

A Novel System for Potential Mosquito Breeding Hotspot Intimation and Monitoring Using MLOps and Improved YoloV3



Sonali Bhutad^{1*}, Kailas Patil²

Department of Computer and Technology, Vishwakarma University, Pune 411048, India

Corresponding Author Email: kailaspatil@vupune.ac.in

<https://doi.org/10.18280/i2m.220105>

ABSTRACT

Received: 12 January 2023

Accepted: 13 February 2023

Keywords:

vector-borne disease control, mosquito breeding site detection, sustainability, cloud infrastructure, object detection, MLOps, deep learning, pruning

Vector-borne disease control is an important issue faced by mankind. Many existing systems perform the detection and prevention of mosquito breeding sites using UAV-based methods. However, they don't provide real-time monitoring and detection of the same on daily basis. This study proposes a cloud-based deep-learning system to control the disease spread at a high scale. The implemented system does continuous monitoring using the existing public cameras for the potential mosquito breeding hotspots, further, the corresponding location coordinates will be forwarded to the local authorities. A history of the location coordinates maintained at the remote server will help monitor hotspots. We evaluated the current approach results and discovered that by layer pruning, the accuracy is improved by 14% and further reduces execution time by 10 sec. For accuracy and execution time calculation, the pruned model was tested on the validation dataset, and then the comparison was done with the original deep learning model. This indicates that the system can accurately detect the number of potential mosquito breeding sites. These results are expected to support decision-making on rapid resource allocation for vector control actions on a regular basis by achieving the sustainability goal of UNSDG (3).

1. INTRODUCTION

Around 50 to 100 million dengue infections occur worldwide yearly, per the World Health Organization [WHO]. In 2020, more than 2.3 million cases of dengue were reported in the Americas alone, most of them in Brazil [1]. The mosquito *Aedes aegypti* is the transmitter of arboviral diseases such as dengue, zika, chikungunya, and urban yellow fever [2]. It is estimated that there are 200,000 yellow fever clinical cases, causing 30,000 deaths per year in the world [3]. Infection during pregnancy such as congenital malformation and microcephaly happens due to the Zika virus; children in these conditions rarely develop normally [4]. These diseases also have a strong economic impact. A survey conducted in 17 Central and Latin American countries estimates that the cost of dengue epidemics exceeds US\$ 3 billion annually [5]. These facts make the arboviruses transmitted by *Aedes aegypti* one of the leading global health problems. Thus, for effective control of vector-borne diseases, it is essential to perform targeted environmental and ecosystem management. However, providing large-scale detailed environmental information to control vector-borne diseases remains a challenge. There are two reasons behind this, the first is local government authorities perform spraying twice or thrice a year. Due to this, it is not possible to control the vector spread as per the occurrence. Even though various initiatives have been taken to involve citizens in collecting and reporting hotspot data [6], we still lack the monitoring of the hotspot. The pivotal piece of such information is the extent of potential mosquito breeding sites that consist of stagnant water. But currently, no system is available for stagnant water notification. Hence, a system for stagnant water notification and monitoring was

created to detect all types of stagnant water surfaces in containers and puddles. This paper proposes a system that will use security cameras in public areas. Whilst video detection will be done throughout the day. Then upon finding the stagnant water presence geotagged images will be sent to give notifications to local authorities and then monitoring of the same can be done. The existing system such as [7] uses UAV-based detection systems to perform the identification of mosquito breeding hotspots. Similarly, in [8] the author has implemented a GPS and UAV-based system for detecting containers, and [9, 10] uses aerial and street-view images for container detection. None of the above systems focus on notifying the authorities and further monitoring the hotspots. In addition to the existing paper survey, we conducted a survey using a Google form questionnaire to understand the need of the hour. This survey was conducted over a group of 306 people residing in Maharashtra (India). Out of 306, 63.7% of responses showed that they informed local authorities about the mosquito breeding site while 56.1% of people accepted negligence in the timing during the pesticide spraying. Figure 1 shows, in some areas of Maharashtra, precautionary measures are still not applied effectively. Hence, we designed a system based on a deep learning algorithm yolov3 and cloud infrastructure. This system not only gives hotspot notifications in the form of longitude and latitude but also keeps monitoring the area. For monitoring, a history of the coordinates will be maintained, and an assessment of the eradication mechanism will be done. Table 1 shows the comparison with the existing systems and clearly indicates the need for automatic intimation and monitoring in order to create a complete system for potential hotspot detection.

