Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 05, No. 03, December 2019.

DEEP LEARNING-BASED RACING BIB NUMBER DETECTION AND RECOGNITION

Yan Chiew Wong[1], Li Jia Choi[1], Ranjit Singh Sarban Singh[1], Haoyu Zhang[2] and A. R. Syafeeza[1]

(Received: 10-Jul.-2019, Revised: 27-Aug.-2019, Accepted: 28-Aug.-2019)

ABSTRACT

Healthy lifestyle trends are getting more prominent around the world. There are numerous numbers of marathon running race events that have been held and inspired interest among peoples of different ages, genders and countries. Such diversified truths increase more difficulties to comprehending large numbers of marathon images, since such process is often done manually. Therefore, a new approach for racing bib number (RBN) localization and recognition for marathon running races using deep learning is proposed in this paper. Previously, all RBN application systems have been developed by using image processing techniques only, which limits the performance achieved. There are two phases in the proposed system that are phase 1: RBN detection and phase 2: RBN recognition. In phase 1, You Only Look Once version 3 (YOLOv3) which consists of a single convolutional network is used to predict the runner and RBN by multiple bounding boxes and class probabilities of those boxes.YOLOv3 is a new classifier network that outperforms other state-of-art networks. In phase 2, Convolutional Recurrent Neural Network (CRNN) is used to generate a label sequence for each input image and then select the label sequence that has the highest probability. CRNN can be straight trained from sequence labels such as words without any annotation of characters. Therefore, CRNN recognizes the contents of RBN detected. The experimental results based on mean average precision (mAP) and edit distance have been analyzed. The developed system is suitable for marathon or distance running race events and automates the localization and recognition of racers, thereby increasing efficiency in event control and monitoring as well as post-processing the event data.

KEYWORDS

Racing Bib Number, You Only Look Once Version 3, Convolutional Recurrent Neural Network.

1. INTRODUCTION

Over the past decades, great popularization of intelligence devices and rapid development of technology have brought forth enormous new outcomes and services that have prompted the huge demand of practical computer vision technologies. As opposed to the scanning of documents, natural scene text localization [1] and recognition contribute a method to directly access and utilize the textual data in the wild, which is apparently the most pressing technology. Nowadays, healthy lifestyle trends are more prominent around the world. Such new trends help organize running activities in order to inspire the awareness of the public of the importance of health. Large numbers of marathon running races in different formats for different situations have been held and inspired interest among people of different ages, genders and countries. Such diversified truths increase more difficulties to comprehending marathon video or images, since such process is often done manually, which yields a process made difficult by the sheer number of available photos. There are multiple methods for text localization and recognition in natural scene images and videos; for instance: parking, border control and analysis of traffic flow. However, these approaches are not good enough to obtain accurate and precise results for RBN localization in marathon images, because they usually rely on characteristics of rich texts, but not numerals [2]. The runners or participants have a single racing bib number (RBN) to indicate themselves and this RBN is usually presented on heterogeneous backgrounds and different materials. Such tag number is generally printed on a cardboard tag and pinned onto T-shirts of different colours and at different parts of each competitor's body during the marathon race. There are some problems faced in the application of RBN detection and recognition. Firstly, competitors run

1. Y. C. Wong*, L. J. Choi, S. S. S. Ranjit and A. R. Syafeeza are with Centre for Telecommunication Research & Innovation (CeTRI), Fakulti Kejuruteraan Elektronik dan Kejuruteraan Komputer (FKEKK), Universiti Teknikal Malaysia Melaka (UTeM), Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia. Email: ycwong@utem.edu.my
2. H. Zhang is with Zhejiang Ocean University, School of Naval Architecture and Mechanical -Electrical Engineering, Zhoushan City, Zhejiang, China. Email: haoyu19871202@163.com

together with different speeds and background complexity varies continuously with moving objects, sky, buildings, trees, among others. Secondly, input images captured in natural scene are usually influenced by occlusion and the loss of information or quality. Thirdly, inconsistent light intensities can influence the images captured, which results in unbalanced illumination throughout the images. Hence, achieving accurate performance for localizing and recognizing RBN in marathon races is a challenging and difficult task.

Along with that, the security of running events is the priority concern for organizers of distance running events. A tragedy incident, Boston Marathon bombing, happened on 15 April 2013. Apart from the extensive video footage and photos from local surveillance systems and witnesses' smartphones and cameras, local, state and federal law enforcement organizations failed to indicate the suspects of this incident by using only face detection technology. As a result, the proposed system in this project can assists in studying and comprehending such images to extract the desired information regardless of age, gender and country. Furthermore, the proposed system of RBN detection and recognition has the capability to assist gait analysis. Gait analysis is the study of motion figures of lower limbs, such as weight of individual, footwear, length of limbs and postures integrated with motion. It can also be employed as a biometric calculation to recognize known individuals and discover unknown subjects. Gait analysis is widely implemented to recognize the performances of athletes in sport training or events such as running and swimming, in order to ensure that the athletes have worn the correct footwear and are performing foot movement that won't result in injury to them. Runner detection included in this proposed system will certainly provide the desired information needed for gait analysis despite of RBN recognition only.

There are many online resources available selling photos of individuals, since runners may want to purchase their photos captured during marathon events as personal memories. Generally, runners shall find and purchase their respective photos by using their RBN. However, there are thousands of images captured by organizers and photographers attending marathon running races, which results in high time consumption and patience needed in sorting such images according to the RBN of each runner. As a result, the proposed system is able to overcome this problem by localizing and recognizing RBN automatically, hence increasing the efficiency in sorting marathon images.

There are a few previous research studied on RBN recognition using image processing methods. Figure 1 is the method proposed by P. Shivakumara using SVM and multiple image processing methods based on 200 plus images [3]. It illustrates the inability of scene text detection methods for RBN detection. Figure 1(i) shows an input image captured from a running race. Figures 1(ii) and (iii) show RBN localization with and without detecting torso. A lot of false positives are detected by text detection method without torso detection, while the desired output is achieved by including torso detection before text detection. Figure 2 shows the result of binarization and the result of RBN recognition with and without torso detection. It can be observed that incorrect binarization and recognition results are obtained without torso detection. Therefore, the accuracy and precision of RBN rely significantly on the text detection methods. P. Shivakumara method [3] combines torso and text detection methods. It is not related to runner movement directions and does not require face information as the method proposed by Ami et al. [4] who used the stroke width transform (SWT) method [5]-[6] to extract characters in input images, where the characters with similar stroke width are grouped together for producing text region. Then, face detection method is applied in order to detect the face of the runner. At the end, the detected torso is used to indicate and recognize the true RBN by using Tesseract OCR engine [7]. The limitation of such method is tremendously large numbers of runners running with different speeds during running races such as marathon races and hence it is very difficult to localize and detect each runner's face. N. Boonsim [8] applied edge-based technique [9]-[10] to extract edges of an input image captured during a marathon race and morphological operations in mathematical form were used to combine the edges in order to generate RBN area and fade other areas. Runner detection method is used to detect the face of the runner first and is then extended to the torso of runner. The detected runner is used to indicate the position of RBN. The verification process applies image intersection between the candidate region image and the runner's body image. After that, image contrast enhancement is employed to enhance the contrast of the image. Local contrast improvement method [9] is employed to refine local contrast instead of global contrast due to that it has the capability to solved images with complicated backgrounds. Text localization and recognition methods alone have failed to obtain high performance for RBN detection and recognition because of background variations and uncontrolled issues. Inspired by these observations, a system which uses the method of runner detection to reduce background complexity and perform RBN detection and recognition based on deep learning is proposed in this work.

