Violence Detection in Drone Surveillance Videos

H. H. Adithya, H. J. Lekhashree, and S. Raghuram

Abstract Drones are increasing in popularity for many applications in the private and government sectors. New applications such as land surveying and wildlife movement monitoring have been made simpler with the advent of drones in these domains. A majority of the drones are fitted with some sort of vision capture system—such as a video camera and are capturing the activity in an area as dictated by the application domain. In this work, we have added analytics functionality to the drone video stream—in particular, we have created a machine learning model which is capable of detecting physical violence between individuals. With the increase in the application domains of drones, their numbers are set to increase dramatically. An important capability is that of vision, and it can be put to good use if analytics are used to detect certain anomalous events in the drone's field of vision. While there is an intended application and journey, this can be altered if such events are detected. For example, in the case of physical violence, if any activity of this nature is detected, the drone can stop and continue to capture the activity, instead of continuing on its usual route. This will have the beneficial effect of both capturing the activity and will also have an indirect effect of the perpetrators ceasing to continue the activity, as they are being watched. The proposed solution applied deep learning models for the detection of events of interest. Deep learning has emerged as the model of choice for classification applications. A primary reason for their adoption is because they obviate the complicated feature engineering process, due to their capability of learning features. From our experiments, we see that we are able to achieve a validation accuracy of over 92% on our dataset.

Keywords Drone · Security · Deep neural networks

H. H. Adithya (✉) · H. J. Lekhashree · S. Raghuram
Department ECE, M S Ramaiah Institute of Technology, 560054 Bengaluru, India
e-mail: hadithyaraj820@gmail.com

S. Raghuram
e-mail: raghuram@msrit.edu

T. Senjyu et al. (eds.), *Smart Trends in Computing and Communications*, Lecture Notes in Networks and Systems 650, https://doi.org/10.1007/978-981-99-0838-7_60

1 Detection of Physical Violence

Physical violence is an activity of extreme importance during surveillance tasks. Currently, CCTV cameras are installed in all spaces such as transportation hubs, malls, and educational institutions. To take their utility to the next level, it is essential to add analytics capabilities to these devices. For example, if an illegal activity is being carried out, the presence of the CCTV is not a major deterrent as the perpetrators can cover their faces and still continue the activity. However, if the CCTV camera also produces a sound, like a loud beeping sound, then it will have a much higher impact and will immediately lead to a cessation of the activities. In this way, the analytics capability has the potential to turn a passive device into an active device, almost the equivalent of a human security guard.

In this work, we propose to add such capabilities to the surveillance video captured by drones during their routine operation. In particular, in the autonomous cases which are more common for applications such as parcel delivery where a path is chosen for the flight, if the drone detects any activity of suspicion during its flight, then it must perform some additional actions such as inform the base station, or stop its normal flight and continue to capture the video. In particular, we focus on how these analytics capabilities can be developed.

1.1 Video Analytics with Deep Neural Networks

Since the advent of Deep Neural Networks [1, 2] and their eventual adoption due to advances in software libraries and Graphical Processing Units, they have become the model of choice for classification tasks. A Deep Neural Network (DNN) [3] is an evolution of the earlier Artificial Neural Network (ANN) which contains a single hidden layer. A DNN contains multiple hidden layers between the input and the output. While traditional ANNs are theoretically capable of learning any arbitrary functions, the depth of the hidden layer becomes such that training is not feasible. In a DNN, the idea of hierarchical learning is applied, wherein the layers learn the features in a bottom-up manner, hence making depth a critical ingredient of the learning process. While image-based classification tasks such as object detection or classification [4, 5] are the primary application domains of deep neural networks, they have expanded to domains such as medical image segmentation [6] and cancer detection [7]. In this work, we have developed a DNN model suitable for the detection of physical violence in drone video streams.

Traditional video analytics approaches used flow vectors to determine actions across the frames. When such techniques are adapted for the particular surveillance problem that we are interested in, then complicated feature engineering is required to detect the particular type of action from the flow vectors of two or more individuals. To avoid this task, we look at Deep Neural Networks, where a large set of training

data is utilized to learn the features of interest, though they may not have any physical significance like flow vectors.

