Human Action Recognition in Still Images

Palak(✉) and Sachin Chaudhary

Punjab Engineering College, Chandigarh, India
palakgirdhar99@gmail.com, sachin.chaudhary@pec.edu.in

Abstract. In this research work, we are addressing the problem of action recognition in still images, in which the model focuses on recognizing the person's action from a single image. We are using the dataset published by V. Jacquot, Z. Ying, and G. Kreiman in CVPR 2020. The dataset consists of the 3 action classes: Drinking, Reading, and Sitting. The images are not classified into these 3 classes. Instead, binary image classification is used on each class i.e., whether the person is performing that particular action or not. To classify the images, we started with the Detectron2 Object detection model for detecting the person performing the activity and the object related to it (foreground) and then we remove everything else (background) from the image. And then, these images without the background are used for the classification task. The classification is done by using various deep learning models with the help of transfer learning. And as a result, the classification accuracy of HAR in still images increases by 10% on VGG16, 7% on InceptionV3, 1% on Xception, and 4% on the Inception-Resnet model.

Keywords: Action recognition · Still images · Detectron2 · Binary image classification · Background removal

1 Introduction

1.1 Human Action Recognition

Human Action Recognition (HAR) is an important problem in the field of computer vision research. The main goal of an action recognition model is to analyze the image/video for identifying the action in the provided data. HAR [7, 15, 37, 40] is used in a variety of applications which includes video storage and retrieval, video surveillance systems [34, 36], human–machine interface, healthcare, image annotation, identity recognition system, elderly patient monitoring system, and a lot more. Although, HAR is used in many applications, it is still a challenging problem [12] in computer vision due to less accuracy and efficiency. Humans can easily understand the action going on in the given image/video through their senses (eyes in this case). To monitor human actions in some real-world situations, a lot of manpower is required which is quite expensive. Thus, we require a machine that can identify the actions with good accuracy. Earlier, hand-crafted approaches were used for recognizing the actions, and these days we are using learning-based approaches. In this research work, we explored the field of HAR in still images by using deep learning-based approaches.

B. Raman et al. (Eds.): CVIP 2021, CCIS 1568, pp. 483–493, 2022.
https://doi.org/10.1007/978-3-031-11349-9_42

1.2 Human Action Recognition in Still Images

Action Recognition in still images has recently become an active research topic in the field of computer vision. In this technique, we try to identify the action performed by the person through a single image. As the image does not contain any temporal information, it makes this problem of HAR in images more difficult than the HAR in videos. There is a huge amount of still images present over the internet. Therefore, it is important to develop an efficient model for recognizing action in still images which will help in better understanding and retrieval of the images. Although, a lot of work has been done in HAR based on videos, HAR in still images is not explored enough. Due to increase in the usage of digital cameras in everyday life, more and more image content is generated and uploaded to the Internet or stored in a large image dataset. Categorizing rich image content based on the actions appearing in the image is a good way to reach the initial goal of organizing these images. Images are used for performing some other important tasks as well like image inpainting [42], image dehazing [44], single image depth estimation [35] etc. Fig. 1 shows some of the activities which can be recognized from a single image only.

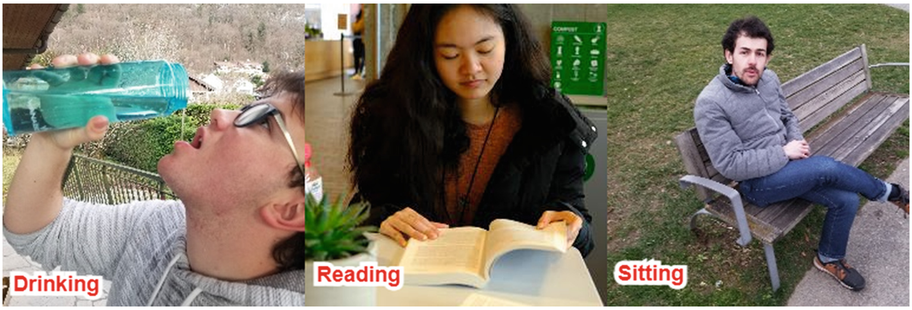

Fig. 1. Activities that can be recognized by single images

According to the survey in [1], the state-of-the-art methods available for recognizing actions in the images are categorized based on the low-level and high-level features available in the images.