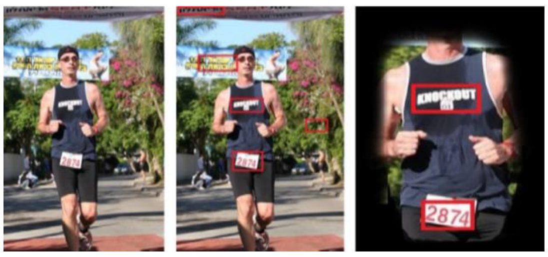


Figure 1. Scene text detection method for RBN detection [3] (i) images of running races (ii) inability of scene text detection method for RBN detection without detecting torso and (iii) RBN localization after torso detection.

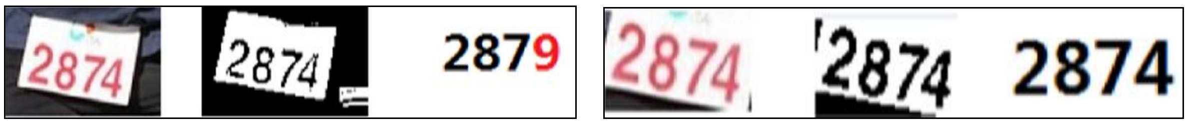
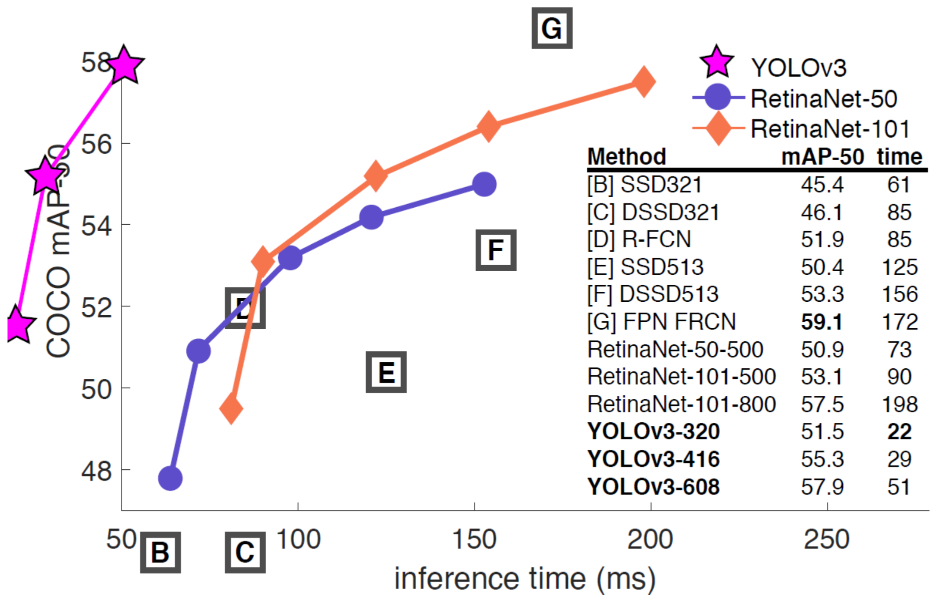


Figure 2. Binarization and recognition results for input images (i) without torso detection (ii) with torso detection [3].

In this work, an artificial intelligence cascade network, which can automatically detect and recognize RBN during marathon races, will be implemented on a graphic processing unit (GPU) by using deep learning. This is the first RBN recognition system implemented by using deep learning. Cascaded network topology, which consists of an object detection algorithm: You Only Look Once v3 (YOLOv3) [12]-[13] and an object-based sequence recognition algorithm: Convolutional Recurrent Neural Network (CRNN), will be developed. YOLOv3 is a pre-processing step that transforms an input image quickly into a sequence of image features for sequence-like object detection through CRNN. YOLOv3 operates dramatically faster than other recent detection methods. As shown in Figure 3, at 320 x 320, YOLOv3 runs in 22ms at 28.2mAP as accurate as Single Shot Detection (SSD), but three times faster.



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

Figure 3. YOLOv3 runs significantly faster than other detection methods with comparable performance [13].

The critical characteristic of bib recognition is that bib sizes may change randomly. As a result, the most famous deep models, such as Deep Convolutional Neural Network (DCNN) [16], are not suitable to apply directly to object-based sequence recognition after the target text has been detected by using a text detector [17]. This is due to that DCNN models generally operate on inputs and outputs with certain dimensions and hence such methods are unable to generate an alterable-length label sequence. Recurrent Neural Network (RNN) is particularly used for recognizing sequences. The main benefit of RNN is that it does not require the location of each component in a sequence object image in both phases of training and testing. CRNN is a combination network of DCNN and RNN. For sequence-like objects, CRNN can be directly trained from sequence labels such as words without any annotation of characters and learning informative data straightly from input images which do not need any hand-craft features or pre-processing steps such as binarization, segmentation and character localization as needed in conventional image processing methods proposed by previous research work.

YOLOv3 will be used for detecting the runner and bib number by multiple bounding boxes. CRNN will be used for recognizing the bib number through generating a label sequence for each input image and then selecting the label sequence that has the highest probability. The work follows with analyzing and enhancing the performance of the system in order to achieve high accuracy and precision. The text language is limited to the representative tag number of the runner only, where the font size and type of the text dataset typically depend on the resources of such dataset. Name of logo and name of runner are not considered in this work.