2 Related Work

Kyrkou and Theocharides [8] focus on the efficient aerial image classification from on-board a UAV for emergency response/monitoring applications. A dedicated database for this task is constructed, referred to as AIDER (Aerial Image Dataset for Emergency Response applications). The dataset construction involved manually collecting all images for four disaster events, namely Fire/Smoke, Flood, Collapsed Building/Rubble, and Traffic Accidents, as well as one class for the Normal case. Through this analysis a lightweight convolutional neural network architecture is proposed, referred to as EmergencyNet, based on atrous convolutions to process multiresolutional features. EfficientNet: A new CNN scaling approach is proposed in this article that uniformly scales all dimensions of depth/width/resolution in a principled manner in order to achieve better accuracy and efficiency. Atrous Convolutional Feature Fusion (ACFF): Atrous (also called dilated) convolutions can capture and transform images at different resolutions depending on the dilation rate which determines the spacing between the kernel points, effectively increasing their receptive field without increasing the parameter count.

In [9], detection of actual fights on long duration CCTV recordings is done by firstly proposing CCTV-Fights 1, a novel and challenging dataset containing 1,000 videos of real fights, with more than 8 h of annotated CCTV footage. The C3D architecture was utilized as a feature extractor, applying directly the weights learned from Sports-1M dataset, and the output from the layer "fc7" as the extracted feature vector. With respect to TRoF, it is used to extract low-level spatiotemporal features from the videos, then the features undergo PCA transformation for whitening and reducing their dimensionality by half. Subsequently, Fisher Vectors (FV) is applied for mid-level generation of features for each snippet of the video. For the snippet prediction stage, the LSTM architecture and training hyperparameters were picked by grid-searching and using a validation split for measuring the performance. The SVM hyperparameter is chosen through grid search and cross-validation during the training phase. Following the protocol from MediaEval, to deem a prediction as a hit (i.e., to contain a fight), it is necessary for it to have at least 50% of its length overlapping with a ground-truth segment.

UCF-Crime dataset [10] is a new large-scale first-of-its-kind dataset of 128 h of videos. It consists of 1900 long and untrimmed real-world surveillance videos, with 13 realistic anomalies including Abuse, Arrest, Arson, Assault, Road Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. These anomalies are selected because they have a significant impact on public safety. This dataset can be used for two tasks. First, general anomaly detection considers all anomalies in one group and all normal activities in another group and second, for recognizing each of 13 anomalous activities.

3 Dataset Preparation and DNN Design

In this work, we are feeding the video dataset to a 3D Convolutional Network—while traditional activities like object detection can be done using images, to classify if contact between two individuals is violent or non-violent requires more information from multiple frames. While previous approaches have used flow prediction algorithms with CNNs (e.g., Lucas-Kanade optical flow) in this work, we are attempting to extract the flow level features using 3D convolutions (Fig. 1).

3.1 Dataset Preparation

Deep neural networks require large amounts of data to be able to learn the appropriate features for the task. While the UCF crime dataset had multiple categories of violence, only 963 of those were related to physical violence involving people. To improve the data samples, datasets provided by [11, 12] were further investigated and the appropriate videos were selected from these datasets. These videos are clipped into 4363 segments, each of length 3–10 s. Of these segments, 1320 are with some violent activity, while 3043 segments are normal activities involving people. The violence segments include only activities that contain two or more individuals involving in violent activities related to physical contact (i.e., guns or knives thrown from a distance are not captured) Sample activities from the dataset are shown in Fig. 2. Further, all of the video segments are authentic, none containing any actors or synthetic videos and the samples are mostly recorded by bystanders from mobile phones. The images have been normalized to have a mean of 0 and a standard deviation of 1 and the image has been resized to 227 $\times$ 227 from the original if 32 $\times$ 32.

3.2 Drone Dataset

A drone was also used to film videos of various actions. Since these videos were not available online or in another dataset, we created these scenes using a drone procured for this purpose. Samples of videos with violence and nonviolence are

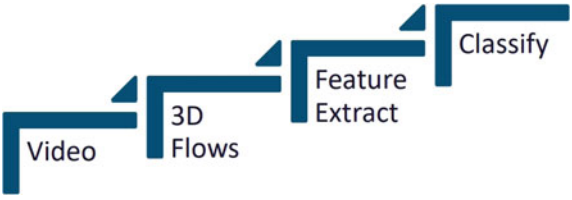

Fig. 1 Proposed video classification methodology

Fig. 2 Sample screenshots of violence videos

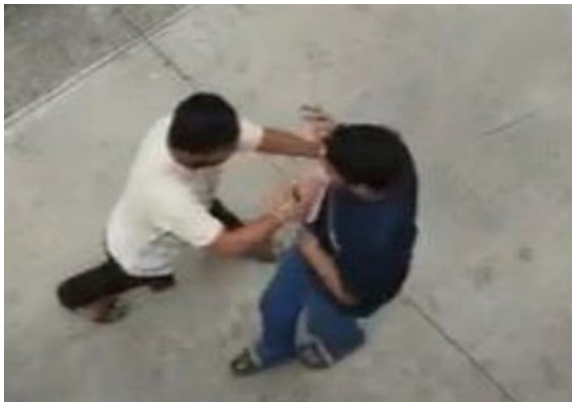

Fig. 3 Sample frame from drone dataset

shown in Fig. 3. A total of 20 videos were created and were equally divided between the two classes.

