CISTER

Conference Paper

Activity Monitoring of Islamic Prayer (Salat) Postures using Deep Learning

Anis Koubâa*

Adel Ammar

Bilel Benjdira

Abdullatif Al Hadid

Belal Kawaf

Saleh Ali Al Yahri

Abdelrahman Babiker

Koutaiba Assaf

Mohannad Ba Ras

Activity Monitoring of Islamic Prayer (Salat) Postures using Deep Learning

Anis Koubâa*, Adel Ammar, Bilel Benjdira, Abdullatif Al Hadid, Belal Kawaf, Saleh Ali Al Yahri, Abdelrahman Babiker, Koutaiba Assaf, Mohannad Ba Ras

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: aska@isep.ipp.pt

https://www.cister-labs.pt

Abstract

In the Muslim community, the prayer (i.e. Salat)is the second pillar of Islam, and it is the most essentialand fundamental worshiping activity that believers have toperform five times a day. From a gestures 19 perspective, thereare predefined human postures that must be performed in aprecise manner. However, for several people, these postures arenot correctly performed, due to being new to Salat or evenhaving learned prayers in an incorrect manner. Furthermore, thetime spent in each posture has to be balanced. To address theseissues, we propose to develop an artificial intelligence assistiveframework that guides worshippers to evaluate the correctnessof the postures of their prayers. This paper represents the firststep to achieve this objective and addresses the problem ofthe recognition of the basic gestures of Islamic prayer usingConvolutional Neural Networks (CNN). The contribution of thispaper lies in building a dataset for the basic Salat positions,and train a YOLOv3 neural network for the recognition ofthe gestures. Experimental results demonstrate that the meanaverage precision attains 85% for a training dataset of 764 imagesof the different postures. To the best of our knowledge, this isthe first work that addresses human activity recognition of Salatusing deep learning.

Activity Monitoring of Islamic Prayer (Salat) Postures using Deep Learning

Anis Koubaa*,¶,‡, Adel Ammar*, Bilel Benjdira*, ‖, Abdullatif Al-Hadid*, Belal Kawaf*,
Saleh Ali Al-Yahri*, Abdelrahman Babiker, Koutaiba Assaf, Mohannad Ba Ras

*Robotics and Internet-of-Things Lab (RIOTU), Prince Sultan University, Riyadh, Saudi Arabia.
† CISTER Research Centre, ISEP, Polytechnic Institute of Porto, Porto, Portugal
¶Gaitech Robotics, China.
‖Research Laboratory SEICT, LR18ES44. National Engineering School of Carthage, Tunisia.

Email: (akoubaa, aammar, bbenjdira)@psu.edu.sa

Abstract—In the Muslim community, the prayer (*i.e.* Salat) is the second pillar of Islam, and it is the most essential and fundamental worshiping activity that believers have to perform five times a day. From a gestures' perspective, there are predefined human postures that must be performed in a precise manner. However, for several people, these postures are not correctly performed, due to being new to Salat or even having learned prayers in an incorrect manner. Furthermore, the time spent in each posture has to be balanced. To address these issues, we propose to develop an artificial intelligence assistive framework that guides worshippers to evaluate the correctness of the postures of their prayers. This paper represents the first step to achieve this objective and addresses the problem of the recognition of the basic gestures of Islamic prayer using Convolutional Neural Networks (CNN). The contribution of this paper lies in building a dataset for the basic Salat positions, and train a YOLOv3 neural network for the recognition of the gestures. Experimental results demonstrate that the mean average precision attains 85% for a training dataset of 764 images of the different postures. To the best of our knowledge, this is the first work that addresses human activity recognition of Salat using deep learning.

I. INTRODUCTION

Human Activity Recognition has been extensively investigated using different techniques including sensing technologies [1]–[3], computer vision [4], and more recently using deep learning [5,6]. It consists of the classification of the activity of a person from the data collected from sensors (*e.g.* accelerometer, camera, and laser scanner). The advances in human activity recognition have enabled different applications in different areas such as healthcare, sports activities, violence detection, older people monitoring, postures' recognition, to name a few.

