COMPUTER METHODS IN MATERIALS SCIENCE

2024, vol. 24, no. 1, 5–14
https://doi.org/10.7494/cmms.2024.1.0824

Development of an object identification algorithm for the forging industry based on standard vision systems

Adrian Litwa*, Łukasz Madej

AGH University of Krakow, Mickiewicza 30 av., 30-059 Krakow, Poland.

Abstract

The work aims to develop an algorithm for identifying objects in a forging plant under production conditions. Particular emphasis is placed on the accurate detection and tracking of forgings that are transferred along the forging line and, if possible, detection will also cover employees controlling and supporting the operation of forging machines, all of this with the use of standard vision systems. An algorithm prepared in such way will allow the performance of effective detections that will support activities related to the control of the movement of forging elements, the analysis of safety in workplaces, and the monitoring of compliance with Occupational Health and Safety Regulations by employees, as well as also allowing for the introduction of additional optimization algorithms that will further enrich the presented model, which may prove to be a long-term goal that will form the basis for subsequent work. Three algorithmic solutions with different levels of complexity were considered during the research. The first two are based on artificial neural network solutions, while the last one utilizes classical image processing algorithms. The datasets for training and validation in the former cases were generated based on the recordings taken from standard cameras located in the forging plant. Data were acquired from three cameras, two of which were used to create training and validation sets, and a third one was used to verify how the developed algorithms would work in a variable environment that was previously unknown to the models. The impact of model parameters on the results is presented at this stage of the research. It has been proven that machine learning-based solutions cope very well with object detection problems and achieve high accuracies after a precise selection of hyperparameters. Algorithms show the performance of detections with excellent accuracy of 92.5% for YOLOv5 and 94.3% for Mask R-CNN. However, a competitive solution using only image transformations without machine learning showed satisfactory results that can also be obtained with simpler approaches.

Keywords: machine learning, artificial neural networks, YOLOv5, Mask R-CNN, forging industry, object identification, vision systems

1. Introduction

Vision systems are currently used in almost every area of life to monitor the behaviour of people, animals, cars or other objects, providing a significant amount of data. As a result, despite the issues related to storage capabilities, the problem of efficient data extraction from the recordings for further processing has become critical. In the past, specially trained personnel performed such analyses, looking through extensive materials to find the appropriate data and draw relevant conclusions. Subsequently, computers took over this role, and now use dedicated algorithms to perform such actions in the same or significantly shorter time (Kühl et al., 2020).

*Corresponding author: nairda6666@gmail.com
ORCID ID's: 0009-0007-9257-9652 (A. Litwa), 0000-0003-1032-6963 (L. Madej)

 ISSN 1641-8581

The current challenge is to create an algorithm that would not only be able to analyze recordings but also learn how to recognize objects with the highest possible precision and shortest time. These algorithms can not only work on prerecorded data but also on live feeds, which ensures real-time analysis. Such approaches allow the development of systems that are used to detect threats, conduct thermal control measurements or optimize and improve work safety in various environments, such as factories and enterprises.

Artificial neural networks (ANN) (Krogh, 2008) have emerged as the most important element used in the operation of such systems. Thanks to graphics cards and deep learning capabilities, ANNs that were initially able to solve elementary problems can now perform complex analyses in a short time. The deep learning (DL) technique (Sarker, 2021) allows the creation of an algorithm resembling the human brain, in which the number of hidden layers (Uzair & Jamil, 2020), unlike the ordinary neural networks, can exceed tens, hundreds or even thousands of layers, which improves accuracy and quality of the results obtained. With the development of artificial neural networks, the idea of more complex networks called convolutional neural networks (CNN) (Indolia et al., 2018), which use deep learning techniques, became more frequently used in practical applications. Such networks allow the precise analysis of images while limiting the number of weights with the use of an appropriate filter called a kernel (Zhuang et al., 2021). Thanks to that procedure, it is possible to reduce the number of network layers several or even dozens of times in some cases.

Every CNN network consists of five layers. An Input Layer is responsible for image representation, a Convolutional Layer is responsible for limiting the number of weights in the network, a Pooling Layer is used to reduce the dimensions of the image while retaining the most important information, a Dropout Layer is used to avoid overtraining of the network and a Flatten Layer, converts a multidimensional surface into a vector. These networks have been successfully used in image analysis to detect, e.g., facemasks (Bhadani & Sinha, 2020), dangers on the road for self-driving vehicles (Goenka et al., 2022) or patients location in hospitals (Sharma et al., 2021).

