***Abstract*—Tracking and detecting of object is one of the most popular topics recently, which used for motion detection of various objects on a given video or images. To achieve the goal of intelligent navigation of a moving platform operating on the sidewalk, our goal is to build the software that is able to detect the pedestrians and predict their trajectories, during which process MOT (multiple object tracking) plays an important role. Although different kinds of approaches have been introduced in the class, even in latest research papers, to tackle similar problem, there still exists many issues unsolved. In this report, we inspect the previous work and propose our method for better combination of detection and MOT algorithms. Here we take advantage of the high efficiency and celerity of Faster RCNN (Region-based Convolutional Neutral Network) and KCF (Kernelized Correlation Filter) for the purpose of real-time performance. To validate the effectiveness of our system, the algorithm is demonstrated on four video recordings from a standard dataset. The experimental results on our test sequences illustrate the reliability of our method.**

***Keywords*—Visual detection, Multiple object tracking, Neutral networks, Kalman Filter, Kernelized Correlation Filter.**

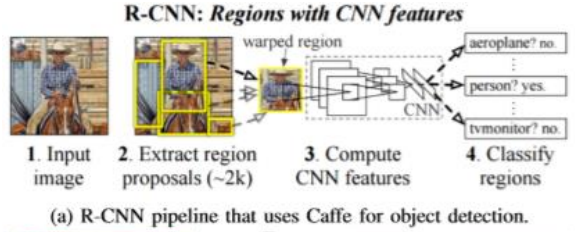
Purpose: Building a software that is able to detect the pedestrians and predict their trajectories for achieving the goal of intelligent navigation of a moving platform operating on the sidewalk. (building a pedestrian assistant system which requires an intellectual Segway to automatically guide itself on the sidewalk and avoid pedestrian simultaneously)

Method: MOT (multiple object tracking) process, For Real-time performance, they use a combination of Faster R-CNN, KCF and Kalman Filtering.

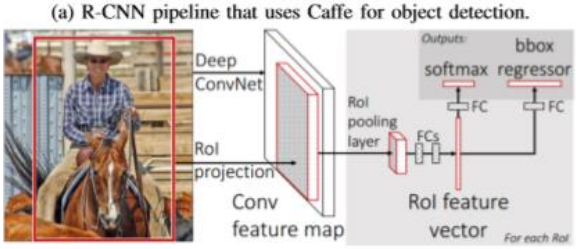
However, the existing algorithms of tracking pedestrians exhibit a low performance when faced with problems like occlusions, moving cameras, illumination changes, motion blur and other environmental variations. In this paper, we propose an algorithm that to apply KCF (Kernelized correlation filters) tracker into pedestrian detection and tracking, combined with R-CNN based object detector.

Problem: a low performance when faced with problems like occlusions, moving cameras, illumination changes, motion blur and other environmental variations.

Suggestion: An algorithm that to apply KCF tracker into pedestrian detection and tracking, combined with R-CNN based object detector



(a) R-CNN pipeline that uses Caffe for object detection.



Faster R-CNN architecture

1. Higher detection quality
2. Training is single-stage, using a multi-task loss
3. Training can update all network layers
4. No disk storage is required for feature caching

Our algorithm combines Faster R-CNN with both Kalman Filter and KCF tracker. Faster R-CNN is used for pedestrian detection (candidates to track), and Kalman Filter or KCF tracker is applied for data association and for tracking objects during occlusions. To improve the overall performance of our method, we added different update steps to solve problems like occlusion and noise reduction. Figure 4 shows the diagram of the steps of our method.

Faster R-CNN => for pedestrian detection

Kalman Filter/KCF tracker => for data association and tracking objects during occlusion

1) object states: If a detected region R_i^t overlaps with only one existed track T_j^t , it means that the object is isolated and currently tracked (state: tracked).

If a detected region R_i^t overlaps with more than one existed track T_j^t, it means that the object is tracked but under occlusion (state: occluded).

If a detected region R_i^t does not overlap with any existed track T_j^t, it means that it is an object not currently tracked (state: new object or noise).