In this paper, we consider a particular human activity application with special interest to the Muslim community around the world, namely the recognition of postures of the Islamic prayer, also known as *Salat*. Salat is the second pillar in Islam and is the most important worshipping activity that is repeated five times a day by any Muslim. Besides its spiritual value, it consists of a series of postures that must be executed in a predefined sequence as instructed by the Prophet Mohamed, peace be upon him. The motivation of

this work is that several people (e.g., kids, beginners) are likely not to perform the postures correctly. There are different reasons for this to happen like, for example, the lack of proper learning of prayer movements, or being careless. Besides, another requirement in Salat is the *tranquility*; that is, every posture should be executed in a proper and sufficient amount of time while making invocations and reading the Quran. Only a few research works in the literature have attempted to address the activity recognition of Salat. In [1,2], the authors used smartphone technology to recognize Salat activities. In [3,7], the authors assessed the activities of Salat using Electromyographic (EMG) signals. On the other hand, several research works used deep learning approaches to address the problem of human activity recognition in different applications such as sports, healthcare, well-being using wearable sensors [5,6,8]. To the best of our knowledge, there is no previous work that addressed the problem of activity monitoring of Islamic prayer using deep learning approaches. In this paper, we propose a novel solution to the posture recognition of Salat using convolutional neural networks (CNN), which aims at identifying the four basic postures of Salat namely, Standing (Qiam), Bowing (Ruku), Prostration (Sujud), Sitting (Julus), using state-of-the-art CNN algorithms. CNN has been used in a wide variety of applications such as vehicle detection [9,10], semantic segmentation of urban environments [11], and self-driving vehicles [12]. The contribution presented in this paper represents the first step towards the main objective of this project that consists in developing an AI-based tool for the assessment of the Islamic prayer and an assistive system that helps beginners and kids to correct the postures during Salat. It has to be noted that we only consider the recognition of correct postures in our neural network model, and we ignore wrong positions, which is left as a future extension of this work in which we will address the anomalies during prayers. The principal added value of this paper is the development of a specific dataset for the human activities in the context of Islamic prayer that provides four classes related to each of the basic postures of Salat. Furthermore, we trained the state-of-the-art YOLOv3 neural network on the constructed dataset to accurately detect and recognize the four postures of

Salat. We have tested the resulting trained network on videos showing people praying, and all the postures were most of the time correctly recognized. The remainder of the paper is organized as follows. Section II discusses the literature on human activity recognition and also related works on Islamic prayer monitoring using sensing activities. Section III gives a brief overview of the YOLOv3 algorithm. Section IV presents the different postures of Salat and the corresponding dataset. Section V discusses the experimental results. Finally, Section VI concludes the paper and presents future extensions.

II. RELATED WORKS

The earliest work that targets the task of automatic recognition of prayer movements is the paper introduced in 2009 by El-Hoseiny et al.[13]. They used a camera to capture the side view of the prayer. Then, they used morphological operations to extract the polygon corresponding to the contour of the prayer body. The polygon information is used to calculate the backbone axis angle and to extract four key-points of the human body. The backbone axis angle is the angle between the y-axis and the back-axis of the detected body. The four key points determined from the polygon are the center point of the polygon, the ankle point, the head point, and the back point. Based on the coordinates of the key-points and the value of the backbone axis angle, the prayer postures and movements can be determined using a set of given inequations. The method is totally based on hand-crafted features without any use of machine learning classifiers. This work is the only work that used ordinary camera sensor for this task. The other works leveraged the use of accelerometer and Kinect sensors. Al Ghannam et al.[2] used a set of classifiers for prayer activity monitoring and recognition based on accelerometer data collected from a mobile phone. They used three machine learning classifiers: J48 Decision Trees, IB1 (Instance-Based Learning) Algorithm, and Naive Bayes. The accuracy of all these algorithms exceeded 90%. Eskaf et al.[14] presented a framework for daily life activities (sitting, standing, walking, ...) from accelerometer data. Then, they aggregated these simple activities for recognition of prayer using supervised machine learning classifiers. Ali et al.[15] presented similarly a system for automatic recognition and monitoring of prayer postures based on triaxial accelerometer data collected from a smartphone. They added an analysis of group prayer activities using dynamic time warping.

