Journal of Computer Science

Original Research Paper

Improving the Detection of Mask-Wearing Mistakes by Deep Learning

Chahinez Mérièm Bentaouza

Department of Mathematics and Computer Science, Faculty of Exact Sciences and Computer Science, Abdelhamid Ibn Badis University, Mostaganem, Algeria

Article history
Received: 11-01-2024
Revised: 24-02-2024
Accepted: 08-03-2024

Email: chahinez.bentaouza@univ-mosta.dz

Abstract: This study focuses on the detection of wearing mask errors after machine learning by a Multi-Layer Perceptron Mixer (MLP Mixer) applied to protect masks from COVID-19. To combat the spread of the COVID-19 pandemic, facemasks have become an essential accessory, so it's necessary to identify individuals who follow this health protection. In this case, the most successful face-detection method Viola-Jones was used combining different techniques, each in one step. To make decisions, image classification aims to detect the presence of masks in images using mathematical methods. The classy design involves partitioning the parameter space based on representative attributes for each class. For this purpose, we used MLP mixer which is a convolutional neural network, also known as CNNs or ConvNets, they constitute deep learning because it is much better at detecting similarities than by an integrated image-to-image comparison. The classification ratio is satisfactory to achieve maximum accuracy in detecting. However, the learning time for network convergence is prolonged due to changes in parameters.

Keywords: Classification, Deep Learning, COVID-19, Face Mask, Machine Learning, MLP Mixer

Introduction

The current health crisis has made it obligatory to wear a hygiene mask in public places (Pentland *et al*., 1994). They seem to be a requirement rather than a choice.

Detection using statistical methods or machine learning methods was a popular subject for many works (Shalev-Shwartz and Ben-David, 2014).

The Multi-Layer Perceptron (MLP) is suggested in this study for its performance in classification (Zinkevich, 2017) and machine learning (Bishop and Nasrabadi, 2006), which can be applied to mask face classification (Mohri *et al*., 2018). MLP-based image classification was used in various domains.

In this instance, there are certain elements that need to be taken into consideration for mask-wearing mistakes (Fig. 1) (ARS, 2017):

- Wear it under your nose
- Leave the mount exposed
- Wear it loose
- Cover only the tip of the nose
- Put it under the chin
- Touch it once positioned

Fig. 1: Type of mask-wearing

Mask Detection

Face Detection

Before the mask detection phase, face detection in the image is a crucial and essential process (Goldstein *et al*., 1971).

OpenCV has a variety of classification methods for detecting faces in frontal poses, as well as profile faces, eye detection, body detection, and more.

Out of the many detection methods available, we chose the Viola and Jones method, which is a predefined function that is available in the OpenCV library.

The Viola and Jones method is a method for detecting objects in a digital image; it is one of the very first methods capable of effectively detecting objects in an image in real-time. The device was originally created to detect faces, but it can also be used to detect other types

of objects such as cars or planes. One of the most well-known and widely utilized methods is the Viola and Jones method, particularly for face and people detection. It is recognized as one of the most important methods for detecting objects (Cayla, 2020).

Deep Face

A system is designed to recognize the faces in an image or video (Abate *et al*., 2007). It uses deep learning methods to process large training sets (Huang *et al*., 2012), with many recent successes in various fields such as vision (Taigman *et al*., 2014), specifically with faces by capturing facial appearance in an exact manner and then learning the RGB values of pixels.

Mask Face

Face masks are usually worn in three different ways, covering half of the face (chin to nose), worn from chin to mouth, or simply placed on the chin, this latter will also be annotated by a rectangle.

To measure the degree of occlusion, we will divide the face into four major regions: The eyes, nose, mouth, and chin. We can conclude three types of occlusion degrees from the major regions we've mentioned above, which means only one or two regions are visible (i.e., chin nose), a medium occlusion (the three regions), or a significant occlusion (which includes all of the four regions).

The model of masks can differ from person to person (Ge *et al*., 2017) which is why it is necessary to consider that while picking our dataset and creating our model, we usually distinguish three types of masks, the simple surgical mask that comes in most cases with a pure color, a complex mask that can contain graphics or logos and a hybrid one which is a combination of the two sorts of masks or one in all the aforesaid mask sort with eyes occluded by the glasses.

