A Low-Power Neuromorphic System for Real-Time Visual Activity Recognition

Deepak Khosla(✉), Ryan Uhlenbrock, and Yang Chen

HRL Laboratories, Malibu, CA 90265, USA
dkhosla@hrl.com

Abstract. We describe a high-accuracy, real-time, neuromorphic method and system for activity recognition in streaming or recorded videos from static and moving platforms that can detect even small objects and activities with high-accuracy. Our system modifies and integrates multiple independent algorithms into an end-to-end system consisting of five primary modules: object detection, object tracking, convolutional neural network image feature extractor, recurrent neural network sequence feature extractor, and an activity classifier. We also integrate neuromorphic principles of foveated detection similar to how the retina works in the human visual system and the use of contextual knowledge about activities to filter the activity recognition results. We mapped the complete activity recognition pipeline to the COTS NVIDIA Tegra TX2 development kit and demonstrate real-time activity recognition from streaming drone videos at less than 10 W power consumption.

Keywords: Activity recognition · Behavior recognition · Foveated detection
Neuromorphic · Aerial surveillance · Onboard video processing
Deep learning

1 Introduction

Visual activity recognition has many applications for surveillance and autonomous vehicles. For these applications it is necessary to recognize activities in unconstrained videos. Much activity recognition research focuses on more constrained videos, where the activity is spatially the center, dominant focus of the video, and temporally the video is trimmed to contain mostly only the activity of interest. Such constrained videos are easier to collect and label, so large datasets for training and evaluation more often consist of videos of this type. Leveraging activity recognition models trained on such datasets can be a benefit for models in the unconstrained domain, but applying them is non-trivial. We propose a visual activity recognition system that can apply pre-trained models and overcome the spatial and temporal challenges of unconstrained videos. We make use of recent advances in convolutional neural network-based object detection and classification systems combined with tracking as an initialization for activity detection candidates. Further, we use a foveated object detection technique to improve object localization and small object detection. We use the objects detected in the foveation phase as context to inform constraints on the activity classification.

G. Bebis et al. (Eds.): ISVC 2018, LNCS 11241, pp. 106–115, 2018.
https://doi.org/10.1007/978-3-030-03801-4_10

We temporally integrate the activity classification model and context information to identify short duration activities in long, untrimmed videos.

2 Related Work

There are many methods for activity recognition in videos [1–9]. The state of the art is in using deep learning methods. One main limitation of many of these methods is that they only address the activity classification problem: they assume the input is an activity video clip that is centered on and contains just the activity of interest. They are not applicable to detect and classify applications where the scene may contain multiple objects, clutter, and the actual activity of interest occupies a small spatio-temporal segment of the video. In this class of problems, the objects of interest first need to be detected, classified and tracked before activity classification can be carried out. In addition, the platform may be aerial or ground and static or moving.

3 Method

Figure 1 shows our system block diagram for real-time activity recognition in streaming or recorded videos from static or moving platforms. We describe the individual components in the following subsections.