Low-Level Features. These are basically the small details of the image like corners or edges. Some of the popular techniques for low-level feature extraction are Dense Sampling of Invariant Feature Transform (DSIFT), Histogram of Oriented Gradient (HOG), Shape Context (SC), and Global Image Descriptor (GIST).

High-Level Features. These features are the complete objects which can be detected by the different object detection models for recognizing the actions in the still images. The various high-level features are: The Human Body, Body Parts, Objects, Human-Object Interaction, Context or Scene. These features are extracted by using machine learning techniques.

2 Related Work

Previous literature on human action recognition in still images comprises the methods of extracting features from the images. The dense sampling of the grayscale images is used to extract low-level features for action analysis, using the Scale Invariant Feature Transform (SIFT) method [2]. DSIFT based feature is used to recognize action classes as discussed in the methods in [3–6]. Still image-based action recognition in [8–11] uses HOG feature proposed by [38]. Approaches in [10, 13, 14] use the shape context feature proposed by [41] to extract the shape features for object matching, to detect and segment the human contour. GIST technique [39] is used for extracting the spatial properties of the scene as an abstract representation of the scene. This feature is used to integrate background or scene information. GIST has been used in [10, 16, 17] for recognizing actions in still images.

High-level features like the human body, body parts, objects, etc. can be extracted using the Object Detection [46] models. These models draw boundaries along the object which helps us in identifying the object. Some of the popular object detection models are R-CNN [29], Fast R-CNN [28], Faster R-CNN [27], YOLO [19], and SSD [30]. R-CNN algorithm [29] puts some boxes randomly in the image and then checks for the objects in those boxes. After choosing the region of the object from the image, the model extracts the features of the object and then combines the regions. Training is a multi-step process and therefore this method is very slow and needs almost 1 min for an image. Fast R-CNN [28] puts a huge number of frames in the image, which results in the duplication of the features extracted by the model. Training speed is increased. Space needed for training is increased. This algorithm puts no constraints on the input data. Better than the R-CNN but still has issues. In Faster R-CNN [27], Deep CNN is used to determine candidate regions. RPN gives the output of the regions with a detection score. This is the fastest of all of its versions and can perform at the rate of 7 images/second. YOLO [19] is different from all the models discussed above. The networks for training and testing are completely different from each other. In this, a single neural network is applied to the complete image which is then divided into different regions, and a bounding box is created among each region containing objects. In SSD [30], a complete image is needed for the input, with the description of the bounding boxes. The default boxes are compared to the ground truth while training the model. It generates a sequence of same-sized boxes and scores for them. It is a newer version of YOLO [19]. Speed is 58 frames/sec with 72% accuracy.

3 Proposed Approach

3.1 Dataset Used

The dataset that we are using in our research work is published by V. Jacquot, Z. Ying, and G. Kreiman in [18]. The dataset consists of 3 action classes: Drinking, Reading, and Sitting where Drinking is defined as liquid in mouth, reading as gaze towards text and sitting as buttocks on support. Every action class is further divided into yes and no category and each category contain images for Training, Testing and Validation. The dataset contains a total of 6804 images, out of which Drinking has 2164 images, Reading has 2524 images and sitting has 2116 images. The complete dataset details are shown in the Table 1.

The images in the dataset are gathered from two different sources:

i) Clicked by the investigators in their lab.
ii) From the internet.