Table 1. Comparison with existing system

Mosquito Breeding Hotspot Detection System					
Author Name	Object	Water Type	Images	Real-Time Detection	Notification and Monitoring
Jared Schenkel [7]	Plastic bottle, Glass bottle, plastic lid, Bucket, Cup, Bag Can	Nil	UAV images with GPS coordinates	No	No
Daniel Trevisan Bravo [8]	Water Tank	Nil	UAV images	No	No
Passos [9]	Bottle, Pool, Bucket, Tire, Puddle, Water tank	Nil	Aerial images	Yes	No
Peter Haddawy [10]	Potted plant, Tire, Jar, Bin, Ceramic Bowl	Nil	Street View	No	No
Implemented System	All containers and Surface water	Black, Green, Muddy, Blue, Shiny	Street View	Yes	Yes

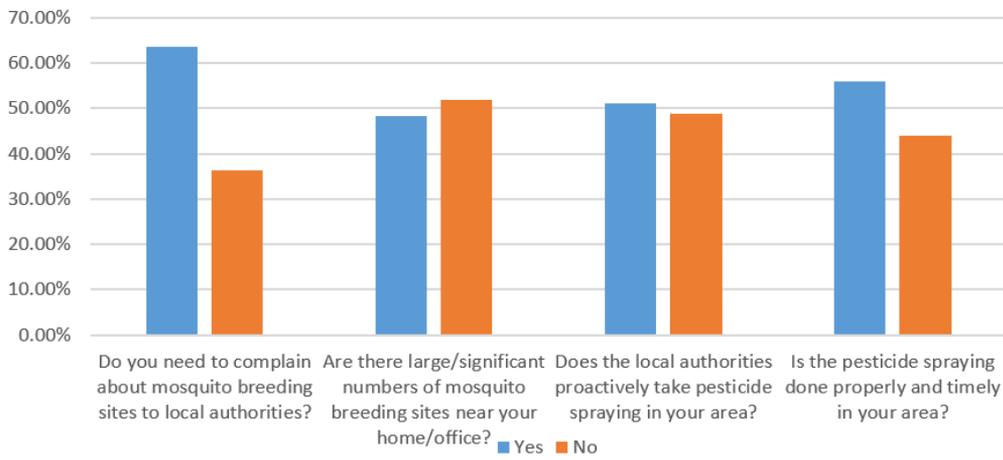


Figure 1. Mosquito breeding site survey

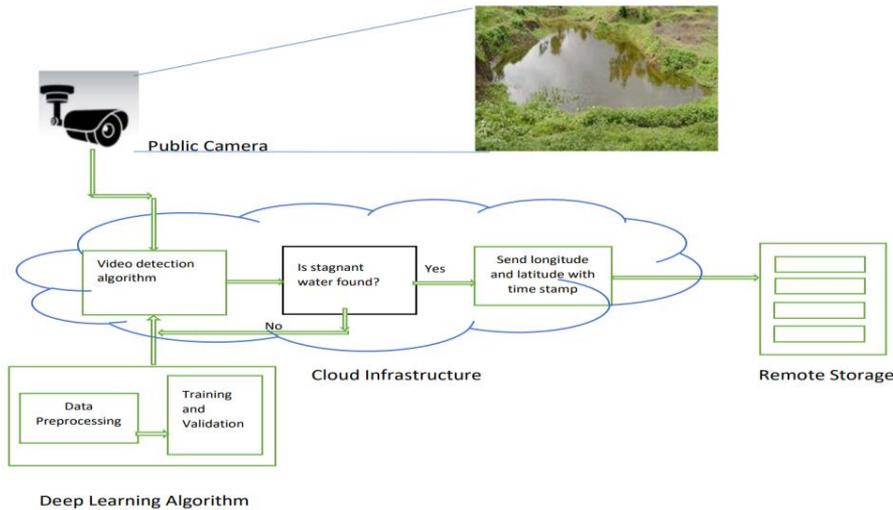


Figure 2. Mosquito breeding site notification and monitoring system

2. METHODOLOGY

Figure 2 shows the mosquito breeding site notification and monitoring system. The longitude and latitude of the detected part will be sent to the remote storage. Remote storage maintains the history of the hotspots. Further, this data can be

verified to assess the impact of the spraying performed by the local authorities. Public cameras can be of two types a) Analog and b) Digital Hd cameras. The minimum resolution would be 1080 pixels. The flow of the activities for the implemented system would be as below.