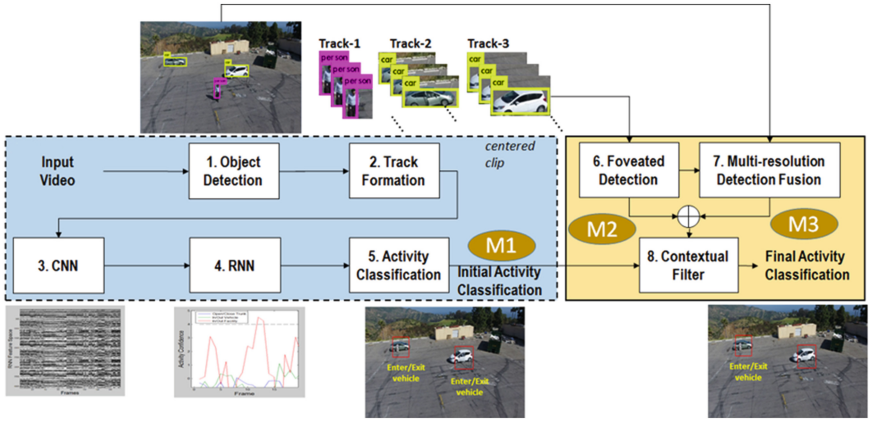

Fig. 1. Block diagram of our activity recognition approach. Blue box (left, dashed) shows the baseline architecture. Yellow box (right, solid) shows improvements via foveation and context. M1, M2 and M3 are the three main methods we compared in Sect. 4.3. (Color figure online)

3.1 Object Detection

The object detection module finds and recognizes objects of interest in the input video and outputs their bounding box location and class label. For example, if the objective is

human activity recognition, then this module detects and classifies all human objects in the incoming video. If the objective is vehicle activity recognition, then this detects and classifies all vehicle objects in the incoming video. This module uses our prior work for real-time object recognition from aerial platforms [10].

3.2 Track Formation

Activity tracks are now formed by tracking detected objects across frames. The matching of bounding boxes from the current frame to the previous frame is done with the Munkres version of the Hungarian algorithm. The cost is computed using bounding box overlap ratio between the predicted bounding box and the previous bounding box. The algorithm is used to compute an assignment which minimizes the total cost. Sporadic detections of moving trees, shadows, etc. are removed by only considering tracks with a minimum duration of T seconds (e.g., T is nominally 2 s). The output of this module is persistent object tracks that have a minimum duration of T seconds. For example, if a person is carrying a gun in the video and visible for 5 s, this module will output a track of that object with a unique track number during those 5 s.

3.3 Convolutional Neural Network Feature Extraction

Persistent tracks are input to a convolutional neural network feature extractor. Track bounding boxes may be enlarged by X% (typically 20%) before feature extraction to help with jitter in the underlying detection bounding boxes. We used an Inception v2 model pre-trained on ImageNet 21K classification task as the CNN for spatial feature extraction.

3.4 Recurrent Neural Network Activity Classifier

The CNN module is followed by a recurrent neural network which extracts temporal sequence features. Since activities may have variable time gap between motion (e.g., person entering a building slowly vs. quickly), we chose the Long Short-Term Memory (LSTM) network as the temporal component. The LSTM RNN takes as input the feature vector from the CNN. The sequence of these features over N frames, typically N = 16 frames, updates the RNN's internal state at each frame. In this invention, we train the 256-hidden-state RNN/LSTM stage on a combination of UCF-101 activity recognition and VIRAT data sets. The RNN's 256-dimensional internal state at the end of the N frame sequence is used as the output of the RNN stage, which is input to a final layer classifier.

3.5 Activity Classifier

Assuming we have K activities to classify, a final fully-connected layer with K outputs gives the final class probability. Alternatively the RNN features can be sent to a Support Vector Machine (SVM) classifier with K outputs. The final output is a probability or confidence score (range 0–1) for each of the K classes. In the case where we only intend to recognize certain types of activity, no softmax is used, and instead a threshold is

placed on the output response of the K output nodes to determine when an activity of interest is detected. Other activities, e.g. a person walking, should have no output above the threshold and receive effectively a label of "no relevant activity". In case of a winner take all embodiment, the activity with the high confidence is the activity label of that track. Modules 3–5 are run in parallel for each track from Module 2.

3.6 Foveated Detection

We leverage the relationship between entity detection and activity detection to design a foveated detection system in which the detection network is first run on the full frame resolution, then for each detected and robust track, the detection network is run again on a foveated region around the track center and expanded larger than the track size (1.5x the size corresponding to the track box). Detections from this second pass replace those in the foveated region from the first pass.

