







Detecting Violent Robberies in CCTV Videos Using Deep Learning

Giorgio Morales^(✉) , Itamar Salazar-Reque , Joel Telles ,
and Daniel Díaz 

National Institute of Research and Training in Telecommunications (INICTEL-UNI),
National University of Engineering, San Luis 1771, 15021 Lima, Peru
{gmorales,jtelles,ddiaz}@inictel.edu.pe,
itamarf.salazar@gmail.com

Abstract. Video surveillance through security cameras has become difficult due to the fact that many systems require manual human inspection for identifying violent or suspicious scenarios, which is practically inefficient. Therefore, the contribution of this paper is twofold: the presentation of a video dataset called UNI-Crime, and the proposal of a violent robbery detection method in CCTV videos using a deep-learning sequence model. Each of the 30 frames of our videos passes through a pre-trained VGG-16 feature extractor; then, all the sequence of features is processed by two convolutional long-short term memory (convLSTM) layers; finally, the last hidden state passes through a series of fully-connected layers in order to obtain a single classification result. The method is able to detect a variety of violent robberies (i.e., armed robberies involving firearms or knives, or robberies showing different level of aggressiveness) with an accuracy of 96.69%.

Keywords: Action recognition · convLSTM · Robbery detection

1 Introduction

Citizen insecurity is one of the most important problems affecting today's people quality of life. This is especially true for developing countries where the problem is exacerbated by the poverty and the lack of opportunities [1]. Of the different ways in which insecurity manifests itself, robberies are the most frequent. To reduce their rate of occurrence, a common solution is to install both indoor cameras – such as in convenience stores, gas stations, or restaurants –, or outdoor cameras – as the public surveillance cameras on the streets managed by the government. Unfortunately, for this solution to be efficient, many resources must be spent. For instance, indoor cameras are normally used just to record the assault and subsequently to identify the robber; but to use them to warn the police when a robbery is being committed, human inspection is needed. This is the approach in public outdoor cameras, where robbery detection relies on the

use of continuously monitored surveillance cameras by security agents. Nevertheless, this is a limited solution due to the small number of agents compared to the number of cameras to be monitored, and to the inherent fatigue caused by this exhausting task.

In this context, the use of artificial intelligence techniques to offer new tools for automatic robbery detection can be of great aid. However, this is a difficult mission since a robbery can happen anywhere in the city, which means that there is a high variety of scenarios that make the solution to this problem a big challenge.

To the best of our knowledge, no studies have been proposed for the automatic detection of robbery. Violence detection, however, do present several previous researches. Some of them used hand-crafted features as in [2] where the authors proposed DiMOLIF, a new feature, based on STIPs [3] and optical flow, to describe violence in surveillance videos. They used this new feature to classify video clips from two datasets using a SVM classifier obtaining an accuracy of 88% and 85% respectively. Deniz et al [4], compute power spectrum of two consecutive frames to detect sudden changes elicited by fast movements to detect violence with an accuracy improvement of up to 12% with respect to the compared methods. It is worth saying that both studies assumed that the actions are fast enough so the difference between frames encoded a violence action. This is not necessarily true for some armed robberies where the criminal can intimidate the victim by holding a gun without a sudden movement.

Some other approaches used deep convolutional neural networks (CNN). In [5], the authors used a 6-layer CNN to classify frame images from videos as normal or abnormal. However, as images were sampled from video sequences there might be a high correlation between samples making the network prone to overfitting. Moreover, as robbery is an action, it will require more than one image to be correctly described. Additionally, the practical use of this approach will require the system to work in every frame which is computationally expensive. In [6], Trajectory-Pooled Deep-Convolutional Descriptors (TDD) were described. They were presented as a new video feature for action recognition that combined the merits of both hand-crafted features (Improved trajectories [7]) and deep-learned features (two-stream ConvNets [8]). This was later used in [9] for violence detection with high accuracy. In [10], the authors proposed an image acceleration field calculated from the optical flow field, which serves as an input to a Convolutional Neural Network, called FighNet. Finally, Sudhakaran et al. [11] developed an end-to-end trainable deep neural network for performing violent video classification using convolutional long short term memory (convLSTM) networks.

