Journal of Artificial Intelligence and Data Science (JAIDA)
e-ISSN: 2791-8335
https://jaida.ikcu.edu.tr

Vol. 1, No. 2, pp. 125-135, 2021
Research Article

Snake Detection and Blurring System to Prevent Unexpected Appearance of Snake Images on Visual Media Sources Using Deep Learning

Selay TEKGÜL, Gokce NUR YILMAZ *

TED University, Faculty of Engineering, Computer Engineering, Turkey

Abstract
This paper proposes an object detection and blurring system using deep learning especially for the detection of snake images since while some of the people does not care about the appearance of the snake images on the visual sources, there is a significant portion of them who are uncomfortable of seeing snakes as well as their images. Therefore, with the system that is proposed in this paper, the snake objects from the images would be detected with YOLOv4 trained model and blurred using OpenCV. The detection f1 score of the selected model is 92% on the test set.
Keywords: blur, deep learning, object detection, phobia, snake.

1. Introduction

It is a fact that there is a significant attention that humans pay to the animals because of the effects of the inheritance [1]. Especially for the snakes, not only the effect of live snakes but also their photographs can cause a great deal of fear, i.e., phobia of snakes. Therefore, their unexpected appearance on the visual media sources e.g., TV, movies, photography is a problem for people who are suffering from that fear. As a result of the problem caused by these unfiltered visual media sources, the quality of their life is affected negatively.

Considering this fact, a model including snake detection on images and videos, as well as blurring the snakes is developed. Therefore, the proposed model consists of two stages:

1. Detection of the snakes on the visual source
2. Blurring the detected objects

For the first stage, YOLOv4 [2], which is a state-of-the-art, real-time object detection system, is used. YOLO has several advantages when it is compared to other object detection algorithms such as EfficientDet developed by Google Brain Team[3], CNN, fast R-CNN and faster R-CNN. One of the most significant advantages is that YOLO considers an image completely. Therefore, it results in better and faster detections with YOLO [2]. After the performance evaluation results, it is observed that 92% f1 score is achieved using YOLO for the snake detection. The second stage relying on the blurring operation is implemented considering with another blurring method from OpenCV that is a software library which includes open-source machine learning and computer vision algorithms [4].

The rest of the paper is organized as follows. In the second section, the proposed model is introduced. The results and discussions are presented in Section 3. Finally, Section 4 concludes the paper and discuss about future work.

2. Proposed Model

In the proposed model for the snake detection, the YOLO approach is exploited due to the advantages it provides compared to other object detection algorithms including CNN, fast R-CNN and faster R-CNN in terms of considering the entire image and faster detection [2]. The proposed model of this study is presented in Figure 1.

As it can be seen from the figure, first of all, dataset is fed into the deep learning model for training purpose. As a dataset, the latest version of Open Image Dataset V6 [5] is used. Open Image Dataset V6 is a dataset of 9M image-level annotated images provided by Google [6]. 1000, 52 and 316 images of the "snake" class by Open Image Dataset V6 are used as training, validation, and test images, respectively, in this study.

Since there is only one class in the proposed model, it is decided to train the model for 6000 iterations based on the recommendations on [7] which particularly explains the selection of the number of iterations as 2000 per class but not less than 6000. During the training process, the losses of the models after each iteration are also observed. When the losses are investigated, a decreasing trend is observed as a function of number of iterations. Moreover, the models that is encountered after each 1000 iterations are saved and ready to be used. Therefore, six models are derived due to the fact that 6000 iterations are divided by each 1000 iteration. As mentioned, there are

*Corresponding author e-mail address: gokce.yilmaz@tedu.edu.tr

Received: 23/11/2021; Accepted: 27/12/2021

six different trained models each encountered after 1000 iterations of training out of total 6000 iterations. The models will be stated as M1, M2, M3, M4, M5 and M6 after each 1000 iterations sequentially trained, respectively as they can be observed from Figure 1.

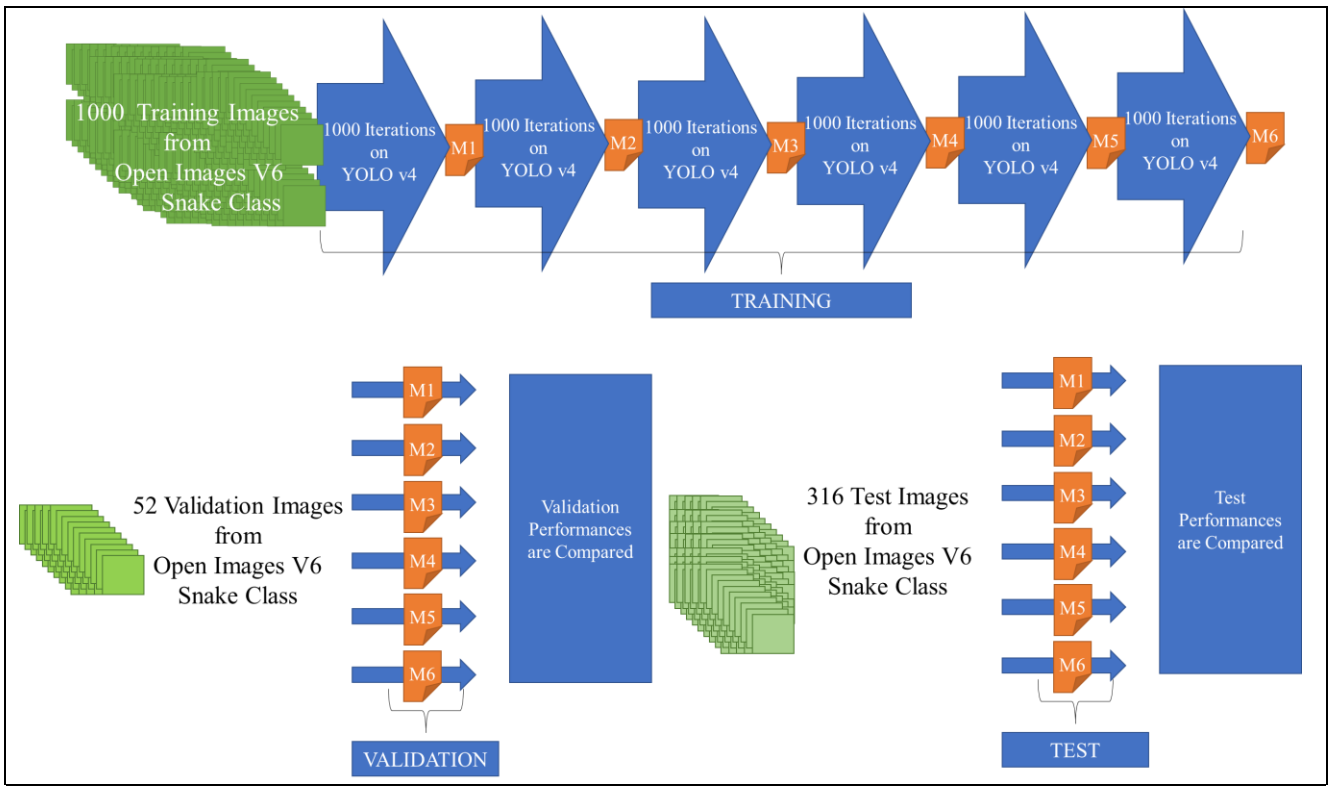

Figure 1. *Proposed model.*

Additionally, these six different models are used to detect the snakes on the validation and test set images. After models are used for detection on the validation and test sets, the performance values of the models are evaluated.