3.7 Multi-resolution Detection Fusion

We run our object detector twice on the incoming video. During the first pass, it analyzes the full video at the native resolution and detects potential objects. A tracker is initiated on every detected object. During the second pass, it analyzes the bounding boxes corresponding to all robust tracks at its resolution to further detect any objects within them that may have been missed in the first pass. This second pass is foveated detection. If the first pass detection is accurate, then no new information is gained in the second pass; it only serves as a confirmation. In some cases (e.g., person in front of car), the first pass misses detection of the smaller object (e.g., person), whereas the second pass run on the car track bounding box detects a new object. Although it is possible, we did not see any instance where the first pass detects more objects than the second pass in our data. We append the detected objects from the first and second pass into a single detected-objects list and use that for context in the next contextual filtering step.

3.8 Contextual Filter

We experimented with combining foveated detection and an entity-based contextual filter on our activity classification probabilities to improve activity recognition. Our activities of interest involve people interacting with vehicles or people alone. So the presence or absence of a person or vehicle is closely tied to what activities are possibly occurring in a given region of interest. Our convolutional and recurrent neural networks don't explicitly have this entity information as input. Our entity detection and localization is generally robust for these two classes. We implemented a filter logic that modifies the activity class probabilities from the neural network based on the detected entities (i.e., context). The logic is based on common sense intuition about the activities. The possible activities are Open/Close Trunk, In/Out Vehicle, In/Out Facility, Person Walking, Person Carrying Weapon, and Person Aiming Weapon. When there are no vehicles or people in a region of interest, no activity is possible. When a vehicle

is present, In/Out Facility is not possible; its class probability is set to 0. When a person is present without a vehicle, Open/Close Trunk and In/Out Vehicle are not possible; their probabilities are set to 0. Softmax is applied after the filter to renormalize the activity class probability distribution.

4 Results

4.1 VIRAT Dataset

We first evaluated our approach on the Video and Image Retrieval and Analysis Tool (VIRAT) dataset. This dataset is designed to be realistic, natural and challenging for video surveillance domains in terms of its resolution, background clutter, diversity in scenes, and human activity and event categories than existing action recognition datasets. We used a subset of the dataset contains several HD videos of people performing various everyday activities. The ground truth annotation specifies the type of activity as well as bounding box and temporal range for activities in each video. There are 12 classes of activities annotated. We combined three pairs of similar activities to pose this as a K = 3-class activity classification problem: Open/Close Trunk, In/Out Vehicle, and In/Out Facility (see Fig. 2).

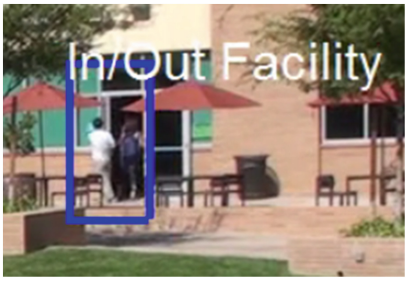

Fig. 2. Example of In/out facility activity classification.

For this evaluation, we focused on activity classification only (i.e., Integrated modules 3, 4 and 5 of Fig. 1). We evaluated four different methods using ground-truth based video clips (16 evenly spaced frames from each activity and rescaling the images to 360 × 360 pixels). We used the CNN-RNN as a 256-dimensional feature extractor and trained a new SVM last layer classifiers for K = 3 activities. The SVMs were trained on either the CNN features-averaged across the 16 frames, RNN features-averaged, RNN features-concatenated, or RNN features selected from the last frame. We evaluated the performance with cross-validation using a split of 80% training and 20% testing. Table 1 shows the activity classification scores with these four methods.

Table 1. Classification accuracy of 3-class VIRAT dataset.

Method	Classification accuracy
1. CNN only	90.8%
2. CNN + RNN averaged	88.6%
3. CNN + RNN concatenated	90.9%
4. CNN + RNN last frame	92.8%

4.2 HRL Parking Lot Dataset

We collected additional activity videos on the HRL campus in order to test the generalization performance of our classifiers. Two pan-tilt-zoom cameras looking down to a parking lot were mounted on a campus building. We recorded 45 min of video while people went through the parking lot, specifically performing the activities of opening/closing a trunk and getting in/out of a vehicle (see Fig. 3). The videos are in color with resolution 704 × 480. We annotated the videos with bounding boxes and start/stop times as the ground truth. This resulted in 47 trunk open/close and 40 vehicle in/out sequences. We used a classifier trained on features extracted by a CNN from the VIRAT dataset on three classes (open/close trunk, in/out vehicle, in/out building). Table 2 shows the activity recognition accuracy.