Convolutional Neural Networks

The CNN evaluates images fragment by fragment. The fragments it seeks are known as features (Zhang *et al*., 2016). CNN is able to detect similarities better than by comparing an entire image to an image by finding approximate features that are roughly alike in two different images.

The CNN model architecture involves multiple convolutions and pooling layers that are stacked one after the other. The use of pooling layers reduces the dimensions of the feature maps. This results in a decrease in the number of parameters to learn and the amount of computation performed in the network.

The pooling layer summarises the features present in a region of the feature map generated by a convolution layer. So, further operations are performed on summarised features instead of precisely positioned features generated by the convolution layer. This makes the model more robust to variations in the position of the features in the input image (Venkateswarlu *et al*., 2020).

Experiments

To detect images, we must go through these steps (Fig. 2).

Libraries Used

There are several reasons for this overwhelming popularity (Statistics and Data, 2022) (Fig. 3):

- Suitable for multiple platforms (Linux, Mac OS, and even Android and iOS)
- Our focus will be on the Python API, which is available in Python, C++, Java, and Go (Brownlee, 2020)
- The backend in C/C++ is responsible for a very short build time
- Is able to support CPU, GPU, and distributed cluster computing
- The documentation is extremely well stocked, with numerous examples and tutorials
- Last but not least, the fact that the Framework comes from Google and that Google has announced that it has migrated almost all of its Deep Learning projects (King, 2009) to TensorFlow (TDS, 2017) is somewhat reassuring

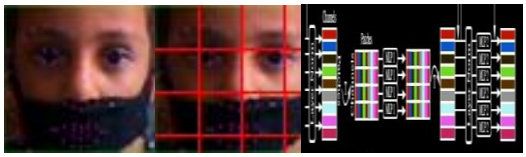

Fig. 2: Experimentation steps; (a) Face detected; (b) Blocks image; (c) MLP Mixer

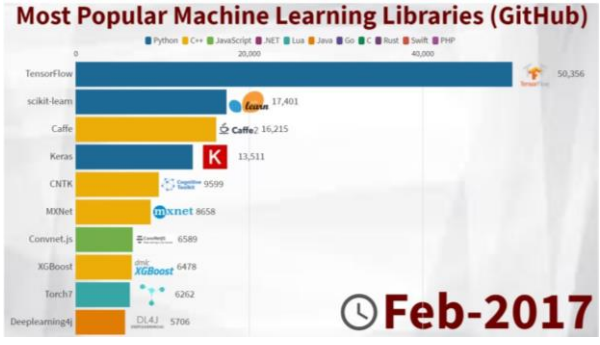

Fig. 3: Most popular machine learning libraries (GitHub)

Table 1: Example of pixel value in a block

RGB Value
1 1:0 2:0 3:0
1 1:0 2:0 3:0
2 1:32 2:32 3:64
2 1:32 2:32 3:64
2 1:32 2:32 3:64
3 1:64 2:32 3:64
3 1:64 2:32 3:64
4 1:64 2:64 3:64
5 1:96 2:64 3:128
5 1:96 2:64 3:128

Pre-Treatment

We take photos of different sizes and divide each image into blocks of 16×16 pixels. Each block contains the same pixels that have the same RGB value in the same class in Table 1 for each image. We begin learning each block with MLP because it's more efficient to learn each block by block than to learn an image.

Data Base

The database (Wang *et al*., 2022) is accessible to the public.

Photos were taken from the beautiful woman's face mask (Getty Images, 2024).

Materials and Methods

The machine that performed these tests has a Mac OS, Processor 1,8 GHz Intel Core i5, 8 Go of RAM for Linux programming and Pentium 4, 2.80 GHz processor and 256 MB of RAM for Windows programming. In this case, programming required the use of two essential tools.

Creating graphical interfaces using simple techniques is made possible with C++ Builder 6 for Windows, which is a powerful, efficient, and fast development tool.

Python under Linux is a tool that was created for scientific research that needed to solve optimization problems that were too challenging to code in a language like C.

The parameter that can be manipulated for machine learning is shown in Fig. 4.

TensorFlow is a machine learning tool that Google developed to make it simple to create ML models that can run in any environment. An interface can be used with both Python and C++.

Python-written neural network components can be found in Keras, an open-source library. It will be executed end-to-end on TensorFlow (Fig. 5) and other free learning platforms. The library was created to be modular and user-friendly, making it easy to understand and assimilat its basic principles.