Table 1. Dataset details

	Training	Testing	Validation	Total
Drinking	862	110	110	1082
Not-Drinking	862	110	110	1082
Reading	1002	130	130	1262
Not-Reading	1002	130	130	1262
Sitting	826	110	122	1058
Not-Sitting	826	110	122	1058
				6804

3.2 Methodology

The major tasks associated with our work includes: object detection, background removal (removing noise), Fine-tuning the pre-trained model, Feature extraction and classification. We worked on the drinking dataset and we are interested in classifying whether the person in the image is drinking or not where drinking is defined as the liquid in the mouth.

Our proposed method is divided into 2 modules: Training and Testing. So, we will be treating both training and testing data differently. Training data is used in the training module as shown in Fig. 2 and Testing data is used in the testing module as shown in Fig. 3.

Training Module. We used training data of both the categories i.e., drinking and not-drinking. Firstly, we separate the google images from the images clicked by them i.e., we are dividing the training data into 2 parts:

1. Images clicked by the investigators are named Training Data 1.
2. Images downloaded from the Internet are named Training Data 2.

After that, we perform object detection and background removal on Training Data 1, combines it with Training Data 2, and then using the entire training data for fine-tuning the CNN model pre-trained on the ImageNet Dataset [21]. This whole process is represented in Fig. 2.

Object Detection Using Detectron2. Object Detection is done with the help of the Detectron2 [25] model which is developed by Facebook AI Research. It is the replacement for Detectron and Mask R-CNN [26] benchmark. Detectron2 [25] includes models like RetinaNet [32], DensePose [33], Mask R-CNN [26], Faster R-CNN [27], TensorMask [43], Panoptic FPN [45] and Cascade R-CNN [47]. Detectron2 [25] can perform a variety of tasks like detecting objects using a bounding box, estimating the pose of the humans present in the image, it can also perform panoptic segmentation which is a combination of semantic and instance segmentation. Detectron2 [25] can be used with the help of PyTorch which was a major drawback for the Detectron model. We have detected the human present in the image along with the object it is related to. Image masks are created by Detectron2 [25], these image masks are used to manipulate the image. The output from the object detection model (Detectron2) consists of these three things:

i) predicted class
ii) bounding box coordinates
iii) mask arrays

We used mask outlining and class ID for selecting the correct mask when there are multiple objects present of same class.

Background Removal. After detecting the relevant objects from the image, we removed the background of the image which is not contributing to the classification task. The motive behind removing the background is that our CNN model will not extract the features which are not contributing to the classification task. For removing the background [31], we generate the binary mask from the mask array that we received from Detectron2 [25] and these pixels with the value = 1 in the binary mask are extracted from the original image and then copied to a new image having blank background.

After background removal [31], all these images (Training Data 1) along with Training Data 2 are used for fine-tuning the CNN models pre-trained on the ImageNet Dataset [21]. We received fine-tuned model as an output of the Training module and this fine-tuned model is then used in Testing module for binary action classification.

Testing Module. We will be using testing data of both the categories i.e., Drinking and Not-Drinking. Firstly, we are dividing the testing data into 2 categories:

1. Images clicked by the investigators are named Testing Data 1.
2. Images downloaded from the Internet are named Testing Data 2.

Then, we perform object detection and background removal on Testing Data 1 by using the same method explained in the Training module used for Training Data. Then feeds Testing Data 1 and Testing Data 2 to the fine-tuned model, which we have received as an output of the training module. This fine-tuned CNN model is used to extract features from the images and then classify them as Drinking or Not Drinking. The block diagram of the Testing module is shown below in Fig. 3.