If a T_j^t does not overlap with any R_i^t, it means that it is an object not currently visible (state: invisible or object that has left).

state: **tracked** - detected region R_i^t와 겹치는 경우, 객체가 고립되어 현재 tracked되고있음을 의미

state: **occluded** - 검출된 영역 R_i^t가 존재하는 하나 이상의 track T_j^t와 겹치는 경우, 이는 object가 tracked되지만 occlusion(폐색)상태임을 판단함

state: **new object or noise** - detected region R_i^t가 존재하는 track T_j^t와 겹치지 않으면, 그것은 현재 tracked 되지 않은 Object임(새로운 object or noise)

state: **invisible or object has left** - T_j^t가 R_i^t와 겹치지 않으면 현재 보이지 않는 객체임을 판단

7) Pseudo Code: Our method is summarized by the following pseudo code.

```
Input: video
Output: trajectories + bounding boxes
    with numbers

for each video frame do
    Detection by Faster R-CNN
    for Each KCF tracker do
        Find Best matching box
    end
    Data assignment based on current
        state
    if State == Tracked
        Update KCF tracker with detection
    end
    if State == Occluded
        Track in straight through manner
    end
    if State == New object
        Create a new tracker
    end
    if State == Invisible or exited
        object
        Delete or retain trajectory
    end
end
```

Conclusion

This paper has proposed the combination model of Object tracker (Kalman Filter && KCF) and Faster R-CNN

Faster R-CNN extract pedestrian objects and get their scale and size in combination with tracker outputs.

KCF/Kalman Filter is used for data association and to handle fragmentation and occlusion problems.

It takes tracking decision at Every Frame

Mixed strategy enables us to get competitive results and reduce errors under complex environment.

Eval function → MOTA && MOTP

Personal View

- 논문에서 결과를 보여준 것에 대해 좀 부족한 느낌..? 잘 나온 데이터를 보여주는 게 아니라 그냥 해본 것을 그대로 쓴 것 같음 (**Table 1**)
- 이전에 읽은 논문(An Effective and Efficient Method for Detecting Hands in Egocentric Videos for Rehabilitation Applications)이 잘 보충해서 넣음
- 논문이 잘 쓰여진 것이 아닌 느낌 (중복이 많다, 기본적 오타, 채워지지 않은 부분)
- **이 논문과 전 논문의 아이디어를 착용해 YOLO v3에 적용해보는 것**
- **several frame마다 tracker, detector를 initializing해서 판단해주는 idea에 몇 프레임을 연속으로 하여 시계열데이터처럼 판단하여 tracking을 예측하는 것에 대한 생각**
- **Tracker에 대해서, KCF, MIL 알고리즘을 통해 Test 해 본 것이 전부이나, TLD Tracker 알고리즘을 도입하는 것도 방법론이 될 수 있을 것 같음 (TLD Tracker의 장점은 여러 장의 Frame에서 occlusion이 있을 때 가장 잘 작동하는 알고리즘, 크기가 변할 때도 가장 좋은 퍼포먼스를 보임)**
  - ⇨ 기존 논문 두개에서도 occlusion 문제를 해결하려는 것이 Point였으므로
  - ⇨ 하지만 TLD Tracker도 문제가 있는데, occlusion된 물체에 대해서 tracking이 다른 object로 track 될 수 있다.
- 다른 tracker는 **MEDIANFLOW Tracker**가 있는데, 내부적으로 전방향, 후방향, 두 쪽 방향 모두 물체를 측정하고 이 두 trajectories 사이의 차이를 측정. ForwardBackward 에러를 최소화하는 것은 믿을 만하게 tracking failure를 감지할 수 있게 하고, 비디오 시퀀스 안에서 믿을 만한 trajectories를 선택할 수 있게 함.
  - ⇨ 테스트에서 이 tracker는 움직임이 예측 가능하고, 작을 때 가장 잘 작동했음. Tracking이 명확하게 실패했을 때 계속 작동하는 다른 tracker과 다르게, 이 tracker는 언제 tracking이 실패했는지 알고 있음**(but occlusion이 없을 때 작동을 잘 함, 큰 움직임이 감지되면 실패)**
  - ⇨ **언제 실패했는지 알면 몇 frame마다 초기화가 아닌, 실패했을 때 tracker를 초기화하고 detector로 잡아준 다음, 다시 tracker를 사용하면 될 듯함.**