlepszych trackerów na danych `tb50`, które były dostępne w bazie benchmarku.



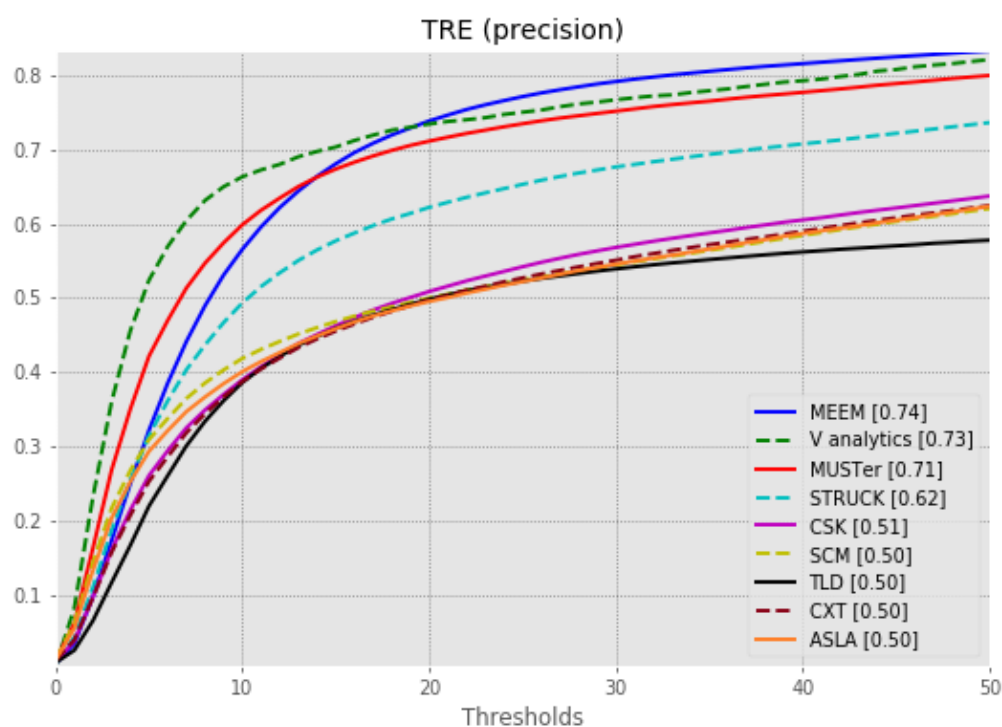
Rysunek 1. Sequence average dla filmów z ludźmi



Rysunek 2. Precision dla filmów z ludźmi



Rysunek 3. Sequence average dla filmów z samochodami



Rysunek 4. Precision dla filmów z samochodami

cars.txt zawiera filmiki:

- BlurCar1
- BlurCar2
- BlurCar3
- BlurCar4
- Car1
- Car2
- Car4
- Car24
- CarDark
- CarScale
- Suv

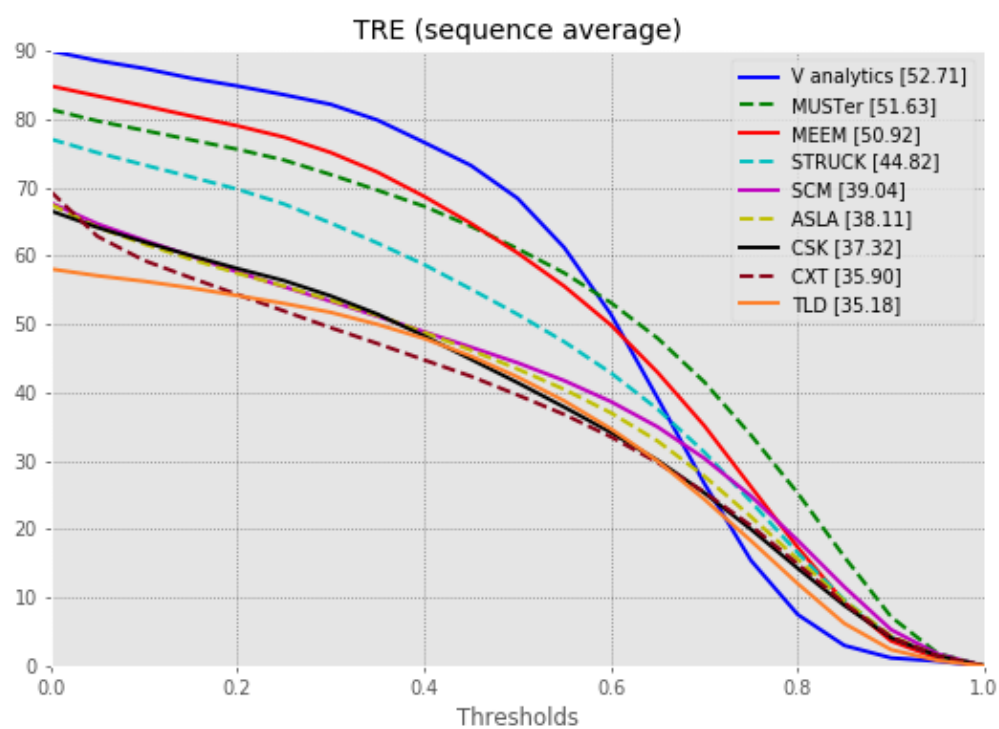
humans.txt zawiera filmy:

- Basketball
- BlurBody
- Bolt
- Bolt2
- Couple
- Crossing
- Crowds
- Dancer
- Dancer2
- David3
- Girl2

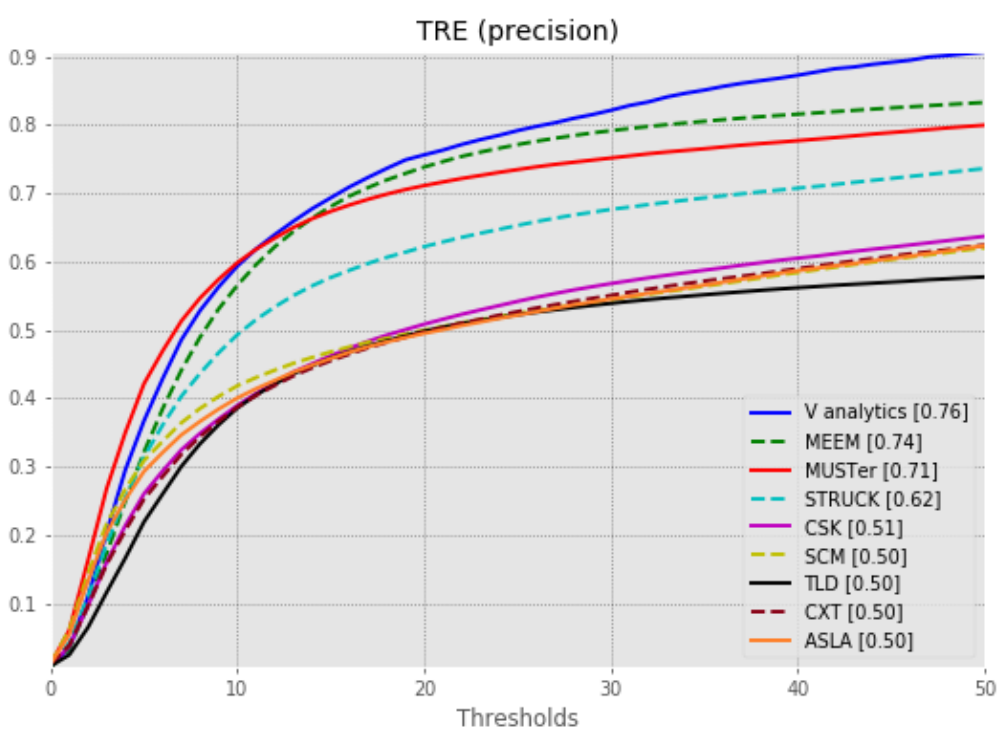
- Gym
- Human2
- Human3
- Human4
- Human5
- Human6
- Human7
- Human8
- Human9
- Jogging1
- Jump
- Singer1
- Skater
- Skater2
- Skating1
- Skating2-1
- Subway
- Walking
- Walking2
- Woman

