Enhanced Deep Learning Framework for Cow Image Segmentation

Rotimi-Williams Bello, *Member*, *IAENG*, Ahmad Sufril Azlan Mohamed, Abdullah Zawawi Talib, Daniel A. Olubummo, and O. Charles Enuma

Abstract— **The applications of deep learning to livestock farming have in recent years gained wide acceptance from the computer vision community due to the continuous achievement of its applications to agricultural tasks. Moreover, the essentiality of deep learning is its practicality in detecting, segmenting, and classifying video and image objects without which precision livestock farming would have been impossible. However, the applications of most of the state-of-the-art models of deep learning to multiple cow objects image segmentation are not accurate and cannot generate colorimetric information due to poor pre-processing mechanism inherent in the associated methods and unequal training of their backbone layers. To overcome the above-mentioned limitations, an enhanced deep learning framework of Mask Region-based Convolutional Neural Network (Mask R-CNN) based on Generalized Color Fourier Descriptors (GCFD) is proposed. The enhanced model produced 0.93 mean Average Precision (mAP). The result shows the performance capability of the proposed framework over the state-of-the-art models for cow image segmentation.**

Index Terms— **Deep learning, GCFD, Image segmentation, Mask R-CNN**

I. Introduction

BOTH dairy cows and beef cows deserve constant monitoring as this is important for first-hand behavioral information regarding their traceability, welfare, and performance records. Through computer vision methods, researchers and breeders can access timely health and performance information of individual cows within the husbandry, this can prevent unnecessary loss and death of the livestock [1]. Moreover, the consumption of dairy and beef cows in recent years has increased due to the human population explosion, and this has caught the attention of many researchers on the need to devise better technological means of monitoring and maintaining both dairy and beef sectors. To achieve this, certain required measures must be considered which could support the individual cow wellbeing and welfare for overall intensive farming [2].

To maintain any large-scale animal husbandry, there is a tremendous management task requirement that involves state-of-the-art methods for precision livestock practice such that there will be an assurance of a lofty input-turnout ratio of cow productivity with a cheap maintenance cost [3], [4]. For precision livestock farming, different techniques of object detection have been applied for tracking cows and monitoring their health status through their behavioral learning [5]. With the instability and variation in the price of beef and dairy products across the globe, the marginal difference between the highly successful husbandry and the fairly successful ones is too close, and to cope with the situation without negatively affecting the animals' wellbeing, much attention from the cow breeders are required on planning an uninterrupted workflow irrespective of the workload [6]. The continuous monitoring of individual cows involves much time which cannot be achieved manually. Moreover, the herculean task involved in daily livestock activity does not encourage efficient physical monitoring and assessment of individual cows [7]. Hence, the need for computer vision applications that can handle the majority of the livestock tasks such as tracking individual cow activities to monitor the cow's spatial distribution, health information, and performance.

Recently, the research community has witnessed the ubiquitous intelligence of machine learning, and this has amounted to advancing its application to features extraction and representation. Some examples to support this are human identifiers such as MPII [8], COCO for human skeleton [9], DeepPose for human body parts detection using images [10], Stackedhourglass network [11], ArtTrack [12], OpenPose [13], [14], Deepcut [15], and Human pose detection [16]. However, machine learning cannot detect each target object as accurately as human beings. The computer vision community over the past few years has recorded reasonable results in detecting and segmenting objects (both instance and semantic) with a Convolutional Neural Network (CNN) playing a vital role. Powerful algorithms such as CNN [17], Fast R-CNN [18], Faster R-CNN [19], YOLO [20], SSD [21], and Mask R-CNN [22] are some of the recently developed algorithms that drive the sudden advancement witnessed in the object detection and segmentation.

Mask R-CNN is an instance segmentation algorithm that recently gained wide recognition in computer vision-based tasks. The semantic segmentation algorithm only segments

Manuscript received December 11, 2020; revised August 30, 2021.

R. W. Bello is a PhD candidate of School of Computer Sciences, Universiti Sains Malaysia, 11800, Pulau Pinang, Malaysia (corresponding author phone: +2348038897840; e-mail: sirbrw@yahoo.com).

A. S. A. Mohamed is a Doctor of School of Computer Sciences, Universiti Sains Malaysia, 11800, Pulau Pinang, Malaysia (e-mail: sufril@usm.my).

A. Z. Talib is a Professor of School of Computer Sciences, Universiti Sains Malaysia, 11800, Pulau Pinang, Malaysia (e-mail: azht@usm.my).

D. A. Olubummo is a PhD candidate of Computer and Information Systems, Robert Morris University, Moon-Township, Pennsylvania, United States of America (e-mail: adanielolu@yahoo.co.uk).

O. C. Enuma is a Head of Department of Planning, Research and Statistics, Ministry of Health, Bayelsa State, Nigeria (e-mail: charlieyogo@yahoo.com).

the objects' class at the pixels level. The semantic segmentation algorithm cannot differentiate between objects that belong to the same class. The instance segmentation algorithm on the other hand differentiates between the objects of the same class in the image pixel-by-pixel by finding mask representation for the objects. As earlier iterated, due to the effects of deep learning applications in many areas such as computer vision [23], image recognition [24]-[26], localization [27], [28], and segmentation [26], research in the domain has received wide acceptance. The most effective technique of deep learning that accurately reports recognition and detection tasks as compared to any other known algorithms is CNN [29].

Newly developed algorithms exist in the literature for objects localization and segmentation [30], [31], and the application of these algorithms was extended to the cow detection and segmentation systems [32]-[34], in which Faster R-CNN [18] was modified to Mask R-CNN [22] and employed for detection and instance segmentation of cow in an image. Xu et al. [33] adopted a Fully Convolutional Network (FCN) [29], ResNet [25], and Mask R-CNN [22] for the detection and pixel-by-pixel instance segmentation of the cow images. The behavior displays by individual cows plays a significant role in detecting their health status; but, enough studies have not been performed on cow behavior-related problems. To precisely manage livestock farming, Mask R-CNN was employed by Xu et al. [33] for cow counting, and Qiao et al. [34] for cow image segmentation. However, the applications of most of the state-of-the-art models of deep learning to multiple cow objects image segmentation are not accurate and cannot generate colorimetric information due to poor pre-processing mechanism inherent in the associated methods and unequal training of their backbone layers.