The losses encountered after 6000 iterations are shown in Figure 2. When the losses are investigated form the figure, a decreasing trend is observed as a function of number of iterations. However, so as to better evaluate the performance of the models after each 1000 iteration, not only the losses but also the true positive (TP), false positive (FP), false negative (FN), precision, recall, F1 score, average intersection over union (IoU), average precision (AP) and mean average precision (mAP) values are calculated and compared. These values will be presented and discussed in Section 4.

Consequently, as a second step of the solution, after determining the best model to be used in the test set, the detected bounded boxes where the snakes are predicted to be on the images are blurred with the blur method from OpenCV. This method uses the normalized box filter and kernel to blur the given image. In this study, the kernel size is selected as 53 to 53.

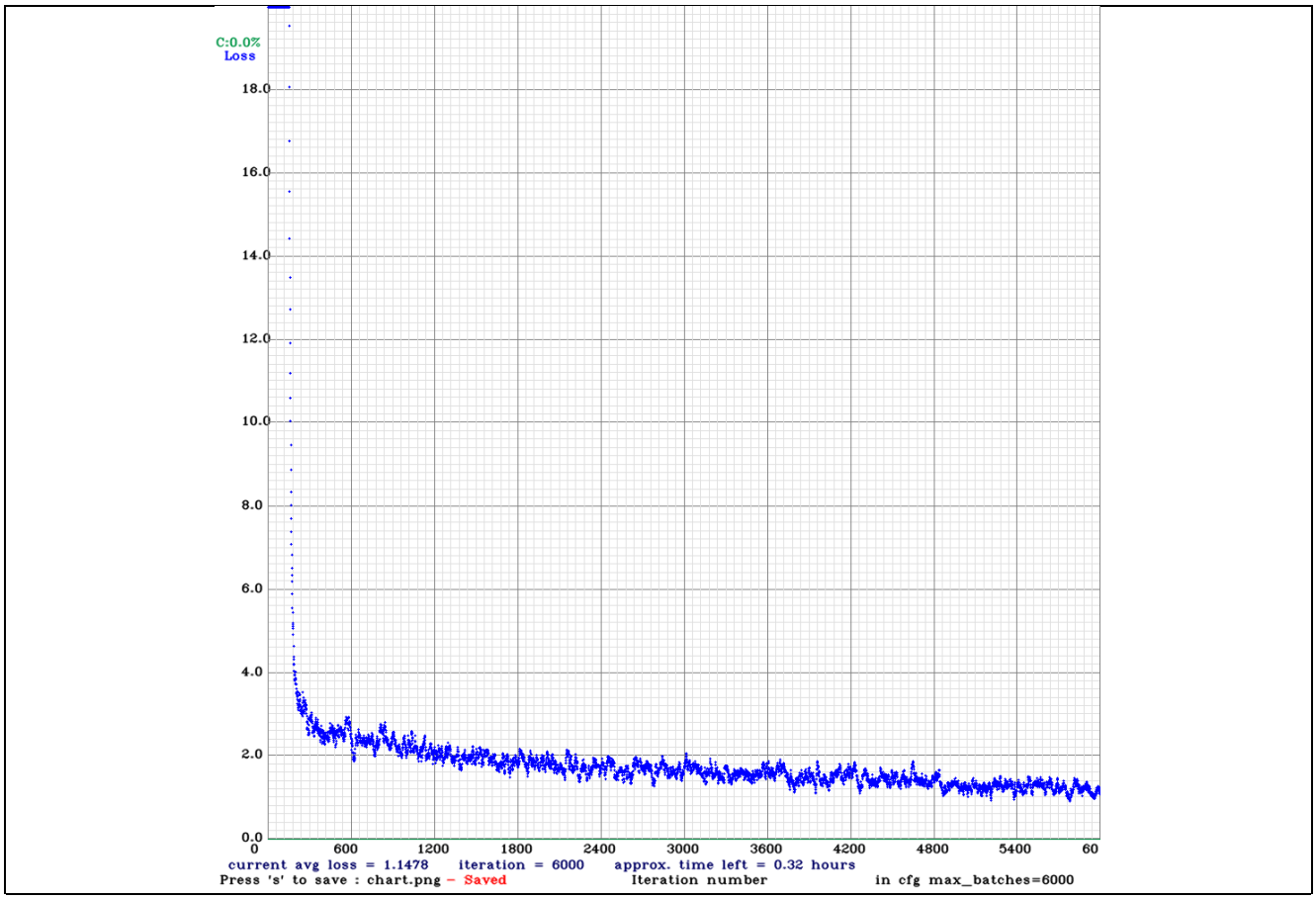

Figure 2. *Losses as a Function of Number of Iterations (Total 6000 iteration).*

3. Results and Discussions

Corresponding performance values related with TP, FP, FN, precision, recall, F1 score, average IoU, AP and mAP of six models on validation set which consists of 52 images are shown in Table 1 by emphasizing the best scores on each column with bold texts. As observed from the bold texts, model M3, has the best TP, FP, FN, precision, recall, F1 score, AP and mAP values among the siz models that are compared. On the other hand, for the average IoU model M5 which is encountered after 5000 iterations gives the best score.

Table 1. *Comparison of the performances of six models on 52 images validation set.*

Models	TP	FP	FN	Precision	Recall	F1	Average IoU	AP	mAP
M1	54	3	12	0.95	0.82	0.88	67.88%	94.28%	94.28%
M2	48	12	18	0.80	0.73	0.76	51.83%	68.39%	68.39%
M3	**60**	**1**	**6**	**0.98**	**0.91**	**0.94**	72.13%	**94.94%**	**94.94%**
M4	58	6	8	0.91	0.88	0.89	61.43%	88.36%	88.36%
M5	58	3	8	0.95	0.88	0.91	**79.12%**	88.76%	88.76%
M6	59	3	7	0.95	0.89	0.92	79.05%	93.66%	93.66%

The precision, recall and F1 score values of six models on the validation set consisting of 52 images are compared in Figure 3 with a chart. As can be observed from the chart, the precision, recall and F1 score values of of the models are at the range of 0.80 to 0.98, 0.73 to 0.91 and 0.76 to 0.94 on the validation set respectively. Moreover, the best precision, recall and therefore F1 score values are gathered with the model M3 on the validation set.