In [16], the authors proposed a human activity recognition technique based on a deep learning model designed for low power devices. The recognition gives significant contextual data for well-being. Alobaid et al.[1] also studied the use of mobile accelerometer data for prayer activity recognition. They made a performance comparison of three feature extraction approaches and eight machine learning classifiers. They concluded that Random Forest is the most appropriate for this task, with an accuracy of 90%. They proposed a two-level classifier to solve the confusion between two similar prayer stages and to improve the accuracy to 93%. They added a brief study on the effect of personal characteristics like height and age on the performance of the classifier. Jaafer et al.[17] used another sensor, the Kinect RGB-Depth camera. Two Kinect sensors are put in a fixed position on the body, and the skeleton information is extracted using Kinect Software Development Kit. They used the Hidden Markov Model as the machine learning classifier for learning the prayer movements from the skeleton information.

We can see from the above works that only one work had used the ordinary camera sensor, although its practicability and easiness of use compared to other sensors. Moreover, this work used only geometric properties without profiting from the potential of machine learning algorithms. The significant advance noted in machine learning and especially the emergence of deep learning algorithms will surely make a significant improvement in the efficiency of prayer postures recognition from a video camera. This is why we targeted this limitation in the current state of the art approaches. We selected one of the most performing one-stage deep learning algorithms in object detection task (YOLOv3[18]), and we applied it in the context of prayer posture recognition. We considered in our study that the camera could be placed in different views from the prayer and not only in a fixed position from him.

III. ALGORITHMS BACKGROUND

Among the deep learning algorithms used in computer vision, YOLOv3[18] is the most attractive for the task of prayer posture recognition. Two facts are defending this choice. Firstly, YOLOv3[18] had proven its efficiency compared to other object detection algorithms[9]. Secondly, it has an efficient inference time (up to 45 frames per second). This allows a real-time recognition of the prayer postures. In the following subsections, we present the architecture of the first version of YOLO (YOLOv1[19]) and the different improvements implemented in YOLOv2[20] and YOLOv3[18].

A. YOLOv1

YOLO was first introduced in 2016 as a different approach to treat generic object detection problems. The approach is based on training only one CNN for the task of localization and the task of classification at the same time. YOLOv1 is composed of 24 convolutional layers for feature extraction and 2 fully connected layers for generating the final output. The total architecture is detailed in $Figure$ 1.

The input image is divided by YOLO into a grid of $S \times S$. Every grid cell can be associated with only one object. The grid cell has also a fixed number B of boundary boxes to be predicted for this object. A confidence score is calculated for every bounding box. As a result, YOLO calculates for every grid cell a vector of class probabilities among the C classes we are targeting. YOLO calculates also for every bounding box of the cell a vector containing 5 parameters: $(x, y, w, h, box\_confidence\_score)$. Hence, the YOLO network generates for every input image a tensor of the form:

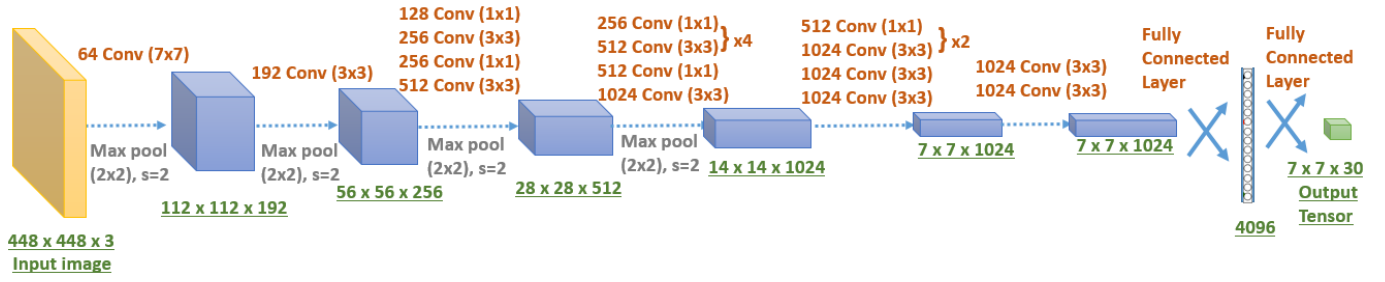$$S \times S \times (B * 5 + C) \tag{1}$$

where:

Figure 1. *YOLO v1 Architecture*

- $S \times S$: corresponds to the number of grid cells.
- B: corresponds to the number of bounding boxes.
- C: corresponds to the number of targeted classes.

To train the YOLO network, we used a combination of three loss functions. The first is the classification loss (loss of the conditional probabilities for every class). The second is the localization loss (the position of the estimated bounding box compared to the ground truth). The third is the confidence loss (the box confidence score compared to the ground truth). To measure the error between the predicted value and the ground truth, YOLO uses sum-squared error as a metric. The expression of the loss is detailed in *Equation* 2.

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

where:

- λ_{coord} is the weight for the bounding box coordinate prediction loss. During the training, it is set to 5.
- λ_{noobj} is the weight for the confidence score prediction loss for boxes that don't include objects. During the training, It is set to 0.5.
- $\mathbb{1}_{i}^{obj}$ indicates if the object appears in cell i
- $\mathbb{1}_{i}^{obj}$ indicates that the bounding box of index j is the responsible for the prediction.
- x_i is the ground truth value of x, $\hat{x}_i$ is the predicted value for x, The same for y, (ω (width) and h (height).
- C corresponds to the confidence score of the bounding box

- $p_i(c)$ is the probability that the cell i belong to the class c. $\hat{p}$ denotes the predicted probability.

When it was introduced, YOLO surpassed other object detection algorithms in terms of speed. Its mAP (mean Average Precision) was comparable or exceeded the mAP of other state-of-the-art algorithms.

B. YOLOv2

In YOLOv2[20], many improvements had been introduced to increase the accuracy and decrease the processing time. Among them, we can note:

- Use of Batch Normalization (BN). This technique was introduced in 2015 to improve the convergence of the loss during the training. BN was added to all convolutional layers in YOLO, which improved the mAP by 2%.
- Replacing the input image size 224×224 by 448×448, which improved the mAP by 4%.
- The adoption of convolution with anchor boxes. The class prediction is moved from the level of the grid cell to the level of the boundary box. This made a small increase in the mAP by a margin of 0.3% but increased the recall from 81% to 88%. This enhanced the ability to detect all the objects existing in the image and reduced the False Negative Rate.
- The use of K-means clustering algorithm to select the best anchor box from the training set of bounding boxes. Euclidian distance is replaced by the IoU (Intersection Over Union) for the clustering.
- The predictions are made on the offsets to the anchors. Instead of predicting (x, y, ω, h, C), YOLOv2 predicts $(t_x, t_y, t_w, t_h, t_C)$. This makes the network converge better.
- The use of fine-grained features. Similarly to the identity mapping in ResNet [21], YOLOv2 concatenates low resolution features to high resolution features to improve the ability to detect small objects. This improves the mAP by 1%.
- The adoption of multi-scale training. Instead of using a fixed size for the input image, YOLOv2 randomly selects an image size every 10 batches. This improves the ability to predict well over different sizes of input images.