Therefore, an enhanced deep learning framework (Mask R-CNN$_{\text{enhanced}}$) is proposed in this study. The study produces results that support the performance capability of the proposed framework over the state-of-the-art models for cow detection and identification. The work in this paper is a contributory step towards precision livestock farming with the following contributions:

1. The application of GCFD to image conversion from 1024×1024×3 (RGB) to 32×32×2048 feature maps whereby the image conversion problems associated with ResNet101 are mitigated.
2. The GCFD is produced by dividing the color of the image into each specific color channel, namely red (r), green (g), and blue (b).
3. The computation of each of these channels produces three sets of descriptors.

The rest of this study is structured as follows: Section 2 presents the materials and methods; Section 3 presents the implementation; Section 4 presents the results and discussion, and Section 5 concludes the study with future work.

II. MATERIALS AND METHODS

The datasets employed for this study were collected from two different sources. The first dataset was collected by constructing a recognizing system to capture cows in the ranch, and this dataset is labeled the Own cow dataset. The second dataset was the standard dataset from Microsoft Common Objects in COntext (MS COCO datasets) [9].

A. Acquisition of Own Cow Dataset

The application of CNN architecture in the past is associated with two main problems, namely the problem of the unlabeled dataset and the problem of low computer computational power. Nevertheless, with the implementation of labeling tools such as LabelMe [35], and improved computational methods using the Graphics Processing Unit (GPU), the associated problems have been mitigated. However, the unavailability of a publicly accessible dataset of cow videos and well-annotated cow images has made the cow instance segmentation process using video datasets a very difficult task. As a result, in this study, a dataset of cow videos was created particularly for the instance segmentation task.

Aside from the standard MS COCO cow dataset [9], that was collected from the public repository, the process used in collecting and generating the video dataset employed in this work was by setting up a recognizing system as shown in Fig. 1. The constructed dataset was made up of 1000 frames (720×1280 sizes) which were extracted from the ten cows that were captured in the ranch by the recognizing camera. Randomly, the dataset was divided into two parts of which 800 (80%) was used as a training dataset and 200 (20%) was used as a testing dataset.

Unlike the MS COCO cow dataset, it was a herculean task to collect and employ the video dataset due to the following reasons:

1. The instability of the cows in the ranch and frequent assumption of different positions affected the quality of the images captured. Therefore, a more capable and generalized model is needed for the segmentation task.
2. The similarity in color and other characteristics that exist among the cow objects made it almost impossible to differentiate one cow from another, making it a challenging assignment for even state-of-the-art methods to differentiate.
3. The placement of the recognizing camera brought about partial occlusion that affected the complete view of the cow.
4. The recognizing system was affected by poor illumination. This was one of the major challenges of the image segmentation tool as it became very difficult for the segmentation tool to differentiate between the objects and the patches in the image.
5. The image's background bearing a resemblance with the image's object makes it a difficult task for the segmentation algorithm to accomplish in detecting cows because of the presence of darkness, noise, and so on in the image.

B. Benchmark Dataset

Some of the commonly used datasets in competition and research involving object detection are Pascal VOC [37],

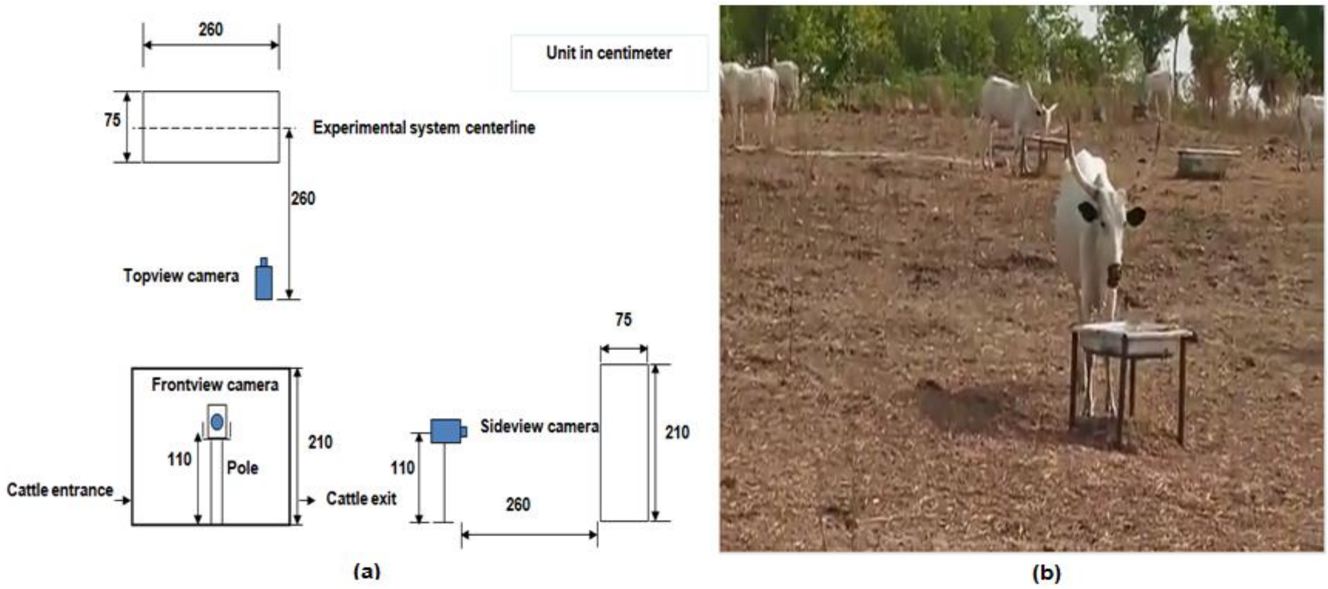

Fig. 1. (a) System for recognizing cattle, (b) Cattle ranch.

MS COCO dataset [9], and ImageNet [36]. This study employs MS COCO datasets because the datasets comprise eighty (80) different classes with over two hundred and fifty thousand (250,000) data of different scenes publicly accessible as training and validation datasets. The MS COCO datasets contain a cow dataset of over two thousand and seventy-one (2071) images from which one thousand nine hundred and eighty-six (1986) images were used for training, and eighty-seven (87) images were used for both validation and testing. While the Own cow dataset was used for implementing and validating the model, both the Own cow dataset and MS COCO cow dataset were used for the testing. Shown in [9] are the images of cattle from the Microsoft COCO dataset.

