APPLICATION OF A CONVOLUTIONAL NEURAL NETWORK TO MULTICLASS IMBALANCED IMAGE CLASSIFICATION OF RESPIRATORY FACE MASKS

Carl Wilson
School of Computing & Engineering
University of Huddersfield
Huddersfield, UK
u0370630@unimail.hud.ac.uk

WORD COUNT: 2915

***Abstract*— This paper applies a convolutional neural network to the task of classifying images of people either wearing, incorrectly wearing, or not wearing respiratory face masks with a focus on classification of the most under-represented class, within the imbalanced dataset, of incorrectly wearing face masks. It will demonstrate the relative performance improvement of class balancing through applying weights and over-sampling – where new samples are created by applying random augmentations to existing samples.**

Keywords—COVID19, Face Mask, Object Detection, CNN, Convolutional Neural Network, Machine Learning.

I. INTRODUCTION

Face masks play a vital role in the reduction of respiratory virus transmission and were a powerful used to minimize the spread of Coronavirus disease 19 [COVID-19]. Automated tracking of mask usage can help better understand the effectiveness of masks for future pandemics.

Convolutional Neural Networks [CNN], or convnets, are the "go to" deep learning algorithm for computer vision-based applications (Chollet, 2021). This report will explore the application of a CNN to classify images of people wearing, incorrectly wearing, or not wearing face masks. The specific challenge and focus of this report are on distinguishing [classifying] incorrectly worn masks from the other two classes.

II. LITERATURE REVIEW

CNN's have successfully been applied to classify images on anything from breast cancer scans (Chen et al., 2019), pulmonary tuberculosis CT scans (Li et al., 2018), lung abnormalities (Kido et al., 2018) to apple species (Nimsuk & Paewboontra, 2021) and building architecture (Guo & Li, 2017). They are not only powerful, but also versatile – solving multi-dimensional datasets such as timeseries problems (S. Singh et al., 2021).

Region CNN [R-CNN] is an object detection method that has been successfully applied and improved upon, primarily through speeding predictions up hence there is also Fast R-CNN and Faster R-CNN. R-CNN works by segmenting images into regions of interest prior to being processed by the CNN. The increased computation time has led to improvements such as Fast R-CNN and Faster R-CNN. These three along with a Single Shot Multibox Detector [SSD] were benchmarked on Manga comics by (Yanagisawa et al., 2018) where it was concluded that SSD performed worse than Faster R-CNN when detecting panels, speech balloons, character faces and text. R-CNN was applied by (Zhang et al., 2018) to identify small objects in images that were typically too small to be presented by feature maps, by introducing deconvolution layers. In this instance feature maps were up-sampled to create new feature maps before applying the R-CNN to divide the image up into areas of interest.

Faster R-CNN and You Only Look Once [YOLO] version were applied to the task of face mask detection, in the context of COVID19 by (Sunil Singh et al., 2021). This was specifically looking to make fast classifications such that it could potentially be applied to video and the researchers saw incredibly fast inferences of 0.045s and 0.15s for YOLOv3 and Faster R-CNN respectively. However, they also only saw average precisions of 55 and 62% respectively. This is relatively low in terms of classification metrics for this binary classification task.

Processing speed was also the focus of (Nagavi et al., 2021) when considering the binary classification problem of with or without masks. The dataset was relatively well balanced with 46% of images with mask and 54% without mask, on a sample size over 4000. Data augmentations was implemented to increase the training sample. A pretrained network, mobileNetV2, was implemented and resulted in training and testing accuracies of 98.6% and 95.7% respectively. There is little to no references to precision, recall or AUC.

In the case of (Gupta et al., 2021) a CNN was created to classify images of faces with or without masks [binary]. It was based on Alex Net architecture and achieved 0.98, 0.99 and 0.84 for accuracy, precision, and recall, respectively. While the accuracy and precision are relatively high, the recall score is relatively low. Due to the inherit trade-off between recall and specificity (Bruce & Bruce, 2017), a better predictor would have a higher recall score thus increasing the area under the curve [AUC] on a receiver operating characteristic [ROC].

Binary classification of with or without mask was examined by (Sarma et al., 2021) where a CNN was developed and benchmarked against VGG16 and InceptionV3 pre-trained networks. While the CNN developed by the researchers achieved high levels of precision, recall and F1-score, all above 95%, the paper concluded that the benchmark models had overfitting problems. However, examining the data I disagree as both benchmark models had training and validation accuracy over 99% and within 0.55% of each other.