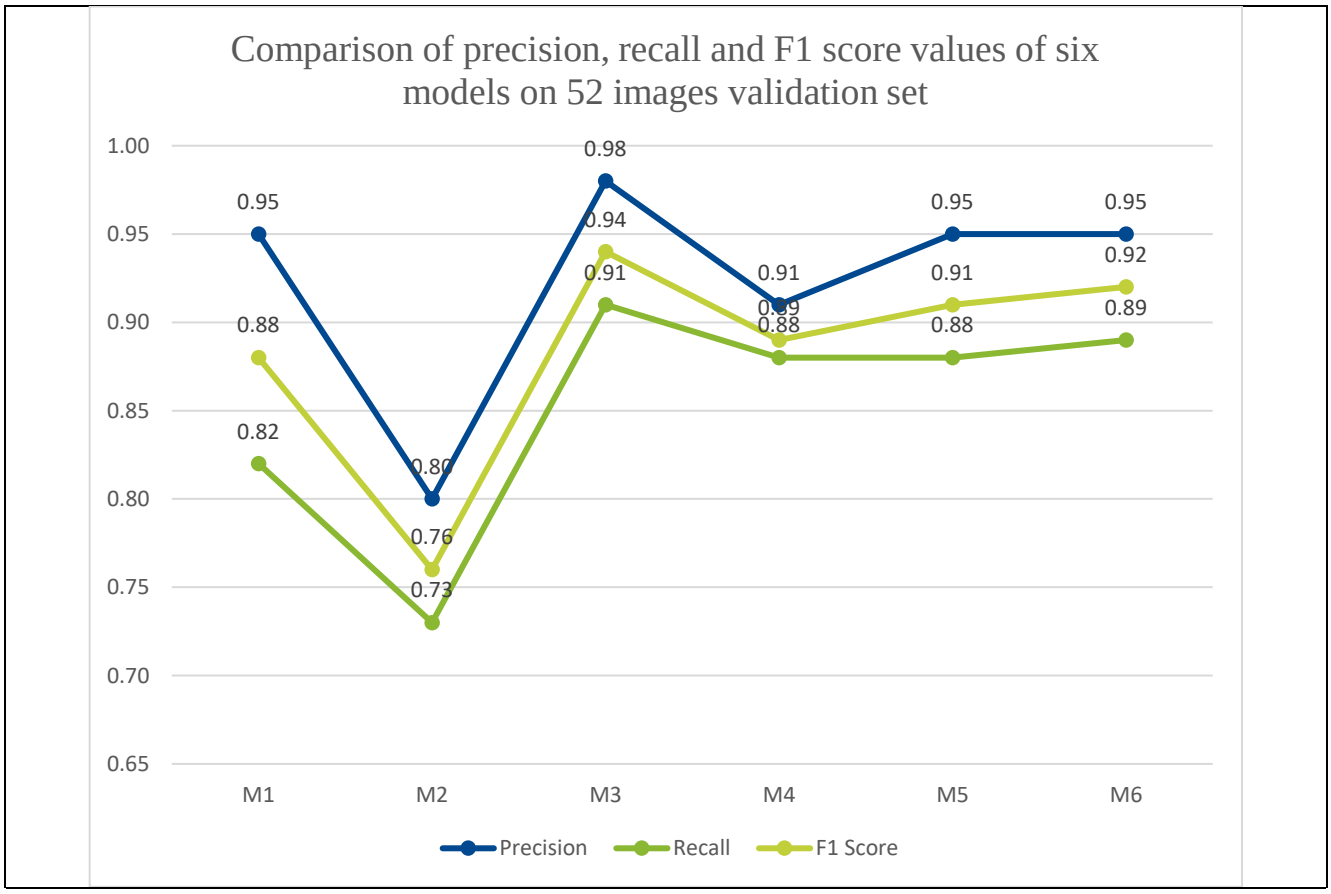

Figure 3. *Chart of comparison of precision, recall and F1 score values of six models on 52 images validation set.*

Moreover, the percentage of average IoU and mAP values of six models on the validation set consisting of 52 images are compared in Figure 4 with a chart. As seen from the chart, the average IoU and mAP values of the six models on the test set are at the range from 51.83% to 79.12% and from 58.39% to 94.94%, respectively. Additionally, the best performance scores are observed with the model M3 for both average IoU and mAP values on the validation set.

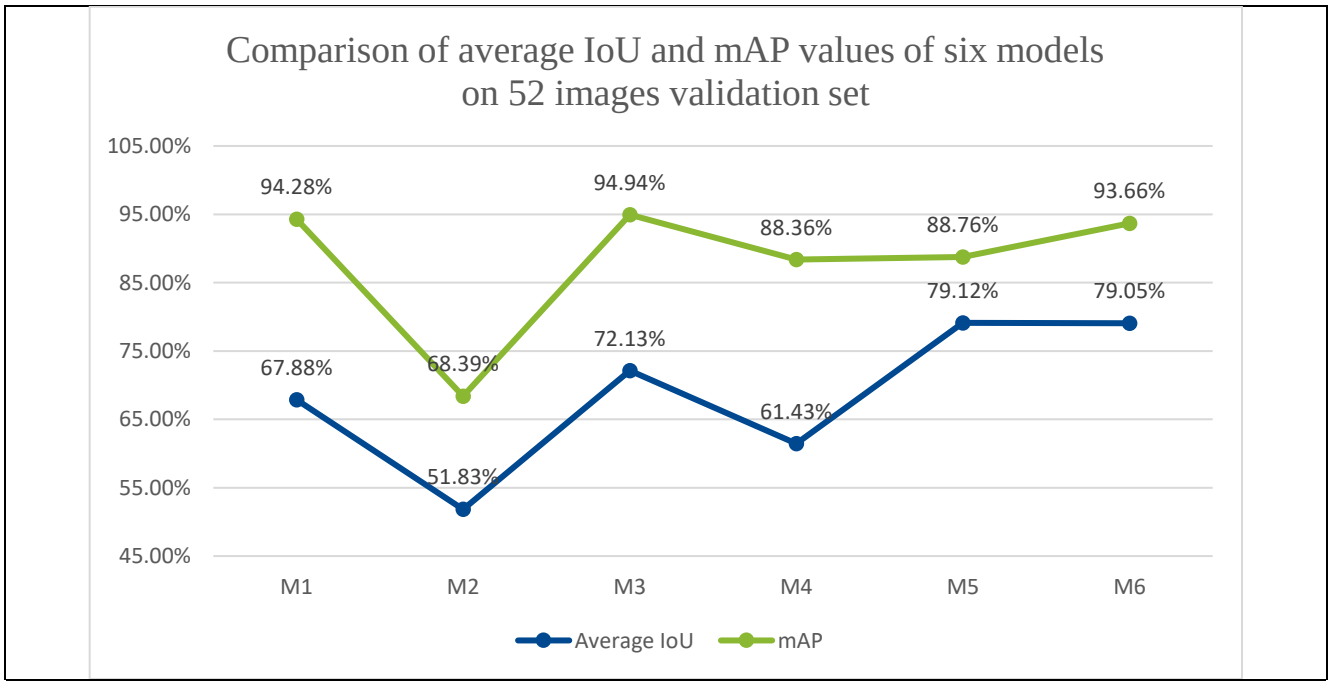

Figure 4. *Chart of comparison of precision, recall and F1 score values of six models on 52 images validation set.*