In this work, we propose an end-to-end trainable sequence model for violent robbery classification similar to that proposed by [11]. We no longer have to design hand-crafted features and feed them into a classifier; instead, the input of our model is a sequence of 30 RGB frames extracted from the CCTV videos of our dataset. The first part of our architecture is the feature extractor, which process each frame using a pre-trained CNN, such as VGG16 or NASNetMobile.

The second part is the sequence network; it is composed of convLSTM layers, which can encode the spatio-temporal changes of the processed features. We tried different configurations and selected the best network, achieving a classification accuracy of 96,69% in our validation dataset.

2 Proposed Method

2.1 UNI-Crime Dataset

One of the difficulties of training an optimal model for robbery detection is the lack of a proper public dataset. Namely, some common problems are the small number of samples, the poor diversity of scenarios, or the absence of spontaneity in simulated scenes [12–14]. Nevertheless, the main problem is that almost none of the previous datasets contains scenes of robberies recorded by CCTV cameras, which is what we aim to detect. On the other hand, the UCF-Crime dataset [15] contains 1900 CCTV videos of normal actions, robberies, abuse, explosions, among other anomalies. However, those videos have different duration and frame rates; what is more, many of them are edited videos that contain multi-camera view, which is not useful for analyzing the continuity of an action, or present advertisements at the beginning or at the end.

For those reasons, we decided to construct the UNI-Crime dataset [16] based on the UCF-Crime dataset. For this, we discarded those multi-camera and repeated videos. Due to the fact that some videos are too long (three minutes or more) and others too short (20 s), we standardized the duration of our videos to 10 s. To do this, we trim each 10 useful seconds of the videos and classify them as robbery or non-robbery; by doing so, we can get multiple scenes of both classes from a single video, which strengthen our model, since it is less prone to over-fitting. We also re-sized all the videos to 256×256 pixels and standardized the

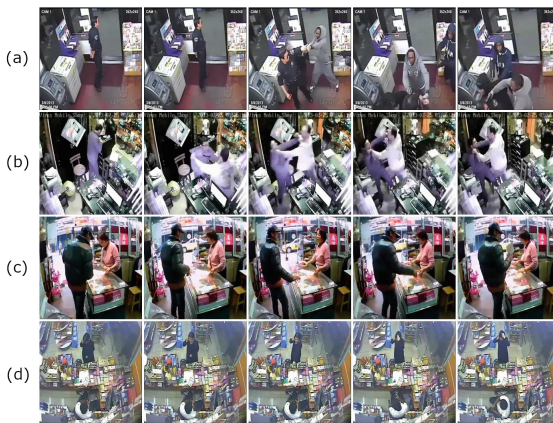


Fig. 1. Sample frames from the UNI-Crime dataset. (a) (b) Robbery samples. (c) (d) Non-robbery samples.

frame rate to 3 frames per second; that is, 30 frames per video, in order to avoid redundant information. In addition, we downloaded extra videos from Youtube, mainly from robberies or normal actions at stores. In the end, we collected 1421 videos: 1001 of non-robbery and 420 of robbery. A sample of frames from the dataset is shown in Fig. 1.

2.2 Neural Network Training

We propose a sequence model for end-to-end robbery detection. The architecture of the model is divided in two parts: the feature extractor, and the recurrent network. We tried the pre-trained VGG16 [17] and NASNetMobile [18] networks as feature extractors, and the Convolutional LSTM network (convLSTM) [19] as the recurrent network. We give further details about these structures below:

ConvLSTM. The Long-Short Term Memory (LSTM) network [20] is a recurrent end-to-end architecture which is capable of efficiently encoding long and short temporal changes of a sequence thanks to the memory cells it presents. Recently, LSTM networks have been widely used for different purposes such as speech recognition, natural language processing (NLP) or even human action recognition [21]. However, a standard LSTM network discards most of the spatial information due to it vectorizes all incoming data. Therefore, the Convolutional Long-Short Term Memory (convLSTM) network [19] was proposed in order to preserve both the spatial and temporal information. The equations that govern a single LSTM unit are as follows:

$$\Gamma_f^{<t>} = \sigma(W_{fx} * x^{<t>} + W_{fa} * a^{<t-1>} + b_f), \quad (1)$$

$$\Gamma_u^{<t>} = \sigma(W_{ux} * x^{<t>} + W_{ua} * a^{<t-1>} + b_u), \quad (2)$$