The researchers made use of image augmentation to expand the training set, but again only focused on two classes of either “with” or “without” mask.

Image augmentation can be used to improve the predictive performance of CNN’s (Chollet, 2021), random image augmentations were experimented with by (Shijie et al., 2017) and benchmarked using the AlexNet CNN architecture and the CIFAR10 and ImageNet datasets predicting multiple classes. Random flips, rotations, cropping, and noise was added as well as an implementation of a Generative Adversarial Network [GAN]. The paper concluded that all enhancements, except adding random noise, improved the predictive performance of the model and that the smaller the initial sample set – the greater the improvement during training. It was also concluded that GAN, random cropping, random flipping, and random rotation were the most influential performance enhancers and combinations of multiple augmentations were more powerful than single augmentations.

Several advanced techniques using state-of-the-art methods, such as Faster R-CNN and YOLO can be used to specifically identify objects in an image at high speed. However, a key challenge with this specific dataset is the very similar, yet different classes of “mask”, “no mask” and “mask worn incorrectly” with “mask worn incorrectly” potentially having a lot of similarity with the other two classes. This literature review has shown there is limited research in this area. This challenge is magnified by the dataset being imbalanced with the “Incorrect Mask” class being underrepresented at only 3% of sample.

The focus of this research will be on correctly identifying this specific class. To accelerate the iterations and solve times, faces will be extracted into small images that can be quickly processed by the algorithm – thus not requiring R-CNN or YOLO type architectures.

III. Exploratory Data Analysis

The dataset used (Larxel, 2020) was taken from Kaggle and is provided as a series of 853 Portable Network Graphics [PNG] and accompanying annotation eXtensible Markup Language [XML] files. The XML files were read using the XML library and images were read and manipulated using Pillow [PIL] in Python.

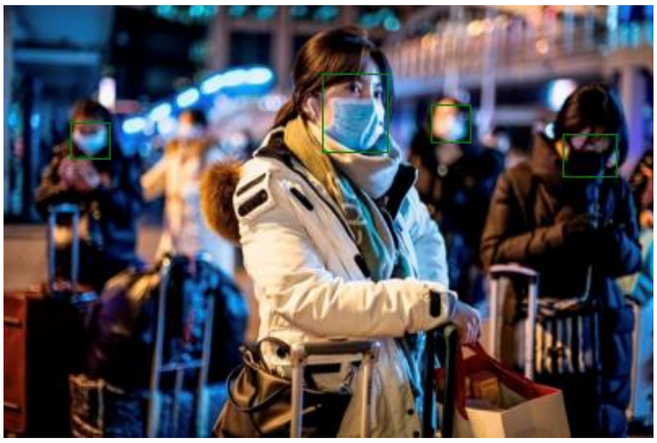
Fig. 1. Sample image with bounding boxes overlaid

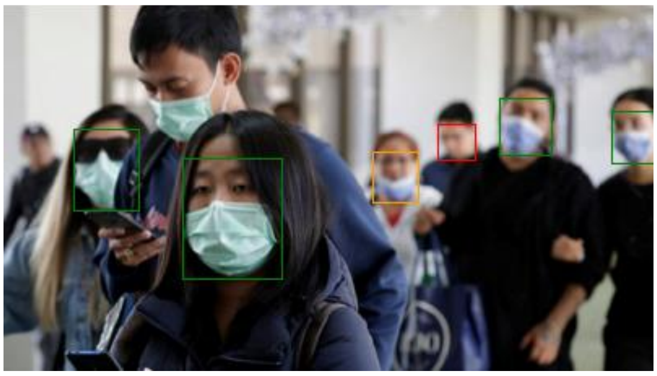
Fig. 2. Sample image with bounding boxes overlaid

Each of the 853 PNG’s may contain one or more targets, for example Fig. 1 shows four targets for “Mask” in one image. Each of the bounding boxes are of varying sizes and provide different facial features, colors of mask and image quality – for example the lack of focus for the rear targets. Specifically, the background images can be difficult to identify, by eye, if the mask is being worn correctly or not. Fig. 2 also illustrates the challenge of classifying “Incorrect Mask”.

One of the challenging aspects of this dataset is the class imbalance, with 79% of the sample belonging to the class “with mask” and only 3% belonging to the class “mask worn incorrectly”. Developing a network capable of reliability differentiating between these two classes – will require additional work and will be the focus of this research.