2. METHODOLOGY

2.1 Phase 1- YOLO v3

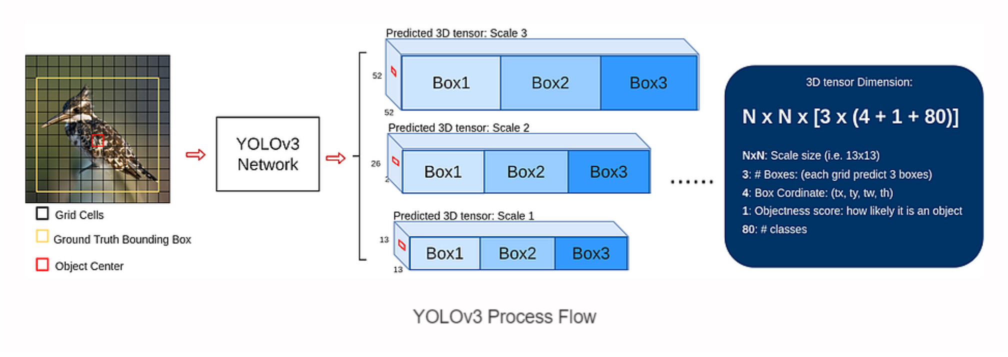
YOLO [12] uses single regression to detect the target object directly from image pixels by predicting multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and straight away optimizes detection performance. YOLOv3 [13] is the third object localization algorithm in the family. There are no fully-connected layers or pooling layers in the network architecture of YOLOv3. Therefore, YOLOv3 has the capability to handle images with variable size.

In this work, the first dataset used to train YOLOv3 has the total amount of 4096 images consisting of 1070 pixels as width and 1600 pixels as height in PNG format and is created by collecting images one by one from an online database [11]. Such images are captured at places near to the finish line during Delaware marathon running race in 2018. Then, a graphical image annotation tool is used in a process named as labelimg. Labelimg tool is a graphical image annotation tool. It is written in Python language and utilizes qt library for its graphical interface. After installing its dependencies, the tool is run to perform the image annotation for the 4096 images. Every image is provided with ground truth values which indicate the detail annotation and temporal localization of each of the target objects that are: runner, bib and number. Therefore, a text file will be generated after saving each of the images in YOLO format. Each row in the text file represents a single bounding box in an input image. The format of the bounding box is shown in Table 1.

Table 1. Format of the bounding box in YOLO format.

<object-class-id><center-x><center-y> <width> <height>	**Object-class-id** is a parameter which represents the class index of an object. It ranges from zero to (number of classes – 1).
x: x-coordinate (in pixels) of the center of the bounding box	**<center-x>** and **<center y>** are the respective coordinates of the center of the bounding box. Each of the values have been normalized by the width and height of the input image.
y: y-coordinate (in pixels) of the center of the bounding box	
w: width (in pixels) of the bounding box	**<width>** and **<height>** represent the normalized width and height of bounding boxes as follows: <center-x> = $\frac{x}{W}$, <center-y> = $\frac{y}{H}$,<width> = $\frac{w}{W}$, <height> = $\frac{h}{H}$
h: height (in pixels) of the bounding box	
W: width (in pixels) of the whole image	

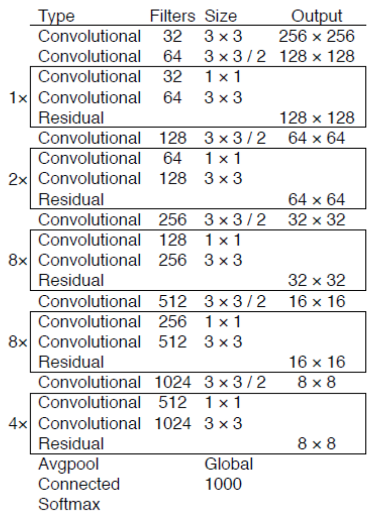
During training, YOLOv3 is applied with input images to predict 3D tensors that act as the last feature map corresponding to three scales that are regions 82, 94 and 106, as shown in Figure 4. Such three scales are constructed to localize target objects with different sizes. For instance, a scale of 25 x 25 means that an input image will be split into 25 x 25 grid cells. Every single grid cell relates to a 1 x 1 x 255 voxel inside a 3D tensor. 255 is calculated from formula of 3 x (4+1+80). By referring to Figure 4(iii), the values in a 3D tensor are shown, which are composed of confidence score, class confidence and bounding box coordinates.


Grid Cells
Ground Truth Bounding Box
Object Center
YOLOv3 Network
Predicted 3D tensor: Scale 3
52
52
Box1
Box2
Box3
Predicted 3D tensor: Scale 2
26
Box1
Box2
Box3
Predicted 3D tensor: Scale 1
13
13
Box1
Box2
Box3
......
3D tensor Dimension:
N x N x [3 x (4 + 1 + 80)]
NxN: Scale size (i.e. 13x13)
3: # Boxes: (each grid predict 3 boxes)
4: Box Cordinate: (tx, ty, tw, th)
1: Objectness score: how likely it is an object
80: # classes
YOLOv3 Process Flow

(i) (ii) (iii)

Figure 4. Process flow of YOLOv3 [12].

In phase 1, runner, racing bib and number are detected by using YOLOv3. The overall process of fetching the labeled images to YOLOv3 model is associated with using OpenCV library [14]. By applying the OpenCV library, all the input images are scaled to the same size. After resizing the input images, such images will be fed to YOLOv3. The network architecture of YOLOv3 is separated into several modules, as shown in Figure 5. First layer is the total of 32 convolutional layers, where the input images have been supplied. These convolutional layers are implemented to extract features from input images. Second layer consists of the residual layers. Such layers are proposed to vary the training process of the deep neural network from layer-by-layer training into phase-by-stage training. Therefore, the deep neural network is separated into few segments in order to realize the problem of gradient explosion or gradient dispersion of the network. Third layer is Darknet-53, which ranges from the 0th layer until the 74th layer. There are 53 convolutional layers and the remaining are the residual layers. Darknet-53 acts as a key component of the network architecture to extract features and implements a series of 3 x 3 and 1 x 1 convolutional layers. Last layer is the feature interaction layer of the YOLO network. It is separated into three scales that are regions 82, 94 and 106 in the feature pyramid network. Local feature interaction is implemented by convolutional kernel layers known as fully-connected layers in each region. It is used to obtain local feature interaction among feature maps and hence accomplish regression and classification.



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
	Convolutional	32	1 × 1	
1×	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
	Convolutional	64	1 × 1	
2×	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
	Convolutional	128	1 × 1	
8×	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
	Convolutional	256	1 × 1	
8×	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
	Convolutional	512	1 × 1	
4×	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 5. Network structure of YOLO v3 [13].

2.2 Phase 2- CRNN

RBN text is recognized by using CRNN model. CRNN [15] is the combination network of DCNN [16] and RNN [18]. For sequence-like objects, CRNN is composed of several unique advantages, such as straight learning from sequence labels, image data for binarization, segmentation or component localization without the need of hand-craft features or pre-processing steps. It also possesses the same characteristic of RNN that makes it able to generate a sequence of labels. In spite of that, CRNN is unlimited to the sizes of sequence-like objects, but is only limited to the demand of height normalization in both phases of training and testing. Besides, it obtains excellent results on scene text recognition compared to other prior arts [17]. Lastly, it has low memory assumption by consisting of lesser parameters compared with a standard DCNN model.

