A Robust Real-time Component for Personal Protective Equipment Detection in an Industrial Setting

Pedro Torres[1], André Davys[1], Thuener Silva[1], Luiz Schirmer[1], André Kuramoto[2], Bruno Itagyba[2], Cristiane Salgado[2], Sidney Comandulli[2], Patricia Ventura[2], Leonardo Fialho[2], Marinho Fischer[2], Marcos Kalinowski[1], Simone Barbosa[1] and Hélio Lopes[1]

[1] *Department of Informatics, PUC-Rio, Rio de Janeiro, Brazil*

[2] *PETROBRAS, Rio de Janeiro, Brazil*

Keywords: Artificial Intelligence, Industrial Application, Computer Vision, Real-time System.

Abstract: In large industries, such as construction, metallurgy, and oil, workers are continually exposed to various hazards in their workplace. Accordingly to the International Labor Organization (ILO), there are 340 million occupational accidents annually. Personal Protective Equipment (PPE) is used to ensure the essential protection of workers' health and safety. There is a great effort to ensure that these types of equipment are used properly. In such an environment, it is common to have closed-circuit television (CCTV) cameras to monitor workers, as those can be used to verify the PPE's proper usage. Some works address this problem using CCTV images; however, they frequently can not deal with multiples safe equipment usage detection and others even skip the verification phase, making only the detection. In this paper, we propose a novel cognitive safety analysis component for a monitoring system. This component acts to detect the proper usage of PPE's in real-time using data stream from regular CCTV cameras. We built the system component based on the top of state-of-art deep learning techniques for object detection. The methodology is robust with consistent and promising results for Mean Average Precision (80.19% mAP) and can act in real-time (80 FPS).

1 INTRODUCTION

Workers, especially in an industrial setting, are continually exposed to various hazards in their workplace. In this context, unfortunately, there are several fatal cases. The Brazilian Protection Statistical Yearbook[1] shows an average of six hundred thousand occupational accidents and 2600 deaths per year, registered between 2010 and 2017.

A company, such as an oil and gas refinery, could avoid injuries by monitoring its workers to prompt corrective measures when the personal protective equipment (PPE) is not used appropriately. Nevertheless, this activity is often performed by a human from a constant visual local inspection or closed-circuit television (CCTV). In this scenario, an industry could benefit from a system powered by Machine Learning and Computer Vision techniques to automate this task in order to prevent accidents, minimize costs, and injuries. Figure 1 illustrates a possible industrial system to monitor the use of PPE automatically and emit alarms when they are missing or not used appropriately. The system is fed with RGB images from a CCTV, each image initially passes through the detection and verification component. This component is the fundamental basis of the system and is responsible for producing evidence of deviations from inappropriate use of the equipment by workers. Finally, the ID association component can match those evidences with the worker's identity in the company database and issue an alert with the worker's identification and type of deviation.

Regarding the detection and verification component, there are two main approaches to address the problem of PPE detection with Computer Vision techniques. One approach uses one-stage classifiers (Bo et al., 2019), which handles the detection and verification phases throughout a single model. Another approach employs a multi-stage classifier (Li et al., 2017), which uses one or more models to handle each

[1] https://bc.pressmatrix.com/pt-BR/profiles/1227998e328d/editions/0e55e8eba33a3ed62b2e/pages/page/40

Torres, P., Davys, A., Silva, T., Schirmer, L., Kuramoto, A., Itagyba, B., Salgado, C., Comandulli, S., Ventura, P., Fialho, L., Fischer, M., Kalinowski, M., Barbosa, S. and Lopes, H.
A Robust Real-time Component for Personal Protective Equipment Detection in an Industrial Setting.
DOI: 10.5220/0010452606930700
In *Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021) - Volume 1*, pages 693-700
ISBN: 978-989-758-509-8

phase. Both approaches show good results but are limited to identify just one PPE (hardhat in the majority of cases). Suggesting that the multiple PPE detection is more challenging than detecting a single PPE (Zheng et al., 2019; Nath et al., 2020).