The problem is a vision based multi-class classification problem, with three imbalanced classes of bounding boxes representing faces either with “Mask”, “No Mask” or “Incorrect Mask”.

IV. Implementation

A. *Data Preparation*

All annotation XML files were read and appended into a Pandas DataFrame, which was then used to drive a short cropping function to crop the 4072 images of faces with, without or incorrectly wearing a mask – all resized and saved as 36x36 PNG files. The dimension of 36 was determined by averaging all the samples along their longest edge.

Classes were ordinally encoded, with 0 representing faces without a mask, 1 representing faces wearing a mask incorrectly and 2 representing faces wearing a mask.

The data was split into training and test datasets on a ratio of 0.8 and 0.2 respectively; the split was done by using SciKit-Learn and had a random seed of 42 for repeatability.

Matrices were created for training and testing using Numpy and each were normalized on the color dimension by diving by 255.

B. *Weights*

For the weighted analysis, an inverse proportion for each class was determined, as per equation 1, where W_A represented the weight for class *a*, N, represents the total number of samples, C represents the number of classes and N_a represents the number of samples in class *a*.

$$W_A = \frac{N}{C \times N_A} \qquad equation\ 1$$

C. *Oversampling*

Oversampling was introduced as a second model performance enhancer by transforming the original training dataset to balance the number of samples in each class.

Oversampling was completed on the training dataset only, by randomly selecting an image then randomly applying transformations, which in turn had random settings – for example the rotation transformation had a random rotation angle between -15° and 15°. The test sample was not oversampled. Each randomly selected image had a 50% chance to have each, subsequent transformation applied for: flipping left to right, rotation, brightness, and Gaussian blur.

The Pillow library was again used for all image modifications.

D. *The Network*

The convolutional neural network consisting of a total of eleven layers was explored in terms of hyper-parameter space. The basic network consisted of three convolutional layers, three max pooling, two drop out, a flatten and finally two fully connected dense layers with one being the output layer. The network was build using Keras and TensorFlow in Python.

The architecture design could be considered similar in structure to LeNet, using ReLu activation functions in place of Sigmoid and one additional pair of convolution and max pool layer.

Convolutional layers could have filter values between 16 and 1024, with a step size of 8 and a kernel of 2 or 4. The default starting filter size was 512. A kernel of 3 was not considered, due to the dimensionality reduction introduced through max pooling potentially leading to configurations of hyper-parameters that would lead to a model that was not be feasible. Activation was set to ReLu for each of the layers, as the default for this type of problem.

Max Pooling layers were of shape 2 x 2. Dropout for both layers were set to 0.5 to regularize the network and reduce the risk of overfitting, which is essential on a small dataset such as this (Chollet, 2021). The output which was a Dense layer of 3 units, for 3 classes, and activation of SoftMax. The Dense layer between flatten and the output layer, also had a range of hidden units between 128 and 1024, with a step size of 16 and an activation of ReLu.

Adam was set as the optimizer with a learning rate option of 0.01, 0.001 and 0.0001. The loss function was Sparse Categorical Cross entropy as this was a multiclass problem and "accuracy" was set as the metric.

E. *Training*

The hyper-parameters outlined in the network were tuned using Bayesian Optimization, from Keras Tuner, with an objective of maximizing the validation accuracy based on a holdout of portion 0.2. The optimization algorithm was given 50 trials and each trial was allowed to run to 50 epochs.

Training was completed using a machine with a water-cooled Intel i9-10900K CPU overclocked at 4.5 Ghz, 64GB of DDR3200 memory and a nVidia GeForce RTX3080 GPU. This allowed the experiments to complete in under six hours.

F. *Post Processing and Results Wrangling*

Once models were developed for each of the three configurations [baseline, weighted and oversamples] – classification metrics, confusion matrices and receiving operating characteristic [ROC] curves were calculated. For the ROC curve, the micro-average area under curve [AUC] for each class was determined due to the class imbalance.

SciKit-Learn metrics was used to calculate all metrics and produce the confusion matrices with matplotlib being used to produce all bar charts, ROC curves and confusion matrices.

V. Results

Precision, recall, F1-score, and AUC was calculated for each class on each model. All three models appear to be able to distinguish between "Mask" and "No Mask" with high precision and recall as can be observed in the bar charts labelled Fig. 3 and Fig. 4.