Second dataset used to train for CRNN is composed of 100,000 images that contain digit numbers only in grey scale, as shown in Figure 6. These images are generated by a text generator python script. Types of font that are similar to the images of the first dataset are searched and downloaded from an online database and the path of font types is specified in the script. Types of font used to generate the second dataset include Identikal Sans Bold, Contax Bold, Roadway and Sofia Pro Semibold, and so on. Only one type of font is utilized to generate texts at one time while keeping switching to another font type to generate the dataset until it accumulates to a total of 100,000 images. In the text generator script, only digit numbers are specified and thus images composed of only digit numbers will be generated in random arrangement. Gaussian blur and rotate functions are applied in the script and hence some of the images are in the condition of blurriness or rotate slightly to assimilate practical real-life conditions, such as insufficient light source and indirectly facing the camera.

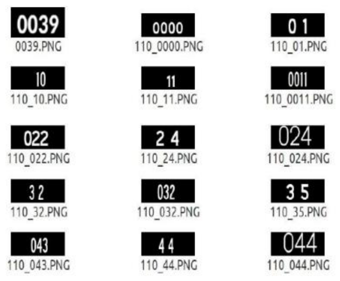


Figure 6. Second dataset generated.

According to Figure 7, the network architecture of CRNN is composed of three components which are from bottom to top convolutional layers, recurrent layers and a transcription layer. At the most bottom of CRNN, the convolutional layers automatically extract a sequential feature from each input image, while on top of the convolutional network, a recurrent network is constructed for making prediction for each frame of the feature sequence which is outputted from the convolutional layers. The transcription layer at the top of CRNN is utilized to translate the per-frame predictions by the recurrent layers into a label sequence.

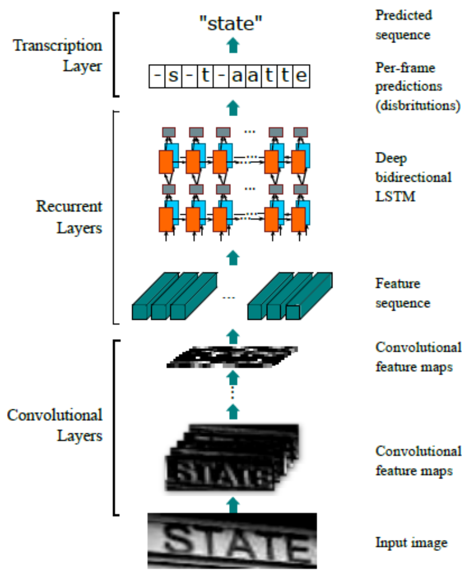


Figure 7. Network structure of CRNN [14].

Before starting training for the CRNN model, it is essential to convert the second dataset composed of 100,000 images into Lightning Memory-Mapped Database (LMDB) format. LMDB is a software library that supplies an embedded transactional database in key-value store format. Key-value store is considered as data storage built for saving, extracting and manipulating associative arrays. LMDB employs memory-mapped files and hence enhances the performance of input and output. A memory-mapped file is a division of virtual memory that has been appointed an undeviating byte-for-byte correlation with few sections of file or resource. As a result, LMDB is operating well when dealing with very large datasets. After separating the second dataset into training subset and testing subset, both subsets are converted into LMDB format. This will generate two files for respective training and testing subsets that are data.mdb and lock.mdb with different sizes.

There are few parameters needed to be adjusted before starting to train. The paths of training and testing images in LMDB format are specified and alphabet trained is set to ‘0123456789’ consisting of 10 classes. CUDA acts as a parallel computing platform and programming model that utilizes GPU for general purpose computing is enabled during training of CRNN model. Besides, several parameters utilized in the training algorithm are unchanged. These are batch size, number of GPU, learning rate, number of data loading workers, width and height of input images. Such parameters are unvaried during the training process while the number of epoch is changeable until the recognition output achieves correct results.

2.3 Cascading Both Models

After finishing collecting, annotating and generating the two datasets, all the images will then be synchronized and inserted into the proposed networks separately to be used as training and testing subsets. Training subset is utilized for learning so as to satisfy the parameters of the classifier. Input images have their respective ground truth files for supervised learning. Testing subset is utilized to evaluate the performance of a fully-trained classifier. No same image should be used as part for both of training and testing subsets. Both datasets have been split in the ratio of 7:3 as training subset to testing subset as shown in Table 2.

Table 2. Training and testing subsets.

Dataset	Training Subset	Testing Subset	Total
1 (RBN)	2868	1228	4,096
2 (Numbers)	700,000	300,000	100,000

The two pre-trained models trained from the two phases that are YOLOv3 for runners and RBN detection and CRNN for RBN recognition are combined together form a cascade network aimed for RBN detection and recognition based on deep learning. Model files, such as classes, model configuration and weight files for YOLOv3 and weight file of CRNN, are specified in a python script and will be parsed by using command line arguments. At the beginning in the script, the demanded packages that include OpenCV, NumPy, Pytorch and utils are imported. Parameters of confidence threshold, non-maximum suppression threshold, width and height of input image are adjusted. Confidence threshold and non-maximum suppression threshold are set to 0.5 and 0.4, respectively. Predicted bounding boxes consist of values lower than that will be discarded. When an input image is parsed to the cascade network, YOLOv3 model will be loading and obtaining the name of classes during training and drawing the predicted bounding box in rectangular form. Hence, confidence scores of each of the predicted bounding boxes are obtained and displayed at the tops of the boxes. Moreover, the predicted bounding boxes with classes of ‘number’ will be cropped by using OpenCV and parsed to the CRNN model. CRNN will be using CUDA to perform the RBN recognition process.

According to Figure 8, the architecture pipeline or network is consists of two phases: runner, racing bib and number detection in phase 1 and number recognition in phase 2. Therefore, two different networks are utilized which are YOLOv3 and CRNN to localize and recognize RBN. During phase 1, YOLOv3 consists of a single convolutional network used to predict RBN by multiple bounding boxes and class probabilities of boxes since there may be more than one RBN in one image. Since the target object detected is only the RBN, runner and racing bib detection is proceeded to avoid that YOLOv3 detects the regions out of the runner, such as background variation when the runners are running. Therefore, the detected RBN can be parsed into CRNN to undergo the process of recognition. For the second phase, CRNN is used to output a label sequence for each input image and then select the label sequence that has the highest probabilities. As a result, CRNN will output the contents of RBN detected.