Therefore, the evaluation of the capabilities of these networks in object tracking under genuine industrial conditions of the forging plant became the motivation for the current research. Two common approaches were selected for the investigations. The first is Mask R-CNN (He et al., 2017), which uses convolutional networks and allows for accurate detection of objects on images in a two-stage process. At the beginning, the algorithm determines regions of interest, which are then classified, and, based on the regression, appropriate boundary boxes are drawn. This approach is additionally extended by one more stage responsible for determining masks for each of the identified objects. A model like this allows for highly accurate detection, which unfortunately has the consequences of long training and operating time (Hassan et al., 2022a). Therefore, the other selected network, YOLOv5 (Karthi et al., 2021), was intended to speed up operation without affecting detection effectiveness. Unlike the Mask R-CNN model, detection only takes place in one stage, so the determination of areas of interest, classification and bounding boxes are generated simultaneously.

However, for comparison purposes, a third method based on classical image processing algorithms was also evaluated during the research. The last method is based only on image transformations, such as colour conversion, blurring or thresholding. This approach should allow for the simple detection of objects in the recordings and, consequently, the determination of bounding boxes in a short time frame.

The presented models will be adapted to detect the position of the forgings in a hot forging line as a case study of a real industrial environment. Each of the tested models will be run on a computer with an Intel Core i9 10900 and a NVIDIA GeForce RTX 3080 Ti graphics card. Additionally, each algorithm uses different libraries and platforms used for machine learning. For YOLOv5, the PyTorch 1.7 library and the Python programming language version 3.9 were used, while in the case of Mask R-CNN, the library responsible for performing machine learning was Tensorflow version 2.2 in combination with Keras 2.3.1. The last algorithm that does not use machine learning techniques was created based on the Python programming language in version 3.9.

2. ANN-based solutions

The development of appropriate datasets is a critical stage during the implementation of ANN models based on supervised learning (Jiang et al., 2020). The current datasets consist of images extracted directly from the standard industrial vision system overseeing activities in the plant. The locations of the forging and, where possible, the employees were marked in each photo and then divided into two datasets, training and validation, respectively. The first is intended for training the neural network, while the second is used to verify the effects of training. With that, it is possible to determine the quality of the developed neural network.

In the case of the YOLOv5 algorithm, each object of interest from the image is marked with simple rectangles (Fig. 1a), while in the case of Mask R-CNN, objects are marked in a more accurate way by generating a polygon that precisely describes the element (Fig. 1b). Initial datasets consisted of 50 images for training and 10 for validation purposes.

a)

b)

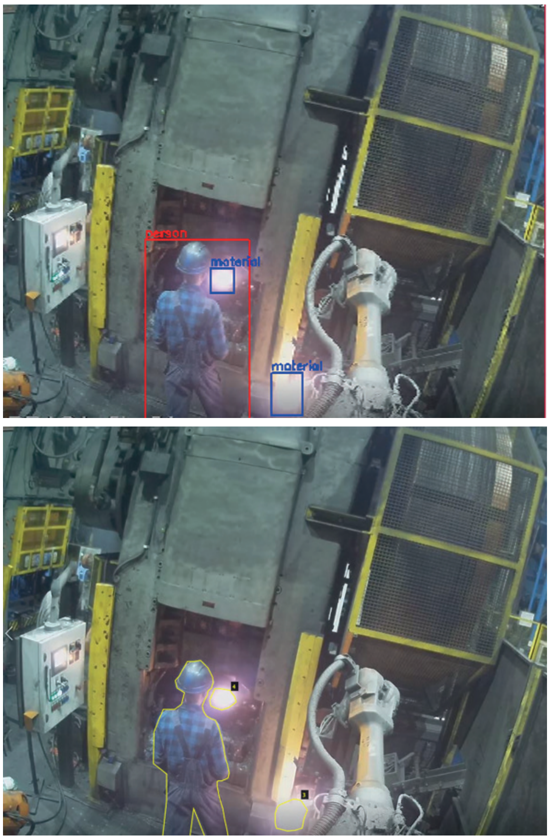

Fig. 1. Example of labeled images for:
a) YOLOv5; b) Mask R-CNN models

For each of the tested models, parameters influencing the quality of training operation and the identification results were investigated. The full list of tested parameters used for the YOLOv5 is presented in Table 1.