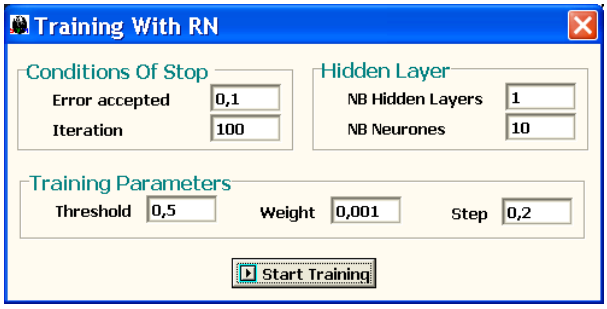

Fig. 4: Machine learning parameters

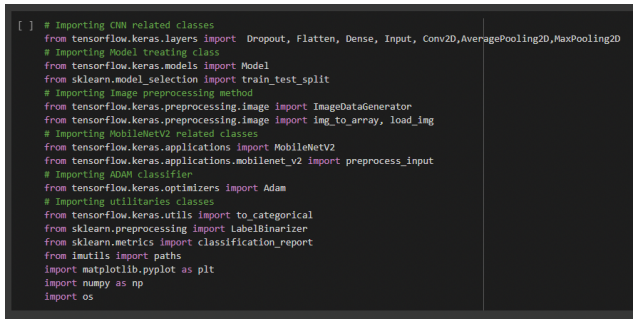

Fig. 5: Machine learning libraries used

Results

The results have been divided into three parts to give a comprehensive comparison of neural network learning, which includes the tests that will determine its effectiveness.

By using Support Vector Machines (SVM) (Vapnik and Vapnik, 1998), we improved the results at the end.

Case 1: Binary Detection

Our initial approach was a binary classification that consists of detecting between wearing:

1. Surgical bib
2. Visor

Variation of several parameters is necessary for neural networks to reach convergence.

In order to calculate the learning time taken for each case, we first vary the number of neurons and hidden layers. Finally, by increasing the number of iterations (ITER), the neural networks manage to converge for learning rate (Tr) and Test rate (Te), without being faced with the problem of forgetting, that is to say, a single class learned. The following figure displays validated results.

We started with a binary classification, which consists of detecting between wearing:

1. Surgical bib
2. Visor

DOI: 10.3844/jcssp.2024.751.757

For neural networks, several parameters need to be varied for network convergence.

We start by varying the number of neurons and hidden layers to calculate the learning time taken for each case (Fig. 6).

Finally, by increasing the number of iterations (ITER) (Fig. 7), the neural networks manage to converge for learning rate (Tr) and Test rate (Te), without being faced with the problem of forgetting, that is to say, a single class learned.

The validated results are represented in the following figure (Fig. 8). Hence, the classification rate is 100%.