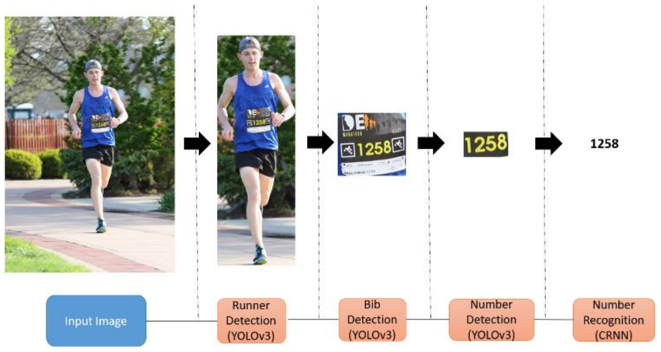

1258
1258
1258
Input Image
Runner Detection (YOLOv3)
Bib Detection (YOLOv3)
Number Detection (YOLOv3)
Number Recognition (CRNN)

Figure 8. Method selection.

2.4 Prediction

After the pre-trained models for the two phases are combined together. The validation and testing subsets will be utilized to predict the RBN localization and recognition. The predicted process is evaluated with combined pre-trained models to enhance the performance of the RBN detection and recognition system. By applying the testing subset, the cascade system has the capability of comparing the predicted bounding box with the actual bounding box for each input image.

2.5 Post-processing

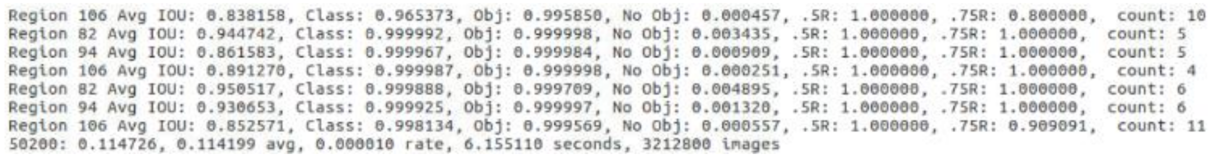
Post-processing is essential for YOLOv3 model only in order to localize RBN from testing images. The predicted bounding boxes from previous step are post-processed by using the method of non-maximum suppression which had been built in the deep neural network implementation of OpenCV. Non-maximum suppression is an important step of post-processing in many computer vision applications, especially for object detection. It is utilized to convert a smooth response map that activates many inaccurate object window hypotheses, which means only a single predicted bounding box for each target object.

3. Results and Discussion

There are two image datasets and each of them will be employed as inputs to train YOLOv3 and CRNN separately. Hence, the actions of localization and recognition of RBN will be acquired from images captured from marathon running events. All the precision results obtained from the models are enumerated to determine the exact location of RBN. Training loss, accuracy and precision of each network will be presented and analyzed in the following sub-sections.

3.1 Training YOLOv3 and CRNN with Own Datasets

In this project, there are two different datasets that are used for RBN detection and recognition during marathon events. In the training process, the parameters utilized in the training algorithm for both neural networks are very important in order to regulate the precision of localization and recognition output. Both neural networks are trained separately with their respective datasets. For YOLOv3, there are several parameters generated during training, as shown in Figure 9. The training process will keep updating the weights of YOLOv3 model from each iteration. Regions 82, 94 and 106 represent distinct sizes of parameters predicted at three distinct scales respectively in the feature pyramid network of YOLOv3. Convolutional layer of region 82 is the greatest prediction scale that utilizes a huge mask that makes it able to detect smaller object. Convolutional layer of region 94 is the medium prediction scale that utilizes a medium mask while convolutional layer of region 106 is the smallest prediction scale for localizing larger objects. Since the batch and subdivision are set as 64 and 16 respectively in the model configuration file, the training iteration consists of 4 groups of regions 82, 94 and 106, where each group consists of 16 images during training output. As a result, the model will simply choose 64 batches of images from all training sets to be fed into the model during each iteration. All of the image batches chosen will be separated into subdivision of 4 times in order to minimize memory consumption.



```
Region 106 Avg IOU: 0.838158, Class: 0.965373, Obj: 0.995850, No Obj: 0.000457, .5R: 1.000000, .75R: 0.800000,  count: 10
Region 82 Avg IOU: 0.944742, Class: 0.999992, Obj: 0.999998, No Obj: 0.003435, .5R: 1.000000, .75R: 1.000000,  count: 5
Region 94 Avg IOU: 0.861583, Class: 0.999967, Obj: 0.999984, No Obj: 0.000909, .5R: 1.000000, .75R: 1.000000,  count: 5
Region 106 Avg IOU: 0.891270, Class: 0.999987, Obj: 0.999998, No Obj: 0.000251, .5R: 1.000000, .75R: 1.000000,  count: 4
Region 82 Avg IOU: 0.950517, Class: 0.999888, Obj: 0.999709, No Obj: 0.004895, .5R: 1.000000, .75R: 1.000000,  count: 6
Region 94 Avg IOU: 0.930653, Class: 0.999925, Obj: 0.999997, No Obj: 0.001320, .5R: 1.000000, .75R: 1.000000,  count: 6
Region 106 Avg IOU: 0.852571, Class: 0.998134, Obj: 0.999569, No Obj: 0.000557, .5R: 1.000000, .75R: 0.909091,  count: 11
50200: 0.114726, 0.114199 avg, 0.000010 rate, 6.155110 seconds, 3212800 images
```

Figure 9. Output parameters produced during training of YOLOv3.

Parameter of ‘Avg IOU’ for the respective region represents the average IOU of the image in the present subdivision. It indicates the overlap of the predicted bounding box to the ground truth of target object and the union. The closer the value to 1, the better the bounding box prediction. For instance, ‘Avg IOU: 0.838158’ produced during training output has achieved quite high accuracy. ‘Class’ indicates the correctness of object classification to respective classes, where the values are expected to approach 1. ‘Obj’ is the average of the objectness or confidence score, which also calculates the probability of that there is an object in the bounding box, while ‘No Obj’ represents the opposite case. To produce a better result of localization, the value of ‘Obj’ shall be closer to 1 and ‘No Obj’ shall be approaching but not zero. ‘.5R’ and ‘.75R’ are referred to as the ratio of true positive objects predicted by the present model to the ground truth of object in each subdivision of images. If all objects are precisely predicted, then ‘.5R’ shall be equal to 1 and ‘.75R’ shall be equal to 0. ‘.5R’ is defined as the ratio with IOU greater than 0.5, while ‘.75R’ is the ratio with IOU greater than 0.75. ‘Count’ indicates the number of training images composed of true positive objects in all present subdivision images which is 4 in our case. By referring to Figure 9, since there are 4 or 6 true positive objects, this means that such division is composed of the images that do not belong in the localized object classes. Last row indicates the batch output, as shown in Figure 7. During the training process, every 1000 epochs will generate one weight file. As the number of epochs is increasing, the frequency of updating the values of weights will also increase until a minimum training loss is obtained.

