 doi:10.20944/preprints202009.0088.v1

YOLOv2 for Pigs Detection in Industrial Farming

Salman Khan and Akif Quddus Khan
akifqk@stud.ntnu.no; salmankh@stud.ntnu.no

Norwegian University of Science and Technology, Gjøvik 2815, Norway.

Abstract. Generic object detection is one of the most important and flourishing branches of computer vision and has real-life applications in our day to day life. With the exponential development of deep learning-based techniques for object detection, the performance has enhanced considerably over the last 2 decades. However, due to the data-hungry nature of deep models, they don't perform well on tasks which have very limited labeled dataset available. To handle this problem, we proposed a transfer learning-based deep learning approach for detecting multiple pigs in the indoor farm setting. The approach is based on YOLO-v2 and the initial parameters are used as the optimal starting values for train-ing the network. Compared to the original YOLO-v2, we transformed the detector to detect only one class of objects i.e. pigs and the back-ground. For training the network, the farm-specific data is annotated with the bounding boxes enclosing pigs in the top view. Experiments are performed on a different configuration of the pen in the farm and con-vincing results have been achieved while using a few hundred annotated frames for fine-tuning the network.

Keywords: YOLOv2, Transfer learning, Pig farming, Object detection.

1 Introduction

Object detection is one of the fundamental tasks in computer vision and form the basis of many high level tasks including but not limited to object tracking [1–9] sports player performance analysis [10–12], crowd analysis [13–16], crowd counting [13, 13, 17, 18], action recognition [19–22], anomaly detection [23–26], detection based facial emotion recognition [27,28], pose estimation [29,30], video scene understanding [31–33]. Technically, object detection is defined as given an image or video, find if there is an instance or instances of the object of interest present. If the object is present, localize its spatial position in the frame/image. In addition to localization, the object of interest is bounded through a geometric shape [34] like rectangle, ellipse, circle, etc [35]. Generally, the object detection community emphasize on detecting a broad range of object categories as compared to specific object category detection where only predefined categories like cats, dogs, faces, airplanes, pedestrians, chair, car, etc [36]. In this regard, many breakthroughs have been achieved and object detectors that could detect up to 2000 object categories have been designed [37].

 doi:10.20944/preprints202009.0088.v1

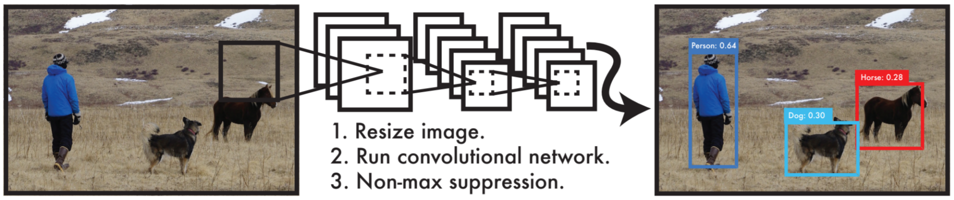

Fig. 1. Figure courtesy [38]: The generic YOLO object detection system. The important steps of YOLO are(1) resizing the input image to 448 × 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence.

Thanks to deep learning-based techniques that have the ability to learn feature representation from the very large volume of data for different object categories. In computer vision, by large, object detection has been mainly used in automatic video surveillance and crowd analysis [39–42] of public places, and the subject of interest was mainly humans. However, with the widespread of AI applications, vision-based surveillance solutions are pervasive in industrial farming. Video-based farm animal monitoring [43] provides an efficient and cost-effective solution [44] to monitor the different activities of the animal and also explore new phenotype for better breeding values and gene selections [45]. Inspired by the success of vision-based solutions in the farming industries, the aim of this paper is to design a pig detector with a minimal amount of labeled data. The detected pigs are represented in a form of bounding boxes. We have followed the approach of transfer learning [46–48] and adopted YOLO [38] model that is one of the best and effective models for generic object detection. The generic YOLO (You Only Look Once) can detect 49 different objects and the localization is represented in the form of the bounding box around the object. However, we transferred the detector into a single class detector and we defined pigs are our predefined class. To get the training data, we created a custom data annotator for labeling the dataset. Initially, we trained the model with three pigs dataset and later with the six pigs dataset. The qualitative results show promising results. in an image and detection was achieved.