## 6. Wyniki v2

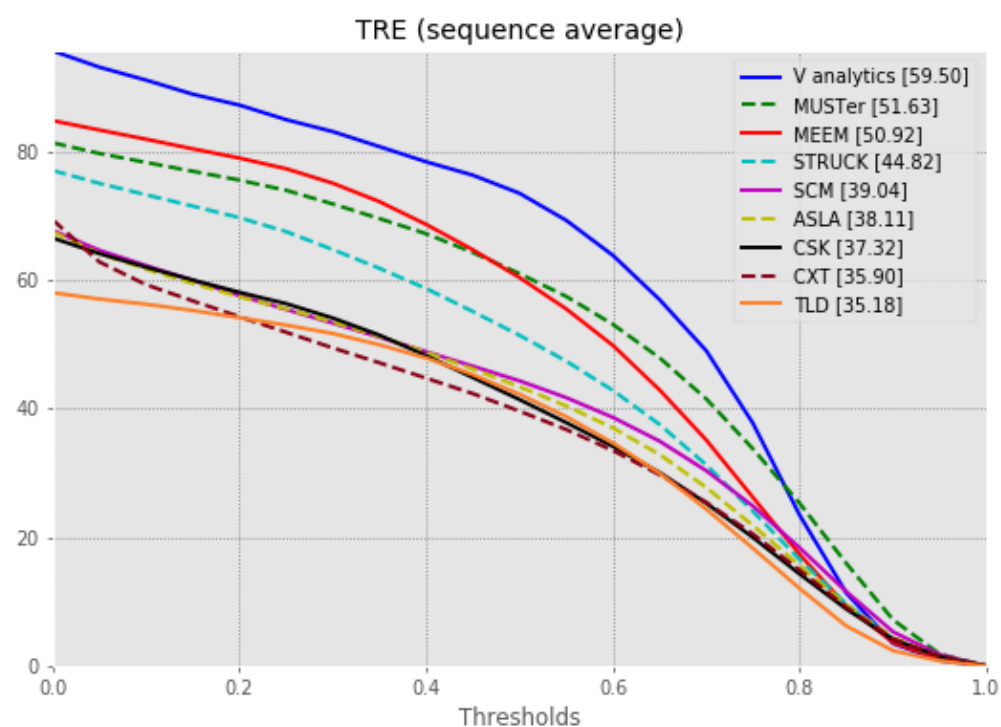
Połączyliśmy ekstraktor cech z detektorem, co polepszyło wydajność całego algorytmu do śledzenia. Również to podejście pozwoliło na wykorzystanie przez ekstraktor cech pierwszych warstw detektora, które są już wytrenowane filtry i mają użyteczne cechy (wykrywanie prostych kształtów...). Przy zwiększeniu wydajności udało nam się nieznacznie ulepszyć wyniki w szczególności nowy model lepiej dopasowuje boksy, co już jest dobrym wynikiem.



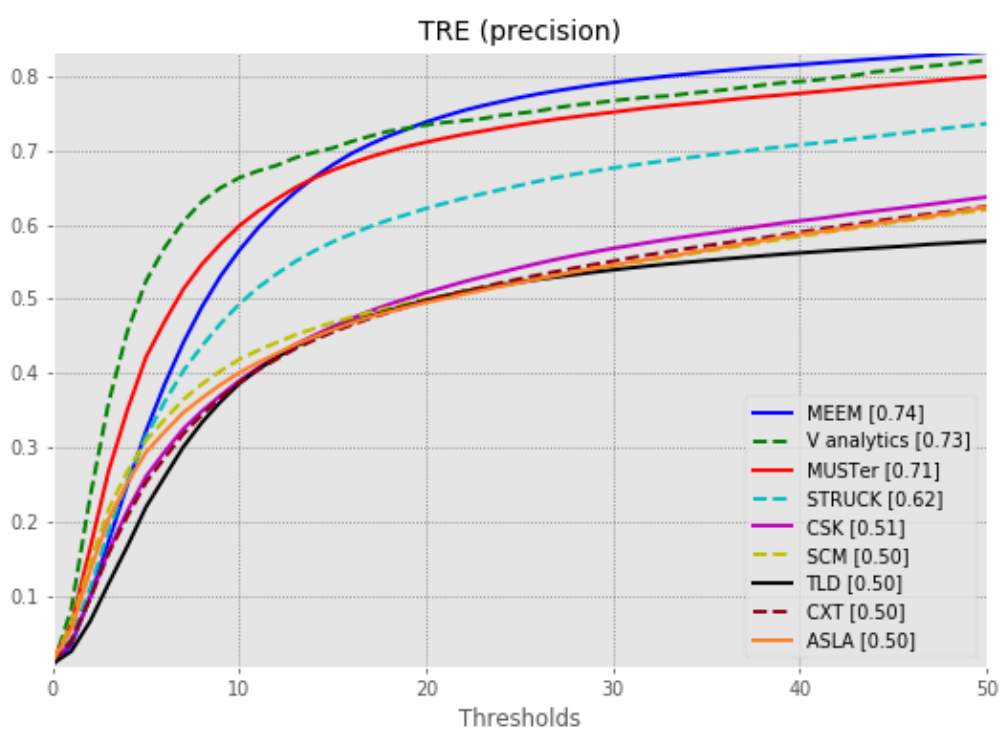
Rysunek 5. Sequence average dla filmików z ludźmi



Rysunek 6. Precision dla filmików z ludźmi



Rysunek 7. Sequence average dla filmików z samochodami



Rysunek 8. Precision dla filmików z samochodami