The training, validation and testing are completed on a laptop computer with a single Nvidia 950M 4GB GPU and 8GB RAM on 64-bit Ubuntu 20.04 operating system using Python version 3.8.5 and OpenCV version 4.5.2. The training is completed within 18 hours and the detection process of the validation set examples took 10-11 seconds for each of the six models which would not affect the comparison of the models drastically. Detection times of the six models on 52 images validation set are compared in Table 2.

Table 2. *Comparison of the detection times of six models on 52 images validation set.*

Models	Total Detection Time (in seconds)
M1	11
M2	11
M3	11
M4	10
M5	10
M6	10

During the evaluation of training and detection times, the given environment characteristics should be taken into consideration. When the usage of GPU changes, the training and detection times would differ.

Predicted best detection performance on the test set is taken with model M3 due to its performance on the validation set examples, however, the best TP, FN and Recall values are encountered with model M6 as 218, 25 and 0.90 on the test set examples, respectively. On the other hand, the best FP, Precision, Average IoU, AP and mAP values are gathered when model M5 is used for testing as 10, 0.96, 80.60%, 92.20% and 92.20%, respectively. Lastly, the best F1 score is the same for these two models, M5 and M6 and is 0.92.

The overall comparison of the performances of the models on the test set is put in Table 3.

Table 3. *Comparison of the performances of six models on 316 images test set.*

Models	**TP**	**FP**	**FN**	**Precision**	**Recall**	**F1**	**Average IoU**	**AP**	**mAP**
M1	205	24	38	0.90	0.84	0.87	66.17%	88.45%	88.45%
M2	160	53	83	0.75	0.66	0.70	49.79%	63.95%	63.95%
M3	214	21	29	0.91	0.88	0.90	67.75%	89.35%	89.35%
M4	212	24	31	0.90	0.87	0.89	63.04%	87.32%	87.32%
M5	217	**10**	26	**0.96**	0.89	**0.92**	**80.60%**	**92.20%**	**92.20%**
M6	**218**	12	**25**	0.95	**0.90**	**0.92**	79.78%	91.81%	91.81%

The precision, recall and F1 score values of six models on the test set consisting of 316 images are compared in Figure 5 with a chart. As presented in the chart, the precision, recall and F1 score values of of the models are at the range of 0.75 to 0.96, 0.66 to 0.90 and 0.70 to 0.92 on the test set respectively. Moreover, the best precision value is gathered with the model M5, the best recall value is gathered with the models M5 and M6 and lastly, the best F1 score is gathered with model M6 on the test set.

The percentage of average IoU and mAP values of six models on the test set consisting of 316 images are compared in Figure 6 with a chart. As seen from the chart, the average IoU and mAP values of the six models on the test set are at the range from 49.79% to 80.60% and from 63.95% to 92.20%, respectively. Furthermore, when the chart is investigated, it is observed that the model that gives the worst performance is model M2 for both of the performance values, however, the performance values gathered by using the other models are close to each other. Therefore, with the chart which compares the precision, recall and F1 Score values of the models provided at the Figure 5 is a better discriminator while selecting the best model than Figure 6.

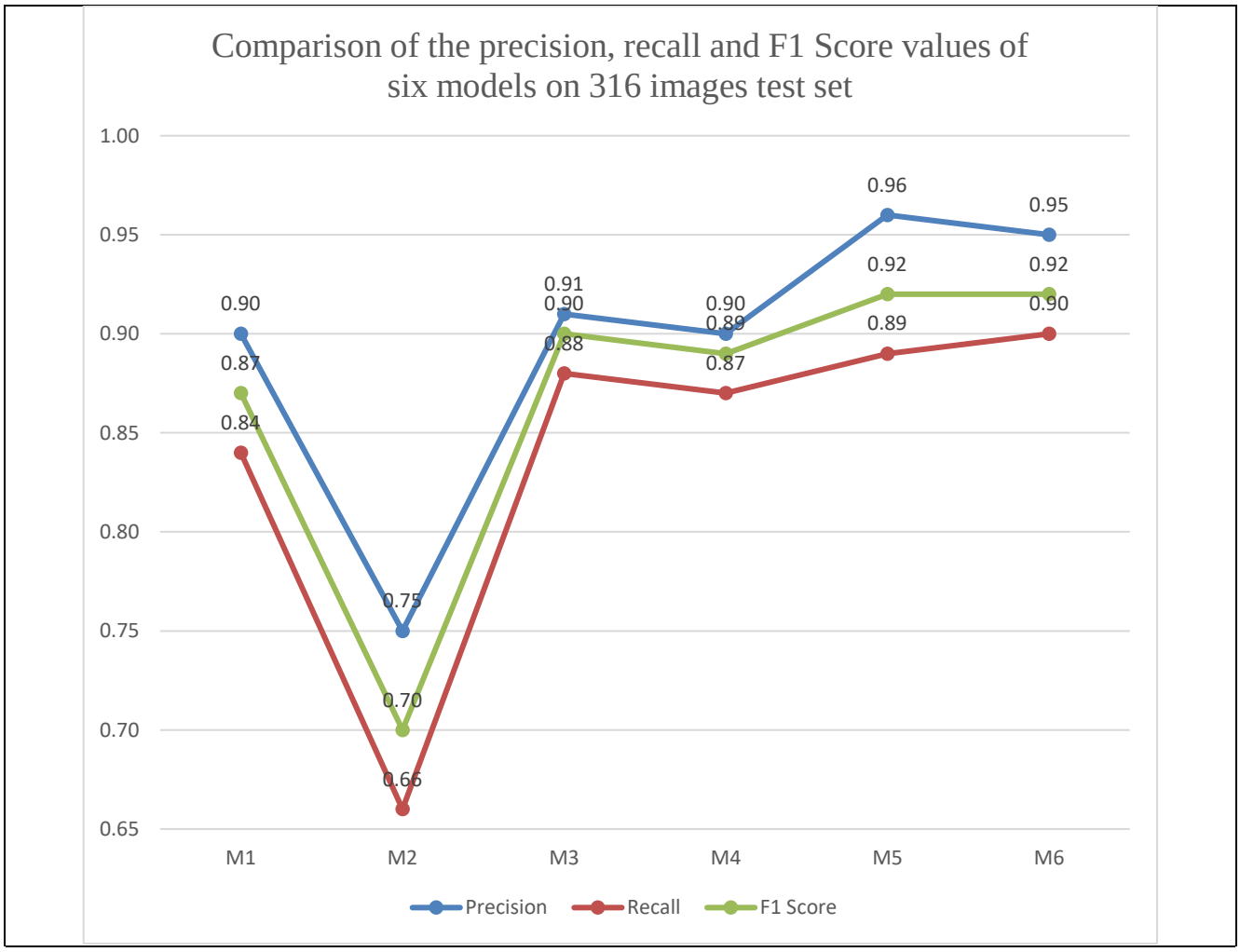

Figure 5. *Chart of comparison of precision, recall and F1 Score values of six models on 316 images test set.*