C. Data Pre-processing

The employed Own cow dataset for this study consists of 1000 cow images (720×1280 pixels). The dataset was not completely processed in its captured form, making it unsuitable concerning the ratio of cow object's pixels to the background. Therefore, the dataset was enhanced to improve training performance. The next step after collecting the dataset from the video was the annotation of each image's object. Segmentation of an individual image's object is difficult and time-consuming; therefore, the LabelMe annotation tool [35] was employed to annotate each cow's image. As earlier said, the collected videos were categorized using the LabelMe annotation tool [35] to ease segmentation and improve segmentation time for the overall validation of the framework. Information about the cow width and the cow length was generated from the annotated images. The normalization of the datasets was between 0 and 255 before converting them into a Microsoft COCO dataset format [9] using Python.

As there is no one-fit-all format for storing the mask of each segmented object, PNG was employed as the format for storing both the original images and their mask representations. The creation of a mask for each image's object was carried out followed by labeling. Bilinear interpolation was used in representing the two values of mask for the two classes; 0 depicts the background object and 1 depicts the cow which represents both the foreground object and the ground truth. The dimension of the mask is the same as the dimension of every other image which is 720x1280 dimensions. The stacking together of the image's ground-truth mask of all the ten (10) cow objects produced an array of 720x1280x10 dimensions; the array was stored and labeled as the image to enable easy identification of the mask representation.

D. Network Architecture of the Proposed Framework (Mask R-CNN$_{enhanced}$)

This section describes the architecture of the proposed framework employed for the cow image instance segmentation task. The algorithm of the proposed framework is an extension of the algorithm of Mask R-CNN [22]. Although, Mask R-CNN algorithm has flexibility and a simple framework for detecting and classifying objects, the colorimetric information of the object in the image is affected during the segmentation process due to the image patches on the processed image. The algorithm of the proposed framework is combined with the GCFD algorithm for effective retention of colorimetric information of the object in the image. Illustrated in [22] is the architecture of the Mask R-CNN with ResNet101 as the backbone, and Feature Pyramid Network (FPN) as the front-end of the network.

The operational structure of Mask R-CNN is such that the first stage of its two-stage frameworks performs extraction operation on the inputted image to generate feature maps which are then passed onto Region Proposal Network (RPN) to generate a class of anchor and a refined bounding box before passing to Region of Interest (RoI) classifier and bounding box regression. The alignment of the RoI (RoIAlign) is necessary for the second stage of the two-stage frameworks where the objects are reduced to the same fixed size feature maps before passing through the Fully Connected Network (FCN) for the generation of the bounding box refinement, object's class, and masks. For every positive region of RoI, the mask gets the positive regions of the RoI to generate a mask of low resolution (28×28 pixels) for each of them.

The generated masks are soft masks with their representation in float numbers, making them holding more details than binary masks. While the ground-truth masks are scaled down to 28×28 during training to enable the computation of loss, the predicted masks on the other hand are scaled up to the size of the bounding box's RoI during inferencing. This process leads to the generation of final masks for each of the objects as shown in Fig. 2.

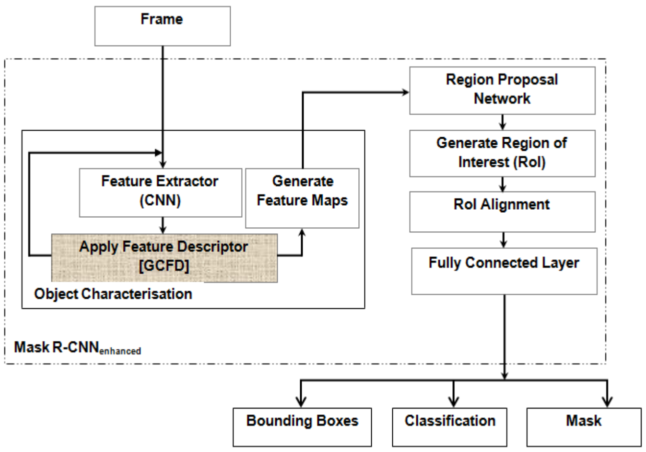

Fig. 2. The proposed framework of cow detection and classification system.

Convolutional Residual Network (ResNet101) Backbone and Feature Pyramid Network (FPN)

He et al. [25] introduced the Residual Network (ResNet) for image classification problems. The network becomes a strong backbone for many deep learning networks. ResNet101 as a variant of Residual Network serves as the CNN of the proposed framework for feature extraction as shown in Fig. 2. This is to enable the detection of higher features such as cow images that are involved in this study unlike what is obtainable in the early ResNet layers which only detect low-level features such as edges. To increase the efficiency of the model, the different values of Intersection over Union (IoU) of the predicted object are taken against the 0.5 value of the ground-truth object to measure the degree of detection accuracy. Using the different values of IoU, any predicted object that is less in value than the value of the ground-truth object is considered invalid and therefore discarded. While any value of the predicted object that is equal or greater than the value of the ground-truth object is considered valid and therefore selected. This process is referred to as Non-Max Suppression (NMS) whereby the algorithm chooses the anchor that contains cow objects with the highest positive score during overlapping and gets their location refined.

The operation of NMS always results to anchor having different sizes of feature maps thereby creating a problem for the RoI classifier. To solve this problem, the size of feature maps must be re-sized to a fixed size by cropping using the method known as RoI pooling. The ResNet architecture, though effective, is not completely efficient due to the following reasons:

1. The unnecessary and insufficient training that some of its layers pass through which as a result wastes computational time and negatively affects the performance of the ResNet as a whole.
2. The unnecessary delay in converting an image from one dimension to another as input to the stages that follow.

These issues of ResNet101 prompted the development of strategies that involve the use of GCFD as shown in Fig. 2 with the mathematical implication of GCFD given in Eq. (2). Illustrated in [25] is the ResNet architecture, and the formula for establishing the two-layer block is presented as follows:

$$H(x)=F(x,\{W_i\}) + x \qquad (1)$$

where,
x = Building block input.
H(x) = Building block output vectors.
$F(x,\{W_i\})$ = The learned residual mapping in the training process.