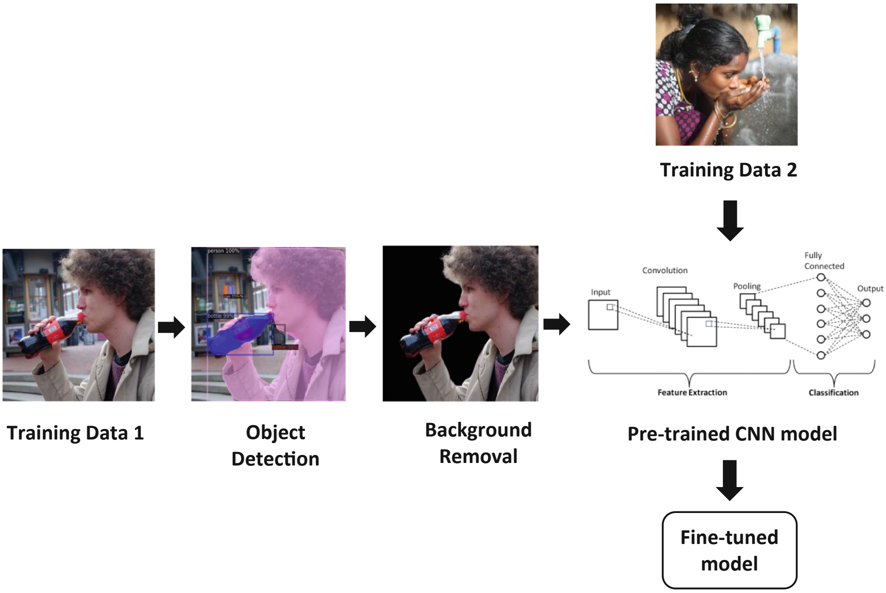

Fig. 2. Training module

We used transfer learning in our research because the dataset we are using is not large enough to train the model. We have also used data augmentation on our dataset before fine-tuning our model to prevent overfitting. We have done all of our work on the GPU.

To evaluate our proposed method, we have used 4 different CNN models pre-trained on the ImageNet Dataset [21]:

VGG16 [20]. It is also known as OxfordNet. It is named after the name of the group (Visual Geometry Group) who developed it from Oxford. This neural network is 16 layers deep. The model uses a set of weights that have been pre-trained on ImageNet [21]. The ImageNet dataset [21] consists of over 14 million images divided into 1000 classes. The top-5 accuracy of the model is 92.7%. The input to the VGG16 model is an RGB image of the size 224 × 224. The image is then processed through a stack of convolutional layers having the filters of size 3 × 3 with stride = 1, max pool layer with stride = 2, and a filter of size 2 × 2.

Inception V3 [23]. It is an Image Recognition model. It consists of symmetric and asymmetric building blocks which include convolution layers, pooling layers (average pooling and max pooling), concats, dropouts, and the fully connected layers. Batch normalization is used in the model. This model has shown an accuracy $> 78\%$ on the ImageNet Dataset [21]. As the deep networks are more likely to be overfitted, in this model, rather than going deep into the network, they have widened the network on different levels. So, there is more than 1 filter present on some levels of this model. So, several filters are applied parallelly on the same level along with the pooling operation. Then, all the outputs from a single level are concatenated and sent to the next level.