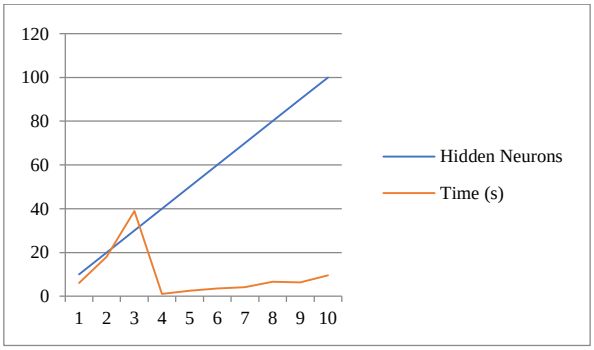

Fig. 6: Time relative to the number of neurons in the hidden layer in case 1

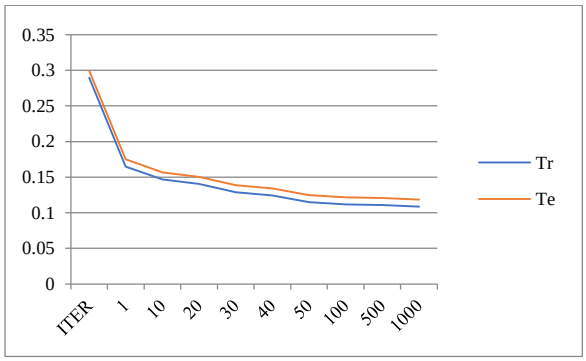

Fig. 7: Tr error and Te error in relation to the number of iterations in case 1

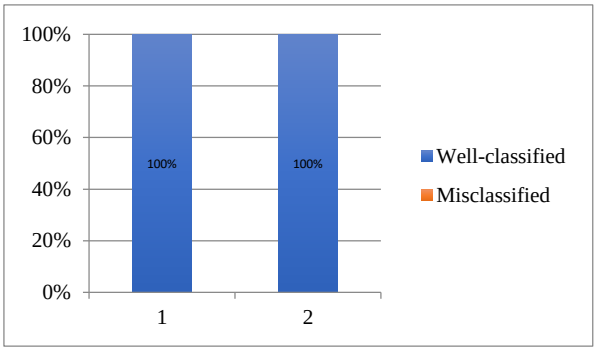

Fig. 8: Validation of results in case 1

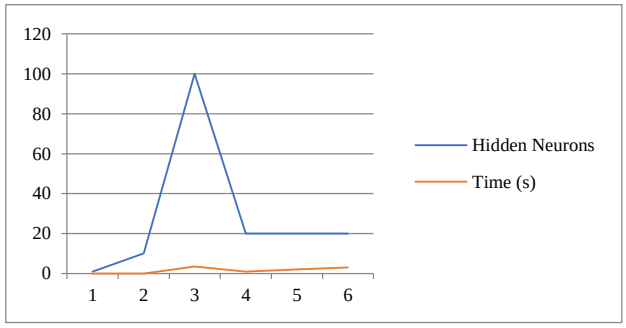

Fig. 9: Time relative to the number of neurons in the hidden layer in case 2

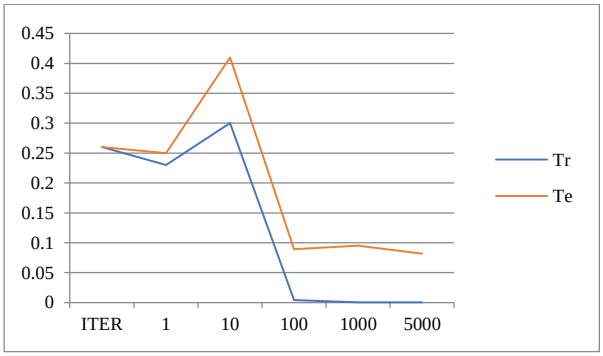

Fig. 10: Tr error and Te error in relation to the number of iterations in case 2

Case 2: Multi Detection

In this case, we are going to use four different types of mask-wearing classes:

1. Leave the mount exposed
2. Wear it loose
3. Cover only the tip of the nose
4. Touch it once positioned

To ensure the convergence of neural networks, a variety of parameters must be adjusted. To find the best learning time (Fig. 9), these values are determined by the number of neurons and hidden layer.

After several attempts, which took a lot of time, we managed to minimize the errors in the test examples (Fig. 10).

Using the selected parameters, we verify the results and observe that the classification rate has not significantly changed (Fig. 11), but the learning time is lengthy and uncertain. So, the classification rate is 98,81%.

Case 3: Multi Classification

In this instance, we will divide our analysis into nine classes that wear masks:

1. Good mask
2. Visor
3. No mask

4. Wear it under your nose
5. Leave the mount exposed
6. Wear it loose
7. Cover only the tip of the nose
8. Put it under the chin
9. Touch it once positioned

Variation of several parameters is necessary for the convergence of neural networks, as previously done. These values include both the number of neurons and the hidden layer (Fig. 12), as well as the number of iterations (Fig. 13).

The error on the test examples was not minimized due to the time wasted in training and converging our network (Fig. 14). The classification rate is as follows 99,19%.