2 Related work

In the recent times, researchers are actively proposing novel machine-vision and specially deep learning algorithms for real time object detection. mechanisms to do real time detection. The algorithms specifically focus on detecting the body posture of the objects such as human, animals and vehicles, etc. and then their movements across time are analyzed to deduce results. The authors in [49] made a model which was able to detect objects using SSD [50] and mobile Nets [51]. There model showed great results also in detecting and tracking an object and then also further understanding the behavior of that object in real time.

 doi:10.20944/preprints202009.0088.v1

The work done by [49] potentially addressed object detection in in unconstrained camera environment where it is harder to detect objects which undergoes body deformation. Sometimes background can be a big issue as there can be a lot of factors that have an effect in detecting the object so to solve this issue some work was done by [52]. A lot of data set is required to train an object detection model, but the work done by [53] combining CNN and SIFT made sure that object detecting can be possible with small amount of data set. A lot of resources of computer were used in object detecting models and a lot of resources were needed but with the introduction of new detection algorithms such as MobileNet has made it possible to do detection with low resources using a simple webcam [53]. Compared to these approaches, we followed the standard approach of YOLO [38] and fine-tuned the model on the customized dataset. In the section 3, the proposed approach is briefly explained. The data annotation and the tool used for such task is also introduced in this section. Model architecture is elaborated in 4. Additionally, the training strategy, and the loss function is discussed. A couple of qualitative results are also listed in section 4 which highlights the effectiveness of the approach. Section 5 concludes the paper with final remarks.

3 PROPOSED APPROACH

3.1 Data Annotation

Data annotator used to label the data was a custom made by ourselves. It annotated images which were directly fed to yolo model for training the model. Every image we had the annotator provided a corresponding xml file in which it had all the coordinates of the objects to create the bounding box. A screenshot of the data annotator is show in figure 2. To make a bounding box of an object we just had to draw a box around the object we can be seen in figure 3

4 MODEL ARCHITECTURE

We are using Yolo model because it is an efficient multi scale deep learning model. Yolo outperformed R-CNN model and all its variants. Yolo is the state of the art in object detection. Yolo looks an image only once but in a very efficient way, Yolo is extremely fast as compared to other object detection models. Yolo is straight forward as you can see in figure 4 Yolo takes a different approach as compared to others object detection models. Yolo mostly divides the image into grid of S x S cell. If the center of the object falls into a grid cell that grid cell should detect the object. Each of these cells predict B number of bounding boxes and confidence scores. Bounding box is a square which encloses the object. Yolo also outputs a confidence score that tells how much chances are there that there is an object in the bounding box. The confidence score does not say what is in the box just tells if the shape of the box is little different the others so there