Improved ResNet101 Using Generalized Color Fourier Descriptors (GCFD)

The unnecessary delay of ResNet101 in converting an image from one dimension to another as feature maps input to the stages that follow reduces the computational speed of the Mask R-CNN in the segmentation process. The application of GCFD in the image segmentation process as shown in Fig. 2 enhances the image such that the corners of the cow image can be easily located. The detection of the edges is based on the color images generated from every frame through which the cow features used in generating the masks were described. As earlier iterated, the generated masks are soft masks with their representation in float numbers, making them holding more details than binary masks. While the ground-truth masks are scaled down to 28×28 during training to enable the computation of loss, the predicted masks on the other hand are scaled up to the size of the bounding box's RoI during inferencing. The descriptors process is effective in improving the detection of the image patches without affecting the patches' color (contrast, brightness, and saturation).

Dividing the color of the image into each specific color channel red (r), green (g), and blue (b) is the ideal technique to produce GCFD. Computation of each of these channels produces three sets of descriptors by combining two descriptors that were computed from both parallel projection and orthogonal projection in two dimensions Fast Fourier Transform (FFT). The formula that defines the method used in combining these two projections [38] is as follows:

$$GCFD\ B(f) = \{GCFD\|B(f) + GCFD^{\perp}B(f)\} \qquad (2)$$

where $GCFD\ B(f)$= Computation of GCFD, $GCFD\|B(f)$ = GCFD in parallel part, and $GCFD^{\perp}B(f)$= GCFD in orthogonal part.

The GCFD is equivalent to the Classical Generalized Fourier Descriptors (CGFD) computation on the parallel part and the orthogonal part of the Clifford Fourier

Transform (CFT). Regarding the parallel part, the computation of GCFD is on the red channel while the computation of the chromatic plane of green and blue is on the orthogonal part. A 16 doubles vector is produced when a segmented cow image is processed using the GCFD feature vector. The first and the ninth values are on the very high side because they represent the first descriptor of GCFD in parallel GCFD$\|$B(f) and GCFD in orthogonal GCFD$^{\perp}$B(f) in that order.

Loss Function

The difference that exists between the predicted value and the ground-truth value is defined by the loss function in the course of training the network. Furthermore, the position of the loss function is essential in training the model for cow image instance segmentation. In our proposed framework, a combination of loss function was applied in the training of the following:

1. Bounding box regression.
2. Object class prediction.
3. Mask branch segmentation.

Eq. (3) represents the loss function used in accomplishing this task:

$$L = L_{ce} + L_{be} + L_{me} \qquad (3)$$

where L represents loss function, L_{ce} represents classification error, L_{be} represents bounding box regression error, and L_{me} represents mask error.

III. IMPLEMENTATION

A Graphics Processing Unit (GPU), TensorFlow [39], Keras, and OpenCV-Python are some of the major hardware and software packages installed on the system on which the proposed framework was implemented. Keras is a popular Python deep learning Application Programming Interface (API) that has the low-level flexibility for implementing arbitrary research ideas while voluntarily presenting high-level expediency features to speed up experimentation processes. TensorFlow on the other hand is an end-to-end open-source Python deep learning application that serves as a platform for machine learning. It possesses an all-inclusive and flexible network of tools and libraries that help research in advancing the state-of-the-art methods in machine learning, and developers effortlessly build and deploy machine learning-powered applications.

The effectiveness of TensorFlow in handling high-performing computation and the ease with which code optimization becomes with TensorFlow make it suitable for this detection and segmentation task. The hyperparameters used in training the network model are shown in Table I. The information about the software and hardware employed in implementing this study is presented in Table II. The optimization algorithm employed for the framework was based on gradient descent [40]. To solve the problem of unnecessary training of some ResNet layers, the entire backbone component layers were fixed, and only the head of the network was left for independent training by using in each case, the training dataset.

TABLE I
NETWORK MODEL HYPERPARAMETERS

Spec	Value
Learning rate	0.001
Weight decay	0.0001
Momentum of learning	0.9
Dimension of image (minimum)	512
Dimension of image (maximum)	512
Detection confidence (minimum)	0.5
Number of batches	5
Size of batch	200
Epochs	5
Iterations per epoch	1000
Mask shape	28×28
Number of anchor classes (cow and background	2

IV. RESULTS AND DISCUSSION

The main goal of instance segmentation lies in detecting and predicting the objects' class using bounding boxes. The results of the experiments that were performed in this study using the proposed framework are discussed in this section. It is clearly illustrated in Fig. 3 (proposed model) and Fig. 4 (existing model) the changes that occurred to the images of the cows after they have passed through the segmentation process. One notable achievement of the proposed model is the brightness and contrast effect of GCFD in the segmentation of the multiple cow objects in the image with their bounding boxes, class predictions, confidence scores, and masks individually generated unlike what is obtainable from the existing segmentation methods. Additional achievement of the proposed framework is the improved time efficiency of the GCFD over ResNet101 algorithm in converting an image from one dimension to another, and the reduction in illumination variation during the Own cow data capturing exercise, thereby mitigating the pixels misjudgment between the foreground (cow body) and the shadow (background).

The method used in the proposed framework mitigates the influence of similarity in coat patterns of external objects on the cow objects used for the experiment. The segmentation accuracy and the computational time of the proposed framework are presented in Table III. The following formula is used in measuring the segmentation process accuracy:

$$\text{Accuracy} = ((A \cap B)/(A \cup B)) \times 100 \qquad (4)$$

where A represents the bounding box of the predicted object and B represents the bounding box of the ground-truth object.

The accuracy of the proposed framework on the datasets that were enhanced is approximately 93% with a processing time of 0.70s, and the accuracy of the datasets that were enhanced is approximately 1% more than the accuracy of

TABLE II
Software and hardware requirements for the research implementation

Software	Type/Version	Hardware	Type/Version
Operating System	64-bit Windows 10	CPU	Intel Core i5 processor@2.4GHz
IDE	Visual studio 2019	RAM	16 Gigabytes
Python library	Keras	Graphics card	GeForce GTX 1080 Ti
MATLAB	R2019b	Hard-disk	2 Terabytes
		Camera module	Vision Datum LEO 640H-200gc High-Speed 200fps Sharp RJ33 CCD Gigabit Ethernet 3d
		Monitor	10.1 inch IPS HD Portable LCD Gaming Monitor PC display VGA HDMI interface for PS3/PS4/XBOx360/CCTV/Camera

the datasets that were not enhanced. Using the proposed framework, the image showed tremendous improvement in quality during the cow image segmentation over the Mask R-CNN and MaskSplitter [41] methods. Fig. 5 shows the GCFD-aided soft-mask result of contour line extraction using the proposed framework; there is a similarity between the result and the actual contour of the cow used for the experiment making the proposed framework preferable to Mask R-CNN [22] which produces output with binary mask [34]. Most of the limitations noticed in this study are due to a few training datasets available for the experiment as deep learning performs segmentation better with huge datasets.