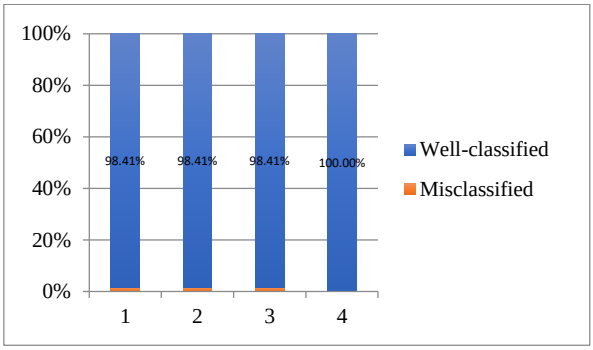

Fig. 11: Validation of results in case 2

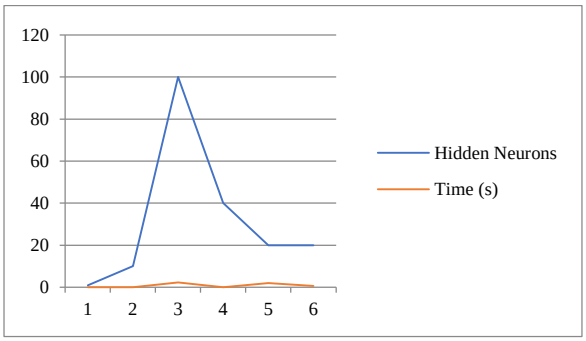

Fig. 12: Time relative to the number of neurons in the hidden layer in case 3

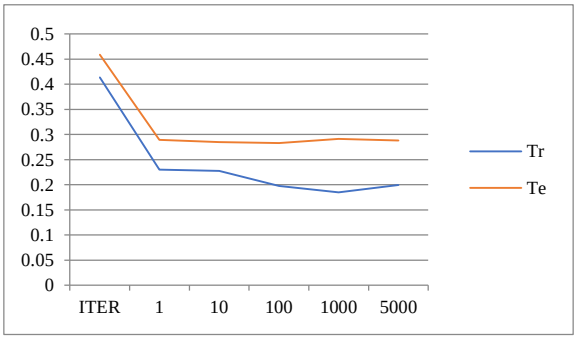

Fig. 13: Tr error and Te error in relation to the number of iterations in case 3

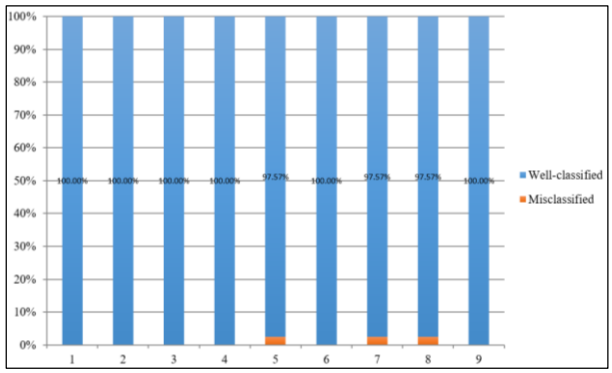

Fig. 14: Validation of results in case 3

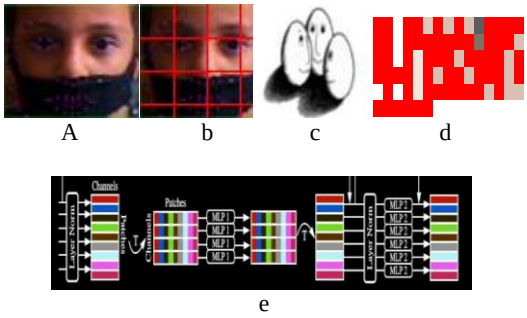

Fig. 15: Experimentation steps; (a) Face detected; (b) Blocks image; (c) SVM; (d) Image compressed; (e) MLP Mixer

Contribution of Support Vector Machines

We employ SVM for image compression because the compressed image's training is too small and provides satisfactory results compared to previous cases.

Each image is divided into blocks of 16×16 pixels by taking pictures of various sizes (Fig. 15). The same pixels with the same RGB value are reassembled in the same class for each image in each block. Afterward, we begin to learn each block with the SVM-Light-Multi class (Bentaouza and Benyettou, 2018) because it's easier to learn by block than by image.

The support vectors (Bentaouza and Benyettou, 2014) are used in image compression to represent the points in the image and the learning results are reflected in these vectors.

As presented in support vector machines for classification (Bentaouza and Benyettou, 2010) the parameter selected for the polynomial function is the one that minimizes learning error and maximizes classification ratio, as described in support vector machines for classification (Fig. 16). The polynomial degree (d) value is 3 and the radial gamma value is 3 as well. The classification rate is 93,50%.

The main point is the decrease in learning time, which is why when using compression; the time is less compared to the percentage of validation.