$$\tilde{c}^{<t>} = \tanh(W_{cx} * x^{<t>} + W_{ca} * a^{<t-1>} + b_c), \quad (3)$$

$$c^{<t>} = \Gamma_f^{<t>} \circ c^{<t-1>} + \Gamma_u^{<t>} \circ \tilde{c}^{<t>}, \quad (4)$$

$$\Gamma_o^{<t>} = \sigma(W_{ox} * x^{<t>} + W_{oa} * a^{<t-1>} + b_o), \quad (5)$$

$$a^{<t>} = \Gamma_o^{<t>} \circ \tanh(c^{<t>}), \quad (6)$$

where $x^{<t>}$ is the two-dimensional input at time t ; $\Gamma_f^{<t>}$, $\Gamma_u^{<t>}$, $\Gamma_o^{<t>}$ are the outputs of the forget, update and output gates, respectively; $c^{<t>}$ is the cell state; $\tilde{c}^{<t>}$ is the candidate for replacing the previous cell state; $a^{<t>}$ is the hidden state; and W_{fx} , W_{fa} , W_{ux} , W_{ua} , W_{ox} , and W_{oa} are two-dimensional convolutional filters.

VGG16. Also called OxfordNet, is a convolutional neural network that is trained on ImageNet [22]. Table 1 shows the performance of the model on the ImageNet validation dataset. The network is 16 layers deep and it exclusively uses 3×3 convolutional filters. It is commonly used as a high-level feature extractor for tasks such as saliency detection [23], action recognition [24, 25] or semantic segmentation [26]. Hence, we consider the convolutional features from the last pooling layer.

NASNet. It is a product of Google’s AutoML project, whose aim is to automate the design of machine learning models. In [18] they proposed a novel search method so that AutoML could find the best layer or combination of layers (like those present in ResNet [27] or Inception [28, 29] models) which can then be stacked many times in a flexible manner to create a final network. One of the networks derived from the large NASNet, trained on the ImageNet and COCO datasets, is called NASNetMobile, which achieved better performance than equivalently-sized state-of-the-art models for mobile platforms [30, 31].

Table 1. Models performance on the Imagenet validation dataset.

Network	Top-1 accuracy (%)	Top-5 accuracy (%)	Parameters
VGG16	0.713	0.901	138,357,544
NASNetMobile	0.744	0.919	5,326,716

Proposed Architecture. Figure 2 illustrates the general architecture of the network for detecting violent robberies. Its input is a sequence of 30 frames of $224 \times 224 \times 3$ pixels. Contrary to previous works such as [8] or [11], we do not use optical flow images or the difference between adjacent frames as inputs because of two reasons: First, some armed robberies does not necessarily involve rapid changes in the scene; instead, a robber could threaten a person holding a gun without sudden movements. Second, we want to keep as much spatial information as we can and let the convLSTM to encode the spatial changes.

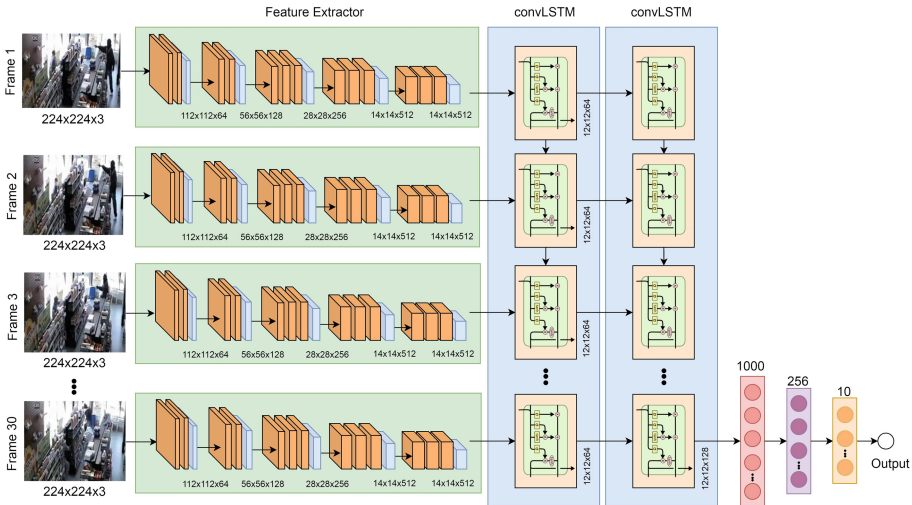


Fig. 2. The proposed network architecture using two convLSTM layers and the first 13 layers of the VGG16 network as feature extractor.

