CIS 581 Final Project
Object Tracking with Dynamic Detection in Sliding Camera

Team Member: Youjia Li, Eddie Wu

Introduction

In an era where there are so many images and videos being taken everyday, there are so many relevant applications, ranging from simple images/videos blending to more complex object detection, and our team would like to create an application that can detect moving objects as well as their classes in an input video, as this can be quite helpful for tasks like accident prevention and autonomous driving. The application consists of three main parts: object detection using Mask R-CNN, Optical Flow of detected objects, and identification of moving objects as well as their classes. In the first section, we used available Mask R-CNN architecture to detect objects with 81 possible classes, such that the bounding boxes and corresponding classes of these objects can be used for optical flow, which is the second section. In optical flow, these objects are tracked throughout, and every once in a while (every 50 frames, for example) objects are re-identified using Mask R-CNN to ensure accurate identification of potentially new objects. In the last section, we detect which of the objects are moving and which ones are not, and also give them respective colored bounding boxes, and the criteria we used is the distance between center of objects' bounding boxes and the center of the background at every frame. To test our work, we generated multiple videos displaying moving objects and non-moving objects, from basketball game clip to kite flying contest, and the outputs look fairly accurate at both detecting & tracking objects and determining whether objects are moving.

Methods

Below is the brief flow chart of our application.

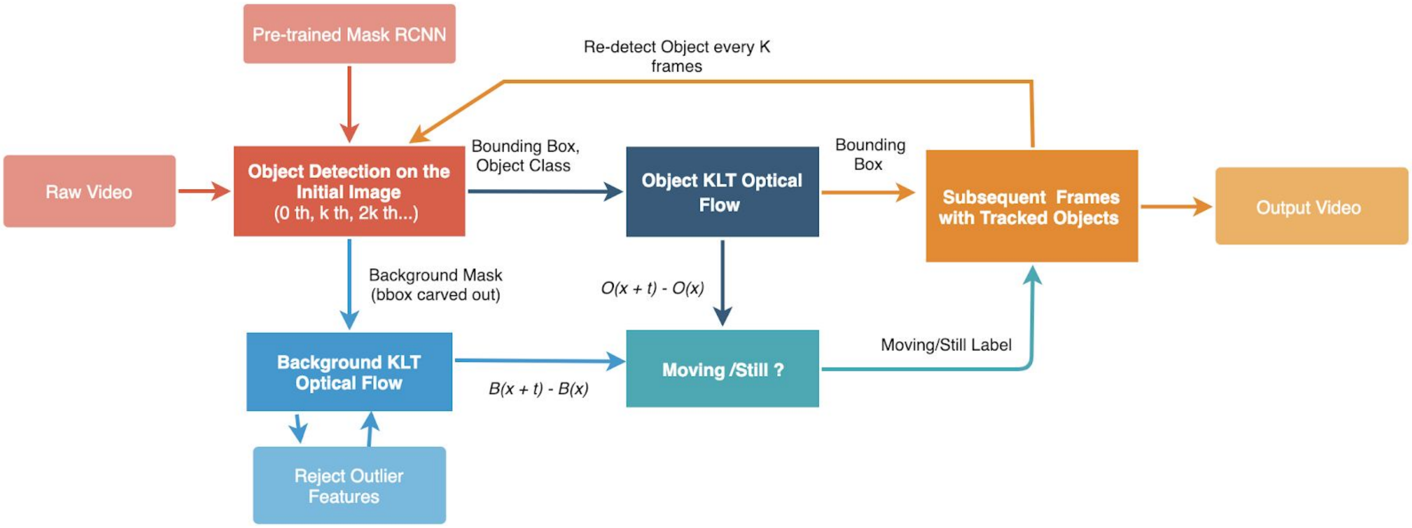
Pre-trained Mask RCNN
Re-detect Object every K frames
Raw Video
Object Detection on the Initial Image (0 th, k th, 2k th...)
Bounding Box, Object Class
Object KLT Optical Flow
Bounding Box
Subsequent Frames with Tracked Objects
Output Video
Background Mask (bbox carved out)
O(x + t) - O(x)
Background KLT Optical Flow
B(x + t) - B(x)
Moving /Still ?
Moving/Still Label
Reject Outlier Features

Figure. Flowchart of object tracking and dynamic detection

Initially, we will apply pre-trained Mask R-CNN (regional convolutional neural network) on input video's first frame to identify obvious objects (only retaining those that can be predicted very accurately). The framework is shown below[1]:

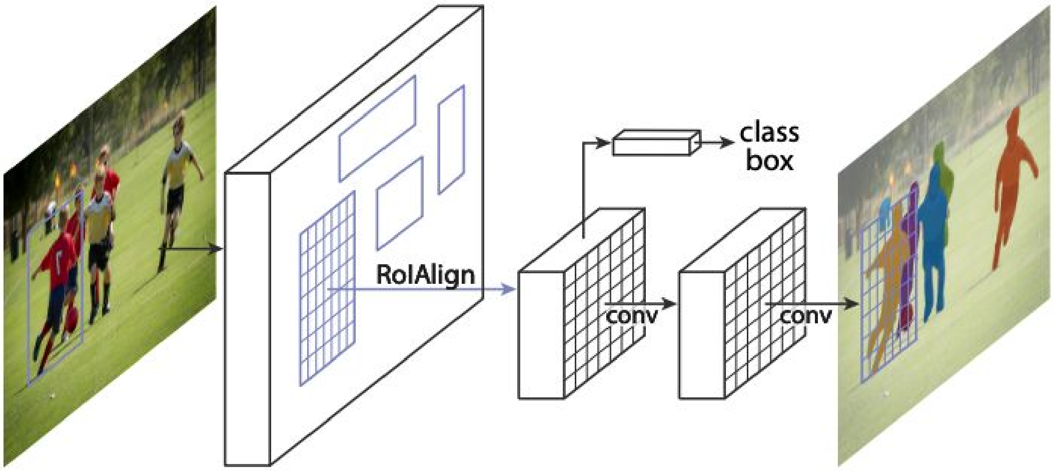

Figure. Framework of Mask R-CNN architecture

In a nutshell, Mask R-CNN excels at instance segmentation (which is a combination of classification, semantic segmentation and object detection) by including a object mask prediction branch in parallel with bounding box recognition branch based on the network called region-based convolutional neural network (R-CNN). Raw R-CNN would evaluate multiple CNNs independently on each regions of interest (RoI), and would also use RoIPool to attend to RoIs on feature maps; Faster R-CNN would incorporate an additional Region Proposal Network (RPN) that can learn the attention mechanism very efficiently, and this is the current leading framework for bounding box detection; Mask R-CNN adds an extra branch for mask prediction on top of Faster R-CNN, and therefore is able to produce an additional mask output (different from class and box outputs) from extraction of very fine spatial layout of objects in images[1].

Subsequently, we apply optical flow on these initially detected objects throughout the video, and along the process it will identify the moving objects and label them differently from remaining objects that are still. Based on the objects' boxes, the whole image is divided into two regions: objects region and background region (with all objects bounding boxes carved out). An example of output starting image from Mask R-CNN model and the corresponding background mask is plotted below.