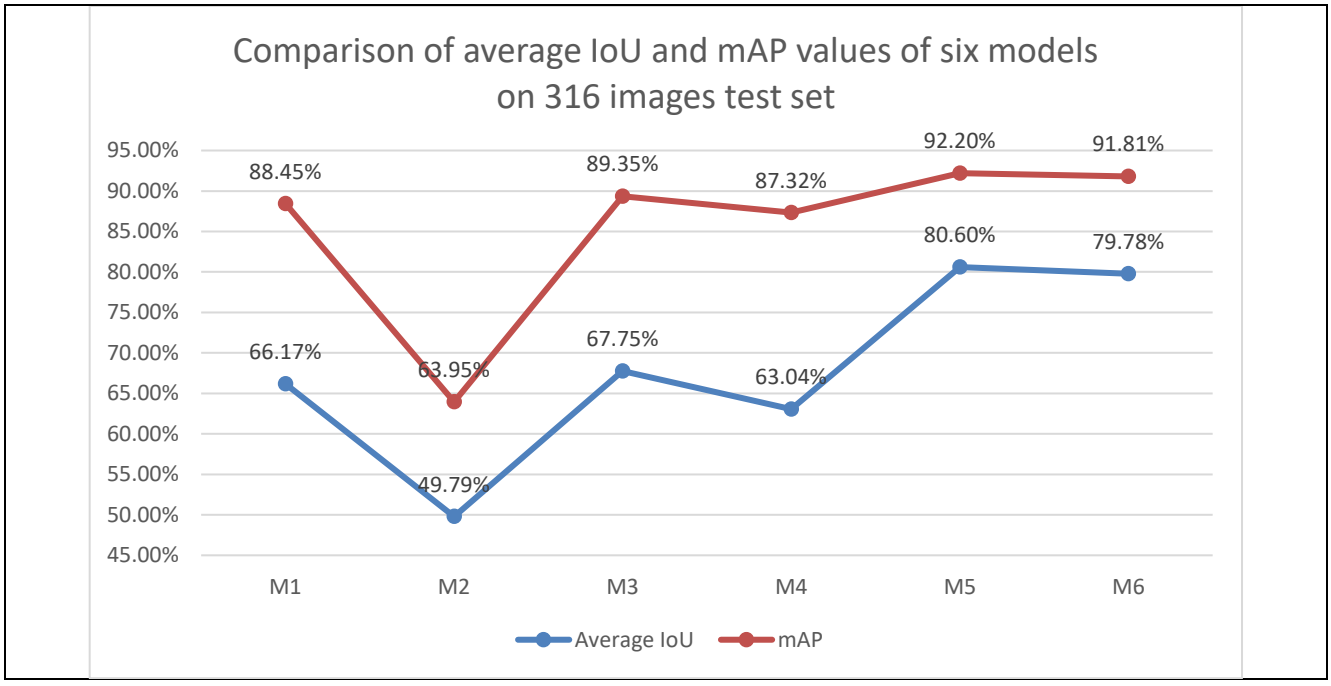

Figure 6. *Chart of comparison of precision, recall and F1 Score values of six models on 316 images test set.*

Problem that is attempted to solve in this paper is to detect as many snake images as precisely as possible when images are tested since when the phobia is considered, it should be guaranteed that none of the snake images will be shown to the viewer. Therefore, when the test results are investigated, since the given values with model M5 and M6 are pretty close to each other, the one whose recall is greater than other should be used before blurring as a final model. Additionally, with YOLO, real-time object detection is possible but requires a better GPU than the one used in the experiment of this study to propose a working model for commercial since the detection of 316 test images took 39-40 seconds to be completed in each of the six models that are shown at Table 4.

Table 4. *Comparison of the detection times of six models on 316 images test set.*

Models	Total Detection Time (in seconds)
M1	40
M2	40
M3	39
M4	40
M5	40
M6	40

In the Figures 7, 8, 9, 10, 11 and 12 the detections on the 12 images selected from the test set that are predicted using M1, M2, M3, M4, M5 and M6 are shown with red bounded boxes representing the predicted snakes, and the green bounded boxes representing the ground truth of the given image.