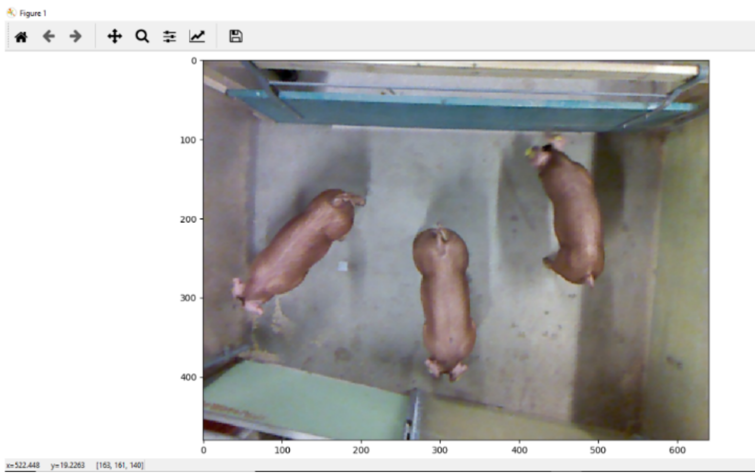

Fig. 2. Depiction of annotation tool

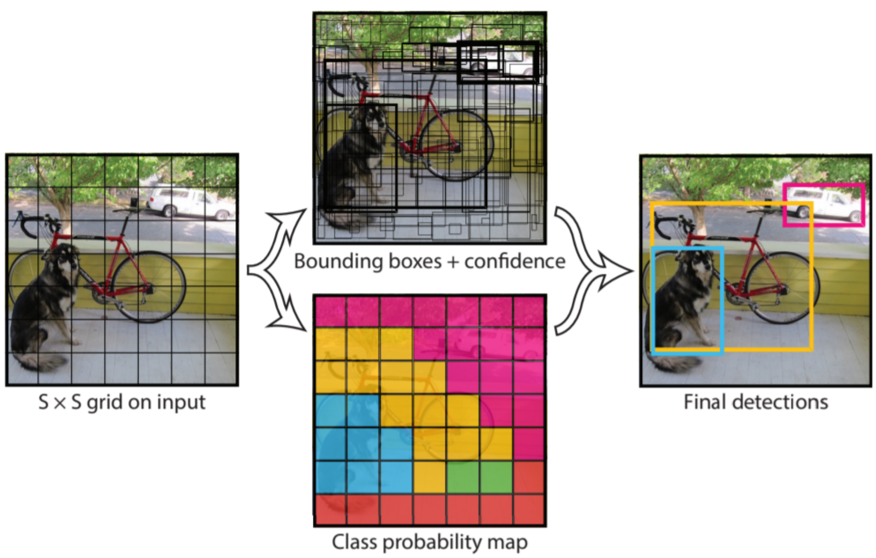

Fig. 3. Figure courtesy [38]: The model perform detection as a regression problem. It divides the image into an S × S grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an S × S × (B × 5 + C) tensor.

might be some object in the bounding box. The formula used for confidence score is PR(object) * IOU truth prediction.

If in in the grid cell there is no object, the confidence score must be equal to zero otherwise the confidence score must be equal to intersection over union between predicted bounding box and ground truth. Each of the bounding box makes 5 parameters x, y which are the coordinates for center of the box in the gird cells. W and H are width and height of the whole image. The last parameter is the confidence score which represents the intersection over union between the predicted bounding box and ground truth. The grid cells also predict conditional class probabilities. These probabilities are higher in the cell where the object appears. The model only predicts one set of class probabilities per grid regardless of how many bounding boxes appear in that grid .

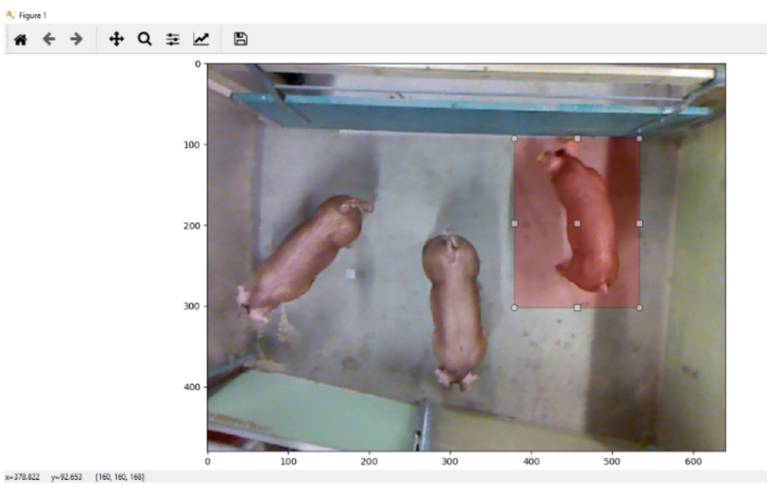

Fig. 4. Depiction of annotation tool

Yolo uses a convolutional neural network. The initial convolutional layers extract the features from the image and the fully connected layers predicts the output probabilities and coordinates. The detection network has twenty-four convolutional layers and two fully connected layers. The full network can be seen in the figure 5 Also, the model uses 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection [38].