Fig. 3. Example video footage collected from HRL campus using EO cameras.

Table 2. Global activity recognition accuracy of 80.5% from HRL parking lot dataset.

	Open/Close Trunk	In/Out Vehicle	In/Out Facility	Accuracy %
Open/Close Trunk	32	15	0	68%
In/Out Vehicle	2	38	0	95%
In/Out Facility	0	0	0	NA

4.3 HRL Drone Dataset

We also evaluated our approach on multiple video datasets collected from a DJI quadcopter drone at a helipad and parking lot. The dataset involves multiple people and cars performing various activities with the drone hovering over and collecting data from two different viewpoints. The videos are in color with 4K resolution. We completed ground-truth annotation of the videos with bounding boxes and start/stop times. We annotated seven classes of activities: {In/Out Vehicle, Open/Close Trunk, In/out Facility, Person walking, Person Carrying Weapon, Person Aiming Weapon, None} (Fig. 4).

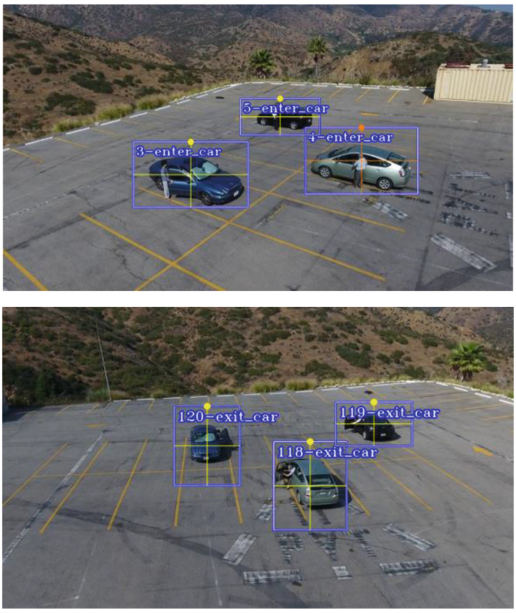

Fig. 4. Example annotations created for HRL August drone data set for In Vehicle (top) and Out Vehicle (bottom) from two different angles.

As described in Modules 3.3 and 3.4, we trained our deep learning architecture based on CNN and RNN for these 7 classes of activities. We used an Inception v2 model pre-trained on ImageNet 21K classification task as the CNN for spatial feature extraction, and a 256-hidden-state RNN/LSTM stage for activity recognition trained on a combination of UCF-101 activity recognition and VIRAT data sets.

The test protocol for the online streaming processing scheme uses an object detector to seed an object tracker. When the tracker has accumulated 16 frames of a tracked object, the activity classifier will be invoked. Since In/out Facility and Person walking are under-represented in the data, we only present results of the other activities in Tables 3 and 4 below. Figure 5 shows a typical result.

Table 3. Summary results across all activities on HRL drone dataset. Method M3 generally performs better than M1 or M2 (high PC, low FPPI).

Method	PD	PC	FPPI
M1	0.87	0.38	2.03
M2	0.91	0.40	1.82
M3	0.88	0.71	1.72

Table 4. Individual class activity results on HRL drone dataset.