The difference in center errors (pixel length) [42] which measures the difference between the predicted objects and the manually labeled datasets (ground-truth objects) using the LabelMe tool, computed as average distance error (ADE) of the extracted contour line is presented in Table IV, where 30.46 ADE of the extracted contour line was achieved by the proposed framework with a processing time of 0.71s, making it significantly better than the results obtained in Ter-Sarkisov et al. [41] and Qiao et al. [34].

Fig. 6 shows the qualitative comparison of the following: (1) Ground-truth cow image; (2) Mask R-CNN-based segmentation; (3) Enhanced Mask R-CNN-based segmentation; (4) SSD-based segmentation; (5) Faster R-CNN-based segmentation; and (6) YOLOv2-based segmentation.

TABLE III
INSTANCE SEGMENTATION RESULTS

Operation	Method	Data type	mAP	Time (s)
1) Instance segmentation	Mask R-CNN	Raw	0.90	0.73
		Enhanced	0.92	0.72
2) Instance segmentation	Enhanced Mask R-CNN	Raw	0.92	0.72
		Enhanced	**0.93**	**0.70**

TABLE IV
CONTOUR EXTRACTION RESULTS

Operation	Minimum	Maximum	ADE	Time (s)
1) Contour extraction using Mask R-CNN	0.035	64.17	35.56	0.77
2) Contour extraction using the proposed model	0.029	61.19	**30.46**	**0.71**

A. Performance Evaluation

The performance of the proposed framework is evaluated using the metric of mean Average Precision (mAP) in finding the area under the Precision-Recall (PR) curve of each cow class. The first PR curve is produced by evaluating the system, and for that specific cow class, an Area Under the Curve (AUC) is computed and called Average Precision (AP). In any problem that involves instance segmentation, the overlapping rate between the value that is predicted, and the value of the ground truth is measured using IoU. The IoU of the predicted instances and the ground truth of the image’s annotated objects must match for the PR curves to be generated. If the value of IoU is greater than the predefined value, and the class of both the instance that is generated and the instance of ground truth possess are the same, this implies a match between the instance that is produced and the instance of the ground truth. Eq. (5) represents the IoU as follows:

$$\text{IoU}= \frac{\text{Area of intersection}}{\textit{Area of union}} \qquad (5)$$

The IoU values from 0.5 to 0.95 with mAP@X notation are considered for this work, where X is the value of the threshold employed to compute the metric. Only after all the matches for the image are established can the precision-recall be computed. Precision is the total number of correct instances that the model produces, and it is computed as follows:

$$\text{P}= \frac{\text{True Positive}}{\textit{True Positive} + \textit{False Positive}} \qquad (6)$$

A recall measures the total positive instances that the model can produce, and it is computed as follows:

$$\text{R}= \frac{\text{True Positive}}{\textit{True Positive} + \textit{False Negative}} \qquad (7)$$

where true-positive predicted as positive as was correct, false-positive predicted as positive but was incorrect, and false-negative failed to predict an object that was there. AP is calculated by taking the area under the PR curve and by segmenting the recalls evenly to different parts. AP is calculated as follows:

$$\text{AP} = \sum_{n=1}^{N} [\text{R}(n) - \text{R}(n-1)].\max \text{P}(n) \qquad (8)$$

where N is the calculated number of PR points.

B. Comparison of Proposed Framework with State-of-the-Art Methods

Fig. 6 and Table V show the qualitative and quantitative comparison of the proposed framework (enhanced Mask R-

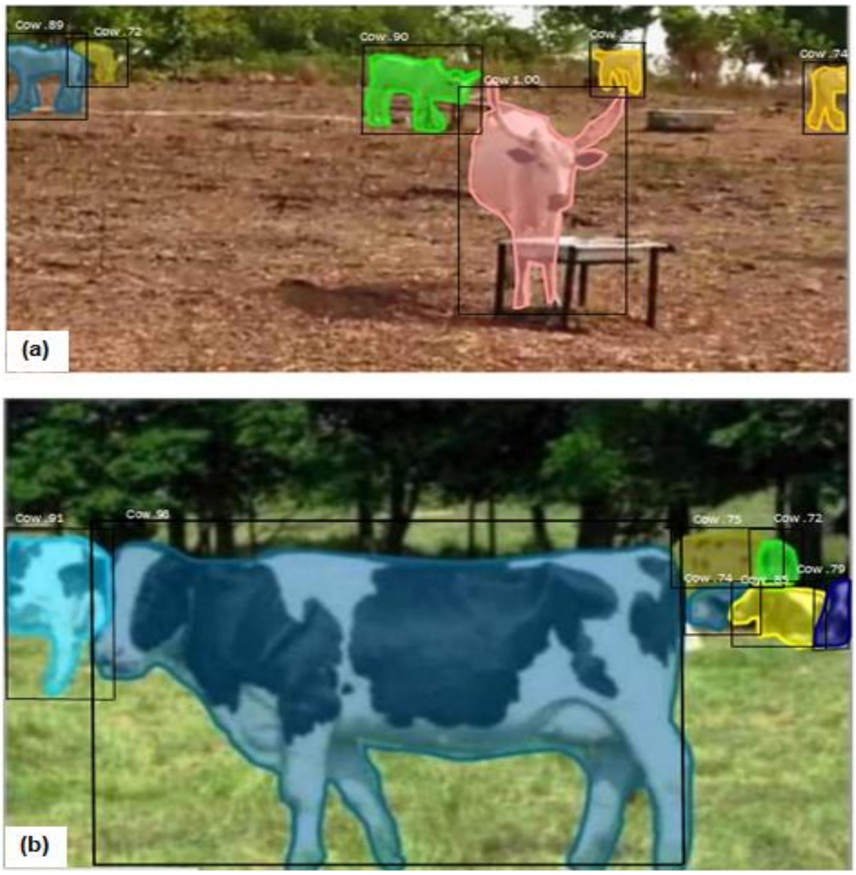

Fig. 3. Segmentation result of the proposed model on (a) Own cow dataset (Fig. 1b) and (b) MS COCO dataset with the generation of bounding boxes, the class predictions, the confidence scores, and the masks.

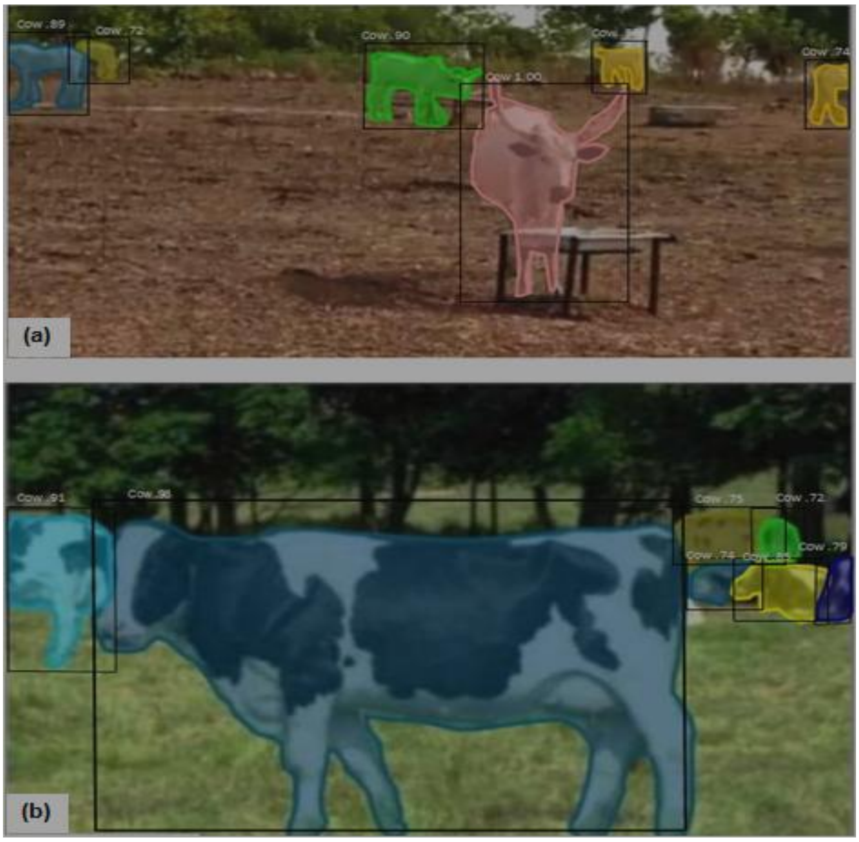

Fig. 4. Segmentation result of the existing model on (a) Own cow dataset (Fig. 1b) and (b) MS COCO dataset with the generation of bounding boxes, the class predictions, the confidence scores, and the masks.

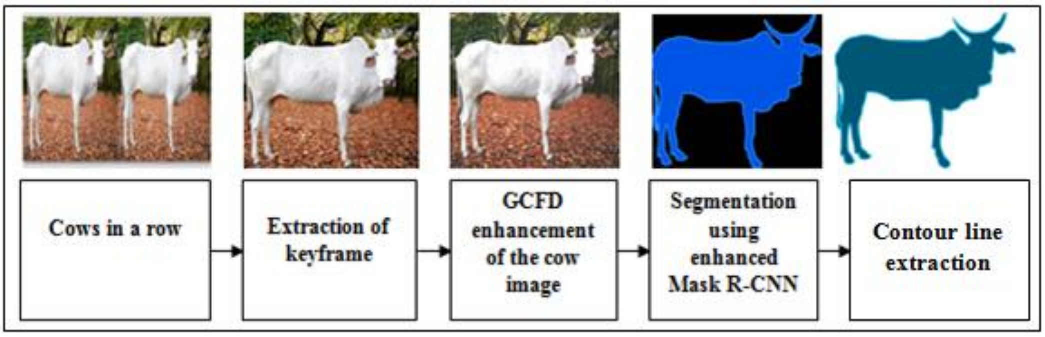
Cows in a row
Extraction of keyframe
GCFD enhancement of the cow image
Segmentation using enhanced Mask R-CNN
Contour line extraction

Fig. 5. GCFD-aided soft-mask result of contour line extraction using the proposed framework.

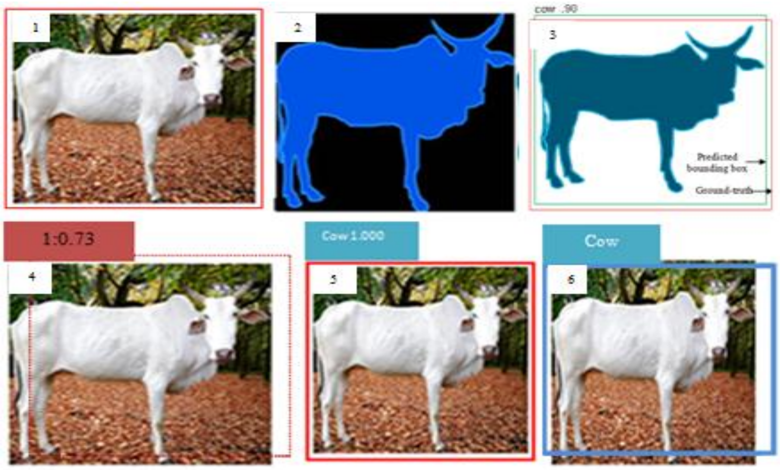

Fig. 6. (1) Ground-truth cow image (2) Mask R-CNN-based segmentation (3) Enhanced Mask R-CNN-based segmentation (4) SSD-based segmentation (5) Faster R-CNN-based segmentation (6) YOLOv2-based segmentation.

CNN) and state-of-the-art segmentation methods to further the performance evaluation of the proposed framework. MaskSplitter as one of the important models used for cow instance segmentation [41] is built using VGG16 as the backbone. MaskSplitter is made up of a Fully Convolutional Network (FCN) and mask of ground truth for refining the mask representation of a cow in an image without predicting any bounding box. Neither bounding box nor Region Proposal Network (RPN) is used by MaskSplitter in instance segmentation. The model of MaskSplitter framework was trained to learn how to generate three different types of mask representations, namely one good and two bad, and the algorithm of the MaskSplitter framework determines the type of mask representations and the number of true cow objects to be predicted; loss functions of the Euclidean and pixel-wise Sigmoid; and a set of Fully-Connected Layers (FCLs) and CNNs, one for every prediction's type.

Based on the characteristics of the MaskSplitter iterated above, the computational time of the model is slow, and the colorimetric information of the masked object is shallow due to the absence of an RPN that serves as a network for scanning the presence of an object in the image generated by feature maps. The proposed framework mitigates the problem of IoU mismatch and false positive, and robust training distribution that translates to minimal over-fitting and its effects commonly found in the Mask R-CNN algorithm, an extension of the Faster R-CNN algorithm without any modification but masks addition. Based on Table V, the mAP for the proposed framework is 0.93, signifying its accuracy and time efficiency in object detection and segmentation. By this, the problem of unnecessary training of layers and image color conversion is overcome. Fig. 7 shows the chart for comparing mAP values across the models.

TABLE V
COMPARISON OF THE PROPOSED FRAMEWORK WITH STATE-OF-THE-ART MODELS

Model	Model network	mAP
Mask R-CNN	ResNet101	0.92
MaskSplitter	VGG16	0.71
FCIS [43]	ResNet101-C5-dilated	0.56
Faster R-CNN	ResNet101-FPN	0.90
YOLO v2	DarkNet19	0.91
Mask SSD [44]	ResNet101-FPN-B6	0.82
DeepMask [45]	VGGNet	0.53
SharpMask [46]	VGGNet	0.82
MNC [47]	ResNet101-C4	0.42
Enhanced Mask R-CNN (Proposed)	ResNet101	**0.93**

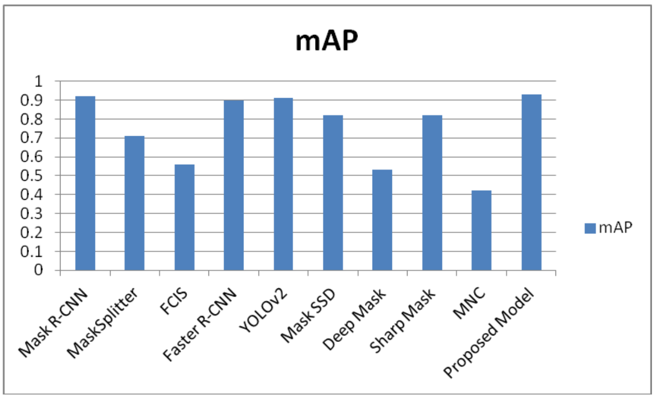

Fig. 7. Chart showing mAP values across the models.

V.CONCLUSION

An enhanced deep learning framework has been proposed in this study to mitigate the limitations of Mask R-CNN. Mask R-CNN instance segmentation involves so much computational time and image patches due to the inefficiency of ResNet101 that was used in constructing its backbone. One notable problem of ResNet101 is the unnecessary delay in converting an image from 1024×1024×3 (RGB) to 32×32×2048 feature maps as input to the stages that follow. This notable problem of ResNet101 prompted the proposed strategies that involve the use of GCFD.

The GCFD is produced by dividing the color of the image into each specific color channel, namely red (r), green (g), and blue (b). The computation of each of these channels produces three sets of descriptors by combining two descriptors computed from both orthogonal projection and parallel projection in two dimensions Fast Fourier Transform (FFT).

For the proposed framework to attain the above achievement, the pre-trained weights of the MS COCO cow dataset were employed in the Own cow dataset after the dataset frames have been annotated using the LabelMe software. The proposed framework architecture performs better when compared to the state-of-the-art segmentation models. Future work includes embedding tracking algorithms into the enhanced Mask R-CNN for multiple cows monitoring in real-time.

REFERENCES

[1] R. Bello, A. Z. Talib, and A. S. A. Mohamed, “Deep Learning-Based Architectures for Recognition of Cow Using Cow Nose Image Pattern,” *Gazi University Journal of Science,* vol. 33, no. 3, pp. 831-844, 2020.

[2] G. Hermans, A. Ipema, J. Stefanowska, and J. Metz, “The Effect of Two Traffic Situations on the Behaviour and Performance of Cows In An Automatic Milking System,” *Journal of Dairy Science,* vol. 86, no. 6, pp. 1997-2004, 2003.

[3] G. Miguel-Pacheco, J. Kaler, J. Remnant, L. Cheyne, C. Abbort, A. French, T. Pridmore, and J. Huxley, “Behavioural Changes in Dairy Cows With Lameness In An Automatic Milking System,” *Appl Animal Behav Sci,* vol. 150, pp. 1–8, 2014.

[4] C. Rutten, A. Velthuis, W. Steeneveld, and H. Hogeveen, “Invited Review: Sensors to Support Health Management on Dairy Farms,” *Journal of Dairy Science*, vol. 96, no. 4, pp. 1928-1952, 2013.

[5] S. M. C. Porto, C. Arcidiacono, U. Anguzza, and G. Cascone, “The Automatic Detection of Dairy Cow Feeding and Standing Behaviours in Free-Stall Barns By A Computer Vision-Based System,” *Biosystems Engineering,* vol. 133, pp. 46–55, 2015.

[6] T. Grandin, “*Improving Animal Welfare: A Practical Approach* (2nd ed),” Oxfordshire, UK, CABI, 2015.

[7] R. W. Bello, D. A. Olubummo, Z. Seiyaboh, O. C. Enuma, A. Z. Talib, and A. S. A. Mohamed, “Cattle identification: the history of nose prints approach in brief,” *in IOP Conference Series: Earth and Environmental Science,* vol. 594, no. 012026, pp.1-9, 2020.

[8] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, “2d Human Pose Estimation: New Benchmark and State of the Art Analysis,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2014, pp. 3686-3693.

[9] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, ..., and C. L. Zitnick, “Microsoft COCO: Common Objects In Context,” *in European Conference on Computer Vision,* Springer, Cham, 2014, pp. 740-755.

[10] A. Toshev and C. Szegedy, “Deep Pose: Human Pose Estimation Via Deep Neural Networks,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2014, pp. 1653-1660.

[11] A. Newell, K. Yang, and J. Deng, “Stacked Hourglass Networks for Human Pose Estimation,” *in European Conference on Computer Vision,* Springer, Cham, 2016, pp. 483-499.