All resulting scored have been tabulated and can be referred to in Table I.

The "Incorrect Mask" class, as expected is the most challenging. The baseline model has a reasonable level of precision, but very low recall and this is translated into the AUC as can be observed in Fig. 5. The weighted model was worse predictor for the "Incorrect Mask" class with no improvement in recall, a slight drop in precision and slightly lower AUC scores. However, the oversampled model managed to significantly improve the predictive performance for the "Incorrect Mask" class with a significant step up in both recall and AUC but a reduction in precision.

While the baseline model struggles, the weighted model manages to slightly improve recall, however the oversampled model significantly improves recall at the cost of accuracy. Both improved models appear to achieve this at the cost of accuracy. it classifies more images as "incorrect mask" but also correctly classifies more. This is also at the cost of recall to the "Mask" class.

An interesting observation for the resulting oversampled model is the number of filters the tuner optimized to. The number of filters for the best of the three types of models, by layer, can be observed in Table II.

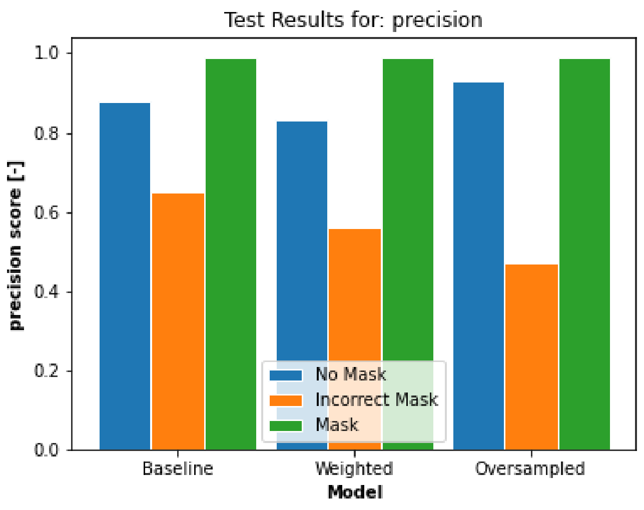

Fig. 3. Precision scores for all three models and classes.

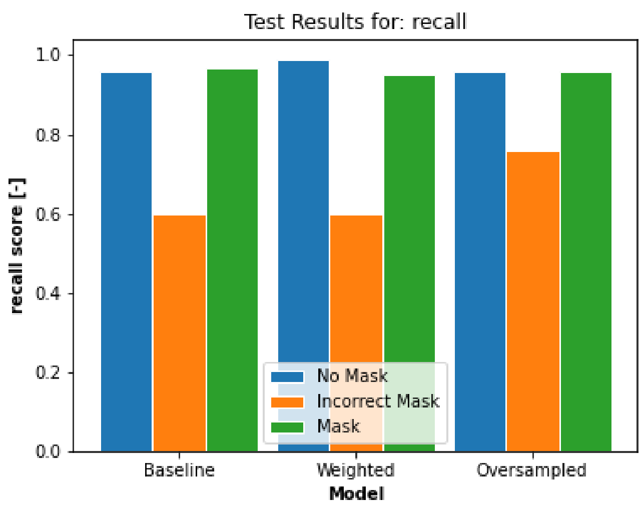

Fig. 4. recall scores for all three models and classes.

TABLE I. EXPERIMENTAL RESULTS

Model	Class	Metric			
		Precision	*Recall*	*F1-Score*	*AUC*
Baseline	No Mask	0.88	0.96	0.92	0.97
	Incorrect Mask	0.65	0.60	0.63	0.79
	Mask	0.99	0.97	0.98	0.96
Weighted	No Mask	0.83	0.99	0.90	0.96
	Incorrect Mask	0.56	0.60	0.58	0.77
	Mask	0.99	0.95	0.97	0.95
Oversampled	No Mask	0.93	0.96	0.94	0.97
	Incorrect Mask	0.47	0.76	0.58	0.87
	Mask	0.99	0.96	0.97	0.96

Both the weighted and oversampled models have hit the hyper-parameter space limit on number of filters, at least once – with the oversampled model hitting the upper limit with three of the four layers available for tuning.