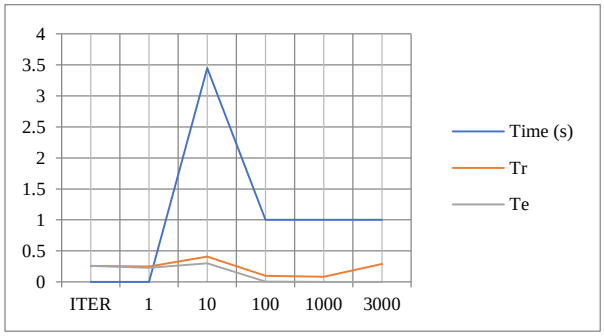

Fig. 16: Tr error, Te error, and time relate to SVM iterations

Discussion

So, we distinguish a big difference between the classification rate of SVMs and that neural network. We conclude this part with the following point.

Neural networks are difficult to manage once we increase the number of classes and we observe a significant difference in the classification rate of SVMs compared to that neural network. The conclusion of this part is that managing neural networks becomes difficult if we increase the number of classes, not the number of examples.

The experimental part's results are achieved by balancing multiple parameters.

In reality, the SVM's behavior is only sensitive to the regulation value if the training data is not separated.

Although the number of neurons in the hidden layer and the number of iterations can direct the neural network towards convergence (Table 2), it also has the following disadvantages:

- The configuration needs to be established
- Local minima
- Random initialization

By comparing the two methods, from the point of view of learning time and guarantee of convergence, we can say that the separators with wide margins intervened to minimize the learning time.

Our experience was not consistent with the accuracy presented (Tomás *et al*., 2021) to the different databases used (Ramesh *et al*., 2021) and the way information was processed (Chen *et al*., 2022).

Table 2: Comparing the parameters employed

	Hidden	ITER	Time	Te	Tr	Rate (%)
Case1	40	5000	1,1	0,1185	0,1085	100
Case2	20	1000	1	0,089	0,0002	98,81
Case3	10	1000	0,6	0,2832	0,185	99,19
SVM	-	100	0,0001	0,0002	0,082	93,50

Conclusion

In recent years, bioinformatics has moved rapidly towards focusing on images, and imagery is frequently used to identify the location and size of an image to provide information.

Therefore, it is advantageous to create practical, reliable, and precise tools for the classification of these images. The difficulties are partially due to the complexity of processing dimensional data with a limited database.

Neural networks are one of the classification models that provide a more connectionist framework for learning theory and have been used to mark pattern recognition.

This model's complexity is determined by the number of support vectors, as well as the addition of SVMs to reduce learning time.

The main purpose of this article is to compare the performance of the different parameters of the neural network in classification. In particular, the requirement to be able to automatically select the model's parameters.

Acknowledgment

Thank you to the publisher for their support in the publication of this research article. We are grateful for the resources and platform provided by the publisher, which have enabled us to share our findings with a wider audience. We appreciate the efforts of the editorial team in reviewing and editing our work and we are thankful for the opportunity to contribute to the field of research through this publication.

Funding Information

The authors have not received any financial support or funding to report.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and that no ethical issues are involved.

References

Abate, A. F., Nappi, M., Riccio, D., & Sabatino, G. (2007). 2D and 3D face recognition: A survey. *Pattern Recognition Letters*, *28*(14), 1885-1906. https://doi.org/10.1016/j.patrec.2006.12.018

ARS. (2017). Together for an Island of Heath. https://www.corse.ars.sante.fr/le-port-du-masque-complete-les-gestes-barrieres

Bentaouza, C. M., & Benyettou, M. (2010). Support vector machines for brain tumours cells classification. *Journal of Applied Sciences*, *10*(16), 1755-1761. https://doi.org/10.3923/jas.2010.1755.1761

Bentaouza, C. M., & Benyettou, M. (2014). Support vector machines for microscopic medical images compression. *Pakistan Journal of Biological Sciences: PJBS*, *17*(3), 335-345. https://doi.org/10.3923/pjbs.2014.335.345

Bentaouza, C. M., & Benyettou, M. (2018). Support Vector Machine Applied to Compress Medical Image. *J. Comput.*, *13*(5), 580-587. https://doi.org/10.17706/jcp.13.5.580-587

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*, (Vol. *4*, No. 4, p. 738). New York: springer. ISBN-10: 978-0-387-31073-2.