C. YOLOv3

In April 2018, the YOLOv3[18] was introduced as an incremental improvement to the previous versions. Among the

improvements made, we can note:

- The use of the multi-label classification. Instead of the mutual exclusive labeling in the previous versions, YOLOv3 uses a logistic classifier to estimate the likeliness of the object being of a specific label. The classification loss is changed to use for every label the binary cross-entropy loss instead of the general mean square loss used in the previous versions.
- The use of another bounding box prediction. During the training, the objectness score 1 in YOLOv3 is associated with the bounding box anchor that best overlaps the ground truth object. Besides, if the IoU (Intersection Over Union) between the bounding box anchor and the ground truth is less than a threshold (0.7 in the implementation), it is ignored. In the end, YOLOV3 associates for every ground truth object one bounding box anchor.
- The change of the output 3d tensor. In YOLOv3, the prediction is done for one grid cell at 3 different scales and then concludes the final bounding box from those scales. This was inspired by the feature pyramid networks. Hence, the new dimension of the output 3d sensor is then:

$$S \times S \times (3 * (5 + C)) \qquad (3)$$

where:
  - $S \times S$: corresponds to the number of grid cells.
  - B: is omitted because only one bounding box anchor is kept at the end.
  - C: corresponds to the number of targeted classes.

- The adoption of a new feature extractor (the darknet-53). It has 53 layers and uses skip connection similarly to ResNet [21]. It uses both 3×3 and 1×1 convolutions. It gave the state of the art accuracy but with better speed and fewer computations.

IV. DATASET

We have built the Salat Postures Dataset from a set of images of students and laboratory members performing different Salat positions using mobile phones, from various view angles, especially captured for the purpose, complemented with images (and video frames) of people in prayer collected from the Internet. In fact, our ultimate scenario is that a person records himself using a mobile phone, and then the video stream will be processed in real-time to identify the different postures and finally provides postures flaws and recommendation. For this reason, most of the collected images are based on mobile phone data collection to serve the final scenario, added to this some more general images. We have also considered images that show the whole body of the person making prayer, as we impose that the person records the whole body from any angle to perform the detection and classification.

The total number of collected images is 849 that we manually labeled into four classes using rectangular bounding boxes. The four classes correspond to the four main positions in Islamic prayer:

Figure 2. Sample images of our dataset, showing the different prayer positions. The images on the top were captured specifically for this study, while the images on the bottom were collected from the Internet.

Table I
NUMBER OF IMAGES AND INSTANCES IN THE TRAINING AND TESTING DATASETS.

	Training dataset	Testing dataset
Number of images	764	85
Percentage	90.0%	10.0%
Instances of class "Standing"	303	37
Instances of class "Bowing"	210	27
Instances of class "Prostrating"	174	11
Instances of class "Sitting"	179	22

- *Qiyam*: it is the act of standing, and this posture is related to when reading the Quran, or after rising from bowing (*Ruku*). It typically spans over one or more minutes. Sub-figure 5 of Figure 2 shows an example of a man in *Qiyam* posture.
- *Ruku*: it means bowing, and this posture happens after Qiam to make invocations. It typically spans over a few seconds. Sub-figures 1 and 3 of Figure 2 shows an example of a man in *Ruku* posture.
- *Sujud*: It is the prostration and happens after raising from bowing for also making invocation. It spans over several seconds. Sub-figure 4 of Figure 2 shows an example of a man in *Sujud* posture.
- *Julus*: it is the position of sitting after finishing *Sujud* for making invocation and closing Salat at its end. Sub-figures 2 and 6 of Figure 2 shows an example of a man in *Julus* posture.

