ONE-SHOT PPE DETECTION FOR BAKE OVENS USING RESNET50 CONVOLUTIONAL NEURAL NETWORK

Michael C. Trinidad
Diego Jose L. Cabalza
Rommel M. Fajardo

Software Development and Automation, Central Engineering
Onsemi, Golden Mile Business Park, SEZ Governor's Drive, Maduya, Carmona, Cavite, Philippines
Michael.Trinidad@onsemi.com, Rommel.Fajardo@onsemi.com, DiegoJose.Cabalza@onsemi.com

ABSTRACT

This paper proposes a novel method for image classification with applications in Personal Protective Equipment (PPE) detection. The method uses the ResNet50 architecture, a prominent Convolutional Neural Network for precise image detection and categorization. The method boasts a complex structure comprising 50 layers and has demonstrated exceptional proficiency across various computer vision tasks. The paper also discusses advanced image preprocessing techniques, such as data normalization, image resizing, and data augmentation, to enhance the model's feature extraction and batch processing abilities.

The paper trains the model on a diverse spectrum of PPE items, from face shields and aprons to masks, gloves, and sleeves. This is a prerequisite before operating high-temperature equipment like bake ovens.

Following an extensive model training spanning a designated number of epochs, the model exhibited a remarkable achievement: attaining a validation accuracy of 99.6%. Interestingly, the model excelled in distinguishing between appropriate and improper utilization of Personal Protective Equipment, demonstrating the project's pioneering effort with implications for other future applications.

1. 0 INTRODUCTION

In the field of computer vision, images come in various sizes, colors, and shapes, displaying diverse visual characteristics. Detecting and classifying operators wearing Personal Protective Equipment correctly is crucial for safety and accident prevention due to exposure to extreme temperatures. At Probe bake ovens, the current PPE detection approach heavily relies on reading QR (Quick Response) codes attached to items like gloves, sleeves, aprons, and face shields with the exception of face mask detection that is performed using a machine learning model trained through transfer learning, which utilizes pre-trained weights from large-scale datasets like ImageNet. The main drawback of the current implementation is the relatively slow detection speed since PPE items are detected one by one (Figure 1). In addition, QR codes have to be re-attached in case the PPE needs to be replenished.

In order to address this challenge, the team proposes an alternative method that leverages Convolutional Neural Networks (CNNs), specifically ResNets, to detect all PPE simultaneously (one-shot detection). QR code detection combined with transfer learning and the CNN-based approach each have their own strengths and limitations. The conventional QR code approach involves scanning the code directly, while face mask transfer learning significantly reduces training time by leveraging pre-trained models that have learned generic features from large-scale datasets. On the other hand, CNNs require a substantial volume of labeled images for effective model training. While CNNs offer a promising solution for automated PPE detection, they pose challenges such as the labor-intensive task of building and annotating a diverse dataset with representative samples and ensuring a balanced sample distribution. Additionally, training deep CNN architectures like ResNets can be computationally demanding and require substantial computational resources.

Figure 1. Old Method of PPE Detection. The photos show sequential PPE detection using QR Codes for face shields, aprons, gloves, and sleeves.

The aim of this study is to explore the implementation of ResNet50 in image classification tasks, incorporating various image processing techniques. In doing so, it has the potential to pave the way for other applications in manufacturing settings, such as defect detection and motion analysis. Figure 2 shows a diagram of the typical implementation of ResNet50.

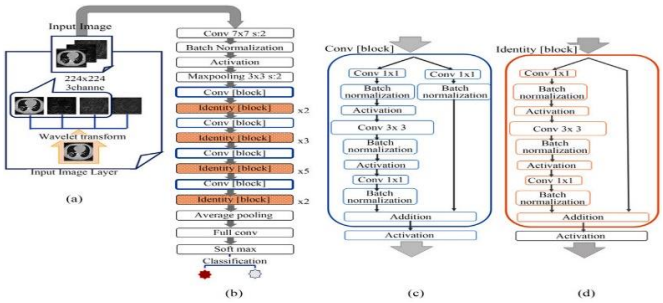

Figure 2. ResNet50 Architecture. This is an example of ResNet50 architecture with a 3-channel image input layer [1].

2. 0 REVIEW OF RELATED WORK

An algorithm named YOLO (You Only Look Once) is another known object detection model introduced in 2016 by Joseph Redmon et al. It works as an end-to-end neural network that makes predictions of bounding boxes and class probabilities all at once. The YOLO model takes an image as input and then uses a deep convolutional neural network to detect objects in the image. Its architecture consists of convolution layers pre-trained using ImageNet and converted to perform detection with the final fully connected layer predicting both class probabilities and bounding box coordinates [2].

A study on Personal Protective Equipment Detection with Camera with Live Camera conducted by Bhing and Sebastian [3], used a YOLO object detection model with a Roboflow computer vision developer framework and TensorFlow platform to detect PPE for COVID-19 by using a dataset consisting of collected images from online datasets and their own captured images. Their results showed that they were able to detect PPE such as face masks, face shields, and gloves from a real-time webcam feed and in uploaded videos.

Another study conducted by Karlsson and Strand [4], created another model with the use of a YOLO model to detect PPE with the following items used often by industry workers: Hardhat, Safety vest, Safety gloves, Safety glasses, and hearing protection. While their results in PPE detection show near-perfect accuracy within a range of three to five meters, the hardware used was limited with a 12-megapixel cellphone camera used for image capturing and the model was run without a GPU, which would make the processing power faster to produce the results.

A project [5] made by Balakreshnan et al. demonstrated the application of Artificial Intelligence (AI) and machine vision to identify PPE. A low-power 8-megapixel camera with Linux OS utility with a low-power graphics processor was used as hardware. Microsoft Azure Custom Vision AI and Intelligent AI Services, in conjunction with low-cost vision devices with lightweight onboard AI capability, were used as a platform for a deep learning neural network model using publicly available images. The results showed poor performance at first but was vastly improved by collecting images specifically from a laboratory environment, where the PPE detection model is expected to be used. The affordability and flexibility of the system showed multiple benefits such as improved safety and better detection and recording of safety violations. The hybrid AI architecture approach allowed for flexibility in training and deployment based on the capability of local computing resources.

A similar paper [6] presented by Nath et al. introduced, tested, and evaluated three Deep Learning-based approaches for detecting PPE attire with one approach using a Convolutional Neural Network (CNN) based classifier model such as ResNet-50. They concluded that another approach, which used a version of a YOLO model, currently provided the most accurate detection of PPE attire. However, it was also added that the approach of using CNN models achieved higher accuracy at first, but errors generated in the latter parts of its simulation lowered its final accuracy score. This indicated that potentially better performances can be achieved by improving the classifiers for detecting PPE with higher accuracy.

3.0 METHODOLOGY

3.1 Data Collection

In the image acquisition process, the team utilized the Microsoft LifeCam Studio USB 5 Mega Pixel 30 Frames per second Webcam. The camera was positioned near the Oven area to collect data. Supervisors were given explicit instructions to capture sample images of operators wearing PPE, both properly and improperly. Various scenarios were simulated, including instances where certain PPE items were missing, such as gloves or face shields. It was crucial to ensure a balanced distribution of data for both classes to prevent bias during the training process. It is important to note that the soundness of the resulting model heavily depends on the quality of the collected data.

To facilitate data collection, the team developed a camera application that included bounding boxes indicating the specific locations where different PPE items should be present in the image as shown in Figure 3. This allowed for the creation of regions of interest for machine learning and provided visual references for the operators during the data collection process. Around 1200 images were collected (50% for each class which made it a balanced dataset).

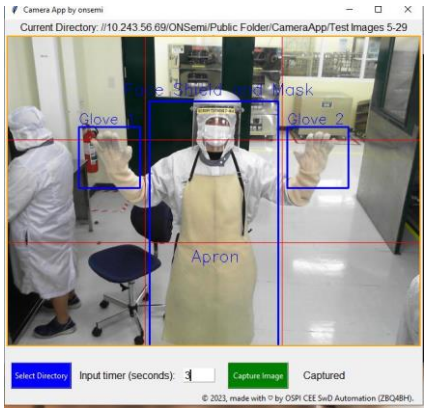

Figure 3. Camera Implementation Set-Up. This is the camera app developed by the team with bounding boxes to enclose different PPE used in Oven Bake, making detection easier for the algorithm.

3.2 Annotation

The process of annotation involves the task of labeling images and categorizing them into separate clusters. In order to achieve this, the images were cropped to isolate specific regions of interest (ROIs). In giving emphasis to these relevant regions, the system can enhance its generalization capabilities and extract distinctive features from the localized areas, see Figure 4.

Figure 4. Proper vs Improper PPE Attire. The photos represent images of properly worn PPE (L) and improperly worn PPE (R).

Proper means complete PPE while improper means incomplete. In the example of improperly worn PPE, the operator is not wearing a face shield, high-temperature gloves, and sleeves. The image was cropped to remove unnecessary objects that offer no value in model training.

3.3 Image Pre-processing

Data Augmentation

Data augmentation involves artificially increasing the number of samples in a dataset by applying transformations to existing images. In our study, we employed various data augmentation techniques, including random distortion, random rotation, random brightness, random contrast and random flips. The distortion technique allows for modifications while preserving the aspect ratio of the image. Rotation involves rotating the image by arbitrary degrees and then taking a crop from the center of the rotated image. Random brightness adjusts the brightness level of an image by adding a random value to the intensity of each pixel. Random contrast, on the other hand, adjusts the contrast by stretching or compressing the range of pixel intensities [7]. Random flips simulate the mirror effect by reflecting the image along the chosen axis as shown in Figure 5. By applying these random or controlled transformations, we introduced variations that mimic real-world scenarios and expose the model to different patterns and variations in the data. After data augmentation, the number of images doubled (2400 images).

Figure 5. Data Augmentation Sample. Examples of Data Augmentation with Original Image (L), Flipped Image (M) and Slightly Darkened Image (R).

Image Resizing

Resizing the images to fixed input sizes is advisable to ensure consistent data handling. This practice offers several benefits, including reducing the computational load on the GPU, optimizing memory usage, and expediting the training process by minimizing the number of pixels to process.

Initially, the images had dimensions of approximately 400 x 430 pixels with three color channels (BGR) after annotation. In order to align with the ResNet-50 architecture, which was pre-trained on the ImageNet dataset, the team chose to resize the images to 224 x 224 pixels while maintaining the three color channels as shown in Figure 6. It is important to note that the training process starts from scratch, meaning pre-trained weights will not be utilized.

Resizing the images not only allows for consistent input sizes but also ensures that the model operates within manageable memory constraints. Additionally, by reducing the image dimensions, computational resources are utilized more efficiently, resulting in faster training. Despite the resizing, the team ensured that the image quality remained suitable for the intended purpose of the study.

Figure 6. Implementation of Resizing Image. The image was resized from the original size of 405x436x3 (L) to 224x224x3 (R).

Image Rescaling

Apart from the image size in terms of aspect ratio, the pixel values themselves can also consume a substantial amount of memory during processing. Rescaling the pixel values by dividing them by 255 serves to scale them down and confine them within the range of 0 to 1 as displayed in Figure 7. This rescaling operation aids in reducing the overall image size, leading to an optimization of memory usage.

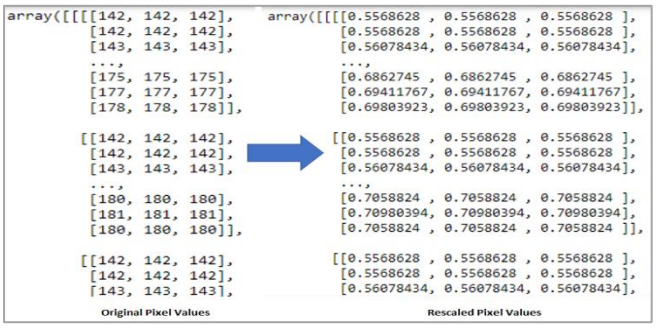

Figure 7. Numerical Output of Re-Scaled Image. The left side is the original pixel values, while the right side, is the rescaled pixel values.

3.4 Splitting of Data and Data Visualization

In this study, it holds significance for the model to grasp the crucial features present in the images it learns from. To achieve this, we have divided the dataset into separate training and validation sets, ensuring that corresponding images in both sets share similarities, albeit not being identical. This approach aims to facilitate the model's ability to generalize its learned patterns and make accurate predictions on unseen data. If the training and validation sets exhibit disparate distributions or characteristics, the model's generalization capability may suffer, resulting in poor performance when applied to real-world data. Data visualizations play a crucial role in verifying whether the training and validation sets contain similar images within their respective groups.

3.5 Model Training

The team utilized NVIDIA's Jetson Tegra Xavier computer to train the data as seen in Figure 8, as conventional laptops and desktop computers are unable to handle large volumes of data, particularly images. The Jetson Tegra Xavier is a 64-bit ARM-based high-performance system-on-a-chip (SoC) designed for autonomous machines. It is equipped with GPUs (Graphical Processing Units) featuring 512 CUDA cores. The model was trained on Keras-Tensorflow deep learning libraries. Keras is a high-level deep learning framework for training and deploying NN (Neural Networks) which is built on top of Tensorflow. Tensorflow on the other hand is an open-source deep learning library developed by Google.

Figure 8. Hardware Set-Up for Machine Learning. The photos show the NVIDIA Jetson Tegra Xavier that was used in the data extensive processing.

During the training phase, the images that were pre-processed and resized to dimensions of 224 x 224 x 3, with pixel values rescaled to the range of 0 to 1 were then fed into a 50-layer convolutional neural network known as ResNet50. The input image goes through the First stage, a convolutional layer with 64 filters of size 7x7, followed by batch normalization and ReLU activation. Zero-padding and max pooling are applied to reduce the spatial dimensions (see Figure 9).

Figure 9a. Sample Image Processing Layer: Input layer.

Figure 9b. Sample Image Processing Layer: Zero padding.

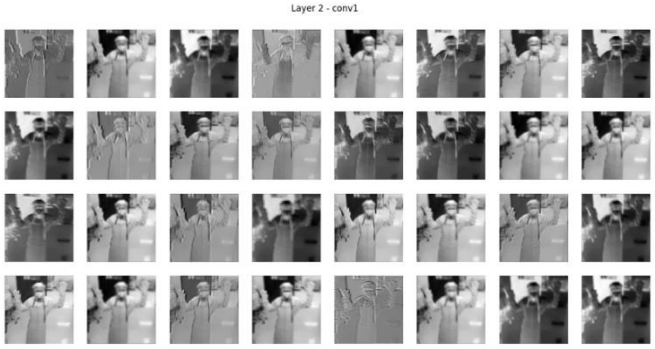

Figure 9c. Sample Image Processing Layer: Convolution.

Convolution is a mathematical operation (dot product) used to extract features from an image (see Figure 10). Padding is done in order to preserve the size of the original image because the convolution process shrinks the size of the image (see Figure 11). Pooling on the other hand is used to reduce the size of the representation [8] as shown in Figure 12.

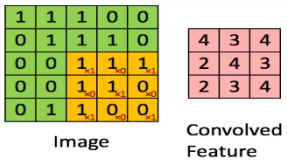

Figure 10. Example of Convolution Process. The matrix colored in green is the image, the matrix colored in yellow (3x3) is the filter or the weight, while the convolved feature colored in red is the output of the convolution process.

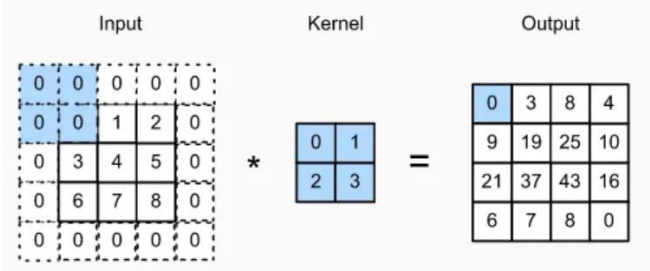

Figure 11. Sample Convolution of Padded Image. Padded image (Input) is convolved in 2x2 matrix (Kernel) producing the Output shown.

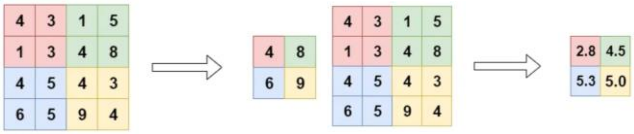

Figure 12. Pooling Illustration Sample. Max Pooling (L) uses the maximum pixel value, Average Pooling uses the averaged value of the neighboring pixels (R).

The second stage consists of multiple blocks that introduce residual connections. Each block is composed of three convolutional layers. The first layer has a 1x1 filter to reduce the number of channels, the second layer applies a 3x3 filter, and the third layer has a 1x1 filter to increase the number of channels. Identity blocks (blocks with no change in spatial dimensions) and convolutional blocks (blocks that change spatial dimensions) are alternated. The output is obtained after several blocks. The third stage consists of multiple blocks with residual connections. The fourth stage further increases the number of filters and continues applying residual connections. The final stage follows a similar pattern but increases the number of filters to an even larger value. After the final stage, an average pooling layer is applied to reduce the spatial dimensions of the feature maps. The output feature maps are flattened into a 1-dimensional vector. The Fully Connected Layer is a dense layer with the desired number of output classes added to the model. The final activation function (e.g., softmax) converts the logits (raw output of a model) into class probabilities (See Appendix A for the visualization of other layers). The output of the model is the predicted class probabilities for the input image. Figure 13 shows the layers of ResNet50 architecture.

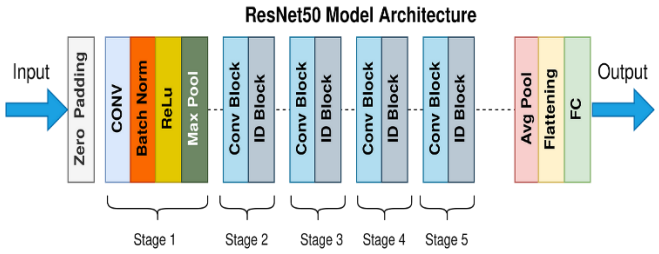

Figure 13. ResNet50 Model Architecture.

Instead of performing binary classification, a multiclass classification approach was employed. The class labels were transformed into one-hot encoded vectors, and the softmax activation function was used in the output layer. This activation function calculates the probabilities for each class, enabling confident predictions across multiple classes.

The optimization algorithm chosen for training was Adam, known for its effectiveness in optimizing deep learning models. The loss function employed was binary cross-entropy, which is well-suited for multiclass classification tasks. To control the weight adjustments during training, a learning rate function was utilized, with different learning rates such as 0.1, 0.01, and 0.001.

The batch size, which determines the number of samples processed in each training iteration, was optimized as well. Furthermore, early stopping was implemented, allowing the training process to halt early if the validation performance did not improve after several defined epochs. In such cases, the weights of the best-performing model will be restored.

The model was trained for 32 epochs, where each epoch involved processing all the training examples, calculating the loss, and updating the model's weights and biases based on the chosen optimization algorithm. Take note that because early stopping is implemented 32 epochs may not be reached especially if the accuracy plateaued and reached the threshold patience (number of epochs where accuracy is on plateau). Figure 14 offers a brief insight with regard to the processes inside a CNN.

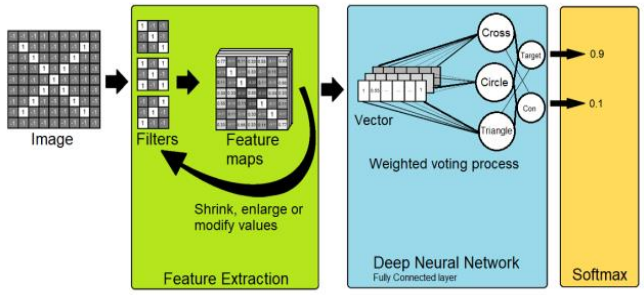

Figure 14. Convolutional Neural Network Workflow. The initial image is subjected to multiple filtering operations to detect different features. Advanced filters then analyze these feature maps, repeating the process for the selected convolutions. The resulting values are then inputted into a Deep Neural Network (DNN) to facilitate learning and prediction. To convert the DNN's output, which starts as random integers, into probabilities, the softmax function is applied [4].

4.0 RESULTS AND DISCUSSION

4.1 Accuracy of Model Training

Initial training already yielded 91% as shown in Figure 15 using the original images and by incorporating additional images through data augmentation techniques, the accuracy of the model training has experienced a significant improvement as seen in Figure 16. The data augmentation technique involves generating more samples and introducing diversity into the dataset, enabling the algorithm to develop a broader understanding and generalize better on previously unseen data.

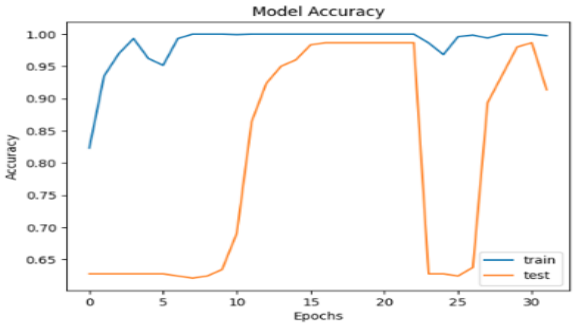

Figure 15. Initial Result. Validation set accuracy at 91% (32 epochs) without data augmentation. Slight overfitting was observed since validation and training accuracy has a slight difference.

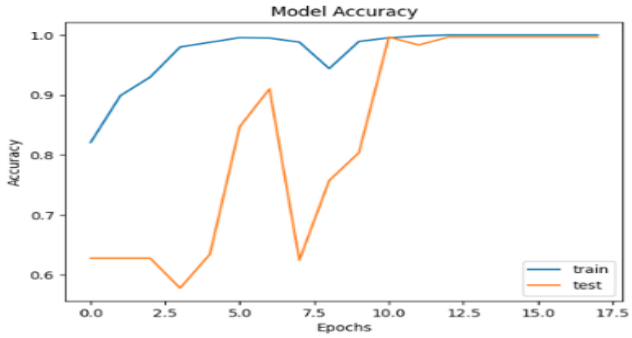

Figure 16. Final Result with Data Augmentation. Data augmentation was implemented which doubled the amount of training data. The validation and training accuracy started to converge on 7 epochs and plateaued until epoch 18. Validation set accuracy at 99.6% which is excellent.

4.2 Model Testing and Prediction Result

Upon the completion of model training, the team continued to gather an additional set of images from Mechanical Finish, containing both proper and improper wearing of Personal Protective Equipment (PPE). The model exhibited highly accurate predictions in all 45 images depicting improper wearing of PPE since all were classified correctly. However, out of the 112 images showing proper wearing of PPE, there was a single misclassification with one image incorrectly labeled as improper wearing of PPE. This is acceptable

because the image in question is also ambiguous in nature as shown in Figure 18. It is seen that the operator wearing the PPE properly was not correctly positioned as required by the bounding boxes within the screen. Overall, the model's performance is highly acceptable with 98.73% accuracy, 0% miss rate, and 0.8% false alarm.

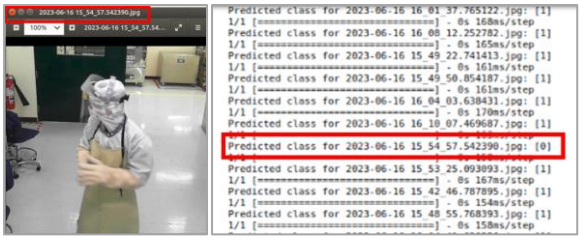

Figure 17. Mislabeled Image. The image was originally labeled as "Proper" but since the face tilted sideways, it did not detect the face shield and face mask properly classifying it as "Improper".

5.0 CONCLUSION

Based on the findings of the case study that employed the ResNet50 convolutional network for image classification of Personal Protective Equipment (PPE) adherence, the results show a remarkable accuracy of 99.6%, a miss-rate of 0%, and a false alarm rate of only 0.8%. These results strongly indicate that utilizing the ResNet50 architecture is a highly effective approach for this particular task. Moreover, there is potential for its application in various other image classification tasks, provided that there is a substantial volume of images for robust training and feature learning.

It is worth emphasizing the crucial role played by factors such as data quality, data balance, and the abundance of images in ensuring the development of a high-quality model. These factors significantly contribute to the overall success and reliability of the ResNet50 convolutional network in accurately classifying PPE adherence based on image analysis.

6.0 RECOMMENDATIONS

Investing in high-performance hardware for model training is highly recommended. While the NVIDIA Jetson Tegra Xavier offers a decent GPU, it may not provide sufficient storage capacity (only 32 GB) for housing a large volume of images required for deep learning. Therefore, it is advisable to ensure ample storage capacity to accommodate the data. Additionally, camera selection should also be considered. In a previous implementation, a low-resolution camera was used (Raspberry Pi) which yielded low-quality images. Image quality played a huge part in obtaining a good model for this study.

Furthermore, it is essential to not limit the training data to what is currently available. There should be a continuous effort to collect new data to regularly update the model. By incorporating new data, the model can be improved and will perform better on unseen data, enhancing its overall effectiveness.

Additionally, it is worth considering other image-processing techniques that were not employed in the experiment. There are numerous techniques available that can supplement or complement machine learning approaches in terms of feature extraction. Exploring these alternative techniques can potentially enhance the performance and capabilities of the model.

7.0 ACKNOWLEDGMENT

We would like to extend our heartfelt appreciation to Sir Peter Gonzales, Myra Enriquez, and Xavier Tacla from Mechanical Finish for their invaluable contribution to the data necessary for the successful completion of this study. We would also like to express our gratitude to the Operators and Supervisors of MF for their unwavering support and cooperation, which played a pivotal role in the data collection process. Lastly, we extend our deepest thanks to the management for their unwavering support of the project and their allocation of adequate budgetary resources, without which this Machine Learning endeavor would not have been possible.

8.0 REFERENCES

[1] Eri Matsuyama, A Deep Learning Interpretable Model for Novel Coronavirus Disease (COVID-19) Screening with Chest CT Images
[2] YOLO Algorithm for Object Detection Explained [+Examples] (v7labs.com)
[3] A. A. Protik, A. H. Rafi and S. Siddique, "Real-time Personal Protective Equipment (PPE) Detection Using YOLOv4 and TensorFlow," 2021 IEEE Region 10 Symposium (TENSYMP), Jeju, Korea, Republic of, 2021, pp. 1-6, doi: 10.1109/TENSYMP52854.2021.9550808.
[4] Jonathan Karlsson & Fredrik Strand, Visual Detection of Personal Protective Equipment & Safety Gear on Industry Workers
[5] PPE Compliance Detection using Artificial Intelligence in Learning Factories – ScienceDirect
[6] Deep learning for site safety: Real-time detection of personal protective equipment - ScienceDirect
[7] https://augmentor.readthedocs.io/en/stable/userguide/mainfeatures.html
[8] https://medium.com/analytics-vidhya/convolution-padding-stride-and-pooling-in-cnn-13dc1f3ada26

9.0 ABOUT THE AUTHORS

Michael C. Trinidad completed his Bachelor of Science degree in Electronics and Communications Engineering from Technological University of the Philippines – Taguig in 2005. He passed the ECE licensure exam in the same year. Currently, he serves as a staff manufacturing engineer at onsemi Carmona Philippines with a primary focus on applying Data Analytics and Machine Learning techniques in the field of semiconductor manufacturing.

Diego Jose L. Cabalza completed his Bachelor of Science degree in Electronics and Communications Engineering from the University of the Philippines – Diliman in 2022. Currently, he serves as a manufacturing engineer at onsemi Carmona, focusing on projects related to Data Analytics and Machine Learning.

Rommel M. Fajardo completed his degree in electronics engineering from First Asia Institute of Technology and Humanities in 2015. He went on to successfully pass the ECE licensure exam in October of the same year. Currently, he holds the position of an Automation Engineer at onsemi Carmona where his primary role entails working on projects related to Data Analytics and Machine Learning.

10.0 APPENDIX

APPENDIX A