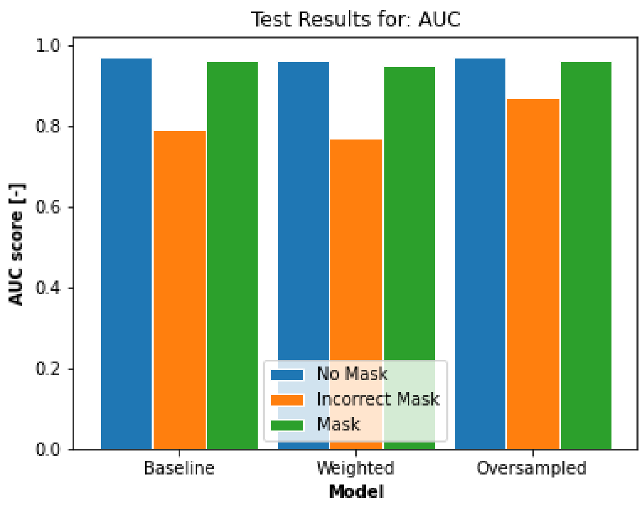

Fig. 5. Confusion matrix for baseline model.

This suggests the oversampled model might be slightly restricted in its performance and with greater freedom on filter number, higher performance could be achieved.

TABLE II. FILTER NUMBERS BY MODEL AND LAYER

Filter Numbers		
Model	***Layer***	***Filters***
Baseline	2D Conv 1	752
	2D Conv 2	720
	2D Conv 3	520
	Dense 1	1024
Weighted	2D Conv 1	720
	2D Conv 2	1024
	2D Conv 3	632
	Dense 1	944
Oversampled	2D Conv 1	1024
	2D Conv 2	480
	2D Conv 3	1024
	Dense 1	1024

Interestingly the weighted model predicts the “No Mask” class with a slightly lower level of precision that both the baseline model and the oversampled model. On balance, this suggests the weighted model is the worst performing predictor of the three – being slightly poorer in overall performance than the baseline model. The oversampled model is clearly better than the baseline overall and significantly better than the baseline and weighted when classifying images of “Incorrect Mask”.

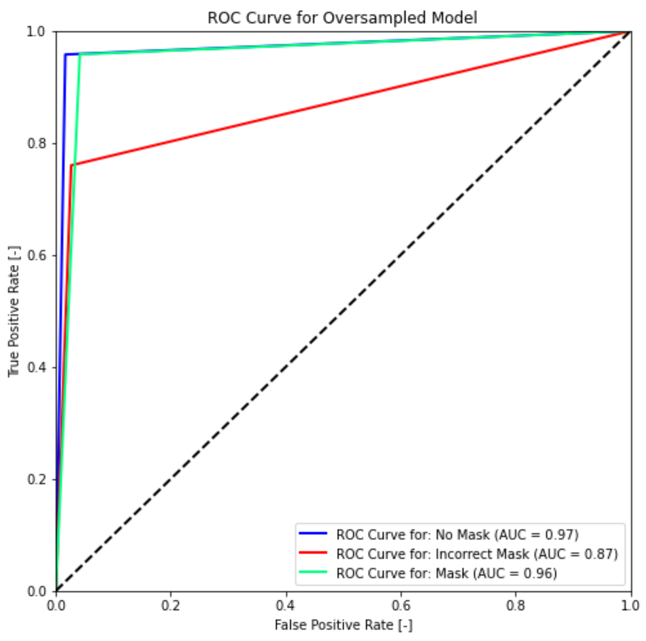

Fig. 6. ROC curve for oversampled model.

The combination of the limited hyper-parameter space and the higher scores also suggests there is entitlement performance that could be unlocked through further optimization. Fig. 6. Illustrates the ROC curve for the oversampled model with the model being an excellent classifier for the "No Mask" and "Mask" classes and a reasonably good classifier for the "Incorrect Mask" class.

Table III provides the CNN configuration for the oversampled model, which was deemed the best predictor. The model had a total of 19,941,859 trainable parameters.

TABLE III. OVERSAMPLED MODEL ARCHITECTURE

Sequential Model		
Layer (type)	***Output Shape***	***Param#***
2D Convolution	(None, 35, 35, 1024)	13312
2D Max Pooling	(None, 17, 17, 1024)	0
2D Convolution	(None, 14, 14, 480)	7864800
2D Max Pooling	(None, 7, 7, 480)	0
Dropout	(None, 7, 7, 480)	0
2D Convolution	(None, 4, 4, 1024)	7865344
2D Max Pooling	(None, 2, 2, 1024)	0
Dropout	(None, 2, 2, 1024	0
Flatten	(None, 4096)	0
Dense	(None, 1024)	4195328
Dense	(None, 3)	3075

VI. CONCLUSIONS & FURTHER WORK

Despite the significant challenges of under-representation and overlap with adjacent classes, a model has been presented with reasonable predictive performance for the class of "Incorrect Mask" without significantly sacrificing the predictive performance of said adjacent classes of "Mask" and "No Mask".