M1	PD	PC	FPPI
In/Out Vehicle	0.90	0.90	2.42
Open/Close Trunk	0.97	0.05	2.60
Aim/Carry Weapon	0.73	0.18	0.26
M2	PD	PC	FPPI
In/Out Vehicle	0.89	0.89	2.09
Open/Close Trunk	0.93	0.25	2.55
Aim/Carry Weapon	0.87	0.27	0.27
M3	PD	PC	FPPI
In/Out Vehicle	0.80	0.73	2.06
Open/Close Trunk	0.91	0.64	2.41
Aim/Carry Weapon	0.92	0.80	0.35

Fig. 5. Typical recognized activity (see text above box) and detected entities (see text below box) using M3.

We evaluated the performance of three methods (M1, M2, and M3) as shown in Fig. 1. Method 1 (M1) is the system without foveated detection. Method 2 (M2) uses foveated detection and contextual filter path. Method 3 (M3) uses multi-resolution detection fusion and contextual filter.

4.4 Hardware Setup

Figure 6 shows a typical setup to collect and process video data from the DJI Inspire quadcopter. In the current invention, we can process results on either a computer with a GPU card or the NIVIDIA Tegra TX1 board. The desktop software and demonstration system runs under Ubuntu Linux 14.04 and requires a NVIDIA GPU to function.

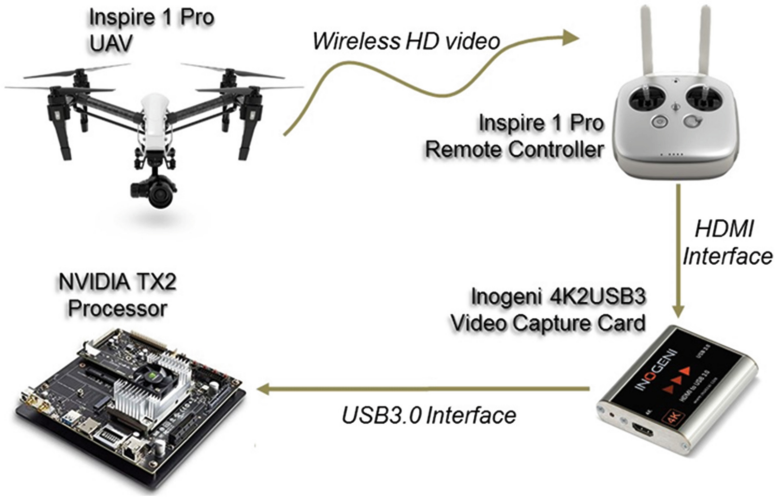

Fig. 6. Drone aerial video processing architecture.

We have mapped the complete activity recognition pipeline to the NVIDIA Tegra TX2 development board and systematically evaluated algorithmic performance in terms of frames per second and the power consumptions. As can be seen in Table 5, we can achieve a throughput of 9.9 frames per second for the full HD video at a processing power of 10 W.

Table 5. Processing throughput and power consumption for full video activity recognition.

	Display live video	Object recognition	Activity recognition (localized)	Activity recognition (full video)
Frames per second	158	16.4	47.8	9.9
Processing power (W)	5.5	8	7.5	10

Acknowledgments. This material is based upon work supported by the Office of Naval Research (ONR) under Contract No. N00014-15-C-0091. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research (ONR).

References

1. Kalogeiton, V., et al.: Action tubelet detector for spatio-temporal action localization. In: ICCV-IEEE International Conference on Computer Vision (2017)
2. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Res. Logist. (NRL) **2**(1–2), 83–97 (1955)
3. Karpathy, A., et al.: Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2014)
4. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems (2014)
5. Vrigkas, M., Nikou, C., Kakadiaris, I.A.: A review of human activity recognition methods. Front. Robot. AI **2**, 28 (2015)
6. Donahue, J., et al.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
7. Oh, S., et al.: A large-scale benchmark dataset for event recognition in surveillance video. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2011)
8. Soomro, K., Zamir, A.R., Shah, M.: UCF101: a dataset of 101 human actions classes from videos in the wild (2012)
9. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. arXiv preprint (2017)
10. Khosla, D., Chen, Y., Kim, K.: A neuromorphic system for video object recognition. Front. Comput. Neurosci. **8**, 147 (2014)