Workflow

- Step 1: Dataset preprocessing was done for the stagnant water dataset.
- Step 2: Training and validating the deep learning model were done.
- Step 3: The trained model was deployed on the cloud.
- Step 4: Public Camera Video streaming was sent to the cloud.
- Step 5: A Deep learning algorithm was executed for detecting stagnant water.
- Step 6: If water was found, then location coordinates will be sent to the remote server else the next video was considered for frame detection.
- Step 7: The history of hotspots was stored on the remote server.

2.1 Cloud infrastructure

We trained our model on Google Colab and then it was deployed on the cloud whilst detection was done for the received video recording from the public cameras.

To provide continuous monitoring the MLOps service of Microsoft Azure Cloud was used [11, 12]. Further, the corresponding location coordinates were stored at the remote server for vector control action to be done by the local authorities.

2.2 Deep learning algorithm

Deep learning is a subset of machine learning methods based on artificial neural networks. Here we used a deep learning algorithm known as YoloV3, to provide accurate real-time detection.

As shown in Figure 2, the algorithm includes two steps: 1) Data Preprocessing and 2) Training and Validation.

2.2.1 Data Preprocessing

To validate the model, the stagnant water image and video datasets were published on the Mendeley repository [13, 14]. This dataset contains 1976 labeled images of stagnant water. These images are street view images taken from top and side view for different water surfaces such as green, muddy, black, and shiny. The images were resized to 256 x 256 pixels and then labeling was done. These images were collected from two cities in India at different times of the day. In the detection step, part of the image will be classified into water or wet surface class. Hence images were labeled in Yolo format with a '0' value for the water class. While '1' is taken for the wet surface class. A few images were rotated by 90 degrees to increase the data samples.

2.2.2 Training and Validation

We trained Yolo family algorithms on Google Colab using transfer learning. For training and validation, the distribution of the dataset was done as per Table 2. For the dataset of 3812 images, the sources used, and the number of images captured are shown in the table. After experimenting with anchor box size [15], we chose YoloV3 out of other existing algorithms.

To increase object detection performance, data fusion was done by adding google images with real-time images [16]. Meng et al. [17] thought that "information fusion is the study of efficient methods for automatically or semi-automatically transforming information from different sources and points in time into a representation that provides effective support for human or automated decision making." The corresponding experimentation results are given in Table 3.

Table 2. Dataset distribution

Training Set			Validation Set
Google	Android	Rotated by 90	Google
Chrome	phone	degrees	Chrome
1200	1976	100	536

Table 3. Model accuracy

Algorithm	Layers	mAP (%)	Detection Time
Yolov3	106	75.67	62 seconds
Yolov4-tiny	36	46.24	32 seconds
Yolov5	224	87.23	46 seconds
Improved Yolov3	94	90.06	52 seconds

The following training parameters were employed: number of batches=64; number of subdivisions=32; maximum number of iterations=45,000; and the learning rate=0.001. After 16 hours, the training was interrupted with a validation loss value of 0.04, at iteration 41,200.

Yolov3 Detection Process

Yolov3 mainly utilizes residual layers and a certain number of convolutional layers to complete the detection process and uses the entire image features to predict each bounding box. This algorithm performs real-time detection at a fast speed. At the same time, it predicts all classes of all bounding boxes for the complete training, which maintains maximum average accuracy and strong real-time performance [16].

It divides the input image into N×N grids and assigns one anchor bounding box for each ground truth object. Equations 1-4 show four coordinates (tx, ty, tw, th) predicted by the network for each bounding box, and then a function was used to predict three corresponding parameters in the form of coordinates like the center point coordinates (cx, cy) of the bounding box, the width- bw, and the height- bh [17]. Where $\sigma(\bullet)$ is the sigmoid activation function, which was used to limit the center like the center point coordinates (bx, by) of the bounding box, the width bw, and the height bh. The confidence in the detected object was calculated by the formula given below,

$$bx = \sigma(tx) + cx \quad (1)$$

$$by = \sigma(ty) + cy \quad (2)$$

$$bw = p_w e^{tw} \quad (3)$$

$$bh = p_h e^{th} \quad (4)$$

Pruning

As Yolov3 has 106 layers, model compression was needed to reduce the model parameters and make it easy to deploy. The methods of model compression are categorized into quantization [18-20], pruning [21, 22], low-rank decomposition [23, 24], and knowledge distillation [25, 26]. Out of mentioned techniques, we used layer pruning, wherein layers were removed according to the loss and Bfvalue. Yolov3 has good detection accuracy, but to reduce time complexity and memory constraints, we performed pruning which contributes to the carbon footprint reduction as well [27].