Yolo predicts b number bounding box per grid cell but out of all the bounding box we need the best bounding box to describe the object to be selected. So, we choose the bounding box with highest intersection of union with the ground truth. This leads to specialization among the bounding box prediction. Each of the prediction keeps describe the object to be selected. So, we choose the bound-

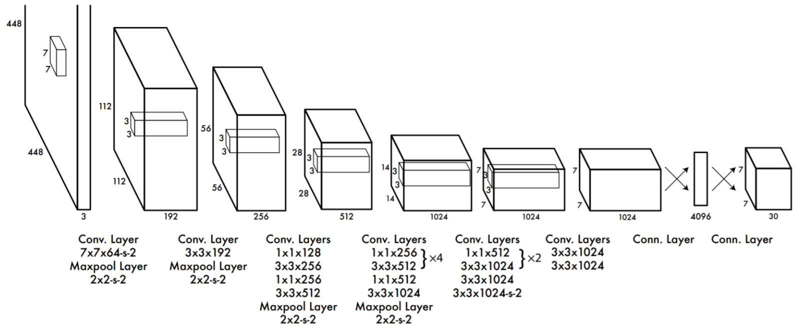

Fig. 5. Figure courtesy [38]: The network consist of 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. The convolutional layers are pretrained on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

ing box with highest intersection of union with the ground truth. This leads to specialization among the bounding box prediction. Each of the prediction keeps getting better and better after some time. YOLO uses sumsquared error between the predictions and the ground truth to calculate loss.

The loss function consists of three things the classification loss, the confidence loss and the localization loss. Classification loss is if an object is detected, the classification loss at each cell is the squared error of the class conditional probabilities for each class. localization loss measures the errors in the predicted boundary box locations and sizes. We only count the box responsible for detecting the object. The confidence loss is if an object is detected in the box, the confidence loss measures the object of the box. So, the loss function can be seen in figure 6

4.1 training model

After we labeled the dataset, we fed the model the trained data. The training dataset was more the fifteen hundred. While training a model on three pigs' data, first a test was run for data generators, Figure 7 is the image of the very first image from the dataset.We train the network for about 100 epochs on the training.Throughout training we use a batch size of 4, a momentum of 0.15 and a decay of 0.00015.

4.2 Reduce Learning Rate

Reduce learning rate parameters saves useless resource utilization and model overfeeding. For this reason, the parameter is set to reduce the learning rate by 0.2 if the loss does not improve after 20 iterations.

doi:10.20944/preprints202009.0088.v1

$$
\begin{aligned}
&\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}
$$

Fig. 6. Mathematical description of the generic YOLO [38] loss function.

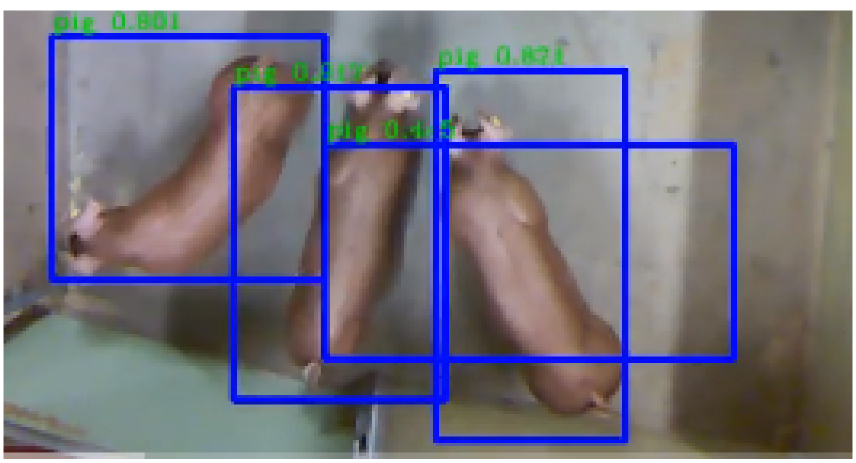

Fig. 7. The progression of the loss function.

4.3 Early Stopping

Another parameter that is used to prevent resource exploitation is Early Stopping. Patience is set 100 iterations that are, training would stop automatically if the loss does not improve after 100 iterations.

4.4 Loss

Training started at a loss over 200, after running 5000 iterations, the loss stopped showing improvement at 8.0. In the test case, when given the same video from which dataset was generated, very accurate results are produced. Below are some results figure 8 of the iterations after running more than almost 25000 times iterations on the dataset. The loss was down to 1.5.

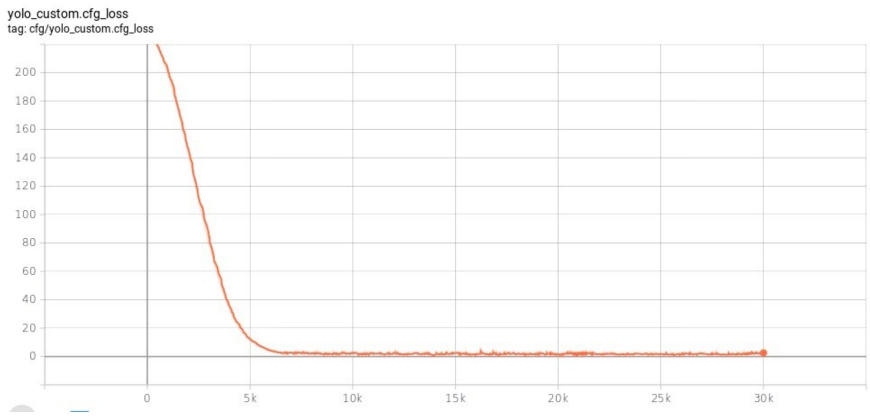

Fig. 8. Qualitative results of three pigs dataset.

The results also started to improve a lot and can be seen at loss 8.0 from the figure 9. After the loss was at 1.5 the results improved drastically As seen from the figure 10. More testing was done with this model on 6 pigs and the results obtain were impressive as seen from figure 11.

5 CONCLUSION

The proposed framework is implemented in Python with the support of Keras backend by TensorFlow. The processing is performed on Nvidia P-100 with 32 GB RAM. We proposed a transfer learning-based deep learning approach for detecting multiple pigs in the indoor farm setting. The proposed framework is implemented in Python with the support of Keras backend by TensorFlow. The processing is performed on Nvidia P-100 with 32 GB RAM. The framework is based on YOLO-v2 and the initial parameters are used as the optimal

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 4 September 2020 doi:10.20944/preprints202009.0088.v1

Fig. 9. Qualitative results of three pigs dataset.

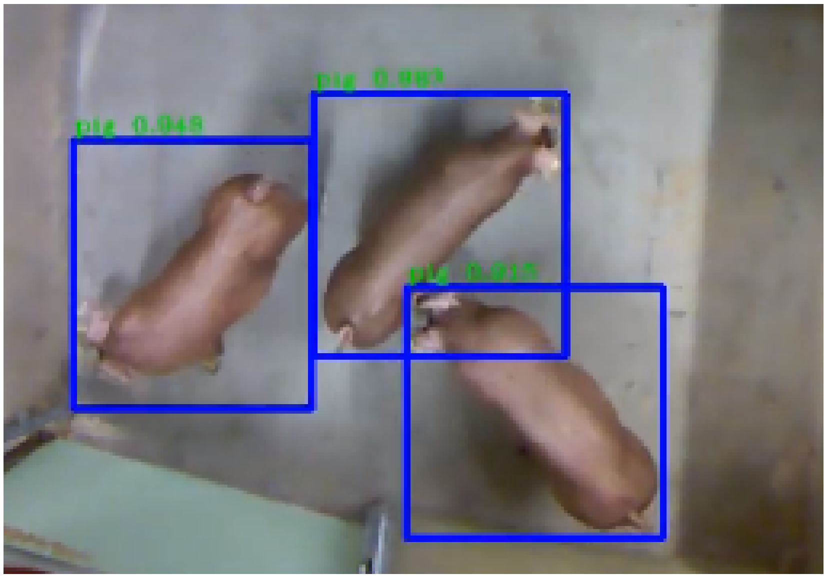

Fig. 10. Qualitative results of three pigs dataset.

 doi:10.20944/preprints202009.0088.v1

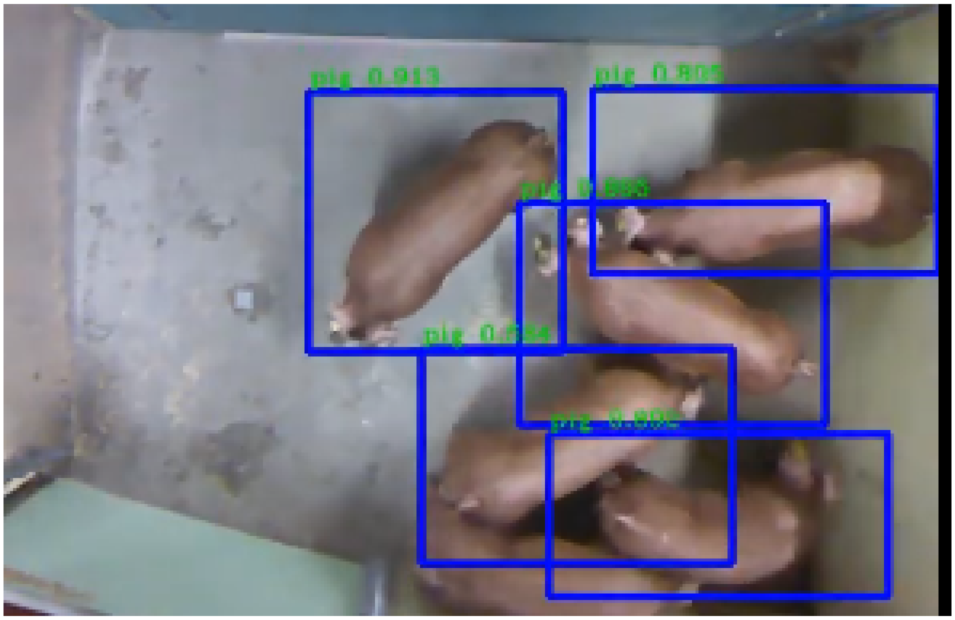

Fig. 11. Qualitative results of six pigs dataset.

starting values for training the network. Compared to the original YOLO-v2, we transformed the detector to detect only one class of objects i.e. pigs and the background. Since the dataset is recorded in a constraint environment, the backgrounds, camera angels, and color of objects (pigs) are very similar. Hence the model showed larger tendency to learn redundant information as compared to training the model on other dataset.For training the network, the farm-specific data is annotated with the bounding boxes enclosing pigs in the top view. Experiments are performed on a different configuration of the pen in the farm and convincing results have been achieved while using a few hundred annotated frames for fine-tuning the network.

References

1. Ju Hong Yoon, Chang-Ryeol Lee, Ming-Hsuan Yang, and Kuk-Jin Yoon. Structural constraint data association for online multi-object tracking. *International Journal of Computer Vision*, 127(1):1–21, 2019.
2. Mohib Ullah and Faouzi Alaya Cheikh. A directed sparse graphical model for multi-target tracking. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1816–1823, 2018.
3. Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 941–951, 2019.

4. Anton Milan, Stefan Roth, and Konrad Schindler. Continuous energy minimization for multitarget tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):58–72, 2014.
5. Mohib Ullah and Faouzi Alaya Cheikh. Deep feature based end-to-end transportation network for multi-target tracking. In *IEEE International Conference on Image Processing (ICIP)*, pages 3738–3742, 2018.
6. Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7942–7951, 2019.
7. Mohib Ullah, Ahmed Kedir Mohammed, Faouzi Alaya Cheikh, and Zhaohui Wang. A hierarchical feature model for multi-target tracking. In *IEEE International Conference on Image Processing (ICIP)*, pages 2612–2616, 2017.
8. Jianbing Shen, Dajiang Yu, Leyao Deng, and Xingping Dong. Fast online tracking with detection refinement. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):162–173, 2017.
9. Mohib Ullah. Siamese neural network based apperance model for multi-target. 2019.
10. Sultan Daud Khan, Habib Ullah, Mohammad Uzair, Mohib Ullah, Rehan Ullah, and Faouzi Alaya Cheikh. Disam: Density independent and scale aware model for crowd counting and localization. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4474–4478. IEEE, 2019.
11. Ana Paula Afonso, Maria Beatriz Carmo, Tiago Gonçalves, and Pedro Vieira. Visualeague: Player performance analysis using spatial-temporal data. *Multimedia Tools and Applications*, 78(23):33069–33090, 2019.
12. William R Johnson, Ajmal Mian, David G Lloyd, and Jacqueline A Alderson. On-field player workload exposure and knee injury risk monitoring via deep learning. *Journal of biomechanics*, 93:185–193, 2019.
13. Sultan Daud Khan, Habib Ullah, Mohib Ullah, Nicola Conci, Faouzi Alaya Cheikh, and Azeddine Beghdadi. Person head detection based deep model for people counting in sports videos. In *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–8. IEEE, 2019.
14. Habib Ullah, Mohib Ullah, and Muhammad Uzair. A hybrid social influence model for pedestrian motion segmentation. *Neural Computing and Applications*, pages 1–17, 2018.
15. Gaurav Tripathi, Kuldeep Singh, and Dinesh Kumar Vishwakarma. Convolutional neural networks for crowd behaviour analysis: a survey. *The Visual Computer*, 35(5):753–776, 2019.
16. Mohib Ullah, Habib Ullah, Nicola Conci, and Francesco GB De Natale. Crowd behavior identification. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1195–1199. IEEE, 2016.
17. Jiwei Chen, Wen Su, and Zengfu Wang. Crowd counting with crowd attention convolutional neural network. *Neurocomputing*, 382:210–220, 2020.
18. Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5099–5108, 2019.
19. Wei Peng, Xiaopeng Hong, Haoyu Chen, and Guoying Zhao. Learning graph convolutional network for skeleton-based human action recognition by neural searching. In *AAAI*, pages 2669–2676, 2020.
20. Mohib Ullah, Habib Ullah, and Ibrahim M Alseadonn. Human action recognition in videos using stable features. 2017.

21. Jun Yang, Zhongke Shi, and Ziyan Wu. Vision-based action recognition of construction workers using dense trajectories. *Advanced Engineering Informatics*, 30(3):327–336, 2016.
22. Mohib Ullah, Habib Ullah, Sultan Daud Khan, and Faouzi Alaya Cheikh. Stacked lstm network for human activity recognition using smartphone data. In *2019 8th European Workshop on Visual Information Processing (EUVIP)*, pages 175–180. IEEE, 2019.
23. Romero Morais, Vuong Le, Truyen Tran, Budhaditya Saha, Moussa Mansour, and Svetha Venkatesh. Learning regularity in skeleton trajectories for anomaly detection in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11996–12004, 2019.
24. Habib Ullah, Ahmed B Altamimi, Muhammad Uzair, and Mohib Ullah. Anomalous entities detection and localization in pedestrian flows. *Neurocomputing*, 290:74–86, 2018.
25. Manassés Ribeiro, André Eugênio Lazzaretti, and Heitor Silvério Lopes. A study of deep convolutional auto-encoders for anomaly detection in videos. *Pattern Recognition Letters*, 105:13–22, 2018.
26. Habib Ullah, Muhammad Uzair, Mohib Ullah, Asif Khan, Ayaz Ahmad, and Wilayat Khan. Density independent hydrodynamics model for crowd coherency detection. *Neurocomputing*, 242:28–39, 2017.
27. Deepak Kumar Jain, Pourya Shamsolmoali, and Paramjit Sehdev. Extended deep neural network for facial emotion recognition. *Pattern Recognition Letters*, 120:69–74, 2019.
28. Abdulrahman Alreshidi and Mohib Ullah. Facial emotion recognition using hybrid features. In *Informatics*, volume 7, page 6. Multidisciplinary Digital Publishing Institute, 2020.
29. Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008*, 2018.
30. Akif Quddus Khan, Salman Khan, Mohib Ullah, and Faouzi Alaya Cheikh. A bottom-up approach for pig skeleton extraction using rgb data. In *International Conference on Image and Signal Processing*, pages 54–61. Springer, 2020.
31. Habib Ullah, Sultan Daud Khan, Mohib Ullah, Faouzi Alaya Cheikh, and Muhammad Uzair. Two stream model for crowd video classification. In *2019 8th European Workshop on Visual Information Processing (EUVIP)*, pages 93–98. IEEE, 2019.
32. Fabio Tosi, Filippo Aleotti, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Luigi Di Stefano, and Stefano Mattoccia. Distilled semantics for comprehensive scene understanding from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4654–4665, 2020.
33. Habib Ullah, Ihtesham Ul Islam, Mohib Ullah, Muhammad Afaq, Sultan Daud Khan, and Javed Iqbal. Multi-feature-based crowd video modeling for visual event detection. *Multimedia Systems*, pages 1–9, 2020.
34. Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 808–816, 2016.
35. Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.
36. Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3974–3983, 2018.

37. Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2):261–318, 2020.
38. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
39. Jianguo Chen, Kenli Li, Qingying Deng, Keqin Li, and S Yu Philip. Distributed deep learning model for intelligent video surveillance systems with edge computing. *IEEE Transactions on Industrial Informatics*, 2019.
40. Habib Ullah and Nicola Conci. Crowd motion segmentation and anomaly detection via multi-label optimization. In *ICPR workshop on pattern recognition and crowd analysis*, volume 75, 2012.
41. Jiayu Sun, Jie Shao, and Chengkun He. Abnormal event detection for video surveillance using deep one-class learning. *Multimedia Tools and Applications*, 78(3):3633–3647, 2019.
42. Habib Ullah. *Crowd Motion Analysis: Segmentation, Anomaly Detection, and Behavior Classification*. PhD thesis, University of Trento, 2015.
43. Miso Ju, Younchang Choi, Jihyun Seo, Jaewon Sa, Sungju Lee, Yongwha Chung, and Daihee Park. A kinect-based segmentation of touching-pigs for real-time monitoring. *Sensors*, 18(6):1746, 2018.
44. Ruilong Chen, Ruth Little, Lyudmila Mihaylova, Richard Delahay, and Ruth Cox. Wildlife surveillance using deep learning methods. *Ecology and evolution*, 9(17):9453–9466, 2019.
45. Wei Geng, Pamela Cosman, Charles C Berry, Zhaoyang Feng, and William R Schafer. Automatic tracking, feature extraction and classification of c. elegans phenotypes. *IEEE transactions on biomedical engineering*, 51(10):1811–1820, 2004.
46. Joseph J Lim, Russ R Salakhutdinov, and Antonio Torralba. Transfer learning by borrowing examples for multiclass object detection. In *Advances in neural information processing systems*, pages 118–126, 2011.
47. Shohei Yonetsu, Yutaro Iwamoto, and Yen Wei Chen. Two-stage yolov2 for accurate license-plate detection in complex scenes. In *2019 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–4. IEEE, 2019.
48. Mohib Ullah, Mohammed Ahmed Kedir, and Faouzi Alaya Cheikh. Hand-crafted vs deep features: A quantitative study of pedestrian appearance model. In *2018 Colour and Visual Computing Symposium (CVCS)*, pages 1–6. IEEE, 2018.
49. G Chandan, Ayush Jain, Harsh Jain, et al. Real time object detection and tracking using deep learning and opencv. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 1305–1308. IEEE, 2018.
50. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.
51. Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
52. Trent Houliston and Stephan K Chalup. Visual mesh: Real-time object detection using constant sample density. In *Robot World Cup*, pages 45–56. Springer, 2018.
53. Jimut Bahan Pal. Real time object detection canbe embedded on low powered devices. 2019.