Real - Time Analytics Dashboard for Machine Maintenance in Legacy Machines Using Deep Transfer Learning and Computer Vision

Venkata Sai Kumar Reddy Aavula and Dr V. Vasu
Department of Mechanical Engineering
National Institute of Technology (NIT), Warangal, India
venkatasaiaavula@gmail.com, vasu@nitw.ac.in, vasuapplepc@gmail.com

Abstract

Productivity losses in the manufacturing industry have to be minimized to perform at total capacity consistently. If productivity losses are neglected, a reduction in overall revenue results from an underperforming manufacturing plant. Deep Learning (DL) and Computer Vision (CV) can serve as an automated surveillance system for continuous monitoring. The combination of DL and CV, which act as eyes and brain, can capture the activities in the manufacturing plant and convert such visual data into meaningful information in the form of object detections. The requirement is to detect the status of maintenance, operator and machines and their interactions based on user-defined thresholds. Metrics such as machine utilization, Mean-Time to Repair, Mean-Time to Failure, Operator in position and maintenance behavior are derived from these detections. This paper compares two approaches using data for a real-time analytics dashboard to track productivity losses in a manufacturing setup. The results are compared to select a better performing approach regarding their contribution to productivity analytics accuracy. The first approach uses weights of the Yolov4 model to detect maintenance and operator as a person in combination with HSV colour filtering for machine status and maintenance operator classification. Later uses Transfer learning on Yolov4 to detect machine status, maintenance and operator. Literature shows that transfer learning is significantly faster and consumes less data than conventional model training, so an approach to compare conventionally trained models was unnecessary. The results from the experiment show that the second approach using transfer learning was accurate compared to the first approach.

Keywords
Productivity monitoring, Yolov4, Deep Transfer Learning, analytics, Legacy Machines

1. Introduction
Monitoring productivity losses and minimising them is critical for the efficient functioning of a manufacturing plant and is more vital in cases where there is a shortage of products, like in the semiconductor industry. Not monitoring productivity losses to minimise them will result in substandard product quality and reduced plant throughput. Hence proper monitoring of machines and working individuals is required for the appropriate functioning of a manufacturing plant.

Industry 4.0 acts as a catalyst in accelerated growth through systematic adaptation of new generation cutting edge tools like Artificial intelligence, Big Data Analytics and cloud computing (R. Dubey et al. 2020). Machines in the latest manufacturing plants practising industry 4.0 come equipped with data collection and data transmission capabilities. The problem is with legacy machines that do not have the data collection, processing and evaluation. These legacy machines also fail to communicate with other machines. However, these legacy machines are in perfect working conditions and can deliver high-quality products as per the defined standards. So, external components are required to enable the machine to be part of the industry 4.0 initiative. These external components help in capturing data needed for productivity monitoring in the manufacturing plant. Literature supports the possibility of this conversion (V. Nguyen et al. 2018). Our goal is to track the maintenance person, operator person, and machine operation for this study. So, a camera for a single line covering ample details is necessary for the data capturing. This data is then processed and fed to a Deep Neural Network to obtain a semantic interpretation of data to display the productivity analytics in a dashboard.

Literature suggests that the transfer learning approach on a Deep Learning model is more efficient than traditional training. Evidence indicates that transfer learning requires lesser data and takes less time to converge to the same accuracy as conventional training methods (S.Li et al. 2020). This study compares two deep learning methods: the transfer learning method and using frozen weights for detections from a state-of-the-art model. These detections can be used as data for deriving productivity analytics. The results are compared, and a better approach is finalised for deriving analytics to be displayed in the dashboard.

The productivity analytics monitored are as follows:

1. Operator tracking: The operator is detected in the manufacturing environment using the deep learning model. The distance of the operator from a machine is used as deciding factor for determining whether he is in the machine's working area. This way, we can track the percentage of the actual time the operator is effectively working on the machine over his shift and issue corrective actions and reports to improve the process.
2. Maintenance tracking: We can make two observations with the detections of maintenance from the Deep Learning Model. The first one is the time taken by maintenance to reach the machine after it has been down. The second is the time taken by maintenance after he arrived at the machine. Both are based on the distance of the maintenance person from the machine.
3. Machine status: Images on Andon light of respective machine are fed to the deep learning model to decide whether the light is red, meaning the machine is not working, or green, meaning the machine is working. The time for which light is on is calculated, and machine utilization is derived.

2. Methodology and Experimental Setup

2.1 Primary Goals

- Detect Maintenance and operator persons and distinguish them from each other
- Identify the status of the machine (On/Off) by detecting the colour of the Andon light
- Find the distance of the machine and operator to be classified if they are working around the machine.