CRNN model requires Connectionist Temporal Classification (CTC) [19] to do the training. CTC is specifically functional for neural networks composed of convolutional layers (CNN) to withdraw a chain of features and recurrent layers (RNN) to distribute information from such chains [20]-[21]. It generates character-scores for each chain component considered as a matrix in an easier way. By referring to Figure 10, there are two operations that need such matrix to proceed with calculating the loss value to train the neural network during training and decode the matrix to obtain the text contents from input images during inference. CTC benefits users by applying CTC loss function to learn the text contents existing in an input image and discard both location and width of characters in the input image. Furthermore, no further post-processing of the recognized text is required.

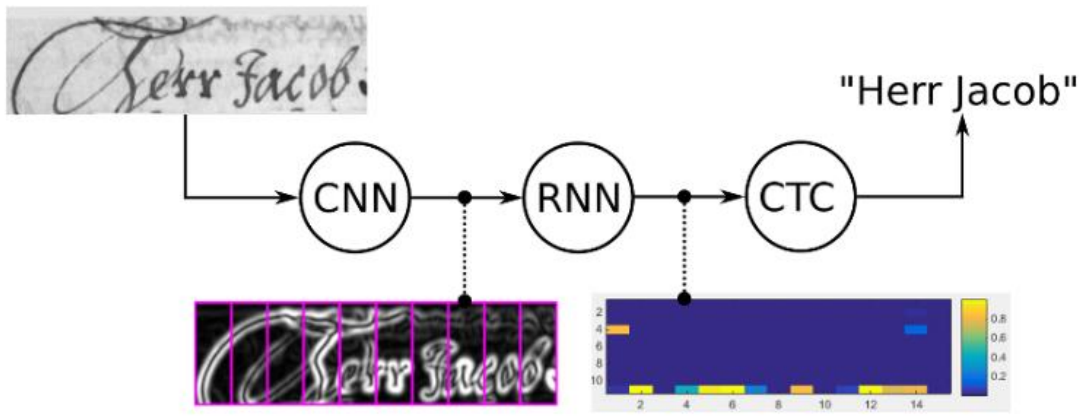
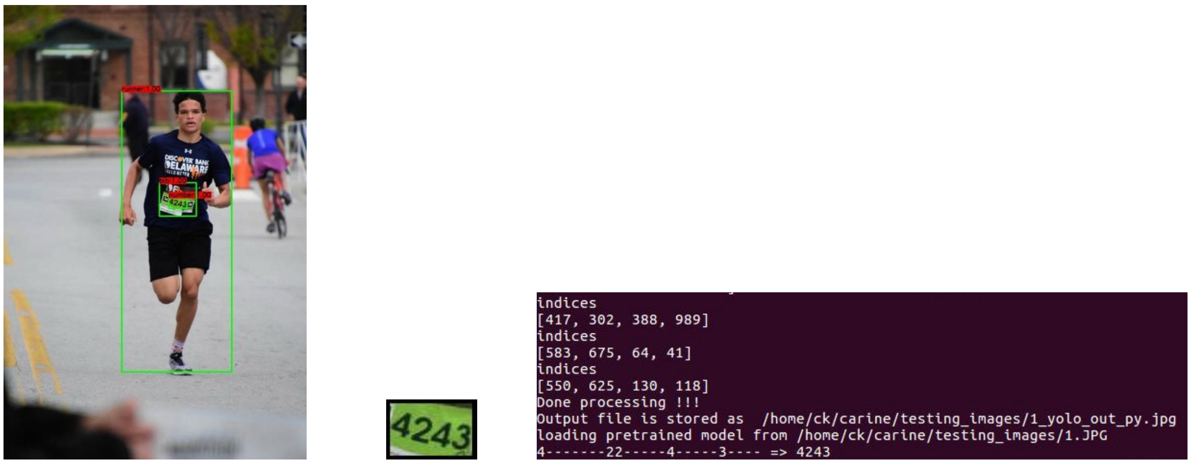


Figure 10. Process flow of neural network for handwriting recognition [15].

3.2 Result of Cascade Network

Since the targeted object is only limited to the RBN, the function of image cropping by using OpenCV is added in the script. Scripts utilized to run the models of YOLOv3 and CRNN are combined to become a cascade network. As a result, YOLOv3 will be used to detect runner, racing bib and number first, then the bounding box of the detected number will be cropped by using the crop function in OpenCV. The cropped image will be parsed to CRNN to undergo the recognition process as illustrated in Figure 11.



```
indices
[417, 302, 388, 989]
indices
[583, 675, 64, 41]
indices
[550, 625, 130, 118]
Done processing !!!
Output file is stored as  /home/ck/carine/testing_images/1_yolo_out_py.jpg
loading pretrained model from /home/ck/carine/testing_images/1.JPG
4-------22-----4-----3---- => 4243
```

(a) (b) (c)

Figure 11. Demonstration of RBN detection and recognition cascade network: (a) Objects detected using YOLOv3 model, (b) Bounding box of RBN cropped image (c) Recognition process using CRNN model.

3.3 Analyzing Predicted Result from Cascade Network

In this work, the prediction accuracy scores are acquired and evaluated for each of the classes localized by YOLOv3 model in the first phase. The metric utilized to implement the results of the temporal localization is named as mean Average Precision (mAP) for object classes, as shown in Table 3. Mean Average Precision (mAP) is usually employed to measure the accuracy of object detectors in terms of how well the predicted bounding box trained by the object detection model overlaps with the ground truth bounding box. The mAP metric is the product of recall and precision of the detected bounding boxes and its value ranges from 0 to 1. Precision determines how accurate the prediction is, while recall determines how well we find all the positives, as shown in Equations (2) and (3).

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

where TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative.

Table 3. Mean average precision before and after fine-tuning with extra training images.

Dataset	Training Subset	Testing Subset	Total	Average Precision	
Training from scratch	2868	1228	4096	Runner	0.8958
				Bib	0.7164
				Number	0.6224
				mAP	0.7449
Fine-tuning	700	300	1000	Runner	0.9786
				Bib	0.8727
				Number	0.7032
				mAP	0.8515

The higher the value of mAP, the better the result. The mAP can be evaluated by calculating average precision (AP) separately for each class then divided by the average over the classes. Detection is reckoned as true positive if the mAP is above 0.5. As a result, the activity localization for each of the classes on 400 testing images is calculated by evaluating the overlap between a ground truth segment and a predicted temporal segment determined by Intersection over Union (IoU) score as shown in the section of training from scratch in Table 2 which summarizes all the prediction results from activity localization of RBN on 400 testing images by using YOLOv3 model trained from the collected dataset. We observe that the average precision of object classes of the runner is the highest, while that of object classes of the number is the lowest. This is due to several reasons, as shown in Figure 12.