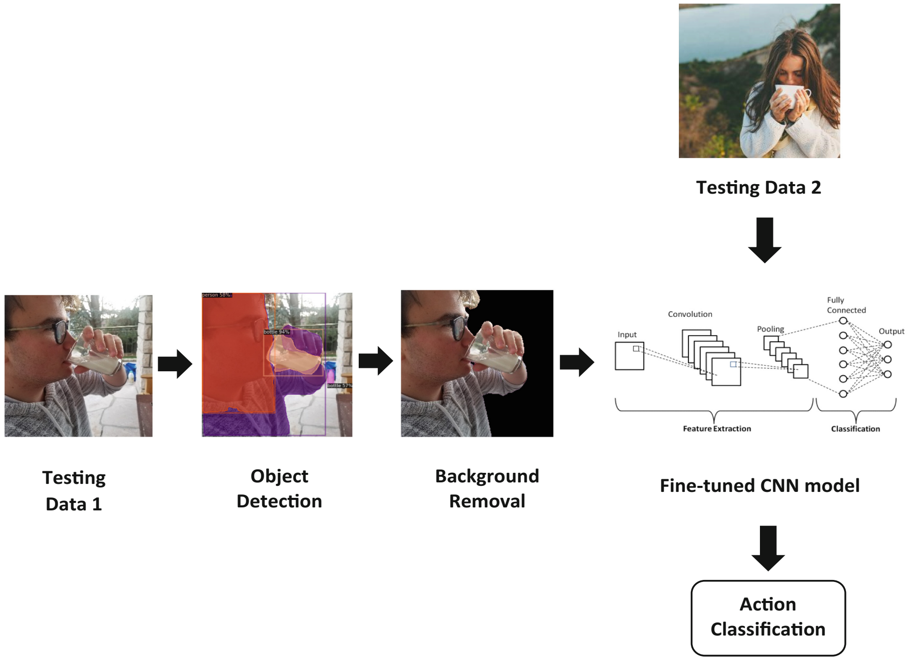

Fig. 3 Testing module

Xception [24]. It is 71 layers deep CNN. This neural network is based on the depth-wise separation of the convolution layers. It stands for Extreme Inception. The concepts of the Inception model are used to an extreme in this model. The depth-wise convolution is followed by pointwise convolution in this model. Its results are somewhat better than the inception v3 model on the ImageNet data [21] but their number of parameters are the same. Thus, this model is efficient as compared to Inception.

Inception-ResNet [22]. This model is a combination of the Inception and the ResNet model because both the model's performances individually are quite appreciable. This model has two versions, the computational cost of the first model is similar to the InceptionV3 [23] and the second's cost is similar to the InceptionV4. Both have different stems. The structure is the same for all the modules and the reduction block. Hyperparameters are different for both. These models can achieve higher accuracies even with the fewer number of iterations.

4 Results

The accuracies of different state-of-the-art models by using our proposed method are calculated and shown in Fig. 4. The accuracies achieved on VGG16 [20], InceptionV3 [23], Xception [24] and Inception-ResNet V2 [22] are 59.09%, 65.91%, 62.75% and 63.18% respectively. These accuracies are then compared with the accuracies of the same dataset on the same deep learning models without using our method and the comparison is shown in Fig. 4.

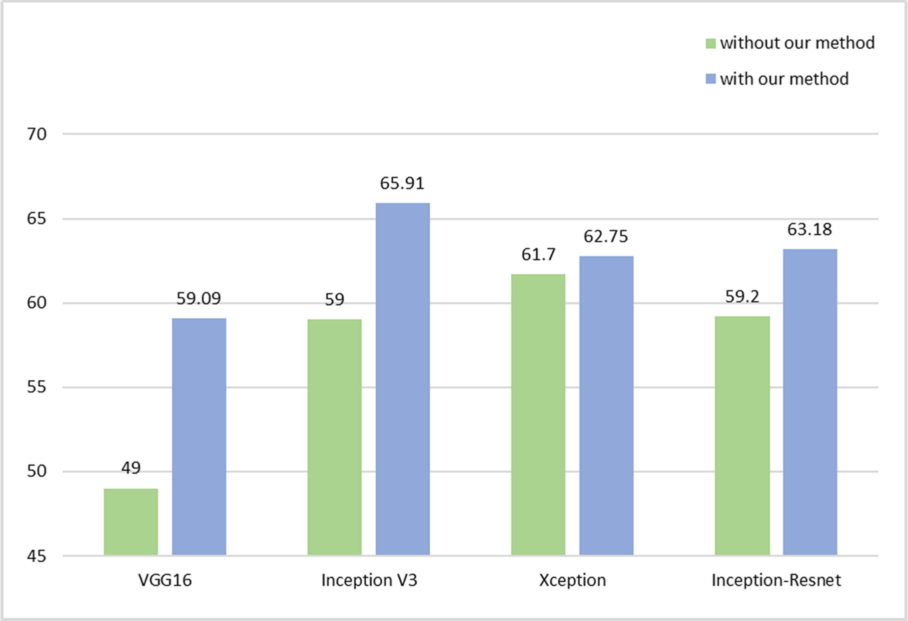

Fig. 4. Comparison of the classification accuracies of the models with and without using our proposed method on the drinking dataset