2.2 Experimental Setup

The view from the camera is shown in Figure 1. Camera placement intends to cover every Andon light on machines in that bay, operators and maintenance whenever they are at their positions. The machines monitored are the legacy type; hence they cannot transmit machine status. So, we use external cameras and deep learning over the video data from cameras to extract the information and use the information for analytics. The person working in the line can be identified based on the shirt they are wearing.

2.3 Model Selection

The main areas that need to be clear before selecting a model are:

Figure 1. Experimental Setup

a) Problem Domain
b) Understanding Dataset
c) Accuracy
d) Architecture

Our domain is object detection from computer vision, and as we are dealing with people being one of the detections, COCO Dataset has the people class labelled. We have very little data available, so the approach has to be in that model gives decent accuracy even with fewer data. Transfer learning can be one such method that facilitates retraining the model with significantly fewer data. After a thorough literature survey and comparison of accuracy vs speed of the model, Yolov4(A. Bochkovskiy et al. 2020) architecture is considered. The following Figure 2 indicates the performance of yolov4 with other state-of-the-art models.

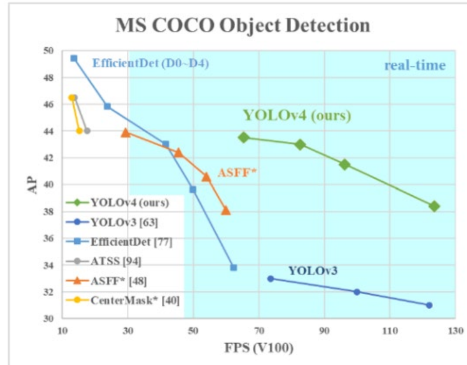


Figure 2. Comparison of Yolov4 with other models (Ref. Alexey Bochkovskiy et al Yolov4)

2.4 Methodology

This study compares two methods to find the optimum approach for productivity analytics. The key is to use images from the camera video data and obtain detections using a deep learning algorithm. The first method uses weights from a model trained on the COCO dataset and HSV colour filtering to differentiate the detections. The second method uses a transfer learning approach on the weights trained on the same COCO dataset.

2.4.1 Method 1

Steps in method 1:

1) Collect the pre-trained weights on COCO Dataset for the Yolo v4 model
2) Use these weights to obtain person detections
3) Use HSV colour Filtering method to detect the state of Andon light and also differentiate between operator and maintenance.

2.4.1.1 HSV Colour Filtering method

The maintenance and operator can be differentiated by the colour of their uniforms. HSV stands for Hue, Saturation and value, respectively. HSV representation is close to human perception of colours rather than RGB representation. HSV is represented in cylindrical geometry, as shown in Figure 3. Hue is on the angular dimension with red at 0°

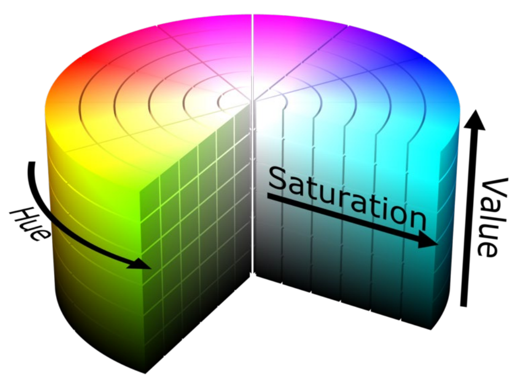


Figure 3. HSV Cylinder (Ref. Wikipedia)

angle, green at 120° angle, blue at 240° angle and closing again with red at 360° angle. The radial direction represents saturation with zero at the centre to 1 at the circumference. Saturation value 1 represents the purest form of the colour. The value is the normal dimension of the cylinder, which increases from 0 at the bottom to 1 at the top. Value 0 represents the blackest tint of that particular colour, and value 1 represents the whitest tint of that colour. In our

Figure 4. HSV Colour Filtering of ROI

experiment, the maintenance and operator are moving objects of interest, and Andon Lights are stationary in the image coordinate system. Hence, we need to obtain these regions of interest (ROI) for both moving and stationary objects for HSV filtering to be applied. The ROI for moving objects is obtained by using a deep learning model trained on the COCO dataset having 'person' as one of its classes for detecting human beings. The Andon lights being stationary can be manually defined in the fixed frame. After obtaining the ROI, we apply HSV filtering of the colours within a range defined by the trial-and-error method and the pixels within the range are extracted and converted to a binary image. The image is void of other colours which are not in the filtered HSV range, as shown in Figure 4. Mathematically, the pixel intensity values of other pixels which do not fall in the range mentioned are turned to zero, which causes the black background. The mathematical manipulation of pixels is shown in Figure 5. few post-processing steps are also required to improve the detection accuracy using this method (Y. Omori et al. 2019) Their sequence of operations is as shown in Figure 6