Evaluation Metrics

As the problems of missed detection and false detection happened in the monitoring process of water detection, hence for experimentation mAP, Precision, and Recall were used as evaluation parameters. The precision was the ratio of the number of correctly detected water/wet areas to the total number of detected water/wet areas. The Recall was the ratio of the number of correctly detected water/wet areas to the total number of water/wet in the data set. The calculation method is shown in Eqns. (7) and (8).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (6)$$

$$mAP = \frac{1}{nc} \sum_{i=0}^{nc} AP_i \quad (7)$$

$$AP_i = \frac{1}{nd} \sum_{j=0}^{nd} P_{ij} \quad (8)$$

In these formulas, True Positive (TP) indicates the number of correctly detected water/wet surfaces, True Negative (TN) indicates the number of correctly detected backgrounds, False Positive (FP) indicates the number of incorrect detections, and False Negative (FN) indicates the number of missed detections, respectively. As shown in equations 9 and 10, mAP is the average of AP (average precision), and AP is the average of all categories, i.e., classes used for labeling. The mean average precision was calculated by taking an average of precision for several recall values.

Sustainability

Sustainability contributes towards fulfilling the needs of present generations without compromising the needs of coming generations, and at the same time maintaining a balance between the environment and economy for social well-being. This system indirectly contributes to the United Nations Sustainability Goal i.e UNSDG (3) ‘‘Good Health and well-being’’ which aims to avoid premature deaths by preventing disease spread.

3. RESULTS AND DISCUSSION

3.1 Prediction of hotspot

For testing, the platform used was a desktop computer with an Intel i5 1035G1 (1.19 GHz) dual-core CPU, a GeForce MX250, 2GB GPU (384 CUDA cores), and 8 GB of memory, running on a Windows 10, 64-bit system. The software tools used included CUDA 10.2, CUDNN 5.0, OpenCV3.0, and Microsoft Visual Studio 2019. For yolov5, Google colab was used for testing and training. To verify the effectiveness of the detection, the test data set of google images was used. Further, we analyzed the experimental data and compared the results. During the training of yolov3, the generated Bfvalue was observed to remove the convolution layers of the network. Therefore, a total of 12 layers were removed to reduce the model size from 234 MB to 164 MB. Eventually, we could

reduce the detection time by 10 seconds, with an increase in the mAP value by 14%. Details of these are given in Table 3.

For the captured image shown in Figure 2, the corresponding result of water detections with accuracy is shown in Figure 3.

The stagnant water detection for a captured image is performed using two classes: wet surface and water. The extra class i.e., wet surface was taken to avoid misclassification in water detection and to increase the accuracy. The accuracy received for each class is given in Table 4. We observed that Improved Yolov3 is more accurate in distinguishing water from wet surfaces as compared to other algorithms. Hence Improved version of Yolov3 was deployed on the cloud for potential mosquito breeding site detection.



Figure 3. Detection result

Table 4. Improved Yolov3 class-wise accuracy

Sr. No	Class	Accuracy
1	Water	95%
2	Wet Surface	86%

The sample database table with the location coordinates is shown in Table 5.

Table 5. Location coordinates

Sr. No	Latitude	Longitude	Img_name	Result
1	18.932500	74.2626845	pic_18.932	water
2	18.931594	74.2628658	pic_18.931	Wet surface
3	7.9806446	49.818794	pic_7.9806	water

The performance evaluation of the developed approach was based on 536 images destined for the test. As in the case of object detection, the AP was computed for each image class. Then, the mAP was calculated from AP values. To classify 536 images, Improved Yolov3 took 52 seconds. From 5,896 ground truth bounding boxes, 5310 were correctly classified (TP values) providing an mAP-50 of 0.9006 and a recall rate of 0.8723. In addition, 586 cases of false negatives (FN) and false positives (FP) were computed as shown in Figure 4, wherein false objects are detected as water. This has happened due to improper sunlight and fewer sample images of a particular type. The results presented in Table 5 demonstrated good performance (95%) for Yolov3 scenarios. Its worst performance occurred in the detection of water with reflection and shiny water, probably due to the low occurrence of this type of scenario in the training images.