Table 1. Summary of tested parameters used for the YOLOv5 model

Parameter	Tested values
Pre-trained model	YOLOv5s, YOLOv5s6, YOLOv5m, YOLOv5m6, YOLOv5l, YOLOv5l6, YOLOv5x, YOLOv5x6
Image size [px]	640, 1280
Number of epochs	50, 100, 200, 300, 500
Batch size	2, 4, 8, 16, 32
Optimizer	SGD, ADAM
Device	GPU, CPU

Each of the parameters set was tested in various combinations to identify the most promising setup. The selection was based on the precision and the time needed to prepare such a model. Overall 21 case studies were evaluated. Various outcomes were obtained during this investigation, starting with the initial model, whose mAP50 (Otani et al., 2022) precision was at the level of 0.285, which did not allow for any detection, ending with a model whose precision was 0.925, allowing for accurate detection of forgings. As it turned out, the parameters responsible for the selection of the pre-trained model and the number of epochs had the greatest impact on the quality of the results. Only selected example models created on the basis of the presented parameters with the provided results are gathered in Table 2. Thanks to this, it can be concluded that even changing only one of the model parameters can significantly affect the detection results and the time needed to execute training.

Table 2. Sample models created based on the presented parameters

Pretrained model	Image size [px]	Number of epochs	Batch size	Device	Optimizer	mAP50	Training time [hours]
YOLOv5s	640	50	16	GPU	ADAM	0.826	0.011
YOLOv5s	640	100	16	GPU	ADAM	0.897	0.023
YOLOv5s6	1280	100	16	GPU	ADAM	0.902	0.235
YOLOv5m	640	100	16	GPU	ADAM	0.893	0.036
YOLOv5m6	1280	100	16	GPU	ADAM	0.859	1.608
YOLOv5l	640	100	16	GPU	ADAM	0.852	0.055
YOLOv5x	640	100	16	GPU	ADAM	0.831	0.670
YOLOv5m	640	200	16	GPU	ADAM	0.904	0.066
YOLOv5l	640	200	16	GPU	ADAM	0.894	0.101
YOLOv5x	640	200	16	GPU	ADAM	0.868	1.045
YOLOv5s	640	200	2	GPU	ADAM	0.747	0.122
YOLOv5s	640	200	8	GPU	ADAM	0.912	0.054
YOLOv5s	640	200	16	GPU	ADAM	0.925	0.042
YOLOv5s	640	200	16	GPU	SGD	0.894	0.038
YOLOv5s	640	200	16	CPU	ADAM	0.910	1.238

As shown above, the first investigated parameter was the type of the pre-trained model. It turned out that the least complex and extensive model is the most effective in the examined problem. This may be due to the simplicity of the objects being analyzed and the straightforward working environment. Based on this result, the YOLOv5s pre-trained model was selected, whose mAP50 efficiency on the validation set was over 0.8 and additionally, the time needed for training was slightly over 2 minutes with a number of epochs of 200. The next parameter tested was the number of epochs. Based on the analysis, it turned out that the model quickly reached the stage where subsequent iterations did not improve effectiveness and even worsened the results. Therefore, it was finally decided to use 200 training epochs. Another element that strongly influenced both the effectiveness and learning time was the batch size (Kandel & Castelli, 2020). When 50 training photos were used, the most optimal value was 16. The last parameter that influenced the model's effectiveness was the optimiser's selection. SGD optimizer (Hassan et al., 2022b) turned out to be less effective than the more popular ADAM optimizer (Jais et al., 2019), and the difference between them was around 3%. The final parameter that only influenced the learning time was the choice of the device on which the learning process was carried out. The GPU turned out to be undefeated, carrying out the process almost 30 times faster than the CPU. The YOLOv5 model developed in such a way was tested in a known environment, as shown in Figure 2.