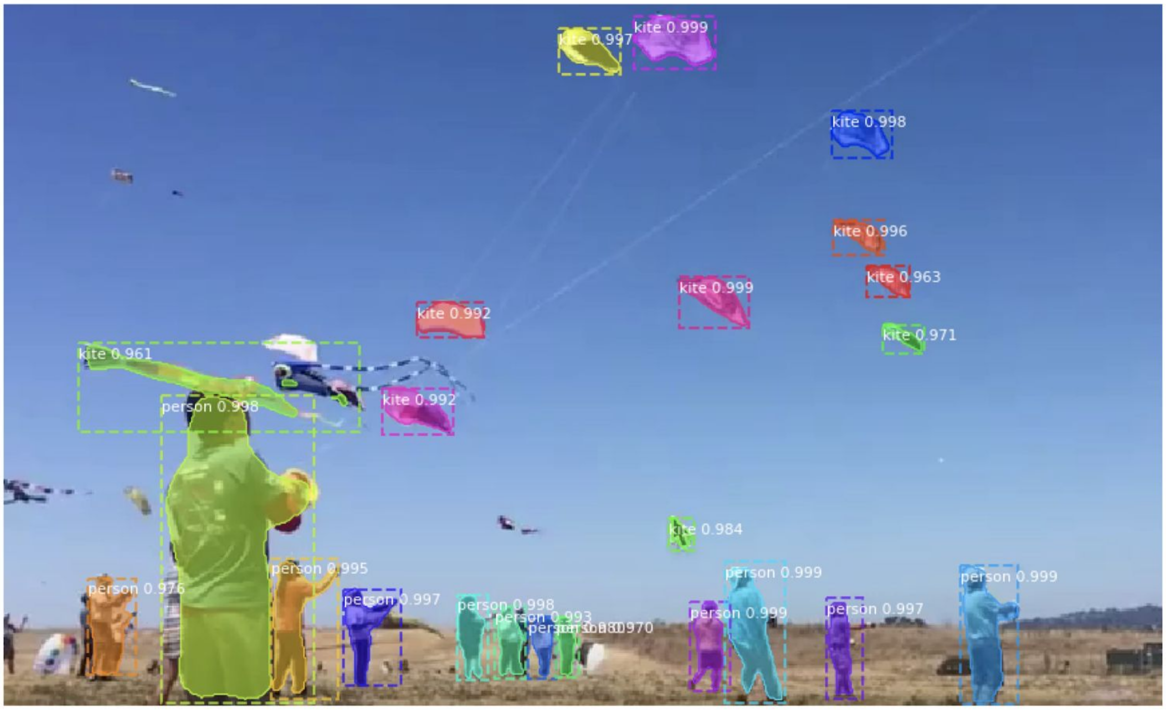

Figure. Output of MRCNN of the first frame of Kites.mp4

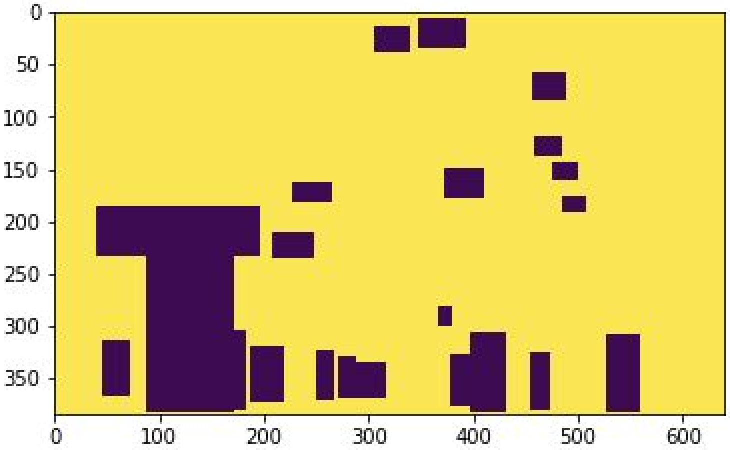

Figure. Background mask the first frame of Kites.mp4

After applying optical flow in both regions with the next frame, appropriate outlier rejection is applied to all background features to minimize the effect of missing any moving object in previous stage. Then the average vector in background region is computed as the baseline movement vector representing the general camera sliding. As for the objects, the bounding box is updated through geometric transformation following the convention from previous optical flow objects. And the vector of box center movements is computed against baseline movements. The L2 norm distance between these two vectors (object and the baseline) is used to decide the whether the object is moving or not. To update features from moving scenes and identify possible new objects entering the frame, for every *k* frames, the next capture from raw video is sent back to RCNN model to re-locate objects and features.

For object detection & instance segmentation portion of the project, the open source code of Mask R-CNN (from a GitHub repository listed in reference #2) is included in “mrcnn_detect.py” and the “mrcnn” directory[2]. For the optical flow portion that includes tracking features, several opencv packages are used as listed below: goodFeaturesToTrack(), calcOpticalFlowPyrLK() and some auxiliary visualization packages.

Result

We have tested our application on the following few videos of different varieties: NBA basketball game clip, donuts eating, kite flying competition, moving vehicles on street with pedestrians and corgi running contest, and the resulting videos have marked moving objects with red rectangle (with “moving” before objects’ class names) and the still ones with blue rectangle (with “still” before objects’ class names). These videos all display good detection and tracking of different objects, and are saved with respective names (with “_result” in the end) in the “output_videos” directory where there are in total six of them.

Conclusion

Through our designed algorithm involving Mask R-CNN for object detection, optical flow for objects tracking and moving objects detection, our application can successfully detect movements in the direction parallel to the (moving) camera, as shown from resulting videos displaying fairly accurate representation of both detecting & tracking objects as well as determining whether they are moving. Thus, we can state that the Mask R-CNN along with optical flow and moving detection using vector gradient work quite well at achieving our goal here.

On the other hand, there are several potential improvements to our existing algorithm & applications, three of which are of the following: enabling to identify an object as moving when it is (or he/she is) moving toward the camera, even though its (or his/her) position relative to the background remains roughly the same (as shown in the figure below, where the moving vehicles are only labelled “moving” if they get very close to the camera); enabling bounding boxes to remain their sizes (not enlarged) when features of bounded objects move in different direction (such as animals or people); and including calculation of objects’ moving speeds to better reflect objects’ moving status.

Reference

1. Kaiming, H., Gkioxari, G., Dollar, P., & Girshick, R. (2018). Mask R-CNN. https://arxiv.org/pdf/1703.06870.pdf

2. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. https://github.com/matterport/Mask_RCNN