Figure 2 shows some sample images of the dataset, and Table I shows the number of images and instances in the training and testing datasets. We have subdivided the training data and test data to be 90% and 10%, respectively.
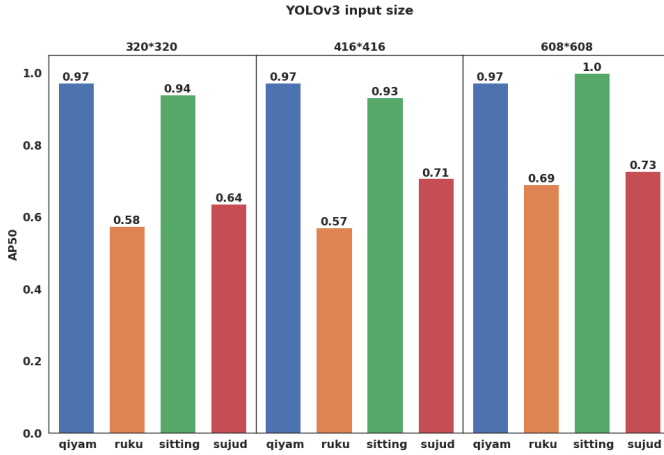
V. EXPERIMENTAL EVALUATION

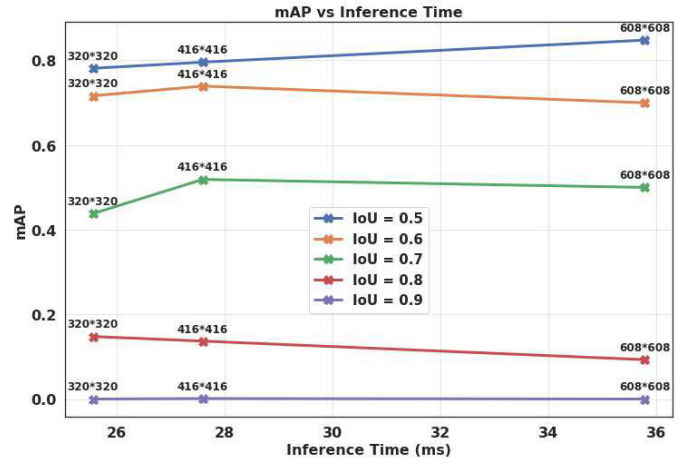Figure 3. Average Precision (AP) at an IoU threshold of 0.5, for each input size and each class.

Figure 4. Comparison of the trade-off between mAP and inference time for the three different input sizes of YOLO v3, and for different values of IoU threshold ranging from 0.5 to 0.9.

For the experimental setup, we used a workstation powered by an Intel Core i9-9900K (Octa-core) processor, with 64GB RAM, and an NVIDIA GeForce RTx2080T (11 GB) GPU, running on Ubuntu 16.04 LTS.

We used the following metrics to assess the performance of the object detection networks:

- IoU: Intersection over Union, which measures the overlap between the predicted bounding box and the ground-truth one.
- mAP: mean average precision, which corresponds to the area under the precision vs. recall curve, averaged over all classes. The mAP was measured for different values of IoU (0.5, 0.6, 0.7, 0.8 and 0.9).
- Inference time (in millisecond per image), which measures the inference processing speed on test images.
- TP (True Positives): number of objects correctly detected and classified.
- FP (False Positives): number of objects detected but wrongly classified.
- FN (False Negatives): number of non-detected objects.

Figure 3 shows the value of AP50 (Average Precision for an IoU threshold of 0.5), for each input size and each class. For all three networks, the performance obtained for the detection of *Ruku* and *Sujud* classes are markedly lower than for qiyam and sitting classes. This may be explained by the visual resemblance between these two classes, which causes them to be conflated.