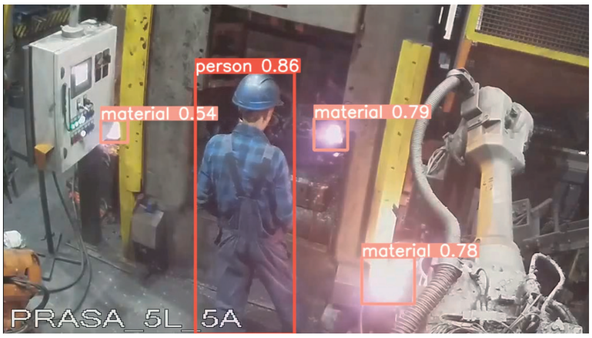

Fig. 2. Network performance in the known environment with a confidence threshold of 0.5

As can be seen, both the worker and the forgings were marked correctly and with sufficiently high precision. Based on this research, a final set of the YOLOv5 model was selected, as shown in Table 3.

The model coped perfectly in an unknown environment in which the entire process is automated by using a robot arm that moves the forgings. The algorithm indicated the location of the material with high precision, and since no employee was there, it did not mistake it for a robot.

Table 3. Summary of the parameters used for the YOLOv5 model

Parameter	Value
Pretrained model	YOLOv5s
Image size [px]	640
Number of epochs	200
Batch size	16
Device	GPU
Optimizer	ADAM

Finally, the capabilities of the developed ANN model were evaluated with a feed from a completely different, fully automatic forging line, as seen in Figure 3.

Fig. 3. The final test of the developed ANN with a confidence threshold of 0.7

The same investigation was realized for the development of the second model based on Mask R-CNN. All tested parameters are summarized and presented in Table 4. Overall, 15 case studies were evaluated.

Table 4. Summary of the parameters used for the Mask R-CNN model

Parameter	Tested values
Number of epochs	5, 10, 20
Number of steps in each epoch	20, 50, 100
Number of validation steps	10, 20, 50
Confidence threshold	0.3, 0.5, 0.7
Learning rate	0.02, 0.005, 0.001
Image size [px]	1024
Device	GPU, CPU

Table 5. Sample models created based on the presented parameters

Number of epochs	Number of steps in each epoch	Number of validation steps	Confidence threshold	Learning rate	mAP50	Training time [hours]
5	20	20	0.3	0.020	0.080	0.20
10	100	20	0.7	0.020	0.103	2.30
10	100	50	0.7	0.005	0.908	0.93
5	20	50	0.5	0.001	0.536	0.16
5	50	50	0.5	0.001	0.669	0.31
10	20	50	0.5	0.001	0.719	0.53
10	50	50	0.5	0.001	0.777	0.72
10	100	50	0.5	0.001	0.939	0.89
10	100	50	0.7	0.001	0.943	0.91

As in the case of the YOLOv5, models were created based on the presented parameters. Examples of the results and the parameters used are collected in Table 5. Also, in this case, changing even a single parameter could significantly impact the results. The first models that were developed did not allow any detection due to the poor selection of the learning rate parameter. It turned out that, apart from the number of epochs, the learning rate parameter has a significant impact on the results, which, when incorrectly selected, strongly limits the effectiveness of Mask R-CNN.

First, the influence of the number of epochs and the number of steps in each epoch was examined. It seems that increasing the number of epochs has a slightly greater impact than changing the number of steps, but both of these parameters significantly influence the quality of the mAP50 indicator. Based on these results, achieving an efficiency score of over 0.85 for the validation set was possible, assuming the number of epochs was ten and the number of steps was 100. The confidence threshold used for the results verification stage turned out to be an equally important parameter. In the case of the most extensive model, the threshold of 0.7 ended up being ideal and allowed to eliminate false indications and determine only the true ones. The learning rate parameter is also important in terms of the obtained results, but determining its value is not straightforward. Its value was initially assumed to be 0.02, which, after 120,000 iterations, was reduced tenfold. In the case of the investigated forging detection problem, the initial value resulted in an efficiency level of 0.1, which did not allow for proper detection. After an in-depth analysis of the operation of this parameter, it was shown that changing its value to 0.005 resulted in an increase in the effectiveness of mAP50 to a value of over 0.9. A further change of this parameter to the value of 0.001 resulted in an improvement to the level of 0.943. The effect of the identification by the most refined model is presented in Figure 4.

Fig. 4. The effect presented by the Mask R-CNN algorithm in the first environment

It can be seen that the algorithm coped perfectly with detection in a known environment, and the precision of the indications was close to or equal to 1. Based on this research, a final parameter set of the Mask R-CNN model is summarized in Table 6.