First reason is the light condition, as some of the testing images are captured under insufficient light condition, because the particular marathon event is held from afternoon until evening and hence results in that the quality of images captured is not so good due to some blurriness. Secondly is that there are wrinkles in the racing bib that runners wear from images captured during approaching the finish line, since runners already ran for hours, which affects the result of RBN detection. Thirdly is that some regions of racing bibs are covered by the hands of the runner during running.

Other than mAP, there is another metric that is implemented for cascade network of YOLOv3 and CRNN models, which is the edit distance. Edit distance is a method of quantifying how dissimilar two strings are by calculating the minimum number of processes needed to convert one string into the other. There are three processes that can be employed to convert the string, which are: insert a character, remove a character and substitute a character. For instance, the edit distance of “2222” and “2221” is 1. Edit distance is generally employed in the field of natural language processing. For instance, automatic spelling correction can evaluate the candidate corrections for a misspelled word. This is done by choosing a word from a dictionary that has a minimum distance to the word by inquiry.

By referring to Figure 12, the graph illustrates the result of RBN detection and recognition of cascaded network before and after the improvement by fine-tuning with 1000 images. Edit distance is calculated for 400 testing images randomly selected from the testing subset of the first dataset. There are 64.25% of correct localization and recognition of RBN which is identical as ground truth numbers; 26.25% achieves one-number difference; 6% achieves two number-difference, 0.0075% achieves three number-difference and 2.75% achieves four number-difference or is completely different from ground truth numbers before improvement by transfer learning. Apart from that, results of edit distance that did not achieve zero edit distance or were not exactly the same as ground truth numbers have been studied and analyzed. There is a total amount of 143 images that did not attain zero edit distance. By referring to Figure 12, the reason of inaccurate text detector has achieved the highest percentage of 71.33%. In this work, YOLOv3 is utilized to localize the RBN from input images. However, since the mean average precision of ‘number’ obtains only 0.6224, it eventually affects the result of RBN recognition. Furthermore, there are other reasons that affect the result of RBN recognition as well, such as wrinkles of racing bib, mix-up between 1 and 7 and between 3 and 9, blurriness of input images captured among others, as illustrated in Figure 9. Although the ground truth number is ‘4203’, the result predicted from CRNN is ‘14203’. This is because of that the position of RBN is slightly rotated in the input image; hence patterns including logos or sketching will be detected by YOLOv3 model. Therefore, this results in that cropped image not only consists of RBN and subsequently affects the result of recognition of CRNN model.

There are some of the images which are still inaccurately recognized due to that the average precision of class of ‘number’ has achieved the lowest value which is only 0.6224. Therefore, the pre-trained model from YOLOv3 is fine-tuned with 1000 images by using transfer learning method, as shown in the section of fine-tuning in Table 3. Such 1000 images are collected from the same source with images from the first dataset used to train YOLOv3 from scratch. Transfer learning is known as a method of machine learning, where a model is implemented for a certain function again as the starting point for such model on another function or new problem. The priorities of transfer learning include saving time needed to develop and reutilizing an already developed model to do training for development on a different task or improvement. By referring to Table 2, it presents the mean average precision (mAP) before and after fine-tuning with 1000 training images captured from marathon running races. Results of localization of target classes about previous and current mAP values are tested on the same 400 training images. It can be observed that the localization accuracy of YOLOv3 is improved after transfer learning. Figure 13 illustrates the result of RBN detection and recognition of the cascaded network composed of YOLOv3 and CRNN before and after improvement by fine-tuning with 1000 images. Although YOLOv3 still achieved the highest result, there is a difference of 26 testing images that obtained accurate RBN recognition with 0 edit distance after transfer learning. Frequencies of other factors for inaccurate RBN recognition still remain unvaried, since the condition of wrinkles of racing bib and blurriness of testing images cannot be avoided.

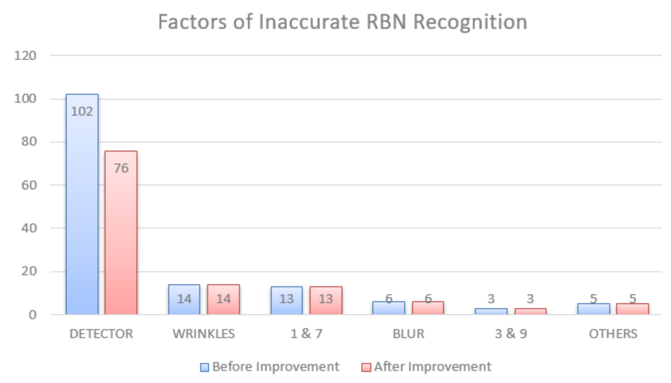


Figure 12. Factors of inaccurate RBN recognition before and after fine-tuning.

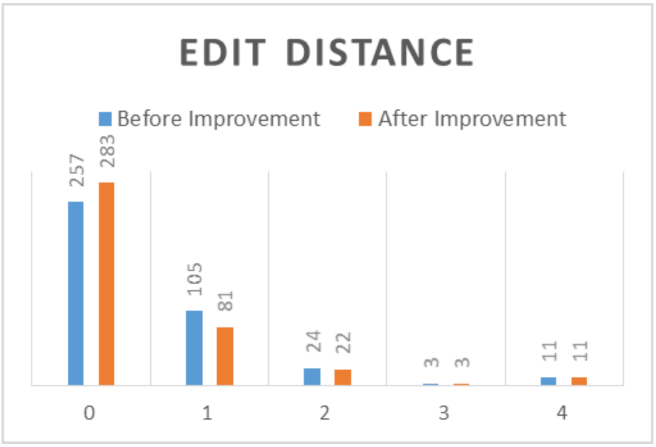


Figure 13. Edit distance before and after fine-tuning.

4. Conclusions

In running races, numerous images of runners are captured by running race organizers. This results in that the task of differentiating individual marathon images of a certain runner from all images becomes very troublesome. As a result, a deep learning-based RBN localization and recognition model for marathon running races is proposed. This is the first RBN recognition system implemented by using deep learning. Network topology implemented in this application is cascaded network composed of YOLOv3 and CRNN. YOLOv3 has been used for detecting the runner and bib number by multiple bounding boxes. CRNN is used to recognize the bib number through generating a label sequence for each input image and then selecting the label sequence that has the highest probability. Mean Average Precision (mAP) achieves 85.15% accuracy and the edit distance with exact output is 283 out of 400 after fine-tuning. The recognition could not reach 100% of accuracy because of that some RBNs in input images are in the condition of wrinkles or patterns such that sketching or logo surroundings around RBN are accidentally predicted together with RBN hence affecting the accuracy of cascaded network. Accuracy of detection has been further improved through transfer learning by fine-tuning with a validation dataset which consists of 1000 images. 9% to 21% of improvement to the accuracy has been achieved. With the demonstrated capability of automating the localization and recognition of racers, the developed system not only could better control large-scale marathon events, but also tighten the security of racing events.