Figure 4 displays the trade-off between mAP and inference time for the 3 input sizes, and for different values of IoU threshold ranging from 0.5 to 0.9. For an IoU threshold of 0.5, passing from an input size of 320*320 to 416*416, the mAP50 increases by 2%, while the inference time increases by 8%. Then, when passing from an input size of 416*416 to 608*608, the mAP50 increases by 7%, while the inference time jumps by 30%. For higher values of IoU threshold,
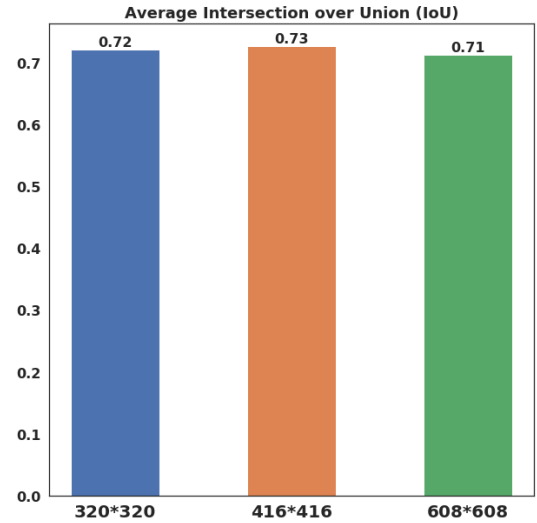

Figure 5. Average intersection over Union (IoU) for the three input sizes.

increasing the input size has little or negative effect on the mAP. This indicates that higher input sizes do not enhance the precision of the bounding boxes, which can also be seen in figure 5 that depicts the average IoU for each input size. The value of mAP90 (at IoU=0.9) is close to zero, which shows that YOLOv3 has difficulties in getting the bounding boxes to be perfectly aligned with the object, as already noticed in the original YOLOv3 paper[18].

On another hand, the increase in inference time for higher input sizes is due to the fact that a larger input size in the network generates a higher number of parameters, which raises the number of operations necessary to obtain the output.

Figure 6 depicts the number of true positives (TP), false positives (FP), and false negatives (FN) for an IoU threshold of 0.5, and an input size of 320x320. We notice that the main issue with the ruku and sujud classes is the relatively high
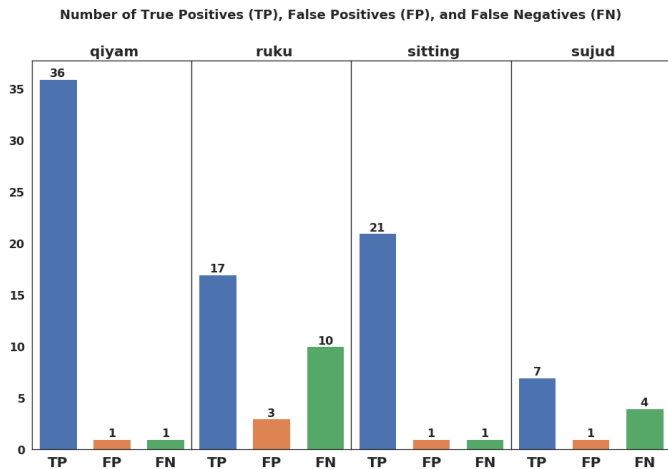
Figure 6. Number of true positives (TP), false positives (FP), and false negatives (FN) for an IoU threshold of 0.5, and an input size of 320x320.

number of false negatives compared to the two other classes. This indicates that we should enrich the training dataset with a higher number of diverse images, especially for these two classes.

VI. CONCLUSIONS

In this paper, we conducted a performance evaluation study for detecting Islamic prayer postures. The detection was performed by training the YOLOv3 network (with different configurations) on a dataset that was especially collected for the purpose. The performance of the different network configurations was assessed using several metrics (mAP, IoU, Inference time, TP, FP, and FN). The mAP at IoU=0.5 ranges from 78% (for a network input size of 320x320) to 85% (for an input size of 608x608), which is very encouraging for a training dataset of only 764 images.

This study is a first step that will be further developed in future works, by enlarging the training dataset, investigating other network architectures, conducting thorough hyperparameter optimization, and integrating the object detection model into a broader artificial intelligence assistive framework that aims to guide Muslim worshippers to evaluate the correctness of their postures during prayer.