Table 6. Summary of the parameters used for the Mask R-CNN model

Parameter	Value
Number of epochs	10
Number of steps in each epoch	100
Number of validation steps	50
Confidence threshold	0.7
Learning rate	0.001
Image size [px]	1024
Device	GPU

Again, the capabilities of the developed ANN model were evaluated with a feed from a completely different, fully automatic forging line, as presented in Figure 5.

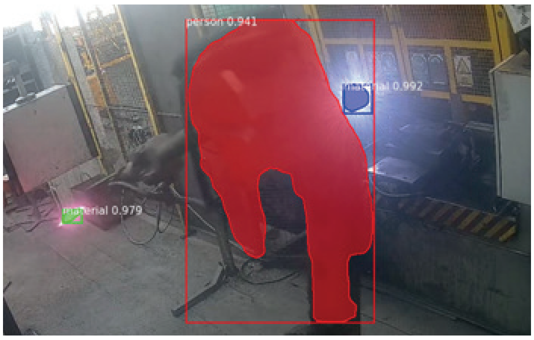

Fig. 5. The final test of the developed Mask R-CNN algorithm

As can be seen in Figure 5, the Mask R-CNN is not able to cope with an environment it has not seen before, which results in marking the robot arm as a human. To improve its operation, it would be necessary to train the network with a scenario in which human work is replaced by a robot. The model prepared in such a way should be able to present more reliable results. Another solution could be to add a new detection class to the model, which, after changes, would be able to detect employees, forgings and robot arms. This would require teaching the network examples of the appearance of such robots, which would increase the complexity of the algorithm but would improve the identification results. Simply expanding the existing model by improving its effectiveness would have a negligible impact on the achieved results.

As mentioned, the capabilities of both approaches developed were compared with the classical approach to data extraction based on image processing algorithms.

3. Image processing-based solution

The third model developed, based on image processing without the use of machine learning, does not allow for simultaneous detection of both the forging and the employees. In this case, only tracking of forgings positions is investigated. The developed algorithm is based on four stages.

The first is responsible for the conversion of the image from the colour scale into the grey one. A blurring operation is also applied to increase the differences between the strongly glowing forgings and the surroundings. Many types of blurring operations (Yamaura et al., 2018) can be used in this process. However, after a set of tests, a Gaussian blurring was selected for further investigation. Then, a thresholding operation based on the Otsu algorithm (Cao et al., 2021) was applied to the image to isolate the forgings from the surroundings. The image containing pixel values of 0 or 255 was finally subjected to a contour detection algorithm, which returns the coordinates of the beginning of the bounding boxes along with their width and height. Examples of the results obtained are shown in Figure 6.

Fig. 6. Positions of forgings detected by the image processing algorithm

As can be seen, the algorithm coped sufficiently well with the detection of the forgings in subsequent positions. The corresponding results from the fully automatic forging line are shown in Figure 7.

a)
b)

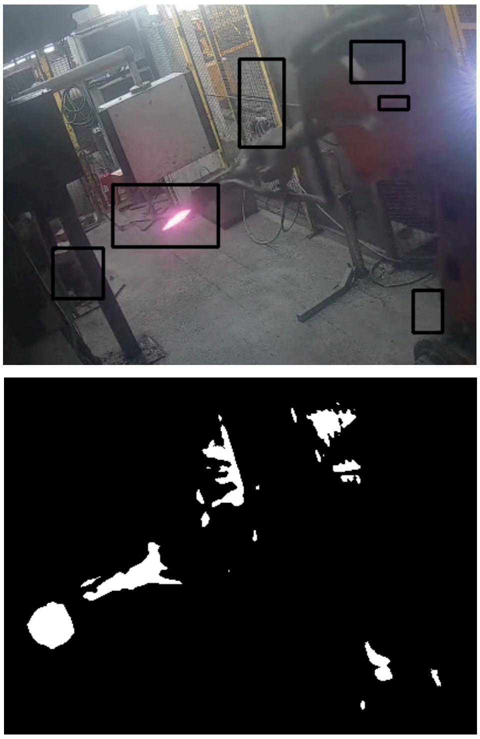

Fig. 7. Positions of forgings in the fully automatic forging line detected by the image processing algorithm and corresponding (a) and binary image (b)

Despite many elements distracting its operation in the test environment, the algorithm managed to indicate the position of the forging with sufficient effectiveness. Without a precise definition of the area in which the model should search for the location of the forgings, incorrect results can be expected due to the strong sources of light visible in the image, which affect the determination of boundary fields.