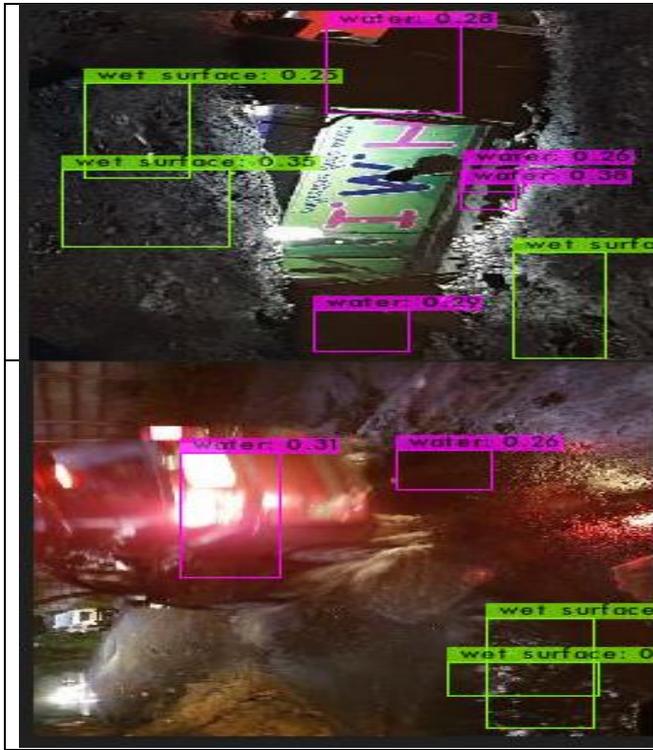


Figure 4. False detections

4. CONCLUSION

The cloud-based mosquito breeding detection model can automatically extract useful information and give intimation to the local authorities. Which will help control the vector spread with the existing public cameras. Detection experiments on stagnant water images show the effectiveness of the pruned YoloV3 deep-learning model. Furthermore, we also provide Hotspot location coordinates to perform spraying for vector control and monitoring. Thus, the system provides a sustainable and cost-effective way to implement vector eradication at a high scale. In the future, mosquito detection and classification can be added to the system for further analysis and prediction of diseases.

REFERENCES

[1] Rückert, C., Weger-Lucarelli, J., Garcia-Luna, S. M., Young, M. C., Byas, A. D., Murrieta, R. A., Fauver, J.R., Ebel, G. D. (2017). Impact of simultaneous exposure to arboviruses on infection and transmission by *Aedes aegypti* mosquitoes. *Nature communications*, 8(1): 15412. <https://doi.org/10.1038/ncomms15412>

[2] Pan American Health Organization, Reported Cases of Dengue Fever in The Americas, <https://www.paho.org/en/topics/dengue>, accessed on Dec. 1, 2023.

[3] World Health Organization, Zika virus, <https://www.who.int/news-room/factsheets/detail/zika-virus>, accessed on Dec. 27, 2022.

[4] Pan American Health Organization, Yellow Fever, <https://www3.paho.org/hq/index.php?option=comcontent&view=article&id=9476:yellowfever&Itemid=40721&lang=en>, accessed on Sep. 23, 2022.

[5] Laserna, A., Barahona-Correa, J., Baquero, L., Castañeda-Cardona, C., Rosselli, D. (2018). Economic impact of dengue fever in Latin America and the Caribbean: a systematic review. *Revista Panamericana de Salud Pública*, 42: e111. <https://doi.org/10.26633/RPSP.2018.111>

[6] Agarwal, A., Chaudhuri, U., Chaudhuri, S., Seetharaman, G. (2014). Detection of potential mosquito breeding sites based on community sourced geotagged images. In *Geospatial InfoFusion and Video Analytics IV; and Motion Imagery for ISR and Situational Awareness II* 9089: 175-182. <https://doi.org/10.1117/12.2058121>

[7] Bravo, D.T., Lima, G.A., Alves, W.A.L., et al. (2021). Automatic detection of potential mosquito breeding sites from aerial images acquired by unmanned aerial vehicles. *Computers, Environment and Urban Systems*, 90: 101692. <https://doi.org/10.1016/j.compenvurbysys.2021.101692>