[12] E. Insafutdinov, M. Andriluka, L. Pishchulin, S. Tang, E. Levinkov, B. Andres, and B. Schiele, “Arttrack: Articulated Multi-Person Tracking in the Wild,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2017, pp. 6457-6465.

[13] Z. Cao, G. Hidalgo, T. Simon, S. E. Wei, and Y. Sheikh, “OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields,” *arXiv preprint arXiv:* 1812.08008, pp. 1-14, 2018.

[14] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, “Realtime Multi-Person 2d Pose Estimation Using Part Affinity Fields,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2017, pp. 7291-7299.

[15] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. V. Gehler, and B. Schiele, “Deepcut: Joint Subset Partition and Labeling for Multi-Person Pose Estimation,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2016, pp. 4929-4937.

[16] D. Ionescu, B. Ionescu, C. Gadea, and S. Islam, “An Intelligent Gesture Interface for Controlling TV Sets and Set-Top Boxes,” *in 6th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI),* 2011, pp. 159–164.

[17] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, “Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks,” *arXiv preprint arXiv*: 1312.6229, pp. 1-16, 2013.

[18] R. Girshick, “Fast R-CNN,” *in Proceedings of the IEEE International Conference on Computer Vision,* 2015, pp. 1440-1448.

[19] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection With Region Proposal Networks,” *in Advances in Neural Information Processing Systems,* 2015, pp. 91-99.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2016, pp. 779-788.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, “SSD: Single Shot Multibox Detector,” *in European Conference on Computer Vision,* Springer, Cham, 2016, pp. 21-37.

[22] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *in Proceedings of the IEEE International Conference on Computer Vision,*2017, pp. 2961-2969.

[23] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” *in European Conference on Computer Vision,* Springer, Cham, 2014, pp. 818-833.

[24] A. W. Rosyadi, and N. Suciati, “Image segmentation using transition region and k-means clustering,” *IAENG International Journal of Computer Science*, vol. 47, no. 1, pp. 47-55, 2020.

[25] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning For Image Recognition,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, ..., and A. Rabinovich, “Going Deeper With Convolutions,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1-9.

[27] R. -W. Bello, A. Z. H. Talib, and A. S. A. B. Mohamed, “Deep belief network approach for recognition of cow using cow nose image pattern,” *Walailak Journal of Science and Technology*, vol 18, no. 5, pp. 1-14, 2021.

[28] R. W. Bello, A. Z. Talib, A. S. A. Mohamed, D. A. Olubummo, and F. N. Otobo, “Image-based Individual Cow Recognition using Body Patterns,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 3, pp. 92-98, 2020.

[29] J. Long, E. Shelhamer, and T. Darrell, “Fully Convolutional Networks For Semantic Segmentation,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2015, pp. 3431–3440.

[30] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “Yolact: Real-Time Instance Segmentation,” *in Proceedings of the IEEE International Conference on Computer Vision,* 2019, pp. 9157-9166.

[31] Z. Cai and N. Vasconcelos, “Cascade R-CNN: High-Quality Object Detection and Instance Segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence, arXiv: 1906.09756v1*, pp. 1-14, 2019.

[32] R. -W. Bello, A. S. A. Mohamed, and A. Z. Talib, "Contour Extraction of Individual Cattle From an Image Using Enhanced Mask R-CNN Instance Segmentation Method," *in IEEE Access*, vol. 9, pp. 56984-57000, 2021.

[33] B. Xu, W. Wang, G. Falzon, P. Kwan, L. Guo, G. Chen, ..., and D. Schneider, “Automated Cattle Counting Using Mask R-CNN in Quadcopter Vision System,” *Computers and Electronics in Agriculture*, vol. 171, pp. 1-12, 2020.

[34] Y. Qiao, M. Truman, and S. Sukkarieh, “Cattle Segmentation and Contour Extraction Based on Mask R-CNN For Precision Livestock Farming,” *Computers and Electronics in Agriculture*, vol. 165, pp. 1-9, 2019.

[35] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, “LabelMe: A Database and Web-Based Tool For Image Annotation,” *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 157-173, 2008.

[36] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, “Imagenet: A Large-Scale Hierarchical Image Database,” *in IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248-255.

[37] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) Challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303-338, 2010.

[38] J. Mennesson, C. Saint-Jean, and L. Mascarilla, “Color Object Recognition Based on a Clifford Fourier Transform,” *in Guide to Geometric Algebra in Practice*, Springer, London, 2011, pp. 175-191.

[39] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, ..., and M. Kudlur, “Tensorflow: A System for Large-Scale Machine Learning,” *in 12th {USENIX} Symposium on Operating Systems Design and Implementation*, 2016, pp. 265-283.

[40] Y. LeCun, L. Bottou, G.. B. Orr, and K. R. Müller, “Efficient Backprop, Neural Networks: Tricks of the Trade,” *Lecture Notes in Computer Sciences*, vol. 1524, pp. 5–50, 1998.

[41] A. Ter-Sarkisov, R. Ross, J. Kelleher, B. Earley, and M. Keane, “Beef Cattle Instance Segmentation Using Fully Convolutional Neural Network,” *arXiv preprint arXiv*: 1807.01972, pp. 1-11, 2018.

[42] M. Kristan, J. Matas, A. Leonardis, T. Vojíř, R. Pflugfelder, G. Fernandez, ..., and L. Čehovin, “A Novel Performance Evaluation Methodology for Single-Target Trackers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2137-2155, 2016.

[43] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, “Fully convolutional instance-aware semantic segmentation,” *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2359-2367.

[44] H. Zhang, Y. Tian, K. Wang, W. Zhang, and F. Y. Wang, “Mask SSD: An effective single-stage approach to object instance segmentation,” *IEEE Transactions on Image Processing, vol.* 29, no. 1, pp. 2078-2093, 2019.

[45] P. O. Pinheiro, R. Collobert, and P. Dollár, “Learning to segment object candidates,” *in Advances in Neural Information Processing Systems*, 2015, pp. 1990-1998.

[46] P. O. Pinheiro, T. Y. Lin, R. Collobert, and P. Dollár, “Learning to refine object segments,” *in European Conference on Computer Vision*, 2016, pp. 75-91.

[47] J. Dai, K. He, and J. Sun, “Instance-aware semantic segmentation via multi-task network cascades,” *in IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3150–3158.