5 Conclusion and Future Work

In this work, we first analyzed all previous work done on HAR in still images and came across a dataset for basic human activities: Drinking, Reading, and Sitting. The dataset contains fewer biases and thus the state-of-the-art methods are showing less accuracy on this dataset.

So, we proposed a classification method, in which we first perform object detection on the images and then we remove the background of images. After that, we used the same state-of-the-art methods to classify those images. Our method performs well and the accuracy is increased by 10% on VGG16 model, 7% on InceptionV3, 1% on Xception and 4% on Inception-Resnet V2 model.

In the future, this model can be applied for action classification tasks having more than 2 action classes to check if it performs better there as well or not. This model can also be used for binary action classification like in medical sciences, where we need to determine whether the person is suffering from some disease or not.

References

1. Guo, G., Lai, A.: A survey on still image-based human action recognition. Pattern Recognit. **47**(10), 3343–3361 (2014)
2. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157. IEEE (1999)
3. Li, L.-J., Li, F.-F.: What, where and who? classifying events by scene and object recognition. In: ICCV, vol. 2, no. 5, p. 6 (2007)
4. Delaitre, V., Laptev, I., Sivic, J.: Recognizing human actions in still images: a study of bag-of-features and part-based representations. In BMVC 2010 (2010)
5. Shapovalova, N., Gong, W., Pedersoli, M., Roca, F.X., Gonzàlez, J.: On importance of interactions and context in human action recognition. In: Vitrià, J., Sanches, J.M., Hernández, M. (eds.) Pattern Recognition and Image Analysis, IbPRIA 2011. LNCS, vol. 6669, pp. 58–66. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21257-4_8
6. Yao, B., Fei-Fei, L.: Grouplet: a structured image representation for recognizing human and object interactions. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 9–16. IEEE (2010)
7. Chaudhary, S., Murala, S.: Depth-based end-to-end deep network for human action recognition. IET Comput. Vis. **13**(1), 15–22 (2019)
8. Desai, C., Ramanan, D.: Detecting actions, poses, and objects with relational phraselets. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) Computer Vision – ECCV 2012. LNCS, vol. 7575, pp. 158–172. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33765-9_12
9. Thurau, C., Hlavác, V.: Pose primitive based human action recognition in videos or still images. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2008)
10. Gupta, A., Kembhavi, A., Davis, L.S.: Observing human-object interactions: using spatial and functional compatibility for recognition. IEEE Trans. Pattern Anal. Mach. Intell. **31**(10), 1775–1789 (2009)
11. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for static human-object interactions. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, pp. 9–16. IEEE (2010)
12. Chaudhary, S.: Deep learning approaches to tackle the challenges of human action recognition in videos. Dissertation (2019)
13. Wang, Y., Jiang, H., Drew, M.S., Li, Z.-N., Mori, G.: Unsupervised discovery of action classes. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2006), vol. 2, pp. 1654–1661. IEEE (2006)
14. Yao, B., Fei-Fei, L.: Modeling mutual context of object and human pose in human-object interaction activities. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition, pp. 17–24 (2010)
15. Chaudhary, S., Murala, S.: TSNet: deep network for human action recognition in hazy videos. In: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 3981–3986 (2018). https://doi.org/10.1109/SMC.2018.00675
16. Prest, A., Schmid, C., Ferrari, V.: Weakly supervised learning of interactions between humans and objects. IEEE Trans. Pattern Anal. Mach. Intell. **34**(3), 601–614 (2011)
17. Li, P., Ma, J.: What is happening in a still picture? In: The First Asian Conference on Pattern Recognition, pp. 32–36. IEEE (2011)
18. Jacquot, V., Ying, Z., Kreiman, G.: Can deep learning recognize subtle human activities? In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, pp. 14232–14241 (2020). https://doi.org/10.1109/CVPR42600.2020.01425

19. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788 (2016). https://doi.org/10.1109/CVPR.2016.91
20. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition (2015)
21. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV), **115**(3), 211–252 (2015)
22. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning (2016)
23. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. CoRR, abs/1512.00567 (2015)
24. Chollet, F.: Xception: deep learning with depthwise separable convolutions. CoRR abs/1610.02357 (2016)
25. Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., He, K.: Detectron (2018). https://github.com/facebookresearch/detectron
26. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988 (2017). https://doi.org/10.1109/ICCV.2017.322
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017). https://doi.org/10.1109/TPAMI.2016.2577031.
28. Girshick, R.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448 (2015). https://doi.org/10.1109/ICCV.2015.169
29. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
30. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) Computer Vision – ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
31. Reaper, T.: Automated image background removal with python. tobias.fyi (2020). https://tobias.fyi/blog/remove-bg-python
32. Lin, T., Goyal, P., Girshick, R., He, K., Dollár, P: Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007 (2017). https://doi.org/10.1109/ICCV.2017.324
33. Güler, R.A., Neverova, N., Kokkinos, I.: DensePose: dense human pose estimation in the wild. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7297–7306 (2018). https://doi.org/10.1109/CVPR.2018.00762
34. Patil, P.W., Dudhane, A., Kulkarni, A., Murala, S., Gonde, A.B., Gupta, S.: An unified recurrent video object segmentation framework for various surveillance environments. IEEE Trans. Image Process. **30**, 7889–7902 (2021)
35. Praful, H., Dudhane, A., Murala, S.: Single image depth estimation using deep adversarial training. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 989–993. IEEE I(2019)
36. Patil, P.W., Dudhane, A., Chaudhary, S., Murala, S.: Multi-frame based adversarial learning approach for video surveillance. Pattern Recogn. **122**, 108350 (2022)
37. Chaudhary, S., Murala, S.: Deep network for human action recognition using Weber motion. Neurocomputing **367**, 207–216 (2019)
38. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition CVPR, vol. 1, pp. 886–893 (2005)
39. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput. Vis. **42**(3), 145–175 (2001)

40. Chaudhary, S., Dudhane, A., Patil, P., Murala, S.: Pose guided dynamic image network for human action recognition in person centric videos. In: 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 1–8 (2019). https://doi.org/10.1109/AVSS.2019.8909835
41. Belongie, S., Mori, G., Malik, J.: Matching with shape contexts. In: Krim, H., Yezzi, A. (eds) Statistics and Analysis of Shapes. Modeling and Simulation in Science, Engineering and Technology, pp. 81–105. Birkhäuser Boston, Boston (2006). https://doi.org/10.1007/0-8176-4481-4_4
42. Phutke, S.S., Murala, S.: Diverse receptive field based adversarial concurrent encoder network for image inpainting. IEEE Signal Process. Lett. **28**, 1873–1877 (2021)
43. Chen, X., Girshick, R., He, K., Dollár, P.: Tensormask: a foundation for dense object segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2061–2069 (2019)
44. Akshay, D., Biradar, K.M., Patil, P.W., Hambarde, P., Murala, S.: Varicolored image de-hazing. In: proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4564–4573 (2020)
45. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6399–6408 (2019)
46. Patil, P.W., Biradar, K.M., Dudhane, A., Murala, S.: An end-to-end edge aggregation network for moving object segmentation. In: proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8149–8158 (2020)
47. Cai, Z., Vasconcelos, N.: Cascade R-CNN: high quality object detection and instance segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **43**, 1483–1498 (2019).