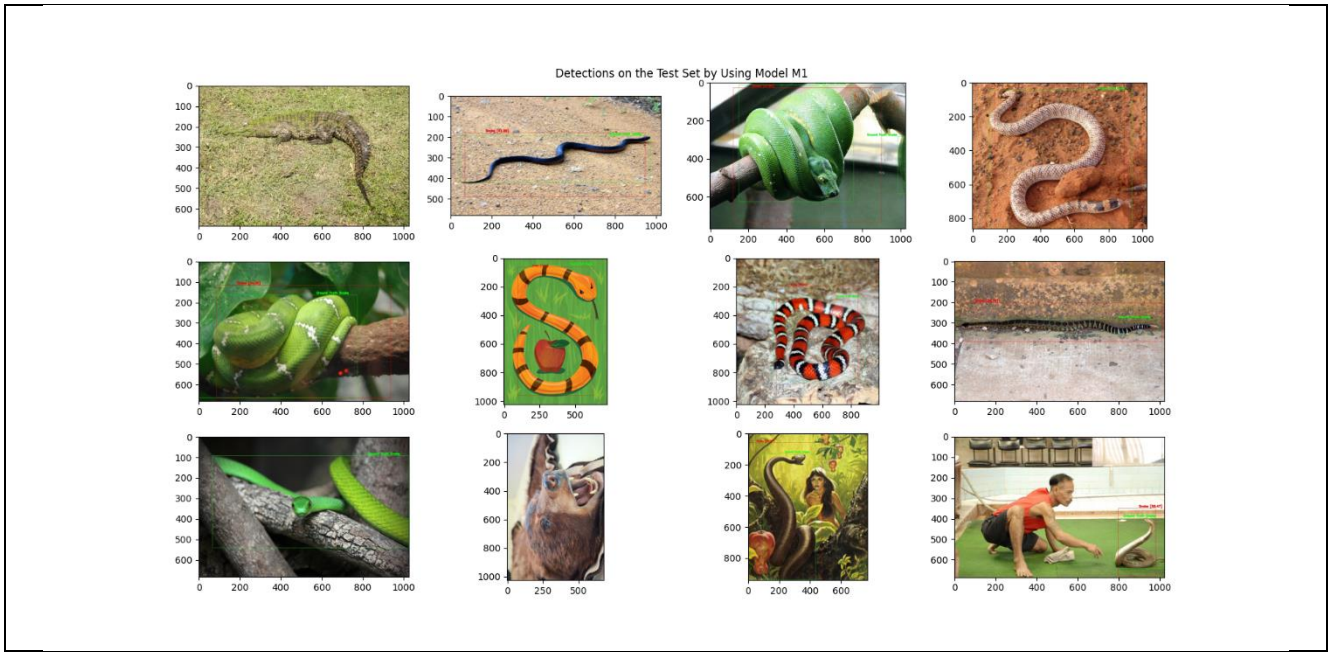

Figure 7. *Predictions of model M1 on the test set.*

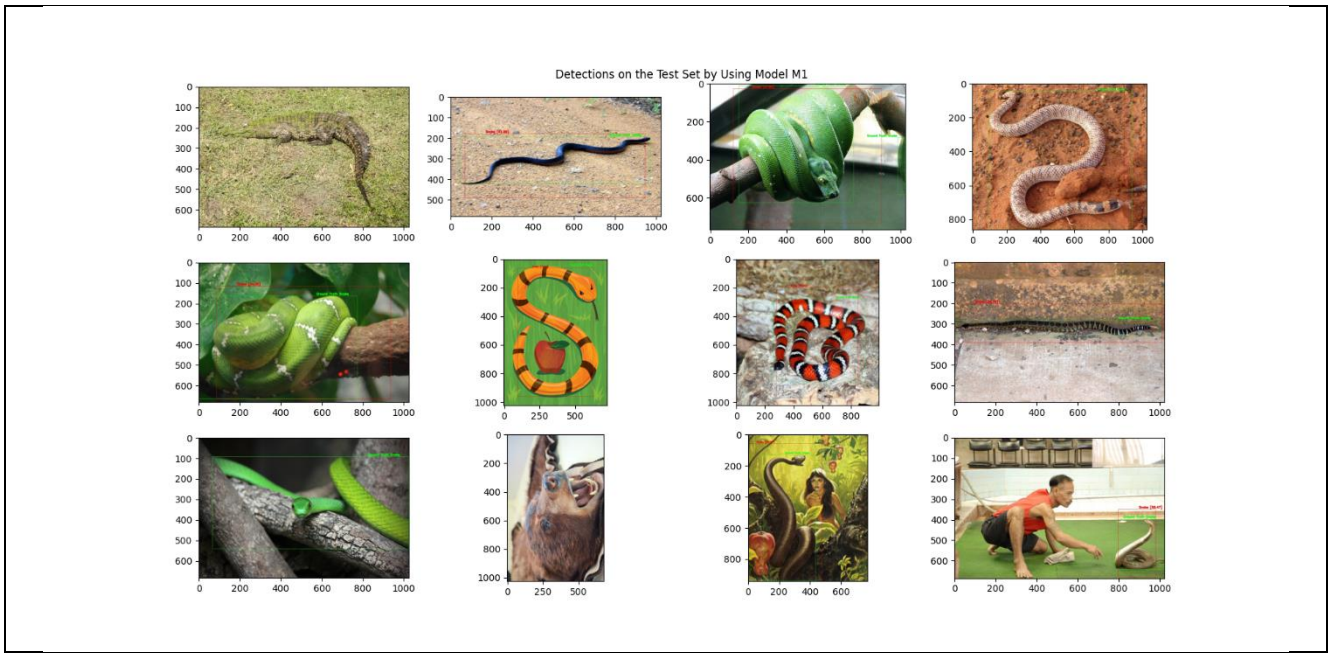

Figure 8. *Predictions of model M2 on the test set.*

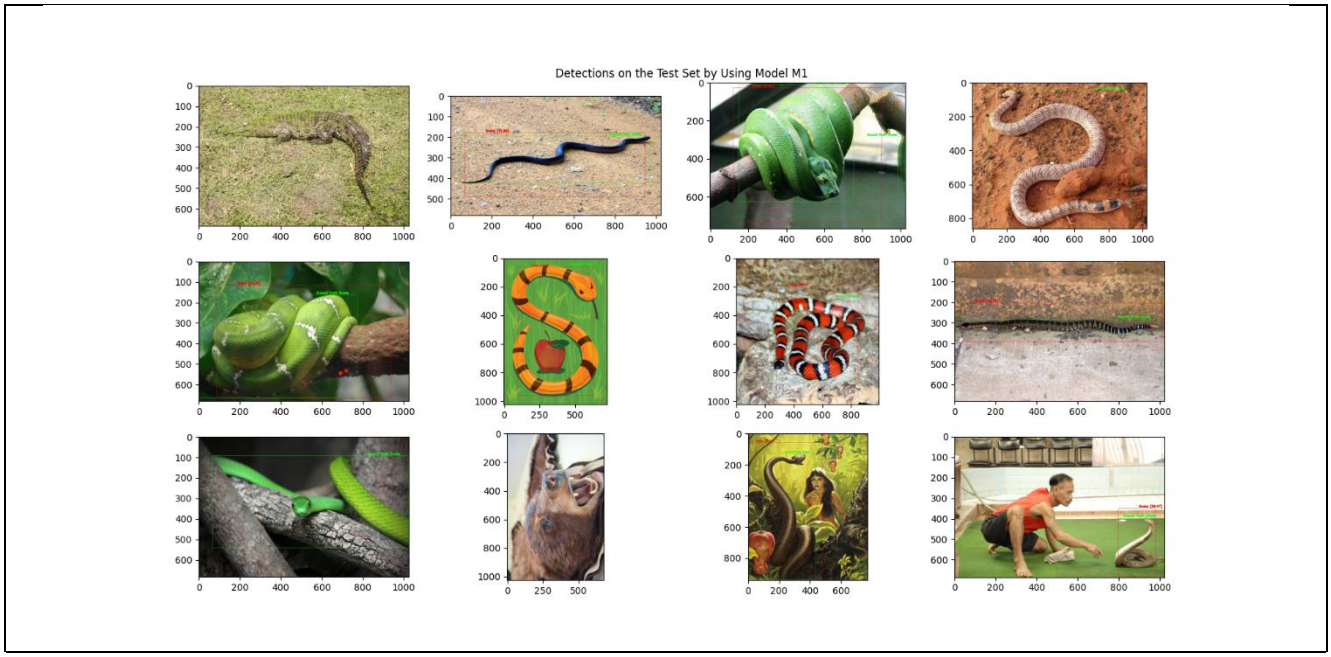

Figure 9. *Predictions of model M3 on the test set.*

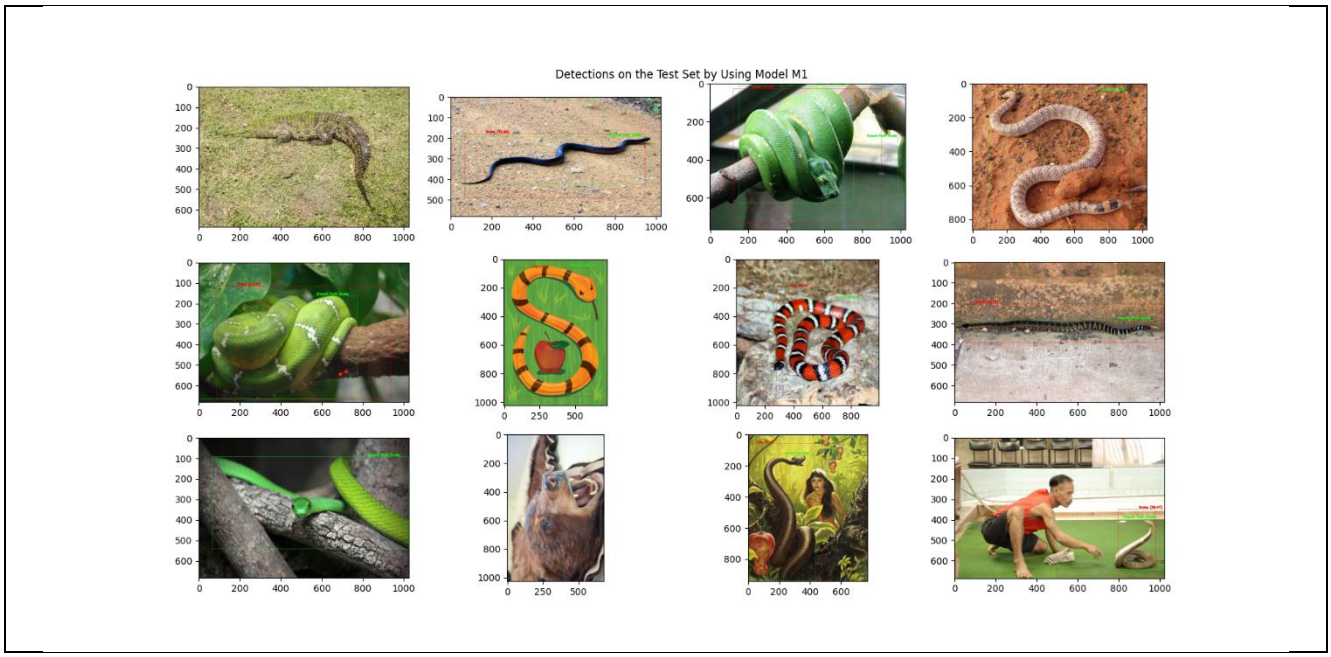

Figure 10. *Predictions of model M4 on the test set.*

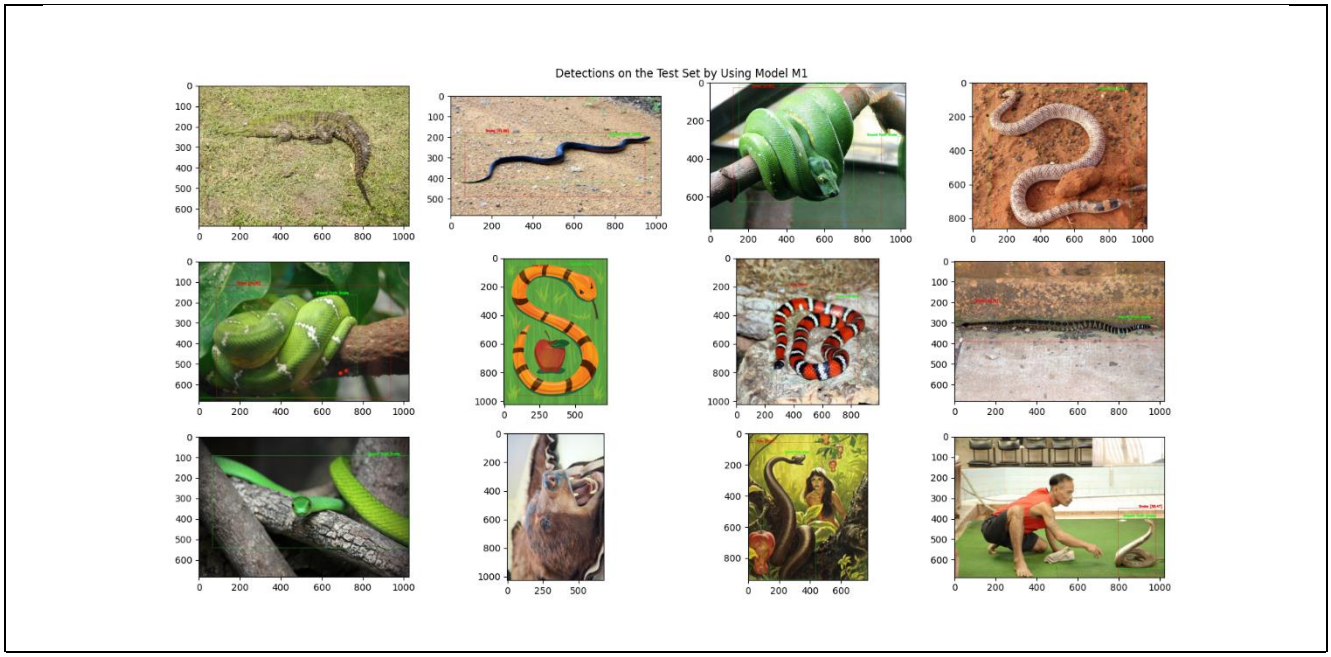

Figure 11. *Predictions of model M5 on the test set.*

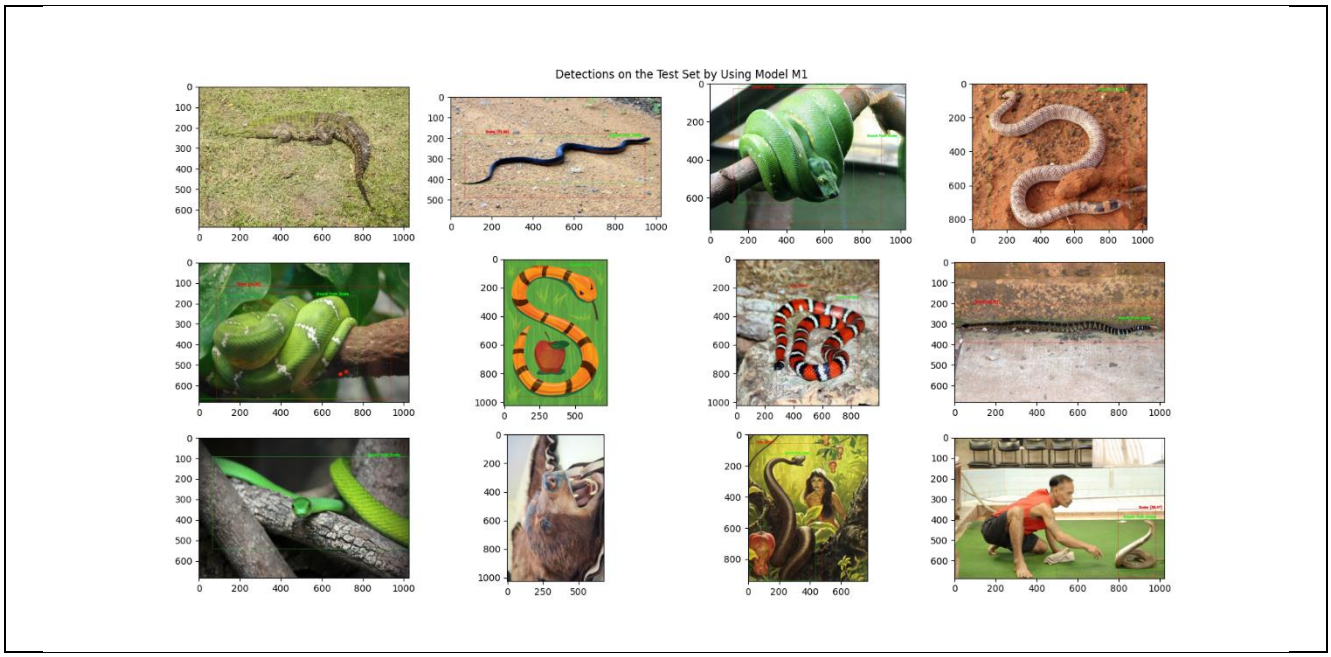

Figure 12. *Predictions of model M6 on the test set.*

With model M1, the non-snake images crocodile (1st row, 1st image on Figure 7) and bear (3rd row, 2nd image on Figure 7) are not predicted as snakes however, there are snake images (1st row, 4th image and 3rd row 1st image on Figure 7) that are not detected as snakes as well. Interestingly, the one and only model who does not predict the crocodile as snake is model M1. Additionally, the snake drawing (2nd row, 2nd image) image, is detected as a snake with only the model M1.

With model M2, bear (3rd row, 2nd image on Figure 8) is not predicted as a snake however, there are snake images (1st row, 4th image and 3rd row 1st image on Figure 8) that are not detected as snakes just like the predictions with model M1. Different than model M1, the image of a crocodile is predicted as snake with model M2.

With model M3, bear (3rd row, 2nd image on Figure 9) is not predicted as a snake however, there are again snake images (1st row, 4th image and 3rd row 1st image on Figure 9) that are not detected as snakes just like the predictions with both models M1 and M2. Also, in the first image the crocodile is predicted as a snake like the prediction of model M2.

With model M4, both the bear and the crocodile are predicted as snakes also, there is a snake image (3rd row 1st image on Figure 10) that is not detected as a snake. In this model M4, as an improvement, the snake in the 4th image at the 1st row of Figure 10 is detected a snake for the first time.