Brownlee, J. (2020). Softmax Activation Function with Python. *Machine Learning Mastery*. https://machinelearningmastery.com/softmax-activation-function-with-python

Cayla, B. (2020). datacorner.fr. Face detection with OpenCV. https://datacorner.fr/reco-faciale-opencv/

Chen, R., Fwu, B. J., Yang, T. R., Chen, Y. K., & Tran, Q. A. N. (2022). To mask or not to mask: Debunking the myths of mask-wearing during COVID-19 across cultures. *Plos One*, *17*(9), e0270160. https://doi.org/10.1371/journal.pone.0270160

Ge, S., Li, J., Ye, Q., & Luo, Z. (2017). Detecting masked faces in the wild with lle-cnns. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 2682-2690). https://openaccess.thecvf.com/content_cvpr_2017/html/Ge_Detecting_Masked_Faces_CVPR_2017_paper.html

Getty Images. (2024). Beautiful woman face mask. https://www.gettyimages.fr/photos/beautiful-woman-face-mask

Goldstein, A. J., Harmon, L. D., & Lesk, A. B. (1971). Identification of human faces. *Proceedings of the IEEE*, *59*(5), 748-760. https://doi.org/10.1109/PROC.1971.8254

Huang, G. B., Lee, H., & Learned-Miller, E. (2012 June). Learning hierarchical representations for face verification with convolutional deep belief networks. *In 2012 IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 2518-2525). IEEE. https://doi.org/10.1109/CVPR.2012.6247968

King, D. E. (2009). Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, *10*, 1755-1758. https://www.jmlr.org/papers/volume10/king09a/king09a.pdf

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). Foundations of machine learning. 2nd Ed. *MIT Press*. ISBN-10: 9780262039406.

Pentland, Moghaddam, & Starner. (1994 June). View-based and modular eigenspaces for face recognition. *In 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 84-91). IEEE. https://doi.org/10.1109/CVPR.1994.323814

Ramesh, P. V., Ramesh, S. V., Ray, P., Aji, K., Ramesh, M. K., & Rajasekaran, R. (2021). The curious cases of incorrect face mask positions in bowl-type perimetry versus enclosed chamber perimetry during the COVID-19 pandemic. *Indian Journal of Ophthalmology*, *69*(8), 2236-2239. https://doi.org/10.4103/ijo.IJO_805_21

Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. 1st Ed. *Cambridge University Press*. pp: 410. ISBN-10: 1107057132.

Statistics, & Data. (2022). Most Popular Machine Learning Libraries-2014/2021. *Statistics and Data*. https://statisticsanddata.org/data/most-popular-machine-learning-libraries/

Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (pp. 1701-1708). https://openaccess.thecvf.com/content_cvpr_2014/html/Taigman_DeepFace_Closing_the_2014_CVPR_paper.html

TDS. (2017). Deep Learning with Tensorflow: Part 2-Image classification. *Towards Data Science*. https://towardsdatascience.com/deep-learning-with-tensorflow-part-2-image-classification-58fcdffa7b84

Tomás, J., Rego, A., Viciano-Tudela, S., & Lloret, J. (2021, August). Incorrect facemask-wearing detection using convolutional neural networks with transfer learning. *In Healthcare*, (Vol. *9*, No. 8, 1050). MDPI. https://doi.org/10.3390/healthcare9081050

Vapnik, V. N., & Vapnik, V. (1998). Statistical learning theory. 2nd Ed. ISBN-10: 0-387-98980-0.

Venkateswarlu, I. B., Kakarla, J., & Prakash, S. (2020, December). Face mask detection using mobilenet and global pooling block. *In 2020 IEEE 4th Conference on Information and Communication Technology* (*CICT*), (pp. 1-5). IEEE. https://doi.org/10.1109/CICT51604.2020.9312083

Wang, C., Fang, H., Zhong, Y., & Deng, W. (2022, October). Mlfw: A database for face recognition on masked faces. *In Chinese Conference on Biometric Recognition*, (pp. 180-188). Cham: Springer Nature Switzerland. MLFW_ A Database for Face Recognition on Masked Faces _ SpringerLink

Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, *23*(10), 1499-1503. https://doi.org/10.1109/LSP.2016.2603342

Zinkevich, M. (2017). Rules of machine learning: Best practices for ML engineering. https://martin.zinkevich.org/rules_of_ml/rules_of_ml.pdf