ACKNOWLEDGEMENTS

The authors acknowledge the technical and financial support by Universiti Teknikal Malaysia Melaka (UTeM).

REFERENCES

[1] S. Roy, P. Shivakumara, P. Mondal, R. Raghavendra, U. Pal and T. Lu, "A New Multi-modal Technique for Bib Number/Text Detection in Natural Images," Proceedings of Pacific Rim Conference on Multimedia, pp. 483-494, 2015.

[2] K. S. Younis, "Arabic Handwritten Character Recognition Based on Deep Convolutional Neural Networks," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 3, no. 3, pp. 186-200, 2017.

[3] P. Shivakumara, R. Raghavendra, L. Qin, K. B. Raja and T. Lu, "A New Multi-modal Approach to Bib Number / Text Detection and Recognition in Marathon Images," Pattern Recognition, vol. 61, pp. 479–491, 2017.

[4] T. Basha, S. Avidan and I. Ben-Ami, "Racing Bib Number Recognition," Br. Mach. Vis. Conf. (BMVC), pp. 1–10, 2012.

[5] B. Epshtein, "Detecting Text in Natural Scenes with Stroke Width Transform," Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2963–2970, 2010.

[6] F. Su and H. Xu, "Robust Seed-based Stroke Width Transform for Text Detection in Natural Images," Proc. of the 13th Int. Conf. Doc. Anal. Recognit., pp. 916–920, 2015.

[7] R. Smith, "An Overview of the Tesseract OCR Engine," Proc. of the 9th International Conference on Document Analysis and Recognition, pp. 629-633, 2005

[8] N. Boonsim, "Racing Bib Number Localization on Complex Backgrounds," WSEAS Transactions on Systems and Control, vol. 13, pp. 226–231, 2018.

[9] Q. Ye and D. Doermann, "Text Detection and Recognition in Imagery : A Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 7, pp. 1480–1500, 2015.

[10] Y. Zheng and A. R. O. Theory, "Edge Detection Methods in Digital Image Processing," Proc. of the 5th Int. Conf. Comput. Sci. Educ., pp. 471-473, 2010.

[11] Celsius, "Delaware Running Festival 2018," [Online], Available: https://pic2go.nascent-works.com/c0ab85474c3597a8dcd1a3d95e917516, [Accessed: 15-Apr-2019].

[12] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 779-788, 2016.

[13] J. Redmon, A. Farhadi and C. Ap, "YOLOv3 : An Incremental Improvement," [Online], Available: https://arxiv.org/abs/1804.02767, 2018.

[14] OpenCV, "AI Courses by OpenCV," [Online], Available: https://opencv.org/ [Accessed: 15-Apr-2019]

[15] B. Shi, X. Bai and C. Yao, "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 11, pp. 2298–2304, 2016

[16] N. Aloysius, "A Review on Deep Convolutional Neural Networks," Proc. of Int. Conf. Commun. Signal Process., pp. 588–592, 2017.

[17] H. Li, P. Wang and C. Shen, "Towards End-to-end Text Spotting with Convolutional Recurrent Neural Networks," IEEE Int. Conf. Comput. Vis., no. 2, pp. 5248–5256, 2017.

[18] A. Graves, M. Liwicki, S. Ferna, R. Bertolami and H. Bunke, "A Novel Connectionist System for Unconstrained Handwriting Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 855–868, 2009.

[19] A. Graves and S. Fern, "Connectionist Temporal Classification : Labelling Unsegmented Sequence Data with Recurrent Neural Networks," Proceedings of the 23rd International Conference on Machine Learning, pp. 369-376, 2006.

[20] Y. C. Wong and Y. Q. Lee, "Design and Development of Deep Learning Convolutional Neural Network on a Field Programmable Gate Array," Journal of Telecommunication, Electronic and

Computer Engineering, vol. 10, no. 4, pp. 25-29, 2018.

[21] T. Vo, T. Nguyen and C. T. Le, "Race Recognition Using Deep Convolutional Neural Network," Symmetry, vol. 10, no. 11, 564, 2018.

ملخص البحث:

يتزايد الاهتمام بأنماط الحياة الصحية في جميع أنحاء العالم. وقد جرى تنظيم أعداد كبيرة من سباقات الماراثون التي استقطبت اهتمام الكثيرين من الناس من مختلف الأعمار من الذكور والإناث من بلدان مختلفة. وما من شك في أن هذا التنوع الواسع يصعب استيعاب الأعداد الضخمة من الصور في سباقات الماراثون؛ نظراً لأن هذه العملية تتم يدوياً. لذا، تقترح هذه الورقة نهجاً جديداًلكشف أرقام المتسابقين في سباقات الماراثون وتمييزها باستخدام التعلم العميق. وفي السابق، كان ذلك يتم باستخدام تقنيات معالجة الصور فقط؛ الأمر الذي يحدّ من جودة الأداء.

يتألف النظام المقترح من مرحلتين هما: كشف أرقام المتسابقين، وتمييز تلك الأرقام. في المرحلة الأولى، تستخدم شبكة التفافية (YOLOv3) مفردة لتوقُّع المتسابق ورقم المتسابق من خلال صناديق تحديد متعددة الى جانب الاحتمالات المتعلقة بصنوف تلك الصناديق. والجدير بالذكر أن تلك الشبكة هي شبكة تصنيف تفوق في أدائها مثيلاتها من الشبكات المستخدمة. أما في المرحلة الثانية، فيجري استخدام شبكة التفافية أخرى (CRNN) لتوليد تتابع من العلامات لكل صورة مُدخلة ومن ثم اختيار تتابع العلامات الذي يحظى بالاحتمال الأعلى. ويمكن تدريب تلك الشبكة مباشرة من العلامات المتتابعة مثل الكلمات غير المحتوية على حواشٍ تفسيرية. وبذلك تقوم شبكة (CRNN) بتمييز محتويات أرقام المتسابقين التي تم كشفها. من ناحية أخرى، جرى تحليل كل من متوسط الدقة ومسافة التحرير. واتضح أن النظام المطّور مناسب لأحداثٍ مثل سباقات المسافات الطويلة ومنها سباقات الماراثون؛ فهو يعمل على أتمتة عملية تحديد موقع المتسابق وتمييزه، ومن ثم فهو يزيد من فاعلية السيطرة على السباقات ومراقبتها، الى جانب تحسين المعالجة البَعدية للبيانات الخاصة بتلك الأحداث.