3.3 DNN Design

A neural network architecture that resembles what we are attempting is shown in Fig. 2, from [13]. The idea of 3D convolutions to capture the flow information across the frames is an ideal way to transform the feature engineering problem into one of learning the appropriate weights. In Fig. 4, all the convolution layers use 3D convolutions instead of the traditional 2D operations used for images.

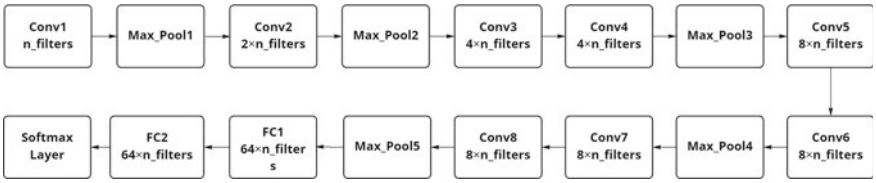

Fig. 4 DNN design for action recognition [13]

3.3.1 Reduction in Filters

The above network was trained on the dataset, with a modification in the number of filters. Since the above network was for general-purpose action reduction, whereas we are interested in a two class problem, it was reasoned that fewer filters will be sufficient for our two class case.

3.3.2 Modification in Depth

Next, we tried modifying the number of layers, along with the number of filters. This is to check the suitability of the deep neural network for our dataset. With respect to the network architecture from [13], we decreased and increased the number of layers from 8 to 5 and 8 to 15, respectively.

4 Results and Discussion

We started with the exact architecture replicated from [13], and trained on our dataset. The filters are changed, and the value of the number of filters in the first layer is given in the table. 64 is the number of filters from the paper, while we have experimented by starting with 8, 16, and 32, in addition to 64. The filters then get doubled in each subsequent layer, as in [13].

The following terminology is used for discussing the results:

- CNVP: % of correct non-violence videos predicted
- CVP: % of correct violence videos predicted
- CDNVP: % of correct drone non-violence videos predicted
- CDVP: % of correct drone violence videos predicted.

In short, the count for violence and nonviolence from dataset collected is CNVP and CVP, while CDNVP and CDVP are for the drone videos created. The number of videos used to generate the results below is as follows (they were not used during training):

Table 1 Training and testing metrix for model with 8 convolution layers

	n-filters = 8	n-filters = 16	n-filters = 32	n-filters = 64
Batch size = 10	Accuracy: 70.51	Accuracy: 70.51	Accuracy: 71.97	Accuracy: 70.51
	Val-Acc: 66.66	Val-Acc: 66.66	Val-Acc: 69.30	Val-Acc: 66.66
	CNVP: 100	CNVP: 100	CNVP: 100	CNVP: 10
	CVP: 0	CVP: 0	CVP: 0	CVP: 64
	CDNVP: 100	CDNVP: 100	CDNVP: 100	CDNVP: 100
	CDVP: 0	CDVP: 0	CDVP: 0	CDVP: 0
Batch size = 12	Accuracy: 95.50	Accuracy: 70.51	Accuracy: 70.63	Accuracy: 70.51
	Val-Acc: 75.14	Val-Acc: 66.66	Val-Acc: 66.89	Val-Acc: 66.78
	CNVP: 0	CNVP: 100	CNVP: 50	CNVP: 100
	CVP: 100	CVP: 0	CVP: 8.3	CVP: 0
	CDNVP: 0	CDNVP: 100	CDNVP: 100	CDNVP: 100
	CDVP: 100	CDVP: 0	CDVP: 0	CDVP: 0
Batch size = 18	Accuracy: 94.21	Accuracy: 76.73	Accuracy: 70.51	Accuracy: 73.75
	Val-Acc: 91.52	Val-Acc: 66.78	Val-Acc: 66.66	Val-Acc: 69.53
	CNVP: 0	CNVP: 100	CNVP: 100	CNVP: 100
	CVP: 100	CVP: 0	CVP: 0	CVP: 0
	CDNVP: 0	CDNVP: 100	CDNVP: 100	CDNVP: 100
	CDVP: 100	CDVP: 0	CDVP: 0	CDVP: 0

1. CVP: 12
2. CNVP: 8
3. CDVP: 4
4. CDNVP: 4.

From Table 1, we see that we get the highest training accuracy when starting with the lowest number of filters, 8, and progressing to 64 for the final two FC layers. However, when testing with the dataset, and drone videos, the accuracies are uniformly poor across all examples. Next, we modify the architecture to have a total of five layers, reduced from the eight layers in [13].