With model M5, there is a snake image (3rd row 1st image on Figure 11) that is not detected as a snake. In this model M5, the snake in the 4th image at the 1st row of Figure 11 is detected a snake like it is detected with model M4. On the other, this model again predicts both the crocodile and bear images as snakes.

With model M6, now there is not a real snake image that is not detected as a snake except the one with the drawing. Also, both the crocodile and bear images are detected as snakes.

In the figures 8, 9, 10, 11, and 12, there is an image of a snake drawing that cannot be detected as a snake with the majority of the models (except the first model M1). Therefore, in order to detect an image of a snake drawing, more snake drawing images should be added to the training set.

4. Conclusions and Future Work

In this paper, a model for snake detection and blurring has been proposed. Six different models using state-of-the-art object detection algorithm YOLOv4 have been explored. The results of this work indicate that different models encountered with different number of iterations on training phase of the deep learning, can vary in terms of detection of different kind of snake images. Generally, it can be said that more iterations would increase the precise detection of the snake images using the dataset that is used in this experiment.

To get the application useful in the real life, there are many possible future directions to improve the work: segmentation and tracking. In the testing of the models, it is observed that there are some body parts of the snakes that are not covered with the bounded box, also there are some areas in the bounded boxes that do not belong to

the area where the snake objects reside. So, when it comes to the real-life application, viewer will see some parts of the snake objects occasionally and after blurring operation is completed some unrelated parts of an image will be unseen. Therefore, the quality of a visual experience will end up decreasing for some level. In order to solve this problem, instead of bounding box detection, segmentation could be performed. With segmentation, more precise blurring operation would be completed, and the experience of a viewer would increase with a training using enough number of segmentation annotated images.

Also, if this approach will be adapted to be used on the videos, by implementing snake objects' tracking, the number of the snakes where the detection and blurring applied, on the frames would be increased.

Declaration of Interest

The authors declare that there is no conflict of interest.

Acknowledgements

An earlier version of this paper was presented at the ICADA 2021 Conference and was published in its Proceedings (Title of the conference paper: "Snake Detection and Blurring System Using Deep Learning").

References

[1] E. Landova, J. Maresova, O. Simkova, V. Cikanova and D. Frynta, "Human responses to live snakes and their photographs: Evaluation of beauty and fear of the king snakes," Journal of Environmental Psychology, vol. 32, no. 1, pp. 69-77, 3 2012.

[2] A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020.

[3] M. Tan, R. Pang and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," 2020.

[4] OpenCV team, "About," OpenCV, 2021. [Online]. Available: https://opencv.org/about/. [Accessed 20 May 2021].

[5] Google Inc., "Open Images V6 Snake Category Detection Type," 2021. [Online]. Available: https://storage.googleapis.com/openimages/web/visualizer/index.html?set=train&type=detection&c=%2Fm%2F078jl. [Accessed 20 May 2021].

[6] Google Inc., "Overview of Open Images V6," Google Inc., 2021. [Online]. Available: https://storage.googleapis.com/openimages/web/factsfigures.html. [Accessed 21 May 2021].

[7] AlexeyAB, "Yolo v4, v3 and v2 for Windows and Linux," 15 5 2021. [Online]. Available: https://github.com/AlexeyAB/darknet. [Accessed 21 May 2021].