Although important in the scenario described in the Figure 1, we will not conduct an in-depth exploration of the challenges related to the ID Identification component, as they lay outside of the scope of this work. We shall focus on exploring the two main approaches for implementing a detection and verification component (highlighted as the blue box of the Figure 1) that is both robust and capable of act in real-time for monitoring systems in a industrial environments, especially oil and gas refineries.

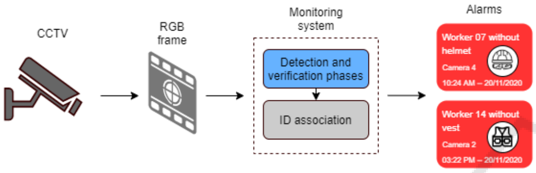

Figure 1: Example of an industrial monitoring system for PPE compliance.

Our main contributions are:

- Evaluating two approaches to solve the PPE detection problem. In our first approach, we built a one-stage classifier. While in our second approach, we build a multi-stage classifier.
- A dataset for PPE detection that addresses multiple types of equipment.
- Exploring how ensemble classifiers performs for the verification stage of a multi-stage implementation.

The paper is structured as follows: Section 2 presents a literature review of Deep Neural Networks and the use of object detection models to solve the PPE detection problem; Section 3 describes the construction and exploration process adopted for the proposed approaches and also includes details regarding dataset generation; Section 4 presents the performance comparison of models and approaches and Section 5, the conclusion.

2 LITERATURE REVIEW

The use of DNN (Deep Neural Networks) has achieved state-of-the-art in different Computer Vision tasks in recent years. Convolutional Neural Networks (CNN) have emerged as an important approach to perform a broad range of visual tasks (Krizhevsky et al., 2012; Ren et al., 2015). CNNs are composed of layers of filters that represent neighbourhood spatial connectivity patterns. Its use of convolutions, non-linear activation functions and downsampling results in a hierarchical understanding of those features. A crucial aspect of this interleaving of operations is that they usually fuse spatial and channel-wise information.

Recent advances of image classification focus on training feedforward convolutional neural networks using “very deep” structure (Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2016). The feedforward convolutional network mimics the bottom-up paths of the human cortex. Several approaches have been proposed to improve further the discriminative ability of deep convolutional neural network. VGG (Simonyan and Zisserman, 2014), Inception (Szegedy et al., 2015) and residual learning (He et al., 2016) are frameworks that are able to train very deep neural networks. VGGNets and Inception models investigated very deep architectures in detail. A complementary approach is ResNets, which applied skip connections also to improve the training of deep networks (He et al., 2016).

Considering PPE detection, to improve the results, some studies use different neural network architectures for dealing with this task. One of the first works in this direction was shown in (Fang et al., 2018), in which the authors use the Faster R-CNN neural network to detect workers’ non-hardhat-use in a construction scenario. According to results, they demonstrate that the use of Faster R-CNN can facilitate improved safety inspection and supervision in a real construction environment.

Bo et al. (2019) use the YOLO-v3 for hardhat detection. You only look once (YOLO) (Redmon et al., 2016) is a state-of-the-art, real-time object detection system that uses a single neural network to predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. The authors tested this detection model in images collected from an electric power construction scenario and are interesting in detect if a worker use hardhat or not.

The work presented above aims to construct a model for hardhat wearing detection. However, in some cases, we are interested in detect multiple PPEs, such as hardhat, gloves, and worker vests. In this sense, Zheng et al. (2019) and Nath et al. (2020) developed models to handle a higher amount of PPE. Nath et al. (2020) propose three different approaches to detect multiple PPEs in construction scenario. In the first approach, they developed a multi-stage model, where the YOLO-v3 model detects workers, hats, and vests and then, a simple machine learning classifier is applied to verify whether a worker is wearing hardhat and vest. In the second approach, they use an one-stage model, based on YOLO-v3, in which the model simultaneously detects individual workers and verifies the PPE usage. The third approach developed a multi-stage model, in which the YOLO-v3 detects only workers in a scenario and then cropped these detections to feed a CNN classifier that is responsible for verifying the PPE compliance.

Deep neural networks have been revealed to be quite useful for solving object identification tasks (Nath et al., 2020). Despite the meaningful results described in the literature, in the industrial domain, this approach has limitations considering the time of inference and precision. In this work, we propose robust approaches that can be used as a fundamental component of a monitoring system for PPE detection that uses YOLO-v4 model as a basis for implementing the proposed approaches. We also experiment with new adaptations, in terms of models and parameters for the multi-stage approach, where we create a solution that uses ensemble classifiers. As one will see in the next sections, our approaches present promising results (in terms of mAP) and still capable of act in real-time.

3 METHODOLOGY

This section presents the construction and exploration process for the two proposed approaches, including dataset generation, used to carry out model training and evaluation. In this work, we evaluate whether these approaches are ready to act in real-time and with robustness when implemented as a fundamental component of a monitoring system that seeks to detect which workers are making the appropriate use of PPEs.

For this study, we choose to focus on two types of PPE, hardhat and protective clothing. These types of equipment are often used to ensure the safety of workers in the oil and gas industry and civil construction. Our approaches are based on techniques for detecting and classifying objects in images. In this way, we have four possible classes to perform the classification, which are: worker with no PPE (W), worker wearing a hardhat (WH), worker wearing protective clothing (vest) (WV), and worker wearing a hardhat and protective clothing (WHV). Notice that the approaches implemented for this work can be extended to any types of PPE, but not without an increase in complexity due to the number of different combinations of equipment.

The detection and verification component is based on models that empower deep learning methods capable of being executed in real-time. In Nath et al. (2020) work, which also addresses the problem of detecting multiple PPEs, the authors adopt as the definition for real-time system one that can process at least five frames per second (FPS). When this is not possible, values ≥ 1 FPS are considered ”near real-time”. These definitions emerge from previous work (Redmon et al., 2016) which also raise this concern. In this work, we follow the same definition since the context is quite similar. We expect that our implementation can act in real-time with a prompt response when a worker’s life is exposed to risk.

In particular, both approaches employ YOLO-v4 architecture and carry some of the steps in a similar manner, such as preparing data and the models’ training. In the approach I, for each image, we annotate one bounding box for each worker, where the class of this bounding box informs which PPE the worker is wearing. This way, we create a single model based on YOLO-v4 architecture for detection and verification phases. Approach II uses one model for the detection phase and another model for the verification phase. First, we detect workers bounding boxes from images based using a model based on the YOLO-v4. Next, a convolutional neural network is applied for PPE compliance verification. The details for both approaches are clarified in Subsection 3.3 and Subsection 3.4.

3.1 Dataset Generation

Supervised machine learning applications requires a large annotated dataset to provide the learning model a way to create and recognize patterns through the data. As we did not have access to a large image dataset available for PPE detection, it was necessary to create the annotated dataset to feed the model.

The dataset used in this work is composed of images from the following sources: Crowd-sourced (as used by Nath et al. (2020)), GDUT-HWD (Wu et al., 2019), Web-scrapped, and images captured by the authors. Of the datasets that already had annotation (Crowd-sourced and GDUT-HWD), the only one that deals with the identification of workers using multiple PPE’s is Crowd-sourced, where the annotation of classes is the same as that used in this work. For GDUT-HWD, only the images were used, since the dataset annotations are only for the individual identification of safety helmets. To compose the Web-scrapped source, images were obtained from public databases using search engines that perform searches by keywords, e.g., ”workers in refinery”, ”workers in platforms". After the complete collection of these images, there was a visual inspection to remove images that were out of context or that had low quality. The images captured by the authors were taken in a controlled environment that sought to reproduce the conditions of an industrial environment. The Figure 2 shows the number of instances for each data source.

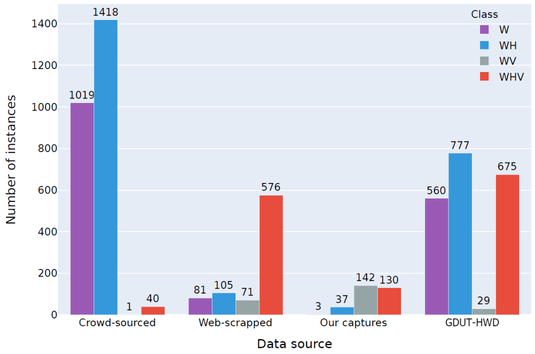

Figure 2: Distribution of instances per class for each data source.

In the annotation procedure to mark bounding boxes of objects in images for training the models, we aimed to minimize the annotation bias by taking the following procedure. For each data source, except for the Crowd-sourced (already annotated), the dataset was split into equal-sized batches. Each batch was initially annotated by a person using the YOLO mark tool[2], an open source library for image annotation. At the end of the annotation of a batch, an annotator review the annotations of another one and separate those that presented divergences (concerning the annotated class, or bounding box region). Each divergence was discussed by annotators until they reached a consensus, defining the ground truth for each image.

3.2 Data Preprocessing

In the dataset preparation, we randomly divided the images into three subsets: training (70%), validation (20%) and testing (10%). This division of the dataset was carefully considered to ensure similar distributions of the classes for each set. To ensure some similarity among the sets, we split the data sources assuring the proportion was the same, i.e., training, testing, and validation sets have the same percentage of each data source, making the sets more homogeneous.

Note that the dataset was split based on the number of images. Thus, to verify whether the instances proportion for each subset is similar, we analyzed the distribution of number of instances, as shown in Figure 3. According to the figure, the same proportion between classes is maintained for all subsets. Moreover, it is possible to observe that the dataset has many examples for WH class and a few examples for WV class, which can hinder the learning model from generalizing these classes.

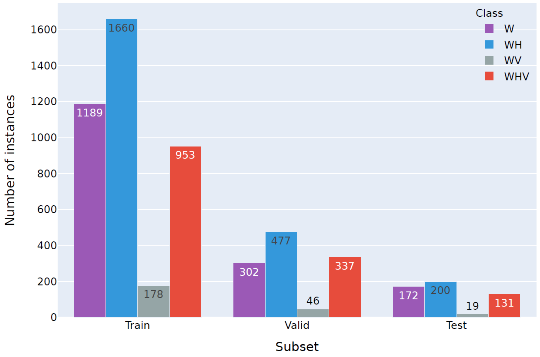

Figure 3: Number of instances per class for each subset.

To improve the ability to detect objects accurately, YOLO uses nine anchor boxes, which must be defined for use during the training and inference phases. With the anchor boxes defined, the model can specialize in objects of certain sizes and objects with a particular aspect ratio (height × width).

In practice, during the training phase, each cell of the feature maps of the network's output layers has an associated anchor box. Thus, the model learns how to shift and scale an anchor box so that the coordinates of the prediction bounding box fits the object of interest. To define the anchor boxes, the K-Means clustering algorithm was used, with $k = 9$. The algorithm was executed with the training set bounding boxes as input.

3.3 Approach I

The first approach uses a YOLO-v4 model, which we will call YOLO-v4-AP1, to perform the identification and verification steps in one-stage. During the annotation phase, the regions where the workers are located and labeled with one of the classes (W, WV, WH, WHV). Once the model is trained, the inference provides a worker's location and the class detected determine which PPE he/she is using.

Figure 4 demonstrates how the approach I is carried out. One of the main advantages of approach I is able to take advantage of the ability that YOLO models make predictions (with the locations of objects and their respective classes) using a single network, which makes a simple and yet effective method.

Following the inference phase, the YOLO-v4-AP1

[2] github.com/AlexeyAB/Yolo_mark

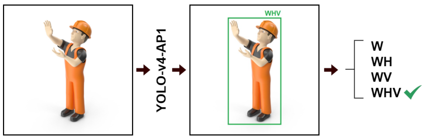

Figure 4: Approach I.

model performs predictions, only detections with a confidence score above 50% are considered as final predictions. It is worth mentioning that the model can make duplicate detections and present different classes for the same worker, but following the problem definition, a worker can only belong to one class. To avoid duplicate detections, the model uses non-maximum suppression (NMS). Usually, NMS adds 2-3% in mAP (Redmon et al., 2016).

3.4 Approach II

In this approach, we use a multi-stage method to perform the identification and verification of the proper usage of PPE's by workers. Initially, a YOLO-v4 model is used (which we will call YOLO-v4-AP2) only to locate the worker (detection stage). Thus, there is only the worker class (W), to carry out the equipment verification stage we use a CNN that receives cropped images for each region in which it is detected a given worker. This CNN classifies each image according to the four possible classes (W, WH, WV, WHV). The Figure 5 illustrate how this multi-stage approach is performed.

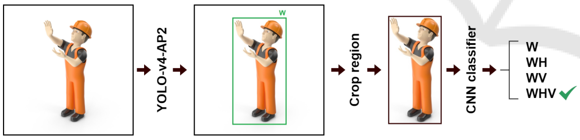

Figure 5: Approach II.

For this approach, we adopted the following classifiers: VGG-16, ResNet50, ResNet101 and Inception. For each classifier, a dropout layer with probability $p = 0.3$, a fully connected layer with 256 nodes and regularization $L1$ and $L2$ with $rw = 10^{-3}$ was added after the convolutional blocks (base model). The layer is followed by the ReLU (Rectified Linear Unit) activation function. Additionally, the input dimensions of the networks were adapted to receive images in the format 150 × 150. A modification was made to the output layer dimensions, where the network must have four nodes, one node for each class. Finally, this layer is followed by the activation function SoftMax.

In the training step, we use a transfer learning strategy, using the weights pre-trained with the ImageNet dataset for the convolutional layers. We freeze these layers and retrain the models with 60 epochs using the Adam optimizer with a learning rate $\eta = 10^{-5}$.

As the classifiers receive only the image cropped with the worker's location, the dataset used was built from each instance of the annotation that belongs to the training set's grounding truth. The images are scaled to the size 150 × 150 to match the networks' input size. For data augmentation, the following transformations were applied to images: random zoom and shear range of up to 20%, a horizontal flip was also applied to up to half of the images.

We also improve approach II by using an ensemble of classifiers. Thus, when performing the verification stage, we do not use the learning of a single classifier, but the learning from a set of classifiers (VGG16, Inception, ResNet50 and ResNet101). We believe that by combining the predictions of multiple classifiers, we can reduce the variance and make the classification less dependent. Further, the classification bias can also be reduced since the classifiers together can make the class distinction criteria more expressive. The ensemble prediction combinations were given from the voting (majority) method. That is, each classifier assigns his vote to the class with the highest probability. The final prediction is given to the class that received the highest number of votes.

3.5 Model Training

Implemented models from approaches I and II, all layers, except the last three output layers, have their weights captured from training the YOLO-v4 model in the COCO dataset. After this first train, a transfer learning method is applied to take advantage of the model's ability to detect up to 80 classes (person, car, motorcycle, etc.) from the COCO dataset and apply the knowledge in our domain. Since this model has already been trained from a more significant number of images, it can generalize its learning ability to distinguish resources for our task, in which there is a much smaller number of images. It may seem that two tasks have no evident intersection, in addition to that a worker is also a person, but in problems that address a classification or detection task, some low-level characteristics, such as, edges, shapes and contours, can be shared between tasks, thus allowing the spread of knowledge between them.

The models of approaches I and II were re-trained following the same definitions of hyperparameters. The re-training was allowed 30 epochs with the learning rate $\eta = 0.0013$, using the Adam optimizer. To accelerate the learning convergence process and to also mitigate the model's overfitting, the value of

$momentum = 0.949$ and $decay = 0.0005$ was adopted.

Additionally, to improve model performance in the real scenario, we use the data augmentation technique, which is performed in real-time during the model's re-training phase. To create more diversification in training images the Mosaic technique was used, which was introduced together with YOLO-v4. This method mixes four training images. Hence, four different contexts are mixed and could allow the detection of objects that are out of their normal context. In addition, we also make random changes to the HSV color-space of the images, where the saturation and exposure values are modified by a factor of up to 1.5.

4 EXPERIMENTAL RESULTS

4.1 Perfomance of the YOLO-V4 Models

The performance of the YOLO models are evaluated using the mAP (Mean Average Precision) metric, which is used in several object detection models (e.g., Faster R-CNN, R-CNN, SSD). One of the advantages of the mAP is that we can quantify how well an object detection model is performing in a data set using a single numerical representation. Figure 6 presents the performance results of the detection models used in the approach I (YOLO-v4-AP1) and approach II (YOLO-v4-AP2) for the test set. The highest value mAP (88.18%) is obtained by the approach II model, which was expected since there is only one class to be detected. A factor that may have contributed considerably to the performance of the model was the use of transfer learning. In this case, we have a notable similarity between the COCO dataset person class and the worker class (W). For approach II, we have a mAP of 80.19%, which is a good result, since this approach makes use of a single model to perform the stages of detection and verification. That is, the result of this approach will not change, while the results of approach II will change when we add the verification step, being directly affected according to the performance of the classifiers.

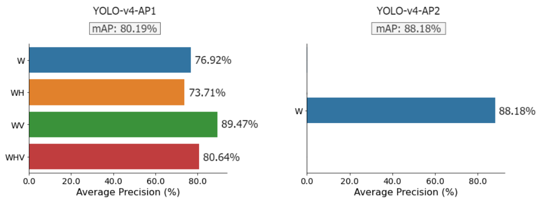

Figure 6: Perfomance of the YOLO-v4 models for each approach.

4.2 Performance of Classifiers

To evaluate the classifiers proposed for the verification step in approach II, instances of the test set were used. The accuracy of the VGG-16, Inception, ResNet50 and ResNet101 models for classifying images in classes W, WH, WV and WHV are 82.85%, 78.35%, 84.02% and 82.75%, respectively. Figure 7 shows the confusion matrix for each model. Note that the WHV class has the lowest accuracy in all classifiers. One reason is confusion with the WH class, which has an average value of 20.05% for false negatives. That is, once a worker is being detected with a helmet, it is difficult to distinguishing between wearing or not wearing safety clothing. The individual accuracy for the rest of the classes (W, WH, WV) presents values higher than 80%, with the exception of class W when evaluated in Inception, which had a value of 6.6% below the average (82.83%).

VGG16

	W	WH	WV	WHV
W	83.1%	14.5%	0.6%	1.7%
WH	4%	85.5%	0%	10.5%
WV	0%	0%	84.2%	15.8%
WHV	2.3%	19.1%	0%	78.6%

INCEPTION

	W	WH	WV	WHV
W	76.2%	18.6%	0.6%	0.47%
WH	5%	83.5%	0%	11.5%
WV	0%	5.3%	84.2%	10.5%
WHV	6.1%	23.7%	0.8%	69.5%

RESNET50

	W	WH	WV	WHV
W	86%	11.6%	1.2%	1.2%
WH	6.5%	83.5%	0%	10%
WV	0%	5.3%	89.5%	5.3%
WHV	3.1%	16.8%	3.1%	77.1%

RESNET101

	W	WH	WV	WHV
W	86%	11.6%	1.2%	1.2%
WH	7%	83%	0%	10%
WV	0%	5.3%	89.5%	5.3%
WHV	5.3%	20.6%	1.5%	72.5%

Figure 7: Confusion matrices for VGG-16, Inception, ResNet50 and ResNet101 classifiers.

4.3 Performance of Approaches

In this subsection, we will evaluate the final performance of the approaches I and II. Note that this result depends directly on the performance of the implemented models. Figure 8 shows that approach I obtained the best mAP (80.19%), even when compared with the different model combinations used in approach II. Although the YOLO-v4-AP2 model presented a MAP of 88.18% in the detection step, the errors of the classifiers for the verification step end up reducing the mAP, since some of the instances will be classified incorrectly, generating false positive. The best result for approach II presents 72.87% of mAP given from the use of the YOLO-v4-AP2 model plus the ensemble with the classifiers VGG16, ResNet50, and ResNet101, which are the classifiers that reported the greatest accuracy. In the scenario where only one classifier was used in the verification step, the best result is obtained using ResNet101 (70.42% of mAP).

We also compared the processing time spent by each approach. We run all models on the same machine, which has the following configurations: Intel

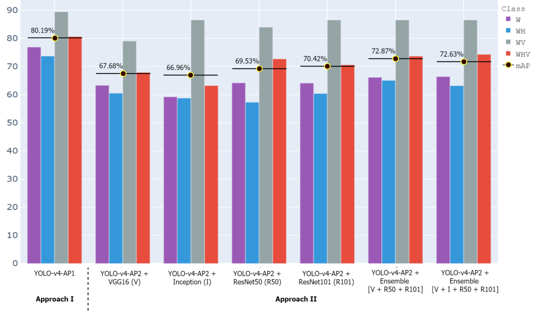

Figure 8: Performance comparison of approaches implementation.

Core i9-7900X, 128 GB RAM and TITAN RTX GPU with 24 GB memory. Figure 9 presents the average processing time for a test image in each approach. Approach I has the best processing time (12.55ms). For approach II the time of the YOLO-v4-AP2 model is slightly less (12.38ms) than the model implemented in approach I, but when adding the classifiers time, that time increases significantly, making approach II the slowest, with times in the interval of 53.4ms to 224.67ms. As the proposed models run at a rate of at least 5 FPS (frames per second), following the definition adopted in this work (same as Redmon et al. (2016) and Nath et al. (2020)), we can say that these approaches are capable to process videos in real-time applications and can be implemented as the base component of a monitoring system for PPE compliance.

4.4 Benchmark of Results

Since the dataset or models used in the work of Nath et al. (2020) were not made available by the authors, we have to find other alternatives. To perform a fair comparison of the methodology adopted in this work and verify if the approaches were effective, we establish a comparison with the results obtained from the YOLO-v3 model. The model was trained and tested in a similar way to the YOLO-v4-AP2 model. Table 1 shows the comparison between models in terms of mAP and FPS. Concerning mAP, we have an increase of 21.52% and 13.96% when comparing the YOLO-v4-AP1 and YOLO-v4-AP2 (with ensemble) models with baseline (YOLO-v3), respectively. On the other hand, there is an increase in FPS. However, since the models of approach I and II are already able to run in real-time, this gain is not very significant.

Table 1: Comparison of mAP and FPS with a baseline model.

Criteria	Baseline model (YOLO-v3)	Approach one-stage (I) (YOLO-v4-AP1)	Approach multi-stage (II) (Ensemble: V + R50 + R101)
mAP	58.67%	80.19%	72.87%
FPS	104	80	7

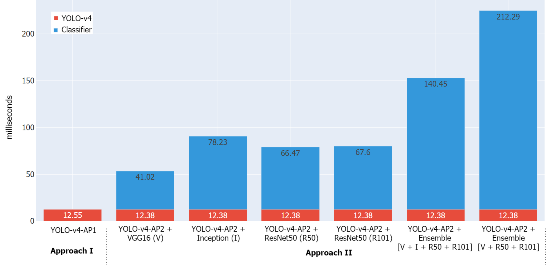

Figure 9: Processing time comparison for approaches implementation.

5 CONCLUSIONS

This work explores the implementation of two different approaches based on deep-learning to perform the task of detecting the usage of PPEs by workers. We aimed to develop and evaluate approaches that are robust and capable of acting in real-time, so that they can be implemented as a fundamental component (detection and verification) of a monitoring system.

Although in this paper, we focus on two types of PPEs (hardhat and protective clothing), our results show that we can employ our methodology to any number of equipment (e.g., goggles, gloves, and masks) changing the networks' output layers. The effect of this, is an increase in complexity due to the number of different combinations of equipment.

To carry out the models' training, we built a dataset from four different sources to supply our models with a more significant number of images from different devices, angles, lighting, and environments. To enable our models to be able to generalize better.

For Approach I, we built a single model based on YOLO-v4 implemented in one-stage; that is, the same model is responsible for identifying and verifying the use of PPE. Hence, when receiving an image as input, the model classifies each founded region that displays a worker, with one of the following classes: W, WV, WH, and WHV. In contrast, Approach II is multi-stage, with at least two distinct models for the identification and verification stages. Initially, a YOLO-v4 based model detects the regions in which workers are located. Then, a convolutional neural network receives the clipping from each region and performs the verification step, which consists of classifying the image into W, WV, WH, and WHV classes.

Both proposed approaches outperforms the baseline results relating to mAP. Our results show Approach I presenting the best mAP for detecting PPEs (80.19%). Although YOLO-v4-AP2 mAP display superior results (88.18%), the classifiers' errors in the verification step decrease the final mAP. This effect is evident even for our best implementation,

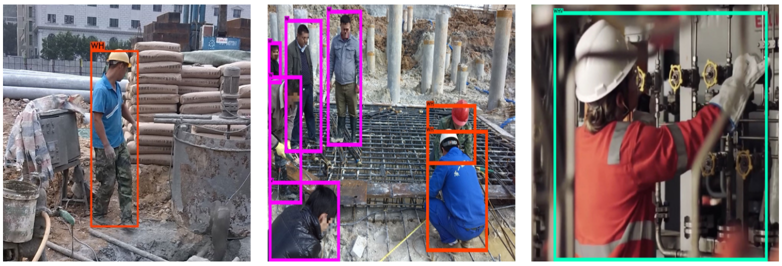

Figure 10: Example of detections obtained from YOLO-v4-AP1 model. The first two images are from the Crowd-sourced dataset. The third image is from the Web-scrapped dataset that is in an industrial setting.

which employs an ensemble of classifiers VGG16, ResNet50, and ResNet101, producing a final mAP of 72.87%. These results may indicate that superior results may be obtained from the individual improvement of the classifiers or methods proposed in this work. The ensemble method achieved an increase of up to 2.45% compared to the best single classifier (ResNet50) mAP (70.42%) of Approach II. Regarding the processing time, Approach I proved to be more effective because of its one-stage implementation, which avoids bottlenecks between the processing phases. Although slower, our results demonstrate that Approach II still feasible to use it in real-time, even with the use of an ensemble of classifiers.

From the implementation carried out for approach I, it is possible to build a monitoring system that has a robust detection and verification component. Since the approach proved to be more efficient, not only in terms of mAP (80.19%) but also in processing time, reaching up to 11x faster (80 FPS) when compared to approach II. Considering that, we believe that the one-stage approach has a high potential for the construction of an effective monitoring system that can contribute to the safety of workers, minimizing the number of accidents and live losses.

Regarding ID association component mentioned in Figure 1, we believe that tracking algorithms such as DeepSORT (Wojke et al., 2017) may present goods results when employed along with the component explored in this work. This happens due to those algorithms working well with robust detection models to track real-time custom objects and assign unique identities for each object.

REFERENCES

Bo, Y., Huan, Q., Huan, X., Rong, Z., Hongbin, L., Kebin, M., Weizhong, Z., and Lei, Z. (2019). Helmet detection under the power construction scene based on image analysis. In *2019 IEEE 7th International Conf. on Computer Science and Network Technology (ICCSNT)*, pages 67–71. IEEE.

Fang, Q., Li, H., Luo, X., Ding, L., Luo, H., Rose, T. M., and An, W. (2018). Detecting non-hardhat-use by a deep learning method from far-field surveillance videos. *Automation in Construction*, 85:1–9.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Li, J., Liu, H., Wang, T., Jiang, M., Wang, S., Li, K., and Zhao, X. (2017). Safety helmet wearing detection based on image processing and machine learning. In *2017 9th International Conf. on Advanced Computational Intelligence (ICACI)*, pages 201–205. IEEE.

Nath, N. D., Behzadan, A. H., and Paal, S. G. (2020). Deep learning for site safety: Real-time detection of personal protective equipment. *Automation in Construction*, 112:103085.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conf. has on computer vision and pattern recognition*, pages 779–788.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Wojke, N., Bewley, A., and Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE.

Wu, J., Cai, N., Chen, W., Wang, H., and Wang, G. (2019). Automatic detection of hardhats worn by construction personnel: A deep learning approach and benchmark dataset. *Automation in Construction*, 106:102894.

Zheng, X., Yao, J., and Xu, X. (2019). Violation monitoring system for power construction site. In *IOP Conf. Series: Earth and Environmental Science*, volume 234, page 012062. IOP Publishing.