Each frame passes through a convolutional feature extractor, which could be derived from a pre-trained VGG16 or NASNet network. These high-level features are processed using the convLSTM layers. The last convLSTM layer is a many-to-one layer. Finally, the hidden state of the last convLSTM unit is processed by three fully-connected layers in order to obtain a single classification result.

We tried different network architectures but show in Table 2 only the five architectures that achieved the greatest performances. The first one, *ROBNet1*, uses the first 13 layers of the pre-trained VGG16 network as feature extractor and two LSTM layers (as shown in Fig. 2). The second one, *ROBNet2*, uses the first 253 layers of the pre-trained NASNetMobile network (i.e. until the *activation_74* layer whose output is $28 \times 28 \times 88$) as feature extractor and one LSTM layer. The third one, *ROBNet3*, uses the same feature extractor as *ROBNet2* and two LSTM layers. The fourth one, *ROBNet4*, uses the first 769 layers of the pre-trained NASNetMobile network (i.e. until the last convolutional layer, *activation_188*, whose output is $7 \times 7 \times 1056$) as feature extractor and one LSTM layer. The fifth one, *ROBNet5*, uses the same feature extractor as *ROBNet4* and two LSTM layers.

Table 2. Parameters of the proposed network architectures.

Network	Feature extractor	#convLSTM	#Filters	#Filters
		Layers	Layer 1	Layer 2
ROBNet1	VGG16 (13 layers)	2	128	64
ROBNet2	NASNet (253 layers)	1	128	-
ROBNet3	NASNet (253 layers)	2	128	64
ROBNet4	NASNet (769 layers)	1	128	-
ROBNet5	NASNet (769 layers)	2	128	64

3 Results

Since the UNI-Crime dataset contains 256×256 - pixel videos and the input size of both the VGG16 and NASNetMobile networks is 224×224 , we applied random cropping four times to each video; two of the cropped videos were horizontally flipped. In this way, we applied data augmentation techniques and got a dataset of 5684 videos. We divided 85% of the dataset to create the training set, and 15% to create the validation set.

The training algorithm was implemented using Python 3.6 on a PC with Intel i7-8700 at 3.7 GHz CPU, 64 GB RAM and a NVIDIA GeForce GTX 1080 Ti GPU. The proposed CNN was trained during 120 epochs using an Adam optimizer [32] with a learning rate of 0.001, a momentum term β_1 of 0.9, a momentum term β_2 of 0.999 and a mini-batch size of 8. Furthermore, we added a 10% dropout rate before the fully-connected layers to prevent overfitting. Figure 3 shows the evolution of network accuracy and loss over training time of all the

Table 3. Metrics comparison of different robbery detection networks

Network	ACC (%)	Parameters
ROBNet1	96.698	20,906,095
ROBNet2	95.401	101,609,199
ROBNet3	94.103	88,025,199
ROBNet4	93.514	11,989,743
ROBNet5	92.453	6,927,471

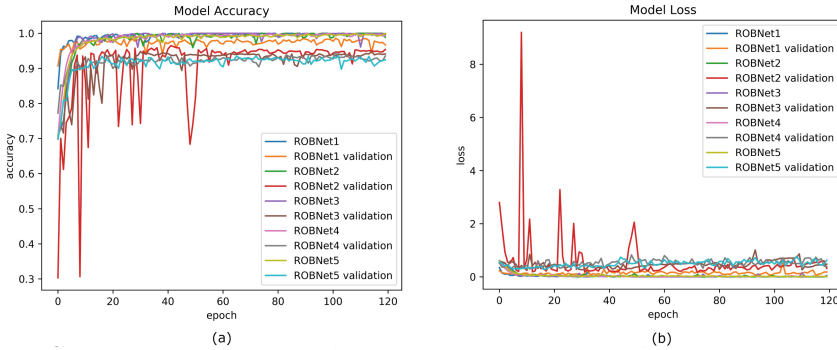


Fig. 3. Comparison of metrics evolution over training time of all networks. (a) Epochs vs. Accuracy. (b) Epochs vs. Loss.