ACKNOWLEDGMENT

REFERENCES

[1] O. Alobaid and K. Rasheed, "Prayer activity recognition using an accelerometer sensor," in *Proceedings on the International Conference on Artificial Intelligence (ICAI)*, pp. 271–277, The Steering Committee of The World Congress in Computer Science, Computer . . . , 2018.

[2] R. Al-Ghannam and H. Al-Dossari, "Prayer activity monitoring and recognition using acceleration features with mobile phone," *Arabian Journal for Science and Engineering*, vol. 41, no. 12, pp. 4967–4979, 2016.

[3] M. F. Rabbi, N. Wahidah Arshad, K. H. Ghazali, R. Abdul Karim, M. Z. Ibrahim, and T. Sikandar, "Emg activity of leg muscles with knee pain during islamic prayer (salat)," in *2019 IEEE 15th International Colloquium on Signal Processing Its Applications (CSPA)*, pp. 213–216, March 2019.

[4] M. Ramzan, A. Abid, H. U. Khan, S. M. Awan, A. Ismail, M. Ahmed, M. Ilyas, and A. Mahmood, "A review on state-of-the-art violence detection techniques," *IEEE Access*, vol. 7, pp. 107560–107575, 2019.

[5] D. Ravì, C. Wong, B. Lo, and G. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, pp. 56–64, Jan 2017.

[6] W. Xu, Z. Miao, J. Yu, and Q. Ji, "Deep reinforcement learning for weak human activity localization," *IEEE Transactions on Image Processing*, pp. 1–1, 2019.

[7] F. Ibrahim and S. A. Ahmad, "Assessment of upper body muscle activity during salat and stretching exercise: A pilot study," in *Proceedings of 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics*, pp. 412–415, Jan 2012.

[8] A. Gumaei, M. M. Hassan, A. Alelaiwi, and H. Alsalman, "A hybrid deep learning model for human activity recognition using multimodal body sensing data," *IEEE Access*, vol. 7, pp. 99152–99160, 2019.

[9] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, "Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3," in *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*, pp. 1–6, IEEE, 2019.

[10] A. Ammar, A. Koubaa, M. Ahmed, and A. Saad, "Aerial Images Processing for Car Detection using Convolutional Neural Networks: Comparison between Faster R-CNN and YoloV3," *arXiv pre-print 1910.07234*, October 2019.

[11] B. Benjdira, Y. Bazi, A. Koubaa, and K. Ouni, "Unsupervised Domain Adaptation Using Generative Adversarial Networks for Semantic Segmentation of Aerial Images," *Remote Sensing*, vol. 11, no. 11, 2019.

[12] B. Schoettle and M. Sivak, "A survey of public opinion about autonomous and self-driving vehicles in the us, the uk, and australia," tech. rep., University of Michigan, Ann Arbor, Transportation Research Institute, 2014.

[13] M. H. El-Hoseiny and E. Shaban, "Muslim prayer actions recognition," *2009 Second International Conference on Computer and Electrical Engineering*, vol. 1, pp. 460–465, 2009.

[14] K. Eskaf, W. M. Aly, and A. Aly, "Aggregated activity recognition using smart devices," in *2016 3rd International Conference on Soft Computing & Machine Intelligence (ISCMI)*, pp. 214–218, IEEE, 2016.

[15] M. Ali, M. Shafi, U. Farooq, *et al.*, "Salat activity recognition using smartphone triaxial accelerometer," in *2018 5th International Multi-Topic ICT Conference (IMTIC)*, pp. 1–7, IEEE, 2018.

[16] D. Ravi, C. Wong, B. Lo, and G. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," in *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 71–76, June 2016.

[17] N. A. Jaafar, N. A. Ismail, and Y. A. Yusoff, "An investigation of motion tracking for solat movement with dual sensor approach," 2015.

[18] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018.

[19] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 779–788, 2016.

[20] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 6517–6525, 2017.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Arxiv.Org*, 2015.