4. Comparison of results

Based on the best results obtained from the three models presented, an analysis of the effects of their operation was carried out. As a result, the most accurate approach for forgings detection was identified. The summary of the models' overall performance is presented in Table 7. The main emphasis was placed on the accuracy and effectiveness of object identification, but additional aspects like the time needed to prepare the model or the complexity of models are also included in Table 7.

The most important parameters that had the greatest impact on the final conclusions and the selection of the best model were the accuracy of the results, clearly indicating the quality of the solution and the ability to operate in the real-time due to the need for the algorithm to operate on a vision camera that continuously monitors the forge. While other parameters, such as implementation complexity, the time needed for training and identification and much more not presented in the table above are not elements that significantly influence the final choice, they do indicate differences between the models and allow for additional conclusions to be drawn that inspire further work on this topic.

Based on Table 7 the YOLOv5 model can be selected as the best approach to forging detection. The model can operate with high efficiency and an accuracy of over 0.9 and can identify each class of objects in various environments. The Mask R-CNN model performed similarly, but when used in an environment that the algorithm did not see at the training stage, it provided incorrect results. More effort is required to develop a reliable Mask R-CNN in this case. The last algorithm that did not require machine learning coped with the problem equally well, but not perfectly. The lack of a learning mechanism means that, in some cases, objects that were not forgings were identified. At the same time, the algorithm does not allow for the detection of employees. Therefore, a model like this can be considered when the execution time rather than the accuracy is the most important parameter.

Table 7. Summary of the performance results of three tested models

Aspect	Model		
	YOLOv5	Mask R-CNN	Image processing
Accuracy of results	Calculated based on the mAP50 parameter. mAP50: **0.925**	Calculated based on the mAP50 parameter. mAP50: **0.943**	No numerical parameter, visual verification of the effects required
Time needed to train the neural network	For the best model: **2.52 minutes**	For the best model, **55 minutes**	No time required for training
Time needed to identify objects	Images are processed immediately. It takes **30 seconds** to process a 48 seconds video	Images are processed immediately. It takes **3 minutes** to process a 48 seconds video	Images and videos are processed in real-time
Possibility to work in real-time	Objects identified in real time with high accuracy	Limitation to 5 frames per second is required for real-time operation	Objects identified in real-time with sufficient accuracy, but incorrect detections may appear
Complexity of implementation	Model requires extensive knowledge of image analysis to create a well-working identification	Model is highly complex and difficult to implement to obtain satisfactory results	Basic knowledge of image processing is required, but the algorithm can be implemented in a dozen or so lines of code

5. Conclusion

As presented, each of the designed algorithms can solve the problem of identifying forgings and workers from industrial vision systems. It has been proven that machine learning-based solutions cope very well with object detection problems and achieve high degrees of accuracy after a precise selection of parameters. The algorithms show the performance of detections with excellent accuracy of 92.5% for the YOLOv5 and 94.3% for the Mask R-CNN. At the same time, all of the approaches can be further improved for the specific environment to achieve the highest quality detection results. In a situation where algorithms were forced to work in a changing environment, it was not possible to design them so that they would be able to work to their full potential.

Both the YOLOv5 and Mask R-CNN models can be improved by expanding the training and validation sets with additional special cases or different environments to improve their detection capabilities. In the case of the third model, it might be more effective to use other techniques to extract objects from the image or to try to expand the existing model to eliminate the disruptions that affected the results.

In the case of each of the presented solutions, it is possible to base the selection of parameters on a less random approach involving the use of appropriate analytical methods based on mathematical principles instead of using the method of changing one or several parameters and examining the result obtained in this way. Additionally, models based on machine learning can be further improved by using the proper optimization methods and tools that improve the operation of graphics cards, which can significantly increase the efficiency and speed of the solutions presented in this work. All of the presented improvements can be the basis for the creation of another work extending the issues presented in this paper in a more detailed way.

Acknowledgements

The work was undertaken within the framework of fundamental statutory research.

References

Bhadani, A. K., & Sinha, A. (2020). A facemask detector using machine learning and image processing techniques. *Engineering Science and Technology, an International Journal*, *11*, 0–8. https://www.researchgate.net/publication/345972030.

Cao, Q., Qingge, L., & Yang, P. (2021). Performance Analysis of Otsu-Based Thresholding Algorithms: A Comparative Study. *Journal of Sensors*, *2021*, 4896853. https://doi.org/10.1155/2021/4896853.