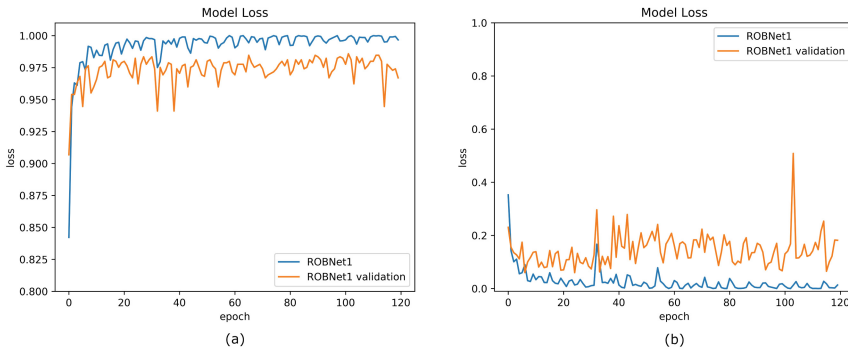


Fig. 4. Metrics evolution over training time of ROBNet1. (a) Epochs vs. Accuracy. (b) Epochs vs. Loss.

networks. Table 3 compares the performance of the five networks in terms of classification accuracy from the validation set.

From Table 4 we chose *ROBNet1* as the best network, as it presented the highest classification accuracy value and the lowest cost when evaluating in the validation set (Fig. 4). What is more, *ROBNet1* is nearly 1.3% more accu-

rate when compared to the second best accuracy and it presents 80,703,104 less parameters. Furthermore, we observe in Fig. 3 that only *ROBNet1* shows a little difference between the training and validation values over the training time, meaning that it prevents overfitting problems and has better performance than the other networks when it comes to predicting new samples outside the training set.

Finally, in Table 4 we compared our results with those achieved using the method of [11] with the UNI-Crime dataset. Although their training accuracy is slightly higher than ours, our validation accuracy is higher than theirs by more than five percentage points. This fact supports our preference of using RGB inputs instead of optical flow inputs.

Table 4. Metrics comparison of different robbery detection networks

Network	Training ACC (%)	Validation ACC (%)
ROBNet1	99.46	96.69
Sudhakaran [11]	99.63	91.25

4 Conclusion

In this paper, we have presented an efficient end-to-end trainable deep neural network to tackle the problem of violent robbery detection in CCTV videos.

We presented a dataset that encompasses both normal and robbery scenarios. What is more, some of the normal videos contained in our dataset correspond to moments before or after the robbery, which ensures that our method can discern between normal and robbery events even in the same environment.

The proposed method is able to encode both the spatial and temporal changes using convolutional LSTM layers that receive a sequence of features extracted by a pre-trained CNN from the original video frames. The classification accuracy evaluated in the validation dataset achieves a value of 96.69%. Although this is a promising result, further improvements need to be done in order to offer a scalable and marketable product, such as severely increasing the number of videos and scenarios of the dataset, reducing the computational cost, among others.

References

1. The Global Shapers Survey. <http://shaperssurvey2017.org/>. Accessed 4 Feb 2019
2. Mabrouk, A.B., Zagrouba, E.: Spatio-temporal feature using optical flow based distribution for violence detection. *Pattern Recognit. Lett.* **92**, 62–67 (2017). <https://doi.org/10.1016/j.patrec.2017.04.015>
3. Laptev, I.: On space-time interest points. *Int. J. Comput. Vis.* **64**(2–3), 107–123 (2005). <https://doi.org/10.1007/s11263-005-1838-7>