[8] Schenkel, J., Taelle, P., Goldberg, D., Horney, J., Hammond, T. (2020). Identifying potential mosquito breeding grounds: Assessing the efficiency of UAV technology in public health. *Robotics*, 9(4): 91. <https://doi.org/10.3390/robotics9040091>

[9] Passos, W.L., Araujo, G.M., de Lima, A.A., Netto, S.L., da Silva, E.A. (2022). Automatic detection of *Aedes aegypti* breeding grounds based on deep networks with spatio-temporal consistency. *Computers, Environment and Urban Systems*, 93: 101754. <https://arxiv.org/abs/2007.14863>

[10] Haddawy, P., Wettayakorn, P., Nonthaleerak, B., et al. (2019). Large scale detailed mapping of dengue vector breeding sites using street view images. *PLoS Neglected Tropical Diseases*, 13(7): e0007555. <https://doi.org/10.1371/journal.pntd.0007555>

[11] Fitzgerald, B., Stol, K.J. (2014). Continuous software engineering and beyond: trends and challenges. In *Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering*, pp. 1-9. <https://doi.org/10.1145/2593812.2593813>

[12] Testi, M., Ballabio, M., Frontoni, E., Iannello, G., Moccia, S., Soda, P., Vessio, G. (2022). MLOps: A taxonomy and a methodology. *IEEE Access*, 10: 63606-63618. <https://doi.org/10.1109/ACCESS.2022.3181730>

[13] Bhutad, S., Patil, K. (2022). Dataset of stagnant water and wet surface label images for detection. *Data in Brief*, 40: 107752. <https://doi.org/10.1016/j.dib.2021.107752>

[14] Bhutad, S., Patil, K. (2022). Dataset of road surface images with seasons for machine learning applications. *Data in Brief*, 42: 108023. <https://doi.org/10.1016/j.dib.2022.108023>

[15] Bhutad, S., Patil, K., Khare, N. (2023). Differentiating stagnant water from wet surface for detecting potential mosquito breeding sites in real time. *Indian Journal of Science and Technology*, 16(5): 331-338. <https://doi.org/10.17485/IJST/v16i5.2111>

[16] Alcorn, M.A., Li, Q., Gong, Z., Wang, C., Mai, L., Ku, W.S., Nguyen, A. (2019). Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4845-4854.

[17] Meng, T., Jing, X., Yan, Z., Pedrycz, W. (2020). A survey on machine learning for data fusion. *Information Fusion*, 57: 115-129. <https://doi.org/10.1016/j.inffus.2019.12.001>

- [18] Ge, J., Zhang, D., Yang, L., Zhou, Z. (2019). Road sludge detection and identification based on improved Yolov3. In 2019 6th International Conference on Systems and Informatics (ICSAI), IEEE, pp. 579-583. <https://doi.org/10.1109/ICSAI48974.2019.9010486>
- [19] Gong, Y., Liu, L., Yang, M., Bourdev, L. (2014). Compressing deep convolutional networks using vector quantization. arXiv preprint arXiv:1412.6115. <https://doi.org/10.48550/arXiv.1608.08710>
- [20] Dettmers, T. (2015). 8-bit approximations for parallelism in deep learning. arXiv preprint arXiv:1511.04561. <https://doi.org/10.48550/arXiv.1511.04561>
- [21] Mozer, M., Smolensky, P., Touretzky, D. (1989). *Advances in Neural Information Processing Systems*; Morgan Kaufman: Burlington. MIT Press USA.
- [22] Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P. (2016). Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710. <https://doi.org/10.48550/arXiv.1608.08710>
- [23] Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I., Lempitsky, V. (2014). Speeding-up convolutional neural networks using fine-tuned cp-decomposition. arXiv preprint arXiv:1412.6553. <https://arxiv.org/abs/1412.6553>
- [24] Oseledets, I.V. (2011). Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(2011): 2295-2317. <https://doi.org/10.1137/090752286>
- [25] Hinton, G., Vinyals, O., Dean, J. (2015). Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531. <https://doi.org/10.48550/arXiv.1503.02531>
- [26] Zagoruyko, S., Komodakis, N. (2016). Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928. <https://doi.org/10.48550/arXiv.1612.03928>
- [27] <https://www.technologyreview.com/2019/06/06/239031/training-a-single-ai-model-can-emit-as-much-carbon-as-five-cars-in-their-lifetimes/>, accessed on 27/12/2022.