From Table 2, we see that the overall training and validation accuracies are increased in almost among all the examples, except when using 64 filters in the first stage. This seems to be a much better fit for the dataset and the classification problem. Table 2 also shows the accuracies for the test videos; we see that for using batch size of 12, we almost get perfect results for number of filters = 16 in the first layer, only one drone video is misclassified, while two videos from the datasets are misclassified, giving a total accuracy of 88% (23/26). Next, we show the results for increasing the number of layers to 10. From Table 3, we see that the overall accuracies are less than both the previous cases, i.e., 3D CNNs with 5 layers, and 3D CNNs with 8 layers. Consequently, the accuracies for the test videos in Table 3 are also lesser than both the cases. Tran et al. [13] did not use any dropout layers, since the 3D CNNs with five layers gave the best results, we added dropout (0.5 uniformly)

Table 2 Training and testing metrix for model with 5 convolution layers

	n-filters = 8	n-filters = 16	n-filters = 32	n-filters = 64
Batch size = 10	Accuracy: 99.57	Accuracy: 98.33	Accuracy: 96.36	Accuracy: 70.51
	Val-Acc: 90.72	Val-Acc: 92.78	Val-Acc: 92.55	Val-Acc: 66.78
	CNVP: 37.5	CNVP: 25	CNVP: 75	CNVP: 100
	CVP: 66.66	CVP: 100	CVP: 25	CVP: 0
	CDNVP: 75	CDNVP: 0	CDNVP: 100	CDNVP: 100
	CDVP: 50	CDVP: 100	CDVP: 0	CDVP: 0
Batch size = 12	Accuracy: 99.36	Accuracy: 97.73	Accuracy: 97.16	Accuracy: 77.96
	Val-Acc: 91.63	Val-Acc: 91.63	Val-Acc: 93.24	Val-Acc: 73.71
	CNVP: 25	CNVP: 75	CNVP: 37.5	CNVP: 75
	CVP: 91.66	CVP: 100	CVP: 100	CVP: 0
	CDNVP: 50	CDNVP: 75	CDNVP: 50	CDNVP: 100
	CDVP: 100	CDVP: 100	CDVP: 100	CDVP: 0
Batch size = 18	Accuracy: 99.45	Accuracy: 99.42	Accuracy: 97.04	Accuracy: 96.93
	Val-Acc: 92.78	Val-Acc: 93.35	Val-Acc: 91.63	Val-Acc: 91.29
	CNVP: 12.5	CNVP: 12.5	CNVP: 0	CNVP: 0
	CVP: 100	CVP: 100	CVP: 100	CVP: 100
	CDNVP: 25	CDNVP: 25	CDNVP: 0	CDNVP: 0
	CDVP: 100	CDVP: 100	CDVP: 100	CDVP: 100

Table 3 Training and testing metrix for model with 10 convolution layers

	n-filters = 8	n-filters = 16	n-filters = 32	n-filters = 64
Batch size = 10	Accuracy: 70.51	Accuracy: 70.51	Accuracy: 70.51	Accuracy: 70.51
	Val-Acc: 66.66	Val-Acc: 66.66	Val-Acc: 66.66	Val-Acc: 66.66
	CNVP: 100	CNVP: 100	CNVP: 100	CNVP: 100
	CVP: 0	CVP: 0	CVP: 0	CVP: 0
	CDNVP: 100	CDNVP: 100	CDNVP: 100	CDNVP: 100
	CDVP: 0	CDVP: 0	CDVP: 0	CDVP: 0
Batch size = 12	Accuracy: 89.36	Accuracy: 70.51	Accuracy: 70.51	Accuracy: 70.51
	Val-Acc:87.74	Val-Acc: 66.89	Val-Acc: 66.66	Val-Acc: 66.66
	CNVP: 50	CNVP: 100	CNVP: 62.5	CNVP: 100
	CVP: 58.33	CVP: 0	CVP: 25	CVP: 0
	CDNVP: 25	CDNVP: 100	CDNVP: 25	CDNVP: 100
	CDVP: 50	CDVP: 0	CDVP: 0	CDVP: 0
Batch size = 18	Accuracy: 91.97	Accuracy: 70.51	Accuracy: 70.51	Accuracy: 70.51
	Val-Acc: 87.74	Val-Acc: 66.66	Val-Acc: 66.66	Val-Acc: 66.66
	CNVP: 0	CNVP: 100	CNVP: 100	CNVP: 100
	CVP: 91.66	CVP: 0	CVP: 0	CVP: 0
	CDNVP: 0	CDNVP: 100	CDNVP: 100	CDNVP: 100
	CDVP: 100	CDVP: 0	CDVP: 0	CDVP: 0