The hyper-parameter space for a relatively simple CNN was fully explored with three sets of inputs: a baseline, a class weighted and an oversampled with image augmentation.

The oversampled model provided the best overall performance, with a noticeable step up from the baseline in AUC and recall for the "Incorrect Mask" class.

The oversampled model architecture exhibited signs that the extents of the hyper-parameter space were limiting its performance, increasing this might provide additional entitlement on performance. An additional extension of this work would be to employ a pre-trained network such as VGG16, as opposed to training from scratch.

REFERENCES

Bruce, P., & Bruce, A. (2017). *Practical statistics for data scientists*. O'Reilly Media.

Chen, G., Chen, Y., Yuan, Z., Lu, X., Zhu, X., & Li, W. (2019, 22-24 Nov. 2019). Breast Cancer Image Classification based on CNN and Bit-Plane slicing. 2019 International Conference on Medical Imaging Physics and Engineering (ICMIPE),

Chollet, F. (2021). *Deep Learning with Python, Second Edition*. Manning Publications Co. LLC. https://go.exlibris.link/LzvRQtY4

Guo, K., & Li, N. (2017, 3-5 Oct. 2017). Research on classification of architectural style image based on convolution neural network. 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC),

Gupta, S., Sreenivasu, S. V. N., Chouhan, K., Shrivastava, A., Sahu, B., & Manohar Potdar, R. (2021). Novel Face Mask Detection Technique using Machine Learning to control COVID'19 pandemic. *Materials today : proceedings*. https://doi.org/10.1016/j.matpr.2021.07.368

Kido, S., Hirano, Y., & Hashimoto, N. (2018, 7-9 Jan. 2018). Detection and classification of lung abnormalities by use of convolutional neural network (CNN) and regions with CNN features (R-CNN). 2018 International Workshop on Advanced Image Technology (IWAIT),

Larxel. (2020). *Mask Dataset* Kaggle. https://www.kaggle.com/datasets/andrewmvd/face-mask-detection

Li, L., Huang, H., & Jin, X. (2018, 19-21 Oct. 2018). AE-CNN Classification of Pulmonary Tuberculosis Based on CT Images. 2018 9th International Conference on Information Technology in Medicine and Education (ITME),

Nagavi, T. C., Sharanya, R. B., Minchu, S., Kumar, R., & Manuprasad. (2021, 7-9 Oct. 2021). COVID-19 Face Mask Detection Using Deep Learning. 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC),

Nimsuk, N., & Paewboontra, W. (2021, 19-22 May 2021). Compact CNN Model for Classifying Rose Apple Species and Detecting Their Skin Defects. 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON),

Sarma, M., Talukdar, A. K., & Sarma, K. K. (2021, 26-28 Nov. 2021). Deep Learning Based Face Mask Detection System for COVID-19 Control. 2021 Sixth International Conference on Image Information Processing (ICIIP),

Shijie, J., Ping, W., Peiyi, J., & Siping, H. (2017, 20-22 Oct. 2017). Research on data augmentation for image classification based on convolution neural networks. 2017 Chinese Automation Congress (CAC),

Singh, S., Ahuja, U., Kumar, M., Kumar, K., & Sachdeva, M. (2021). Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment. *Multimedia tools and applications*, *80*(13), 19753-19768. https://doi.org/10.1007/s11042-021-10711-8

Singh, S., Singh, K., & Saxena, A. (2021, 8-9 Jan. 2021). Remaining useful life (RUL) prediction for FDIA on IoT sensor data using CNN and GRU. 2021 International Conference on Advances in Technology, Management & Education (ICATME),

Yanagisawa, H., Yamashita, T., & Watanabe, H. (2018, 7-9 Jan. 2018). A study on object detection method from manga images using CNN. 2018 International Workshop on Advanced Image Technology (IWAIT),

Zhang, W., Wang, S., Thachan, S., Chen, J., & Qian, Y. (2018, 22-27 July 2018). Deconv R-CNN for Small Object Detection on Remote Sensing Images. IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium,