JOURNAL OF SCIENCE, TECHNOLOGY AND ENGINEERING RESEARCH
Bilim, Teknoloji ve Mühendislik Araştırmaları Dergisi
ISSN (Online) 2717-8404
Available *online* at https://dergipark.org.tr/en/pub/jster

RESEARCH ARTICLE

Human Instance Segmentation Based on Omega- Shape Using Deep Learning

* Huma Sheraz, [1] Zuhaib Ahmed Khan, and [2] Muhammad Awais[3]

*National University of Modern Languages NUML, Faculty of Computing, Software Engineering Department, Islamabad, Pakistan
Huma.sheraz@numl.edu.pk https://orcid.org/0009-0002-8562-7947
[1]Mirpur University of Science and Technology, Mirpur Azad Kashmir, Pakistan
engr.zohaib01@gamil.com https://orcid.org/0009-0007-4849-1303
[2] Capital University of Science and technology, Faculty of Computing, Software Engineering Department, Islamabad, Pakistan
Mawaiskhan1808@gmail.com https://orcid.org/0009-0006-3101-524X

Citation:
Sheraz, H., Khan, Z.A., Awais, M. (2025). Human Instance Segmentation Based on Omega- Shape Using Deep Learning, *Journal of Science Technology and Engineering Research*, 6(1): 31-41. DOI: 10.53525/jster.1469697

HIGHLIGHTS

- *Advances detection accuracy*
- *Collaborative insights from varied*
- *Refinement of human detection methods*
- *Pivotal for future advancements*

Article Info

Received **:** April 17, 2024
Accepted **:** February 7, 2025

DOI:
10.53525/jster.1469697

***Corresponding Author:**

Muhammad Awais
mawaiskhan1808@gmail.com
Phone: +92 3418977385

ABSTRACT

Human detection and segmentation in an unconstrained environment is a very important and difficult task, having many important applications including tracking human beings, pedestrian detection, head count etc. Human detection in a single object environment is quite easy, but the problem becomes quite cumbersome in a crowded and cluttered environment. Most of the existing algorithms work on the detection and segmentation of the whole person in a scene. However, the performance of such algorithms degrades in case of occlusion and cluttered environment. To increase the performance in such an environment a technique is available that detects a distinct part of the human body which is "omega shape" instead of the full human body. However, the detection of bounding box also includes background pixels which limit the performance of the high-end applications such as tracking. Therefore, the objective of this research is to accurately segment the omega shape, so that the high-end applications have no background clutter in the human appearance model. We have trained and evaluated Mask R-CNN and YOLO+UNET and got a trade-off between accuracy and computation cost. The testing accuracy of Mask R- CNN and YOLO+UNET is 92.6% and 88.4%, while the computation cost is 6fps and 29fps, respectively.

Keywords: *Deep Learning, Omega-shape, Pedestrian-Detection, Instance Segmentation*

I. INTRODUCTION

This document In this century, with the increase in computational power, the concept of Computer Vision(CV) has caught attention in the field of security, agriculture, and segmentation. Segmentation is the process of segregation of a digital image into multiple homogeneous parts. The objective of segmentation is: 1) to change the illustration of an image for ease of analysis and 2) recognition. Mask R-CNN [1] is the state-of-art segmentation algorithm but it lags in term of execution time due to its multi stage network. So, this paper pays attention on computation time and proposes a method for real time human segmentation. It is very important because it has been widely used in security, head-count, pedestrian detection, driver-less cars, and human tracking.

Human tracking is the process of following and locating person in the video. With the recent advancements in the field of object detection, most researchers performed multiple human tracking by the process of tracking-by-detection. Tracking by detection has two steps. The first step is human detection and the second step is instance matching. Human detection in a scene is quite a cumbersome task even today due to the variation in pose and occlusion of the human body. To tackle pose variation and occlusion, we propose a detection technique in which only omega shape (it includes head, torso, and arms) is detected. The bounding box (with white boundary) in fig 1 shows the omega-shaped of the human body. This helps vastly as variations in pose and occlusion caused by a crowded environment does change and occupy the omega shape in most cases. Another major challenge in human detection is back- ground clutter which results in problematic instance matching. Therefore, this paper also focuses on instance segmentation of the human body using omega-shape features.

The contributions of our work can be summarized as:

- Implementation of an automated system for the instance segmentation of human.
- To tackle the pose and occlusion problem.
- To speed up the instance segmentation process.

Figure 1. Omega-shape

II. LITERATURE REVIEW

Human Detection is of vital importance in many fields. These fields include pedestrian detection, tracking, head count etc. Pedestrian detection is gaining wide popularity in applications like driverless technologies. Whereas, tracking is equally important in the problems like security. In complex environment where a large population and high occlusion is present, human detection becomes a harder job. A substantial amount of research in human detection and segmentation has been done in recent times. The researchers have opted three kinds of methods to achieve

aforementioned task which are, rule-based method, traditional machine learning based methods and deep learning-based methods.

There are several methods for solving computer vision problems by manipulating the images based on a specific rule without applying any learning. Such methods are known as rule-based methods. Among those, there are only few techniques which work in real conditions for a long period of time [2]. Methods for human detections demonstrated in [3], [4]. Works on a supposition that a person will appear in an occlusion free environment for a time period, giving enough time for a model of the pedestrian to be built up while it is isolated from occlusion. The techniques presented in [3], [4], [5], [6] are the examples of algorithms which fail in real time environments which are unconstrained. The reason behind the failure of algorithms in a real environment is that they rely on color intensity. The techniques described in [7], [8], [9], [10] employ periodic features attained from a time-based set of frames for human detection. In [11] a system with high reliability is introduced which integrate various nods such as face, shape pattern and color to detect humans. Nevertheless, face detection can detect only pedestrian facing towards camera and color is very sensitive to illumination changes. The problems with Rule based algorithms are quite evident in the above-mentioned algorithms i.e. they require presence of constrained environment, and their performance suffer in the unconstrained environment.

In the last decade, some quite efficient traditional machine learning based algorithms for pedestrian detection are created. There are different detectors which employ different sets of features and classification methods. A popular technique which involves linear SVM classifier in combination with HOG features is presented by Dalal and Triggs [12]. This algorithm, when applied on the INRIA dataset provided superior performance. Following the technique produced by the researchers in [12], many researchers have modified the HOG based method by introducing extra features which include color information [13], LBP [14], second order statistical measures [15], [16], and many other methods to boost the performance in comparison with the original detector. However, it also increased the computational cost of detectors. A cascade-based detector ACF proposed by Dollar et al [17], uses histograms of augmented orientation with color information treated as differentiating structures and subsequently, they used Adaboost based classifiers for the classification purpose [13]. To enhance the speed of detection, ACF estimates features among various scales to avoid unnecessary computations. This detector was a breakthrough in the field of human detection. Authors in [15] proposed a SPF detector which is a significant enhancement on ACF detector. This algorithm improves the collections of features used by ACF to take in an LBP variant, full 360-degree orientations of calculated gradients and spatial gradients of second order. Authors in [18], [19] demonstrated a detailed analysis of the individual features used by various known detectors such as HOG and ACF and their influence on the final detection performance. In all of the above algorithms full humans are detected. These algorithms provide degraded performance in environment with high occlusion and clutter. This problem can be rectified by using Omega-Shape features. The method proposed by researchers in [20] used such features by deploying Viola-Jones Adaboost classifier along with local HOG features. The most important factor while implementing the traditional machine learning algorithms mentioned is the selection of features. These selected features, if not selected properly will make the model over-fit during the process of learning, which results in loss of performance during test time.

Convolutional neural networks (CNN) algorithms for human detection are propelled by the accomplishment of CNNs in various detection-based tasks. Some of the works of incorporating CNNs in pedestrian detection are [21], [22], [23], [24], [25], [6]. They mostly employed R-CNN [26] which is a two- stage network for detection. Furthermore, in these techniques, rule- based detectors such as ICF [27] and its improvements [28], [29], [30] were deployed for region proposals, followed by a CNN to classify the pre-planned areas once more. CNN presented in [31] employed convolutional sparse coding to train the model basic parameters, and then fine-tuned on the Caltech dataset. Very Fast proposed in [32] created a CNN based cascade, that uses a small CNN to filter candidates before passing through to a deep CNN. F-DNN explained in [33] employed a fused multiple CNNs and soft cascade mechanism in the stage two cascade. The Faster R-CNN presented in [34] has become a standard model for detection of general object, but it underperforms when applied to pedestrian detection problem without any modification. This is because of the background clutter and low resolution of object [35]. Higher performance can be achieved when boosted decision forest was deployed on top of convolution feature maps [35], [36], [37]. There are still other approaches which still show top performance. These algorithms involve customized CNNs derived from Faster R-CNN architectures [38], [39]. Researchers in [40] revealed that after proper modification, a simple Faster R-CNN model can be similar to the state of the art in terms of performance. The problem with previously mentioned techniques is that they detect a whole human body, which reduces the performance of the CNNs significantly in an environment with partially occluded human

bodies. The human bodies have a distinct omega shape which can be detected even in occluded environment. The authors in [41] used a deep Omega-shape feature learning and multi-paths detection to make the detector more efficient to human pose variations and scale changes. The problem with omega shape human detection is that the presence of background pixels in the bounding box reduces the performance of high-level application of human tracking, because appearance model has background pixels in addition to foreground human pixels. The algorithm presented in [42], rectifies the problem created by background pixels in an environment where whole human shape is segmented. Our proposed methodology performs instance segmentation of human, based on omega shape features using deep learning techniques to overcome the above-mentioned problems.