Table 4 Training and testing metrix for model with 8 convolution layers and 1 dropout layer at the end

	n-filters = 8	n-filters = 16	n-filters = 32	n-filters = 64
Batch size = 10	Accuracy: 97.90	Accuracy: 98.30	Accuracy:70.51	Accuracy: 70.51
	Val-Acc: 91.86	Val-Acc: 92.32	Val-Acc: 66.66	Val-Acc: 66.66
	CNVP: 37.5	CNVP: 50	CNVP: 62.5	CNVP: 100
	CVP: 91.66	CVP: 83.33	CVP: 0	CVP: 25
	CDNVP: 75	CDNVP: 50	CDNVP: 100	CDNVP: 100
	CDVP: 100	CDVP: 100	CDVP: 0	CDVP: 75
Batch size = 12	Accuracy: 98.73	Accuracy: 98.05	Accuracy: 87.30	Accuracy: 83.29
	Val-Acc: 91.98	Val-Acc: 93.01	Val-Acc: 80.29	Val-Acc: 76.52
	CNVP: 37.5	CNVP: 12.5	CNVP: 0	CNVP: 87.5
	CVP: 91.66	CVP: 100	CVP: 100	CVP: 25
	CDNVP: 0	CDNVP: 25	CDNVP: 0	CDNVP: 100
	CDVP: 100	CDVP: 100	CDVP: 100	CDVP: 0
Batch size = 18	Accuracy: 99.02	Accuracy: 99.22	Accuracy: 97.76	Accuracy: 98.25
	Val-Acc: 93.35	Val-Acc: 92.66	Val-Acc: 91.63	Val-Acc: 92.78
	CNVP: 12.5	CNVP: 25	CNVP: 0	CNVP: 50
	CVP: 91.66	CVP: 100	CVP: 100	CVP: 66.66
	CDNVP: 0	CDNVP: 50	CDNVP: 0	CDNVP: 0
	CDVP: 100	CDVP: 100	CDVP: 100	CDVP: 0

Table 5 Training and testing metrix for model with 8 convolution layers and 2 dropout layers

	n-filters = 8	n-filters = 16	n-filters = 32	n-filters = 64
Batch size = 10	Accuracy: 73.29	Accuracy: 66.70	Accuracy: 59.71	Accuracy: 60.20
	Val-Acc: 86.48	Val-Acc: 78.57	Val-Acc: 66.66	Val-Acc: 66.66
	CNVP: 100	CNVP: 87.5	CNVP: 75	CNVP: 100
	CVP: 0	CVP: 58.3	CVP: 0	CVP: 8.33
	CDNVP: 100	CDNVP: 100	CDNVP: 100	CDNVP: 100
	CDVP: 0	CDVP: 100	CDVP: 0	CDVP: 0
Batch size = 12	Accuracy: 71.51	Accuracy: 59.68	Accuracy:67.67	Accuracy:62.63
	Val-Acc: 83.61	Val-Acc: 66.66	Val-Acc: 72.50	Val-Acc: 70.21
	CNVP: 62.5	CNVP: 75	CNVP: 62.5	CNVP: 100
	CVP: 91.66	CVP: 0	CVP: 25	CVP: 50
	CDNVP: 50	CDNVP: 100	CDNVP: 100	CDNVP: 100
	CDVP: 100	CDVP: 0	CDVP: 0	CDVP: 0
Batch size = 18	Accuracy: 80.20	Accuracy: 78.28	Accuracy: 73.61	Accuracy: 66.81
	Val-Acc: 92.21	Val-Acc: 88.65	Val-Acc: 88.54	Val-Acc: 81.67
	CNVP: 37.5	CNVP: 12.5	CNVP: 12.5	CNVP: 100
	CVP: 100	CVP: 91.66	CVP: 58.33	CVP: 25
	CDNVP: 25	CDNVP: 25	CDNVP: 0	CDNVP: 100
	CDVP: 100	CDVP: 100	CDVP: 0	CDVP: 0