4. Deniz, O., Serrano, I., Bueno, G., Kim, T.K.: Fast violence detection in video. In: 2014 International Conference on Computer Vision Theory and Applications (VISAPP), pp. 478–485. IEEE, Lisbon (2004)
5. Tay, N.C., Connie, T., Ong, T.S., Goh, K.O.M., Teh, P.S.: A robust abnormal behavior detection method using convolutional neural network. *Computational Science and Technology*. LNEE, vol. 481, pp. 37–47. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-2622-6_4
6. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4305–4314 (2015). <https://doi.org/10.1109/CVPR.2015.7299059>
7. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 3551–3558. IEEE, Sydney (2013). <https://doi.org/10.1109/ICCV.2013.441>
8. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 568–576. MIT Press, Montreal (2014)
9. Meng, Z., Yuan, J., Li, Z.: Trajectory-pooled deep convolutional networks for violence detection in videos. In: Liu, M., Chen, H., Vincze, M. (eds.) *ICVS 2017*. LNCS, vol. 10528, pp. 437–447. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68345-4_39
10. Zhou, P., Ding, Q., Luo, H., Hou, X.: Violent interaction detection in video based on deep learning. *J. Phys. Conf. Ser.* **844**, 012044 (2017)
11. Sudhakaran, S., Lanz, O.: Learning to detect violent videos using convolutional long short-term memory. In: 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, pp. 1–6 (2017). <https://doi.org/10.1109/AVSS.2017.8078468>
12. Hassner, T., Itcher, I., Kliper-Gross, O.: Violent flows: real-time detection of violent crowd behavior. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, Providence (2012). <https://doi.org/10.1109/CVPRW.2012.6239348>
13. Bermejo Nievas, E., Deniz Suarez, O., Bueno García, G., Sukthankar, R.: Violence detection in video using computer vision techniques. In: Real, P., Diaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W. (eds.) *CAIP 2011*. LNCS, vol. 6855, pp. 332–339. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23678-5_39
14. Li, W., Mahadevan, V., Vasconcelos, N.: Anomaly detection and localization in crowded scenes. *IEEE Trans. Pattern. Anal. Mach. Intell.* **36**(1), 18–32 (2014)
15. Sultani, W., Chen, C., Shah, M.: Real-world anomaly detection in surveillance videos. [arXiv:1801.04264](https://arxiv.org/abs/1801.04264) (2018)
16. UNI-Crime Dataset. <http://didt.inictel-uni.edu.pe/dataset/UNI-Crime.Dataset.rar>. Accessed 25 Jan 2019
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
18. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8697–8710. IEEE, Salt Lake City (2018)

19. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.C.: Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Cortes, C., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1 (NIPS 2015), vol. 1, pp. 802–810. MIT Press, Cambridge (2015)
20. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997). <https://doi.org/10.1162/neco.1997.9.8.1735>
21. Salehinejad, H., Sankar, S., Barfett, J., Colak, E., Valaee, S.: Recent advances in recurrent neural networks. [arXiv:1801.01078](https://arxiv.org/abs/1801.01078) (2018)
22. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Miami (2009). <https://doi.org/10.1109/CVPR.2009.5206848>
23. Lee, G., Tai, Y., Kim, J.: Deep saliency with encoded low level distance map and high level features. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 660–668. IEEE, Las Vegas (2016)
24. Lan, Z., Zhu, Y., Hauptmann, A.G., Newsam, S.: Deep local video feature for action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE, Honolulu (2017)
25. Li, Z., Gavriluyk, K., Gavves, E., Jain, M., Snoek, C.G.: Video LSTM convolves, attends and flows for action recognition. *Comput. Vis. Image Underst.* **166**, 41–50 (2018). <https://doi.org/10.1016/j.cviu.2017.10.011>
26. Liu, T., Stathaki, T.: Faster R-CNN for robust pedestrian detection using semantic segmentation network. *Front. Neurorobot* **12**, 64 (2018). <https://doi.org/10.3389/fnbot.2018.00064>
27. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. IEEE, Las Vegas (2016). <https://doi.org/10.1109/CVPR.2016.90>
28. Szegedy, C., et al.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9. IEEE, Boston (2015). <https://doi.org/10.1109/CVPR.2015.7298594>
29. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inceptionv4, Inception-Resnet and the impact of residual connections on learning. In: AAAI Conference on Artificial Intelligence, San Francisco (2017)
30. Howard, A.G., et al.: Mobilenets: efficient convolutional neural networks for mobile vision applications. [arXiv:1704.04861](https://arxiv.org/abs/1704.04861) (2017)
31. Zhang, X., Zhou, X., Mengxiao, L., Sun, J.: Shufflenet: an extremely efficient convolutional neural network for mobile devices. [arXiv:1707.01083](https://arxiv.org/abs/1707.01083) (2017)
32. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations (ICLR 2015), San Diego (2015)