Present-day deep learning methods, such fully convolutional neural networks, address several important remote sensing issues, including the identification and segmentation of artificial land objects on extremely high-resolution hyperspectral map sceneries and the utilization of non-visible bandwidths. The three components of the solution are the creation of the dataset, the neural network, and the neural network fine-tuning. The initial section of the solution proposes a semi-automated approach for quickly creating datasets with contemporary technologies while maintaining the benefits of remote sensing imaging, such extremely high resolution and the ability to use a range of visual and thermal hyperspectral imagery bandwidths. The current research offers a method for solving the problem of remote sensing images segmentation using deep fully convolutional neural networks, which can be used to demonstrate the use of a dataset. The binary classifier for a dataset is constructed based on Mask RCNN, and the best deep learning architectures for instance segmentation are taken into consideration and examined. Using several configurations and optimizers, the final neural network architecture with the new classifier is adjusted to maximize the benefits of the created hyperspectral remote sensing dataset [47].

Design of a revolutionary iterative deep reinforcement learning agent that can simultaneously learn to distinguish between several items. Using a graph coloring approach, we have built our reward function for the trainable agent to favor grouping pixels that belong to the same item. We show that instance segmentation of several objects may be accomplished rapidly with the suggested technique without requiring extensive post-processing [48].

Introduce Poly-YOLO, an improved version of Yolo that incorporates instance segmentation for improved efficiency. Expanding upon the concepts of YOLOv3, Poly-YOLO eliminates two of its drawbacks: an excessive number of rewritten labels and an ineffective anchor distribution. By employing stair step up sampling and a hyper column approach to aggregate data from a light SE-Darknet-53 backbone and provide a single scale output with high resolution, Poly-YOLO lessens the problems. While only 60% of Poly-YOLO's parameters are trainable, it increases the mean average accuracy by a relative 40% when compared to YOLOv3. Additionally, we propose Poly-YOLO lite, which has a lower output resolution and fewer parameters. While it is twice as quick and three times smaller as YOLOv3, it has the same accuracy and is therefore appropriate for embedded systems. Lastly, Poly-YOLO uses bounding polygons to do instance segmentation. The network is taught to identify polygons on a polar grid that are size-independent. Poly-YOLO generates polygons with different numbers of vertices since each polygon's vertices are predicted together with their confidence [49].

Introduce SeqFormer, a tool for segmenting video instances. SeqFormer constructs instance connections among video frames by utilizing the vision transformer idea. However, we note that, in order to capture a temporal sequence of instances in a video, a stand-alone instance query is sufficient; however, attention methods must be applied to each frame separately. In order to accomplish this, SeqFormer finds an instance in every frame and then combines temporal data to learn a strong representation of a video-level instance. This representation is then utilized to dynamically anticipate the mask sequences on every frame. Without post-processing or branch tracking, instance tracking occurs organically. Without bells and whistles, SeqFormer achieves 47.4 AP on YouTube-VIS with a ResNet-50 backbone and 49.0 AP with a ResNet-101 backbone. This accomplishment greatly outperforms the prior state-of-the-art by 4.6 and 4.4, respectively. Furthermore, when combined with the recently suggested Swin transformer, SeqFormer attains a significantly higher AP of 59.3 [50].

Introduce OSFormer, the first one-stage disguised instance segmentation (CIS) transformer framework. OSFormer is built around two main concepts. Firstly, by incorporating location-guided queries and blend-convolution feed-forward network, we construct a location-sensing transformer (LST) to retrieve the location label and instance-aware parameters. Secondly, we create a coarse-to-fine fusion (CFF) to combine various context data from the CNN backbone

and LST encoder. By combining these two elements, OSFormer may effectively combine local characteristics and distant context dependencies to forecast instances that are concealed. Our OSFormer exhibits high convergence efficiency and reaches 41% AP when compared to two-stage frameworks without requiring a large amount of training data [51].

Figure 1. Images from PASCAL-part Dataset [43]

III. METHOD

As discussed, our work is about the instance segmentation of omega-shape using deep learning techniques. We used the Pascal-part dataset, which is publicly available [43]. We trained Mask R-CNN and YOLO+UNET and used pre-trained weights for the training of Mask R-CNN and YOLO, while we trained UNET from scratch. Dataset and proposed model architecture are discussed in the following sections.

A. Dataset

As discussed in the previous section that pascal-part is a publicly accessible dataset. This dataset is a set of additional annotations for PASCAL VOC 2010 [44]. The dataset consists of 20 classes i.e. human, vehicle, boat, etc. We have used 3,503 images of human class in which 2,452 (70%) are used for training and 1051 (30%) for validation and testing. Fig 2 shows some sample images of human class.