Table 6 Training and testing metrix for model with 8 convolution layers and dropout after each layer

	n-filters = 8	n-filters = 16	n-filters = 32	n-filters = 64
Batch size = 10	Accuracy: 65.15	Accuracy: 65.67	Accuracy: 60.97	Accuracy: 60.80
	Val-Acc: 66.66	Val-Acc: 66.66	Val-Acc: 66.66	Val-Acc: 66.66
	CNVP: 75	CNVP: 100	CNVP: 100	CNVP: 100
	CVP: 33.33	CVP: 0	CVP: 0	CVP: 0
	CDNVP: 50	CDNVP: 100	CDNVP: 100	CDNVP: 100
	CDVP: 0	CDVP: 0	CDVP: 0	CDVP: 0
Batch size = 12	Accuracy: 68.36	Accuracy: 61.28	Accuracy: 59.57	Accuracy: 62.32
	Val-Acc: 68.15	Val-Acc: 66.66	Val-Acc: 66.66	Val-Acc: 69.75
	CNVP: 75	CNVP: 100	CNVP: 100	CNVP: 100
	CVP: 66.66	CVP: 0	CVP: 0	CVP: 0
	CDNVP: 50	CDNVP: 100	CDNVP: 100	CDNVP: 100
	CDVP: 0	CDVP: 0	CDVP: 0	CDVP: 0
Batch size = 18	Accuracy: 62.89	Accuracy: 66.61	Accuracy: 60.11	Accuracy: 59.02
	Val-Acc: 66.66	Val-Acc: 66.66	Val-Acc: 66.66	Val-Acc: 66.66
	CNVP: 100	CNVP: 100	CNVP: 100	CNVP: 100
	CVP: 0	CVP: 0	CVP: 0	CVP: 0
	CDNVP: 100	CDNVP: 100	CDNVP: 100	CDNVP: 100
	CDVP: 0	CDVP: 0	CDVP: 0	CDVP: 0

to different layers to check the accuracy. Interestingly by observing Tables 4, 5 and 6, dropout does not affect the accuracy positively. This may probably be due to the relatively low number of layers in the architecture.

5 Conclusion

3D convolutional neural networks are viable models for video action recognition tasks. While general networks are created for multiple action recognition, these networks can be pruned in terms of the filters and the layers for the violence detection case. This will lead to an optimized implementation capable of running in resource-constrained devices on the edge. From our experiments, we see that a 5-layer 3D CNN network, starting with 16 filters and progressing up to 256 filters gives the best result on the test videos and the videos captured from the drone. Interestingly, no additional modification may be required for a network trained on videos taken on mobile phones by bystanders, to detect violence from a video captured from a drone. This result is from the accuracy of the above network on the drone test set, and it correctly classifies 7 of the 8 drone captured videos, classifying a normal video as anomalous.

References

1. LeCun Y et al (1989) Handwritten digit recognition with a back-propagation network. In: Advances in neural information processing systems, vol 2
2. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
3. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press
4. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision
5. Redmon J et al (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition
6. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, Cham
7. Burt JR et al (2018) Deep learning beyond cats and dogs: recent advances in diagnosing breast cancer with deep neural networks. Br J Radiol 91(1089):20170545
8. Kyrkou C, Theocharides T (2020) EmergencyNet: efficient aerial image classification for drone-based emergency monitoring using atrous convolutional feature fusion. IEEE J Select Top Appl Earth Obs Remote Sens 13:1687–1699
9. Perez M, Kot AC, Rocha A (2019) Detection of real-world fights in surveillance videos. In: ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE
10. Sultani W, Chen C, Shah M (2018) Real-world anomaly detection in surveillance videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition
11. Soliman M, Kamal M, Nashed M, Mostafa Y, Chawky B, Khattab D (2019) Violence recognition from videos using deep learning techniques. In: Proceedings of 9th international conference on intelligent computing and information systems (ICICIS'19), Cairo, pp 79–84
12. Peng W et al (2020) Not only look, but also listen: learning multimodal violence detection under weak supervision. In: ECCV
13. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3D convolutional networks. In: 2015 IEEE international conference on computer vision (ICCV). https://doi.org/10.1109/iccv.2015.510