Goenka, U., Jagetia, A., Patil, P., Singh, A., Sharma, T., & Saini, P. (2022). Threat detection in self-driving vehicles using computer vision. In R. Doriya, B. Soni, A. Shukla, X.-Z. Gao (Eds.), *Machine Learning, Image Processing, Network Security and Data Sciences. Select Proceedings of 3rd International Conference on MIND 2021* (pp. 617–630). Springer Singapore. "Lecture Notes in Electrical Engineering", vol. 946. https://doi.org/10.1007/978-981-19-5868-7_45.

Hassan, E., El-Rashidy, N., & Talaa, F. M. (2022a). Review: Mask R-CNN models. *Nile Journal of Communication & Computer Science*, *3*(1), 17–27. https://njccs.journals.ekb.eg/article_280047.html.

Hassan, E., Shams, M. Y., Hikal, N. A., & Elmougy, S. (2022b). The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study. *Multimedia Tools and Applications*, *82*, 16591–16633. https://doi.org/10.1007/s11042-022-13820-0.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). *Mask R-CNN*. ArXiv. https://doi.org/10.48550/arXiv.1703.06870.

Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual understanding of convolutional neural network – a deep learning approach. *Procedia Computer Science*, *132*, 679–688. https://doi.org/10.1016/j.procs.2018.05.069.

Jais, I. K. M., Ismail, A. R., & Nisa, S. Q. (2019). Adam optimization algorithm for wide and deep neural network. *Knowledge Engineering and Data Science*, *2*(1), 41–46. http://dx.doi.org/10.17977/um018v2i12019p41-46.

Jiang, T., Gradus, J. L., & Rosellini, A. J. (2020). Supervised machine learning: A brief primer. *Behavior Therapy*, *51*(5), 675–687. https://doi.org/10.1016/j.beth.2020.05.002.

Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express*, *6*(4), 312–315. https://doi.org/10.1016/j.icte.2020.04.010.

Karthi, M., Muthulakshmi, V., Priscilla, R., Praveen, P., & Vanisri, K. (2021). Evolution of YOLO-V5 Algorithm for Object Detection: Automated Detection of Library Books and Performace validation of Dataset. In *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICSES52305.2021.9633834.

Krogh, A. (2008). What are artificial neural networks?. *Nature Biotechnology*, *26*(2), 195–197. https://doi.org/10.1038/nbt1386.

Kühl, N., Goutier, M., Baier, L., Wolff, C., & Martin, D. (2020). Human vs. supervised machine learning: Who learns patterns faster?. *Cognitive Systems Research*, *76*, 78–92. https://doi.org/10.1016/j.cogsys.2022.09.002.

Otani, M., Togashi, R., Nakashima, Y., Rahtu, E., Heikkilä, J., & Satoh, S. (2022). *Optimal correction cost for object detection evaluation*. ArXiv. https://doi.org/10.48550/arXiv.2203.14438.

Sarker, I. H. (2021). Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, *2*(6), 420. https://doi.org/10.1007/s42979-021-00815-1.

Sharma, S., Dubost, F., Lee-Messer, C., & Rubin, D. (2021). *Automated detection of patients in hospital video recordings*. ArXiv. https://doi.org/10.48550/arXiv.2111.14270.

Uzair, M., & Jamil, N. (2020). Effects of hidden layers on the efficiency of neural networks. In *2020 IEEE 23rd International Multitopic Conference (INMIC)*. IEEE. https://doi.org/10.1109/INMIC50486.2020.9318195.

Yamaura, H., Tamura, M., & Nakamura, S. (2018). Image blurring method for enhancing digital content viewing experience. In M. Kurosu (Eds.), *Human-Computer Interaction. Theories, Methods, and Human Issues. 20th International Conference, HCI International 2018, Las Vegas, NV, USA, July 15–20, 2018, Proceedings, Part I* (pp. 355–370). Springer Cham. "Lecture Notes in Computer Science", vol. 10901. https://doi.org/10.1007/978-3-319-91238-7_29.

Zhuang, J.-X., Tao, W., Xing, J., Shi, W., Wang, R., & Zheng, W.-S. (2021). *Understanding of kernels in CNN models by suppressing irrelevant visual features in images*. ArXiv. https://doi.org/10.48550/arXiv.2108.11054.