Mask R-CNN

Mask R-CNN [1] is the extended version of Faster R-CNN for pixel-level instance segmentation. Instance segmentation is the task to locate each object in the class, predict its class and provide a binary mask for each object in that class. At the high level, the Mask RCNN consists of feature pyramid network and backbone layer, followed by region proposal network which generates positive region (object) and bounding box refinement.

We selected the Mask R-CNN as our model [1]. The model choice is based on the following consideration; first Mask R-CNN can achieve advanced results on a range of object detection tasks. One such example is that Mask R-CNN has beaten the top models in the most recent COCO competition for object detection. Secondly, the Mask R-CNN algorithm is a quite simple and bendable technique system for object detection and object instance segmentation. In addition, the model can be easily implemented with pre-trained weights in Keras and TensorFlow library, and the model can be accessed on GitHub, which happens to be an open-source repository.

The current work implemented the same network architecture as defined in the Mask R- CNN paper. Mask R-CNN model is based on two base phases. The first phase scans the image input, and then provides an output called proposal (also called the region of interest (ROI)), which is the area where object is likely to be present. The second phase classifies the region and generates the bounding box and masks. At high level, the Mask RCNN consists of Feature Pyramid Network plus Backbone, followed by region proposal network which generates positive region (object) and bounding box refinement. A mask basically is, a CNN based network which, out of ROI, digs out the positive regions and then generates a mask for them. The various components of our Mask R-CNN algorithm design and their functionality are discussed below.

The Convolutional Res-Net backbone and FPN: Residual Network (Res-Net) [51] was introduced initially as CNN to perform the image classification task but it became a popular choice for other deep learning tasks. Residual networks enable us to efficiently train deep neural network simply by introducing skip connections, in which weights coming from previous layers, are copied into more deep layers. In our model, ResNet101 (a variant of Res-Net) serves as a

basic CNN network of our network which serves as a feature extractor in this specific application. The initial layers detect the parameters like edges, corners etc. (low-level features) and the deep layers detect higher features. This backbone layer processes the images at the input, and outputs them into the feature map in the last layer. He et al. used RestNet101 in combination with Feature pyramid network (FPN) to serve as a backbone layer.

Region proposal Network (RPN): The next RPN layer in Mask RCNN architecture scans the input images in "sliding window" method and is an easy-to- go neural network. It finds the area which contains the object. RPN scans over the extracted feature of backbone and it increases the performance and efficiency of the network. The RPN scan over the region is called anchors. There are more than 200k anchors for an image. The RPN generates two classes from each anchor. These two classes are; bounding box refinement and anchor class. Anchor class has one of two classes i.e. foregrounds (object) and background. While bounding box ensures to fit the object better by refining the anchor perfectly.

Non-max Suppression (NMS): Using RPN, the algorithm picks the top anchor in which the objects are probable to exist and then approximate their location in close proximity. If there are multiple overlapped anchors, the one with the highest foreground score will be chosen and this process is known as NMS. After this, the final Region of interest (ROI) is passed to the next stage.

ROIAlign: It is used to extracts ROIs on feature maps and then scale them to the same size. The ROIAlign is designed to fix the misalignment problem, which is that the scaled size output is not the same position as the original ROI's position.

Classifier and Regressor: This layer generates two outputs for each ROI i.e. it's class and bounding box around each class.

Segmentation mask: Another CNN based network, the Segmentation Mask branch takes the fixed-size feature map and generates the mask for them. During the inference mode, the predicted mask is resized to the ROI bounding box of original size and provides the final mask for each object.

B. Yolo+UNET

Mask R-CNN is quite accurate, but it lags in terms of execution speed. To increase the execution speed of segmentation, we proposed a new technique which is a combination of YOLO [45] and UNET [46]. YOLO+UNET can be divided into 3 phases. Yolov3 which is the first part, suggests the region of interest (ROI), and regresses for the classes and confidence scores. In the thesis, a modified YOLOv3 is implemented, which also output the feature maps from different layers of the residual blocks. The second part is ROIAlign that takes different YOLO bounding box outputs and feature maps as inputs and generates the fixed-size ROI feature maps. UNET is the last part, which transforms the ROI inputs to the mask outputs. The whole model architecture is shown in the below fig 3. The detailed architecture is discussed below.s