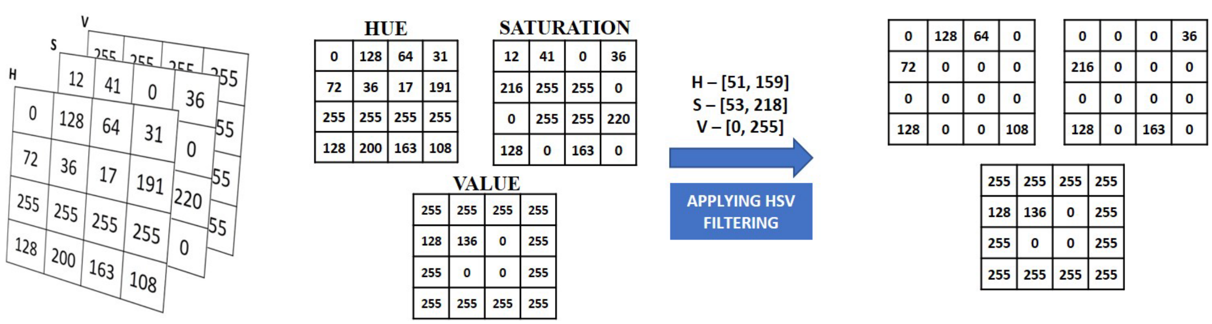


Figure 5. Mathematical representation of HSV Filtering

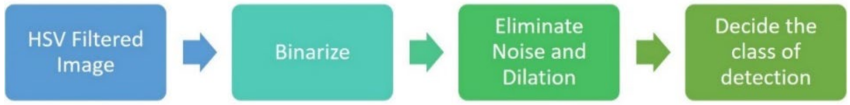


Figure 6. Post-Processing for improving HSV detection results

a) Binarize: After filtering the image using HSV, it is converted to grayscale. This grayscale image is converted to a binary image using OpenCV. The final image is as shown in Figure 7(a)
The mathematical equation governing the binarisation is as follows:

$$dst(x,y) = \begin{cases} maxVal, & if\ src(x,y) > threshold \\ 0, & otherwise \end{cases}$$

Figure 7(a). Binarization

b) Eliminate noise and dilation: There can be noise in the binary image due to various external factors and sensor inaccuracy. Removing this noise will help in achieving better detections. A sample image after noise removal is shown in Figure 7(b). The blur used here is a gaussian blur
The mathematical equation governing Gaussian blur is as follows:

$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

c) Decide the class of detection: In this step, we obtain the class of input ROI by using the final image in binary. After the HSV filtering, only the objects having colour within the defined range are displayed, and the rest are discarded. Hence, we can say that if there is a white pixel even after filtering, then there is the desired class which we have defined using the HSV range. For this process, we are calculating the average pixel intensity value of the ROI and setting a threshold based on trial and error. If the average pixel intensity value crosses a certain threshold, we can come to the conclusion that our desired object is in the ROI. The results of this method are discussed at the end.

Figure 7(b). Gaussian Blur

2.4.2 Method 2

We are using transfer learning on the pre-trained COCO weights of the Yolov4 model in this method. There are specific steps we follow in this method for preparing data to feed into the Yolov4 model, and they are as follows:

a) Data Collection: Data in the form of images is essential for a Deep Learning (DL) model to learn. A DL model is only as good as the training samples it has been trained on. So, balanced data with all the class instances can be crucial in helping a model achieve reasonable accuracy. The objects that should be in our images for the DL model to learn are
   - Instances of the maintenance person
   - Instances of operator person
   - Instances of Red Andon Light
   - Instances of Green Andon light

   These instances can be collected from the video cameras placed in the production line.

b) Data Pre-Processing: After the raw data has been collected from the scene, it has to be cleaned before feeding to the deep learning model. The video data has to be made into images, and the images have to be scaled into the desired resolution for feeding into the model. Any blank images or images not readable by the computer can be removed. Images with noise can be denoised or deleted and made sure that images are free from errors and ready for labelling. Labelling is a process where we tell the model that at this particular location in this

Figure 8. Data Labelling

image, this particular class is present. We draw rectangular boxes and mention that particular class so that model will be able to identify the class and location. The labelling has to follow the Yolo format so that the generated text file will be in the prescribed format for the model, as shown in Table 1. The x,y denotes the centre of a bounding box. The labelling procedure in LabelImg is shown in Figure 8

Table 1. Yolov4 labelling format

Class	X	Y	Width	Height
0	0.633984	0.703472	0.100781	0.420833
1	0.314844	0.540278	0.07125	0.305556
2	0.46875	0.418056	0.028125	0.072222

c) Data Augmentation: Data augmentation is a technique where transformations are applied to the existing image data to prevent the deep learning model from overfitting or learning unnecessary data features. Our data consists of 1500 images, out of which an 80:20 split was applied to divide it into training and testing data. The instances of each class are as shown in Table 2. It can be seen that there is an imbalance in the data. Hence there is a chance for the model to overfit onto the data with more instances. The goal of data augmentation is to increase the variability of the input images so that the designed object detection model has higher robustness to the images from different environments (A. Bochkovskiy et al. 2020). CutOut(T. DeVries), MixUp(H. Zhang et al. 2018), CutMix(S. Yun et al. 2019) are a part of Bag of freebies in the Yolov4.

Table 2. Training Data Statistics

Class	Data Instances
Operator	1200
Maintenance	500

Andon Red Light	1000
Andon Green Light	200

2.4.2.1 Transfer learning process on Yolov4

Transfer learning in Deep learning is where a model trained on a vast dataset similar to our application is taken and retrained on the scarcely available data for our particular use. This approach is usually taken when the available data

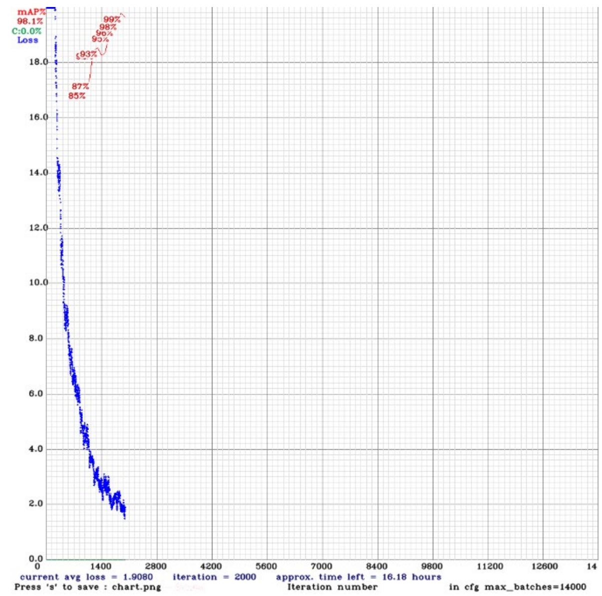


Figure 9. Graph of mAP vs iterations

is scarce. Yolov4 model trained on COCO Dataset taken and is trained on the image data processed from the camera. The training was done for 2000 iterations, and mAP value of 98% was achieved. This high mAP shouldn't be considered an actual performance of the model as there is class imbalance. In the instances with class imbalance, a confusion matrix is plotted from which precision, recall and accuracy are calculated for each class individually. This estimation will give a true performance of the model. The graph shown in Figure 9 indicates the decreasing trend in loss and increasing trend in mAP with the increase in iterations.

2.5 Distance Calculation using Perspective transform

After obtaining the detections from the model, the distances between subjects are needed to evaluate if the maintenance or operator is in the working area attending the machine. To calculate the distance between machine and maintenance as well as machine and operator, we need to obtain the birds-eye view of the image. Birds-eye view is a projection of the environment as if we are looking from the top of the scene. A graphical view of the birds-eye view is shown in Figure 10, where all the image pixels are projected onto the bottom plane. The advantage of this view is that whenever we calculate distances between objects, the pixels not being in contorted positions give us a distance proportional to the real-world distance.

The statistics that we need to track from the detections are:

a) Time of operator away from the machine
b) Time spent by the maintenance to repair the machine
c) Time taken by maintenance to respond to the machine down signal

By tracking these, we can determine the attentiveness of the operator, response time of maintenance for the machine down signal and time for which machine is not working in the line. These can indirectly help us in deriving KPIs such as machine utilization, operator efficiency etc. By identifying processes where the losses are significant and can be optimized, we reduce them and thereby reduce the productivity losses.

2.5.1 Obtaining a Birds-eye view of the image

We can use the open-source library OpenCV for the perspective transformation of the image, which will give us birds-eye view. Before using cv2. getPerspectiveTransform(src, dst) we need to identify 4 points on the image, two on the x-axis and two on the y-axis such that lines passing through them are perpendicular to each other in real life.
The following is a mathematical equation that governs the perspective transformation of points from src to dst

$$\begin{bmatrix} t_i x'_i \\ t_i y'_i \\ t_i \end{bmatrix} = M * \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

$$dst(i) = (x'_i, y'_i), src(i) = (x_i, y_i), i = 0, 1, 2, 3$$

Parameters
src - Coordinates of quadrangle vertices in the source image.
dst - Coordinates of the corresponding quadrangle vertices in the destination image.

Now we have a M which is a transformation matrix that we can use to calculate the warping of the image
After we obtain the matrix M, we can warp the image so that all the pixels are shifted to the new locations, forming a birds-eye view of the image. We can use cv2.warpPerspective(image, mapmatrix, (maxWidth, maxHeight)) from OpenCV library to warp the image.

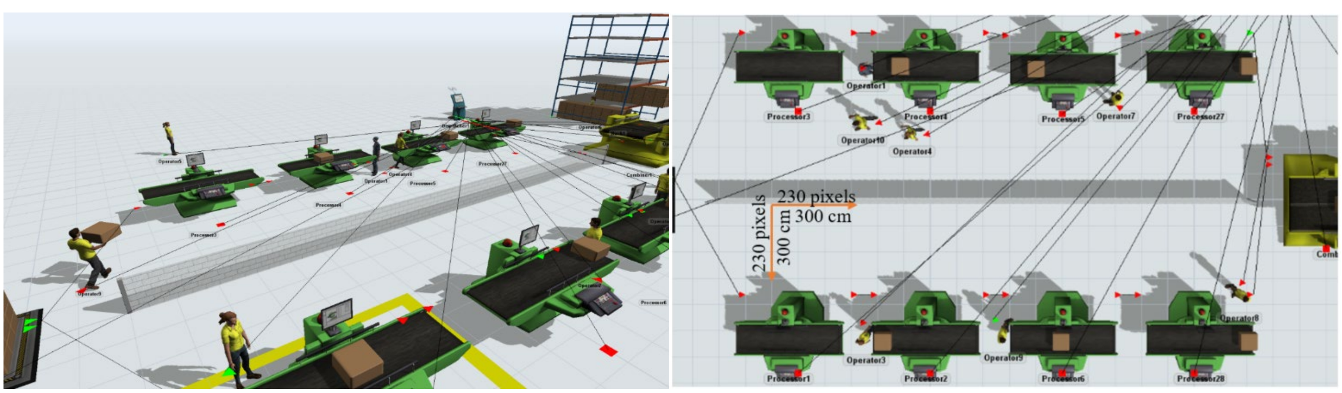
$$dst(x, y) = src\left(\frac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}}, \frac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}}\right)$$

Where M_{ij} is the element in the ith row and jth column in the transformation matrix obtained from the perspective transform.
Parameters:
src - input image
dst - output image that has the size dsize and the same type as src
M - transformation matrix
(maxWidth, maxHeight) - size of the output image

After performing these two operations, we obtain an image similar to Figure 10(b).

2.5.2 Distance Calculation

After obtaining the perspective transform, we have to calculate the distance between the centroid of the bounding box and the machine. This can be done by taking (x,y) coordinates of the bounding box centroid and (x,y) coordinates of the machine reference point. But this distance would be the pixel distance. The actual distance can be obtained by multiplying this pixel-to-pixel distance with a multiplication factor. The multiplication factor can be derived by taking a reference element like a scale or any object of standard size and placing it in our experimental environment. Keeping the camera angle the same, we can capture the standard element and measure the pixel-to-pixel distance. The standard



(a) Camera View (b) Birds eye view

Figure 10. Graphical representation of Camera view and Birds eye view

element is whose dimensions we know in real world. From the Figure 10(b) we can see that tiles are taken as reference elements and the pixel dimensions are calculated from the image. As we already know the size of those tiles in real world, we can use the below formula for calculating the multiplication factor

$$unit\ distance = \frac{real\ world\ distance\ covered\ by\ reference\ element(centimenters)}{pixel\ distance\ covered\ by\ reference\ element\ (pixels)}$$

$$corrected\ distance\ = unit\ distance * d$$

Where d is the pixel distance obtained from the detection and machine

3. Results and Discussions

The results for both approaches are discussed here. The test data used in both the approaches is same. As there is a class imbalance in the data, we need to obtain precision and recall for each class to know the model's actual performance concerning each class. Precision is a measure of the percentage of correct detections, whereas recall gives us the proportion of correct detections compared to ground truth. The accuracy considers both predicted positives and negatives and actual positives and negatives (Table 3).

Table 3 Test Data Statistics

Class	Data Instances
Operator	250
Maintenance	100
Andon Red light	250
Andon Green Light	50

3.1 Object detection results

a) Operator:
The precision for operator detection in the HSV method is 0.708, and the transfer learning was 0.9794. This means only 70.8% of the detected were correct, and the remaining were false detections in HSV, and 0.9794 for transfer learning indicates that out of the detections made, all were correct. However, the metric does not discuss missed detections. The recall value for HSV is 0.388 % which indicates that the model could predict only 38.8 % of the actual ground truth labelled operators and missed the rest, and 0.9520 for transfer learning indicates that the model could predict 95.2% of the ground truth labels for the operator. A similar interpretation can be made with the following results.

HSV APPROACH			
TOTAL N = 300		ACTUAL	
		NEGATIVE	POSITIVE
PREDICTION	NEGATIVE	10	153
	POSITIVE	40	97

Figure 11. Confusion Matrix of Operator Class using HSV method

TRANSFER LEARNING			
TOTAL N = 300		ACTUAL	
		NEGATIVE	POSITIVE
PREDICTION	NEGATIVE	45	12
	POSITIVE	5	238

Figure 11. Confusion Matrix of Operator Class using DTL method

	PRECISION	RECALL	ACCURACY
HSV	0.7080	0.3880	0.3566
DTL	0.9794	0.9520	0.9433

Figure 13. Results of operator class using HSV and DTL methods

b) Maintenance

HSV APPROACH			
TOTAL N = 300		ACTUAL	
		NEGATIVE	POSITIVE
PREDICTION	NEGATIVE	102	43
	POSITIVE	98	57

Figure 14. Confusion Matrix of Maintenance Class using HSV method

TRANSFER LEARNING			
TOTAL N = 300		ACTUAL	
		NEGATIVE	POSITIVE
PREDICTION	NEGATIVE	194	6
	POSITIVE	6	94

Figure 15. Confusion Matrix of Maintenance class using DTL

	PRECISION	RECALL	ACCURACY
HSV	0.3677	0.5700	0.5300
DTL	0.9400	0.9400	0.9600

Figure 16. Results of maintenance class using HSV and DTL method

c) Red Andon Light

HSV APPROACH			
TOTAL N = 300		ACTUAL	
		NEGATIVE	POSITIVE
PREDICTION	NEGATIVE		
	POSITIVE		

TRANSFER LEARNING			
TOTAL N = 300		ACTUAL	
		NEGATIVE	POSITIVE
PREDICTION	NEGATIVE		8
	POSITIVE		242

	PRECISION	RECALL	ACCURACY
HSV	1.0000	0.7680	0.8066
DTL	1.0000	0.9680	0.9733

Figure 17. Conf... Andon usi...

...ix of Red ...thod

Figure 19. Results of Red Andon using both the methods

d) Green Andon Light

HSV APPROACH			
TOTAL N = 300		ACTUAL	
		NEGATIVE	POSITIVE
PREDICTION	NEGATIVE	250	11
	POSITIVE	0	39

Figure 20. Confusion matrix of Green Andon using HSV method

TRANSFER LEARNING			
TOTAL N = 300		ACTUAL	
		NEGATIVE	POSITIVE
PREDICTION	NEGATIVE	250	4
	POSITIVE	0	46

Figure 21. Confusion matrix of Green Andon using HSV method

	PRECISION	RECALL	ACCURACY
HSV	1.0000	0.7800	0.9633
DTL	1.0000	0.9200	0.9866

Figure 22. Confusion matrix of Green Andon using HSV method

3.2 Analytics Dashboard Results

3.2.1 Machine Analytics

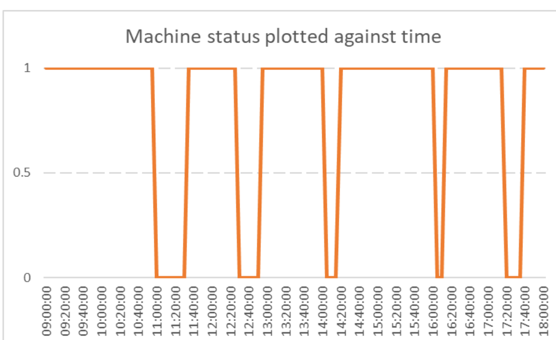


Figure 23. Machine status

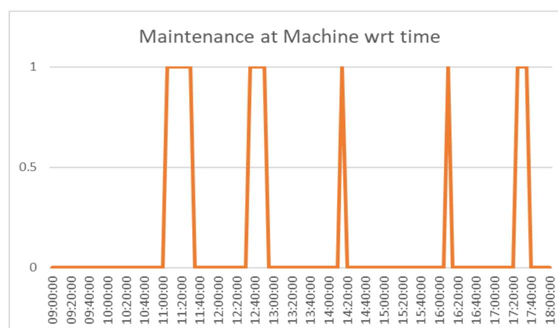


Figure 24. Maintenance Status

The detection data is collected in real-time and displayed as a dashboard using Power BI. In Figure 23, it can be seen that time is on the horizontal axis, and Boolean value 0 or 1 is taken on the vertical axis. Here 0 represents off, and 1 represents on. The status of the machine with respect to time is displayed in the Figure 24. From here, we can calculate the statistics such as Mean Time to Failure (MTTF), Mean Time to Repair (MTTR) and Availability

$$MTTF\ (m_f) = \frac{number\ of\ operational\ hours}{number\ of\ failures}$$

$$MTTR\ (m_t) = \frac{Total\ maintenance\ time}{total\ number\ of\ repairs}$$

$$A = \frac{m_f}{m_f + m_t}$$

3.1.2 Operator Analytics

Figure 25 gives us information about how much time an operator is spending at the machine based on how far he is from the machine. A threshold value is fixed by trial and error. As long as the operator is within this threshold distance, it is considered to be in the working area. From this dashboard, we can analyse the efficiency of the operator during shift hours.

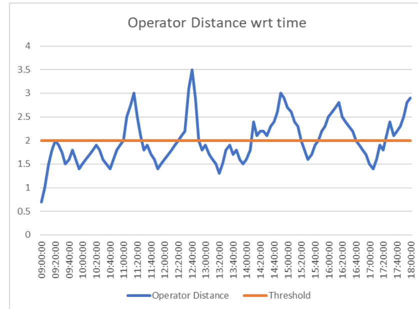


Figure 25. Operator Distance

4. Summary and Conclusions

4.1 Summary

From the results, it is clear that there are low precision and recall values for individual classes in the HSV method when compared to the Transfer Learning method. This indicates that the Transfer learning approach would be accurate for detection than the HSV colour filtering method. The weights of the Transfer learning method were used to obtain the dashboard results. The transfer learning method consumes fewer data compared to traditional deep learning methods. But HSV colour filtering required even fewer data. It is important to note that the results of the HSV method are not inaccurate because of using lesser data compared to the Transfer learning method. It is worthy to note that transfer learning approach results can be improved by improving the quality of data we supply. The reason for the poor performance of the HSV method lies in the colour filtering method itself. When we filter unnecessary colours from the region of interest, certain elements creep into the region of interest as the subject moves. When they are in the same colour as that of the person of interest, these elements affect the result, causing the model to perform poorly. For example, in Figure 26, the bounding box indicating the person marked in white when in front of the machine has the machine's blue colour making its way into the bounding box, which will cause the person in the white-coloured box also to be classified as an orange class due to domination by blue colour over the brown colour within ROI.

Figure 26. Background intrusion into HSV results

4.2 Conclusions

We compared two methods and concluded that the transfer learning approach is better to generate data for dashboards to monitor productivity losses in a manufacturing setup. The Transfer learning method is faster to train and lighter in computation with the perks of consuming less data than traditional Deep Learning methods. Transfer learning performs better consistently throughout the testing for each class, as shown in the results. From the dashboards generated, strict monitoring can be done on the processes in the manufacturing, and processes contributing to a loss in productivity can be rectified by conducting a root cause analysis.

References

Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, *YOLOv4*: Optimal Speed and Accuracy of Object Detection*, 2020.*

Christos Manettas, Nikolaos Nikolakis, Kosmas Alexopoulos, Synthetic datasets for Deep Learning in computer-vision assisted tasks in manufacturing*, Procedia CIRP,* vol. 103, pp. 237-242, 2021.

Dennis Bakir, Robin Bakir, Florian Engels, Industry_Integrator as retrofit solution for digital manufacturing methods in existing industrial plants*, Procedia Manufacturing,* vol. 17, pp. 1009-1014, 2018

H. Zhang, M. Cisse, Y.N. Dauphin, D. Lopez-Paz, MixUp: Beyond empirical risk minimisation*, 6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.* pp. 1–13, 2018.

Jisun Baek, WooRam Park, The impact of improved passenger transport system on manufacturing plant productivity*, Regional Science and Urban Economics,* 2022.

Luke Scime, Jack Beuth, Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm*, Additive Manufacturing,* vol. 19, pp. 114-126, 2018.

Nicolas Jourdan, Tobias Biegel, Volker Knauthe, Max von Buelow, Stefan Guthe, Joachim Metternich, A computer vision system for saw blade condition monitoring*, Procedia CIRP,* vol. 104, pp. 1107-1112, 2021.

Pinku Yadav, Olivier Rigo, Corinne Arvieu, Vibhutesh Kumar Singh, Eric Lacoste, Data processing techniques for in-situ monitoring in L-PBF process, *Journal of Manufacturing Processes,* vol. 81, pp. 155-165, 2022.

R. Dubey, A. Gunasekaran, S.J. Childe, D.J. Bryde, M. Giannakis, C. Foropon, D. Roubaud, B.T. Hazen, Big data analytics and artificial intelligence pathway to operational performance under the effects of entrepreneurial orientation and environmental dynamism: A study of manufacturing organisations, *Int. J. Prod. Econ.* Vol. 226 2020.

Subham Kailthya, Uma Kambhampati, Road to productivity: Effects of roads on total factor productivity in Indian manufacturing, *Journal of Comparative Economics,* vol. 50, no. 1, pp. 174-195, 2022.

S. Li, P. Zheng, Transfer Learning for Smart Manufacturing: A Stepwise Survey, *IFAC-PapersOnLine.* vol. 53 pp. 37–42, 2020.

S. Yun, D. Han, S. Chun, S.J. Oh, J. Choe, Y. Yoo, CutMix: Regularisation strategy to train strong classifiers with localisable features, *Proc. IEEE Int. Conf. Comput. Vis.* pp, 6022–6031, *2019-October*

Sri Sudha Vijay Keshav Kolla, Diogo Machado Lourenço, Atal Anil Kumar, Peter Plapper, Retrofitting of legacy machines in the context of Industrial Internet of Things (IIoT), *Procedia Computer Science,* vol. 200, pp. 62-70, 2022.

T. DeVries, G.W. Taylor, Improved Regularization of Convolutional Neural Networks with Cutout, Thorsten Wuest and Daniel Weimer and Christopher Irgens and Klaus-Dieter Thoben, *Machine learning in manufacturing: advantages, challenges, and applications, Production \& Manufacturing Research*, pp. 23-45, 2016

V. Nguyen, A. Dugenske, An I2C based architecture for monitoring legacy manufacturing equipment, *Manuf. Lett.*vol. 15, pp. 67–70, 2018

Xin Lin, Qisheng Wang, Jerry Ying Hsi Fuh, Kunpeng Zhu, *Motion feature-based melt pool monitoring for selective laser melting process, Journal of Materials Processing Technology,* Volume 303, 2022,

Yuanbin Wang, Pai Zheng, Xun Xu, Huayong Yang, Jun Zou, Production planning for cloud-based additive manufacturing—A computer vision-based approach, *Robotics and Computer-Integrated Manufacturing,* vol. 58, pp. 145-157, 2019

Y.Omori, HSV Color Space Based Lighting Detection for Brake Lamps of Daytime Vehicle Images, *J. Comput.* vol.14 pp. 25–30, 2019.

Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 6154–6162, 2018.

Biographies

A Venkata Sai Kumar Reddy is a Masters Student from Department of Mechanical Engineering, National Institute of Technology Warangal (NITW). He has great interest pursuing research in the field of applications of artificial Intelligence to Manufacturing Systems.

Dr. V. Vasu is graduated in Mechanical Engineering from V R Siddhartha Engineering College, Vijayawada in 1999. Earned his post-graduation in Manufacturing Engineering from BITS, Pilani in 2003, PhD from JNTU, Hyderabad in 2010 and POST-DOC under Erasmus Mundus fellowship from Politecnico Di Milano, Italy, Europe 2014. Presently, he is serving as a Professor at National Institute of Technology, Warangal, India. He has published several papers in international journals and has participated in several conferences and presenting technical papers. He developed online pedagogy course for "Mechatronics" and Systems & Controls under ICT MHRD, India. He has guided 4 PhD works and is currently guiding 3. He was awarded Young Engineer 2010 Award by IE India and Govt. of AP and Outstanding researcher in Nanotechnology Awards 2016. He is also a member in Siemens Centre of excellence Industry 4.0, NITW.