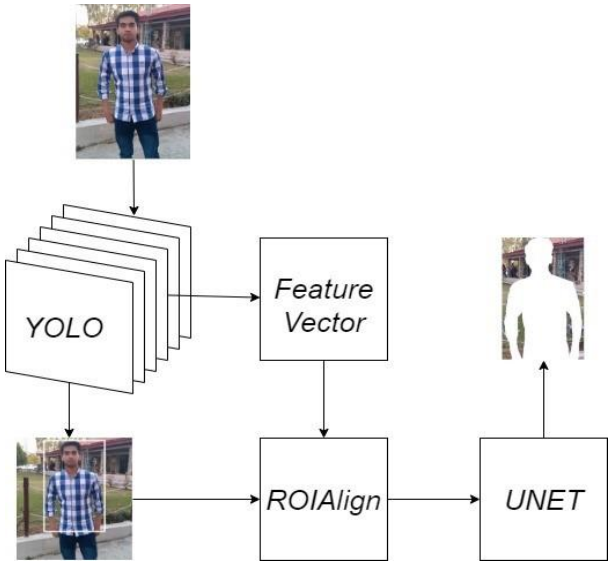

Figure 3. YOLO+UNET Model Pipeline

Yolo: You only look once (YOLO) is an end-to-end network for object detection. Many cutting- edge technology is generated by the 3rd generation of YOLO, which is YOLOv3. YOLO splits the image to SxS block, and then it is the responsibility of each block to detect those targets whose center points fall within the grid. To eliminate the duplicate bounding boxes Non-Maximum Suppression is used after detection. The network includes residual block-based backbone, feature pyramid network like network head for multi-scale prediction, batch normalization, anchor boxes prediction etc. These details of the network are discussed below.

	Type	Filter	Size	Output
	Convolutional	32	3x3	256x256
	Convolutional	64	3x3/2	128x128
1x	Convolutional	32	1x1	
	Convolutional	64	3x3	
	Residual			128x128
	Convolutional	128	3x3/2	64x64
2x	Convolutional	64	1x1	
	Convolutional	128	3x3	
	Residual			64x64
	Convolutional	256	3x3/2	32x32
8x	Convolutional	128	1x1	
	Convolutional	256	3x3	
	Residual			32x32
	Convolutional	512	3x3/2	16x16
8x	Convolutional	256	1x1	
	Convolutional	512	3x3	
	Residual			16x16
	Convolutional	1024	3x3/2	8x8
4x	Convolutional	512	1x1	
	Convolutional	1024	3x3	
	Residual			8x8

Figure 4. Darknet53 Architecture

Darknet53:Darknet53 is an efficient backbone for performing feature extraction. It is a very deep backbone that contains 53 convolutional layers. It conveys many advanced structures including: (a) Residual blocks which add shortcut layers to make the network easier to train; (b) Inception structure, that contains 3x3, 1x1 convolutional kernel, which keeps the respective field and decreases the computation cost; (c) Batch normalization layer makes the learning of layers in the network more independent of each other. With the high efficiency of the Darknet53, it is selected as the feature extractor for both YOLOv3 head and for later segmentation done by us. Darknet53 architecture is shown in fig 3.

Feature Vector Formation for Segmentation: As shown in fig 5, We extract the feature vector of different size from the backbone layers, which are 5,10,26 and we name them V1, V2, and V3 accordingly. When the layer goes more in-depth, the size of the feature map reduces to half of the previous layer. Then we up-sample V2, V3 and do the feature concatenation with their previous feature map. Because in networks when we go deeper into the layers we lose lower-level information about boundaries, edges, and contours. So, to recover this information we take features from lower layers and concatenate them with higher.

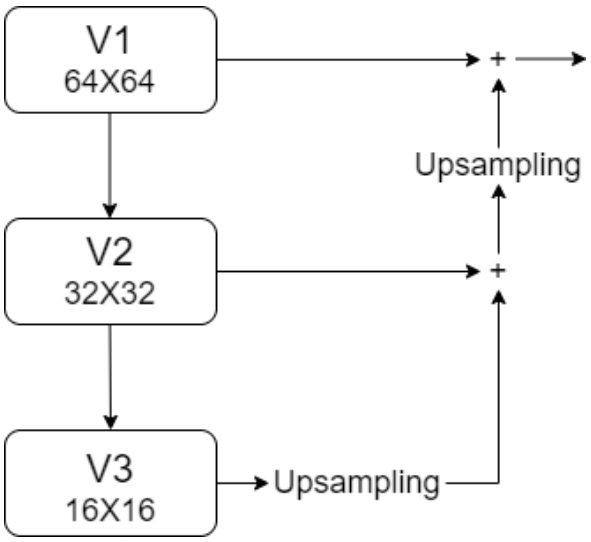

Figure 5. Feature Vector Extraction

UNET: UNET is an object segmentation algorithm that does not depend on a region proposal network. It was designed by Olaf Ronneberger et al. [46] for biomedical image segmentation. It is an encoder-decoder framework for object segmentation.

UNET architecture comprises of two phases. First phase is the construction phase (encoder) that captures the context in the image. It is just a stack of CNN and max-pooling layers. The second phase is the expanding path (decoder). It is used for exact localization using transposed convolutional. UNET is an end- to-end a fully convolutional network as it just comprises of convolutional layers. We reduced layers of UNET from 26 to 14 because the input size is 28x28 and at sixth layer feature vector size reduces to 7X7. In our modified YOLO+UNET model, YOLO detects the bounding boxes of each instance than ROI-Align crops these instances from the feature vector. UNET takes each cropped instance one by one as input and gives the segmented output.

IV. RESULT

Our models are trained on a NVIDIA GeForce RTX2060. We have implemented Mask R- CNN with FPN plus Res-Net-101 as a backbone network, and YOLO with backbone Darknet53 with modified UNET for segmentation. We used open-source deep learning framework Keras and TensorFlow for model implementatoin. For the training of Mask R- CNN and YOLO, we used the model transfer learning method.

A. Model Performance

We compare Mask R-CNN with YOLO+UNET. We trained both the models using the same dataset and the same GPU. Speed and accuracy performance are discussed in the following sections.

Speed Performance: First, we performed a speed comparison between Mask R-CNN and YOLO+UNET. Since the YOLO is a single-stage network that's why we achieved real-time 29 frames per second (FPS) segmentation with YOLO+UNET, and it is 4.8 times faster than Mask R-CNN segmentation.

Accuracy Performance: Secondly, we conducted an experiment regarding accuracy. Mask R-CNN achieved 92.6% and YOLO+UNET achieved 88.4%. Experimental results show that YOLO+UNET's accuracy is less with the difference of 4.2% which is not a big difference compared to the speed improvement of YOLO+UNET over the Mask R-CNN.

In terms of speed as it is 4.8 times (48%) faster than Mask R-CNN. We also compute the segmentation accuracy of both models by dropping the instances from the ground truth which are not detected by the models. The segmentation accuracy. The difference is only 1.2% which shows that YOLO+UNET and Mask R-CNN has almost the same segmentation performance. Some sample qualitative results of Mask R-CNN and YOLO+UNET.

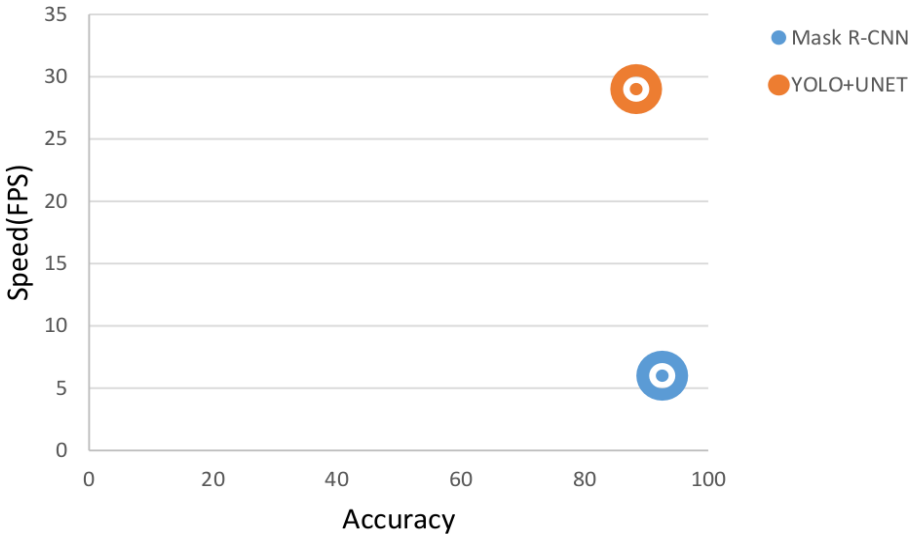

Figure 6. Speed Accuracy Tradeoff

Detection Performance: We also conducted experiments regarding the detection of omega-shape. The detection performance for both the models. The accuracy of YOLO is 3.09% less than Mask R-CNN which affect the segmentation accuracy of YOLO from these experiments we concluded that Mask R-CNN and YOLO+UNET has almost the same segmentation accuracy.

This hybrid technique to human instance segmentation has never been employed before, combining state-of-the-art models Mask R-CNN, YOLO, and UNET. The precision of Mask R-CNN in region-based segmentation, YOLO's real-time detection capabilities, and UNET's potent feature extraction for fine-grained segmentation are all tapped into in this innovative combination to provide a more reliable and precise framework. The study illustrates its advancements by contrasting its findings with those of earlier research. In particular, it compares performance to current human instance segmentation models in terms of speed (FPS), accuracy (mIoU, Precision, Recall, F1-score), and detection (AP, AP@50, AP@75). By incorporating Omega-Shape features into segmentation, the limitations of conventional methods are addressed and boundary refinement and object discrimination are further improved. This study outperforms traditional strategies by introducing a novel fusion mechanism that maximizes both detection and segmentation accuracy.

V. CONCLUSION

In this research, we trained Mask R-CNN and YOLO+UNET for the instance segmentation of human using omega shape feature on Pascal part dataset. For the training of Mask R- CNN and YOLO, we used pre-trained weights (from COCO Dataset), but not for UNET because we changed its layers. Furthermore, we evaluated our models over accuracy and speed performance. During the experiment, we concluded that Mask R-CNN gives results with an accuracy of 92.6% which is good enough and shows its reliability. However, the speed of Mask R-CNN is 6FPs, so to enhance the speed we developed another model namely YOLO+UNET. Experimental results showed that YOLO+UNET is 4.8 times faster than Mask R-CNN while, its accuracy is 88.4%, which is 4.2% less than that of Mask R- CNN. This is a very good trade off in terms of speed over accuracy. Therefore, a segmentation model for real-time usage is developed which can be used for real-time high- end applications like tracking.

CONFLICTS OF INTEREST

All the authors of this paper declare that there is no conflict of interests regarding the publication of this manuscript.

RESEARCH AND PUBLICATION ETHICS

The research conceptualization and methodology were done by Huma Sheraz. The technical and theoretical framework was prepared by Zuhaib Ahmed Khan. The technical review and improvement were performed by Muhammad Awais.

ACKNOWLEDGMENT

Thankfully, we are aware of our parent's affection for us. Last but not least, we would like to thank our family and friends whose prayers made it possible for us to finish this project.

REFERENCES

[1]. M. Harville, "Stereo person tracking with adaptive plan-view templates of height and occupancy statistics," Image and Vision Computing, vol. 22, no. 2, pp. 127–142, 2004.

[2]. A. Senior et al., "Tracking people with probabilistic appearance mod- els," in ECCV workshop on Performance Evaluation of Tracking and Surveillance Systems. Citeseer, 2002, pp. 48–55.

[3]. A. Elgammal and L. S. Davis, "Probabilistic framework for segmenting people under occlusion," in Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, vol. 2. IEEE, 2001, pp. 145–152.

[4]. W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, and S. Maybank, "Principal axis-based correspondence between multiple cameras for people track- ing," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pp. 663–671, 2006.

[5]. J. Rittscher, P. H. Tu, and N. Krahnstoever, "Simultaneous estimation of segmentation and shape," in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2. IEEE, 2005, pp. 486–493.

[6]. C.-J. Pai, H.-R. Tyan, Y.-M. Liang, H.-Y. M. Liao, and S.-W. Chen, "Pedestrian detection and tracking at crossroads," Pattern Recognition, vol. 37, no. 5, pp. 1025–1034, 2004.

[7]. B. Heisele and C. Woehler, "Motion-based recognition of pedestrians," in Proceedings. Fourteenth International Conference on Pattern Recog- nition (Cat. No. 98EX170), vol. 2. IEEE, 1998, pp. 1325–1330.

[8]. R. Cutler and L. S. Davis, "Robust real-time periodic motion detection, analysis, and applications," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 781–796, 2000.

[9]. S. A. Niyogi, E. H. Adelson et al., "Analyzing and recognizing walking figures in xyt," in CVPR, vol. 94, 1994, pp. 469–474.

[10]. ——, "Analyzing and recognizing walking figures in xyt," in CVPR, vol. 94, 1994, pp. 469–474.

[11]. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), vol. 1. IEEE, 2005, pp. 886–893.

[12]. P. Dolla´r, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," IEEE transactions on pattern analysis and machine intelligence, vol. 36, no. 8, pp. 1532–1545, 2014.

[13]. X. Wang, T. X. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in 2009 IEEE 12th international conference on computer vision. IEEE, 2009, pp. 32–39.

[14]. S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Pedestrian detec- tion with spatially pooled features and structured ensemble learning," IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 6, pp. 1243–1257, 2015.

[15]. T. Watanabe and S. Ito, "Two co-occurrence histogram features using gradient orientations and local binary patterns for pedestrian detection," in 2013 2nd IAPR Asian Conference on Pattern Recognition. IEEE, 2013, pp. 415–419.

[16]. S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in 2008 IEEE conference on computer vision and pattern recognition. IEEE, 2008, pp. 1–8.

[17]. R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool, "Seeking the strongest rigid detector," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3666–3673.

[18]. S. Zhang, R. Benenson, B. Schiele et al., "Filtered channel features for pedestrian detection." in CVPR, vol. 1, no. 2, 2015, p. 4.

[19]. M. Li, Z. Zhang, K. Huang, and T. Tan, "Rapid and robust human detection and tracking based on omega-shape features," in 2009 16th IEEE International Conference on Image Processing (ICIP). IEEE, 2009, pp. 2545–2548.

[20]. J. Hosang, M. Omran, R. Benenson, and B. Schiele, "Taking a deeper look at pedestrians," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4073–4082.

[21]. S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?" in Proceedings of the iEEE conference on computer vision and pattern recognition, 2016, pp. 1259–1267.

[22]. Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1904–1912.

[23]. ——, "Pedestrian detection aided by deep learning semantic tasks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 5079–5087.

[24]. D. Ribeiro, J. C. Nascimento, A. Bernardino, and G. Carneiro, "Im- proving the performance of pedestrian detectors using convolutional learning," Pattern Recognition, vol. 61, pp. 641–649, 2017.

[25]. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[26]. P. Dolla´r, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," 2009.

[27]. P. Dolla´r, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," IEEE transactions on pattern analysis and machine intelligence, vol. 36, no. 8, pp. 1532–1545, 2014.

[28]. W. Nam, P. Dolla´r, and J. H. Han, "Local decorrelation for improved pedestrian detection," Advances in neural information processing sys- tems, vol. 27, pp. 424–432, 2014.

[29]. R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool, "Seeking the strongest rigid detector," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3666–3673.
[30]. P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in Proceed- ings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 3626–3633.
[31]. A. Angelova, A. Krizhevsky, V. Vanhoucke, A. Ogale, and D. Ferguson, "Real-time pedestrian detection with deep network cascades," 2015.
[32]. X. Du, M. El-Khamy, J. Lee, and L. Davis, "Fused dnn: A deep neural network fusion approach to fast and robust pedestrian detection," in 2017 IEEE winter conference on applications of computer vision (WACV). IEEE, 2017, pp. 953–961.
[33]. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 6, pp. 1137–1149, 2016.
[34]. L. Zhang, L. Lin, X. Liang, and K. He, "Is faster r-cnn doing well for pedestrian detection?" in European conference on computer vision. Springer, 2016, pp. 443–457.
[35]. Q. Hu, P. Wang, C. Shen, A. van den Hengel, and F. Porikli, "Pushing the limits of deep cnns for pedestrian detection," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 6, pp. 1358– 1368, 2017.
[36]. Z. Cai, M. J. Saberian, and N. Vasconcelos, "Learning complexity-aware cascades for pedestrian detection," IEEE transactions on pattern analysis and machine intelligence, 2019.
[37]. J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast r- cnn for pedestrian detection," IEEE transactions on Multimedia, vol. 20, no. 4, pp. 985–996, 2017.
[38]. Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in European conference on computer vision. Springer, 2016, pp. 354–370.
[39]. S. Zhang, R. Benenson, and B. Schiele, "Citypersons: A diverse dataset for pedestrian detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3213–3221.
[40]. Y. Xu, X. Zhou, P. Liu, and H. Xu, "Rapid pedestrian detection based on deep omega-shape features with partial occlusion handing," Neural Processing Letters, vol. 49, no. 3, pp. 923–937, 2019.
[41]. K. Chen, X. Song, X. Zhai, B. Zhang, B. Hou, and Y. Wang, "An integrated deep learning framework for occluded pedestrian tracking," IEEE Access, vol. 7, pp. 26 060–26 072, 2019.
[42]. Pascal voc 2010. [Online]. Available: http://roozbehm.info/pascal-parts/ pascal-parts.html
[43]. "Pascal voc 2010." [Online]. Available: http://host.robots.ox.ac.uk/ pascal/VOC/voc2010/index.html
[44]. J. Redmon and A. Farhadi, "Yolov3: An incremental improvement,"arXiv preprint arXiv:1804.02767, 2018.
[45]. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.
[46]. Hnatushenko, V., & Zhernovyi, V. (2020, August). Method of improving instance segmentation for very high resolution remote sensing imagery using deep learning. In International Conference on Data Stream Mining and Processing (pp. 323-333). Cham: Springer International Publishing.
[47]. Anh, T. T., Nguyen-Tuan, K., Quan, T. M., & Jeong, W. K. (2020). Reinforced coloring for end-to-end instance segmentation. arXiv preprint arXiv:2005.07058.
[48]. Hurtik, P., Molek, V., Hula, J., Vajgl, M., Vlasanek, P., & Nejezchleba, T. (2022). Poly-YOLO: higher speed, more precise detection and instance segmentation for YOLOv3. Neural Computing and Applications, 34(10), 8275-8290.
[49]. Wu, J., Jiang, Y., Bai, S., Zhang, W., & Bai, X. (2022, October). Seqformer: Sequential transformer for video instance segmentation. In European Conference on Computer Vision (pp. 553-569). Cham: Springer Nature Switzerland.
[50]. Pei, J., Cheng, T., Fan, D. P., Tang, H., Chen, C., & Van Gool, L. (2022, October). Osformer: One-stage camouflaged instance segmentation with transformers. In European Conference on Computer Vision (pp. 19-37). Cham: Springer Nature Switzerland.