Visual Inspection of Collective Protection Equipment Conditions with Mobile Deep Learning Models

Bruno Georgevich Ferreira, Bruno Gabriel Cavalcante Lima and Tiago Figueiredo Vieira
Institute of Computing, Federal University of Alagoas, Maceió, Alagoas, Brazil

Keywords: Deep Learning, Object Detection, Visual Inspection, Collective Protection Equipment.

Abstract: Even though Deep Learning models are presenting increasing popularity in a variety of scenarios, there are many demands to which they can be specifically tuned to. We present a real-time, embedded system capable of performing the visual inspection of Collective Protection Equipment conditions such as fire extinguishers (presence of rust or disconnected hose), emergency lamp (disconnected energy cable) and horizontal and vertical signalization, among others. This demand was raised by a glass-manufacturing company which provides devices for optical-fiber solutions. To tackle this specific necessity, we collected and annotated a database with hundreds of in-factory images and assessed three different Deep Learning models aiming at evaluating the trade-off between performance and processing time. A real-world application was developed with potential to reduce time and costs of periodic inspections of the company's security installations.

1 INTRODUCTION

Deep Learning (DL) has been presenting excellent performances for the past decade in subjective tasks due to its capability of adapting to large amounts of data. This is particularly true in Computer Vision, given the high dimension and variability of images and videos. As a consequence, different Deep Learning models have been applied to a wide range of supervised learning tasks and its popularity is increasing considerably (Dargan et al., 2019). Nevertheless, models can often be better tailored to tackle highly specific demands (Aggarwal, 2018) from several kinds of industries if an adequate partnership is articulated between research lab and market.

In this context, we aim at solving a typical task many companies rely on; the periodic visual inspection of Collective Protection Equipment (CPE). Under a partnership firmed with a multinational company focused on manufacturing devices for installation of optical fiber systems, we tuned DL models to inspect various specific conditions of fire extinguishers, emergency lamps, horizontal and vertical signalization, among others. An overview of the application can be seen in the Figure 1. The solution can be applied remotely on the company's surveillance system or embedded on a tablet attached to a mobile robot responsible for navigating the factory. It has the potential of reducing costs and time associated with the inspection of security systems.

More specifically, we present the following contributions:

1. We collect and annotate a database containing characteristics specifically aimed at tackling the company's demand. To the extent of our knowledge, no database containing such features has been presented so far.
2. We assess the performance of three DL models and evaluate the trade-off between precision and processing time. The system was embedded on a tablet that can potentially be attached to a mobile robot.

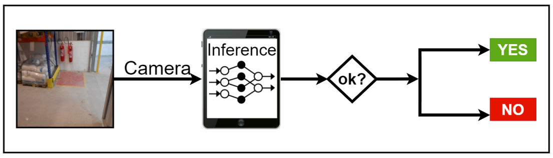


Figure 1: Overview of the application.

2 RELATED WORK

According to (Hocenski et al., 2016), using Computer Vision in pottery can result in promising and satisfactory results. Even though no deep learning algorithms were used, they faced the necessity of automated as-

Ferreira, B., Lima, B. and Vieira, T.
Visual Inspection of Collective Protection Equipment Conditions with Mobile Deep Learning Models.
DOI: 10.5220/0009834600760083
In *Proceedings of the 1st International Conference on Deep Learning Theory and Applications (DeLTA 2020)*, pages 76-83
ISBN: 978-989-758-441-1

sessment techniques for production analysis. Authors used well traditional in image processing to create three routines that detected problems in ceramic tiles such as those related to surface, edge and corner. The system worked in real time and was tested in a real factory for validation with satisfactory results.

As stated in (Veeraraghavan et al., 2017), Battery Management System (BMS) is a critical component in electrical vehicles. Accurate detection of the actual status of the battery is relevant, since this evaluation has impact over the control of many stages of the car functioning. Current models of battery estimation are complex and unable to provide result information in real time. Thus, authors applied deep learning techniques to develop an less complex estimator which would allow processing in real time. Complex battery model functionalities were simulated with using neural networks and the models developed presented high accuracy and real-time processing capabilities.

According to (Li et al., 2018), with the advance of Internet of Things (IoT) have been largely incorporated into industry facilities. Allowing better monitoring of their processes, sensors are now easier to install on machinery and to be connected to local fog via wireless networks. From sensors readings, often with high sampling frequency, a big volume of data is generated. From these data, authors proposed a classifier based on neural networks capable of detecting faulty products. However, in order to be suitable for industry, the classifier should be capable of running inferences about all data in real time which could be tackled using fog network.

Authors in (Rao and Frtunikj, 2018) elucidate difficulties that deep learning will find in automobile industry, bearing in mind the enormous effort that has been applied in the production of autonomous vehicles. One of the hardest difficulties is related to the safety, since autonomous cars should not take decisions harmful to passengers or people nearby.

For (Zhang et al., 2018), proposed the utilization of a YOLOv2 for automated detection of oil facilities to improve safety in extraction and production performance. Results were compared to traditional techniques with a combination of Haar features combined AdaBoost classification. YOLOv2 presented higher efficiency and accuracy.

As stated in (Choi et al., 2019), shipping industry is one of the most dangerous and there are many safety policies and techniques for decreasing the number of accidents. Besides the risk of not wearing safety equipment appropriately, there exists the risk of accidents due to unpredictable environment variables. In this manner, authors proposed a model capable of estimating the current risk of an environment, aiming to evaluate which safety measures are more adequate to make the workplace safer. A deep learning model was trained and allowed the identification of dangerous zones, measuring risk automatically.

In (Chou et al., 2019), a detection scheme for faulty coffee grains was proposed, together with a Generative-Adversarial Network (GAN) which both augmented the database and labeled new data. This approach improved the generalization capability of the model decreasing significantly the cost of creating the database, making it easier to train different DL models for the task.

3 METODOLOGY

The steps that were followed in order to obtain our results were;

1. building a database with labeled images for each class;
2. designing deep network models considering architectures more suited to the problem at hand;
3. testing the trained models.

3.1 Collecting the Database

The first stage of building the database was collecting the images. For that, many photos of fire extinguishers were taken in diverse environments, such as, the factory itself, the university facilities, many buildings, etc. One challenge was the difficulty in finding fire extinguishers really rusty, or with noticeable defects. This was due to safety laws that obligate near-to-expire fire extinguishers to be replaced. Thus, images were also collected from the internet, with some examples of faulty extinguishers. Another approach was taking pictures of different rusty object while annotating only the rust. This strategy was aimed at teaching the model characteristics of rust images instead of teaching only what rusty extinguishers look like.

After collecting fire extinguisher images and emergency lamps, the annotation process begun. The hose, the signaling plate, the rusty marks, the extinguisher body and the floor signaling were annotated in extinguisher pictures. On the emergency lamp photos, the male socket plug, the female socket plug, the status led and the body of the lamp were annotated. These characteristics were chosen by considering the demand presented by the company, which stated that those are most common faults occurring on their facilities.

With respect to the extinguisher, only the hoses in good conditions were annotated, considering that the faulty hoses (including the ones placed inappropriately) were very different and, hence, should not be detected by the model. Another reason for not annotating the situations where the hose was found in a bad position was due to a proximity that the classes would present in feature space, resulting in increased difficulty for the classification. From the emergency lamps, the power plug and the female socket on the wall were annotated. If one of them was detected, we assumed that the emergency lamp was not plugged into the electrical power-line and, hence, was not being used.

Since the database was built from scratch, it did not present many images per class, which led us to apply some augmentation techniques to reduce overfitting. The first pre-processing technique was resizing all images to 300×300 pixels and the conversion from PNG to JPEG. For data augmentation, the following techniques were used;

- vertical and horizontal mirroring;
- 90 degrees rotation;
- bright adjustment;
- resizing
- cropping

All augmentation techniques were applied randomly during training. In the case of resizing and cropping, the bounding boxes were taken into consideration. When there were multiple objects of interest in the same image, regions containing one or more objects were extracted to generate a new sub-image sample. This way, one image could generate sub-images with combinations of its objects. At least one object was visible for each sub-image.

3.2 Topologies

Three neural network topologies with very distinct characteristics were chosen aiming to assess the pros and cons of each one, in different circumstances. The first was the MobileNet V2 SSDLite, proposed by (Sandler et al., 2018). The second was FPN Resnet-50 SSD, presented by (Lin et al., 2017). The third was the Inception Resnet V2 Faster R-CNN with Atrous Convolution, adapted from (Szegedy et al., 2017). All architectures were trained using the same database, for unbiased comparison.

3.2.1 MobileNet V2 SSDLite

The MobileNet V2 SSDLite (MV2) was chosen as the first topology to be tested by the fact that it was designed to run in mobile devices, which makes it faster than other topologies. According to (Sandler et al., 2018), the most important contribution presented in MV2 was the new layers known as Inverted Residual with Linear Bottleneck. This new layer presents an input with reduced dimension, that is first expanded to an increased dimension and filtered with depth-wise separable convolutions. Next, filtered features have their dimension reduced through linear convolutions. The author also proposes a SSDLite that is a variation of the SSD, proposed by (Liu et al., 2016), with the convolutional layers being replaced by depth wise separable convolutions.

Despite the new layers Inverted Residual with Linear Bottleneck being the most important contribution of MV2, they still inherit some very important characteristics from its predecessor: the MobileNet V1. The main inherited features are the depth-wise separable convolutions, which decrease the necessary number of mathematical operations in one inference, making the topology faster to train and test. Depth-wise separable convolutions consist of replacing conventional convolution by a factored version with two separated layers. The first layer is the depth-wise convolution, which executes a low-cost convolution applying only one filter per input layer. The second layer is a 1×1 convolution, which is called point-wise convolution, used to compute new features from the linear combination of input layers. This is how the depth wise separable convolutions is done. First, it applies only one convolutional filter per each input layer. Then, it summarizes the features generated on previous layer with a 1×1 linear convolution.

3.2.2 FPN Resnet-50 SSD

The FPN Resnet-50 SSD (FPN50) has the following characteristics: presents feature pyramid network (FPN) as a generic extractor of features; has 50 residual layers; uses SSD as a multi-box detector.

The FPN is important by the fact that it aggregates invariance to scale for the model. With respect to the Resnet-50, its incorporation on the chosen model was relevant since residual layers allow for more deep networks while preventing over-fitting. This is possible because residual layers only apply convolutions when strictly necessary. If not necessary, the layer will reproduce the input on the output.

The FPN Resnet-50, proposed by (Lin et al., 2017) was modified by replacing the Faster-RCNN by the SSD. The FPN50 was chosen due to its capability of detecting more complex features.

3.2.3 Inception Resnet V2 Faster R-CNN with Atrous Convolution

Choosing the Inception Resnet v2 Faster R-CNN with Atrous Convolution (IRV2) was due to the fact that it presents, currently, one of the best results in object detection according to (Huang et al., 2017). This topology has a slower inference time than the previously cited ones, since it is large, but also comes with larger learning capabilities.

The IRV2 is a combination of the Inception Resnet v2, proposed by (Szegedy et al., 2017), with the Atrous Convolution, proposed by (Chen et al., 2017). A model trained with this topology is not able to compete with MV2 and FPN50, since the region proposal architecture from it is the Faster-RCNN. Faster-RCNN presents a bigger inference time when compared to SSD, as showed by (Huang et al., 2017).

Thus, the main objective of using this topology was to evaluate the performance of a model well known by its high generalization capabilities and high quality of bounding box prediction and classification. This would bring a best case scenario in order to compare previous models with it.

3.3 Training and Tests

For training and testing, it was defined which techniques of database augmentation would be used, taking into consideration the low number of images per class. Variables such as split ratio between training and test sets, as well as evaluation metrics for the model performance are chosen here. Training time was also recorded for each model.

3.3.1 Data Augmentation

Chosen techniques for the augmentation were: vertical and horizontal mirroring; 90 degrees rotation; bright adjustment; resizing; and cropping.

Mirroring and rotations were made aiming at generalizing shapes of objects that were present on the training base. However, using this kind of manipulation excessively may produce the inverse effect, making the model filters account more for color and texture.

Resizing and cropping had as objective the aggregation invariance to scale for the models that do not present this characteristic inherently, such as the MV2.

With respect to the bright manipulation on the image, this technique was aimed at making the model less sensible to color, accounting more for contour patterns and shapes.

3.3.2 Hardware and Software Infrastructure

A video-graphics card RTX 2080 Ti, with 11 GB of RAM GDDR6, was used along with Object Detection API from TensorFlow V1. The choice of a GPU of family 20XX from Nvidia for training the models was due to the presence of special cores in it, called Tensorcores, which decrease training time significantly.

Tensorflow API for object detection provides all topologies that were discussed previously, among others. It allows for rapid prototyping, including easy adjustments in the parameters. This API also provides pre-trained models with well known data-sets, allowing for techniques such as transfer learning. It is extremely useful for limited custom databases.

In the adopted training process, pre-trained models were used for the three topologies on the MSCOCO dataset, developed by (Lin et al., 2014). We used 75% and 25% of the database for training and testing ratio, respectively. Tests were carried out over an Android smartphone, with a Snapdragon 845 processor.

3.3.3 Performance Metrics

In order to evaluate models at training and testing time, the following metrics were defined: loss in training set; loss in test set; Average Recall (AR) and mean Average Precision (mAP) in test set; AR Across Scales and mAP Across Scales in test set; frame rate per second (FPS) on a mobile device and on RTX 2080 Ti.

Analysis of loss, both on training and test set, focused on evaluating if the models were generalizing well or if they presented some over-fitting characteristics. The AR metric calculation has three main configurations: (1) using one detection per image (AR@1); (2) using ten detections per image (AR@10); and (3) using 100 detections per image (AR@100). AR results using more detections tend to be better.

Similar to the AR, mAP calculations have three main variants: (1) the mean of the Average Precision (AP) over the limits of Intersection over Union (IoU), with values between 50% and 95% and step of 5%; (2) the mean AP with IoU limits set to 50% (mAP@0.5IoU); (3) the mean AP with IoU limits set to 75% (mAP@0.75IoU).

With respect to Across Scales metrics, they are calculated for three sets of objects: (1) small size objects, with area less than 32^2 pixels; (2) medium size objects, with area values between 32^2 e 96^2 pixels; and (3) large size objects, with dimensions larger than 96^2 pixels. Thus, the AR Across Scales is calculated for images with 100 detections for small objects (AR@100 small), medium objects (AR@100 medium) and large objects (AR@100 large).

By the other hand, mAP Across Scales is calculated using mAP's first configuration for small images (mAP small), medium images (mAP medium) and large images (mAP large). The analysis over the FPS indicator, which consists of the number of images per second that the model able to analyze, is also used for speed and efficiency analyses.

4 RESULTS AND DISCUSSION

As the training and testing activities were being executed, results were collected and evaluated, assessing whether it was necessary to modify any configurations. Thus, in this section, obtained results from each methodology section will be presented.

4.1 Database

Collected database resulted in 137 photos of emergency lamps, 147 photos of rusty objects and 256 photos of fire extinguishers. Even though we collected photos from emergency lamps, models were trained without this class of objects. This was due to the fact that some of the objects from lamp emergency class did not appear very clearly, which was the same problem for the status led class. Trained models were not able to find the power socket from the emergency lamps either, explained by the lack of sufficient images for the proper generalization. Therefore, we decided to train the models only for the classes strictly related to fire extinguishers. Some examples from the tailored dataset can be seen at Figure 2.

4.2 Models Performance

Performance for each of the trained models will be analyzed in this section considering the metrics discussed in Section 3.

4.2.1 Losses

Losses for each model for both training and testing sets can be seen on Figure 3. Results show that losses for FPN50 and IRV2 are significantly smaller than MV2.

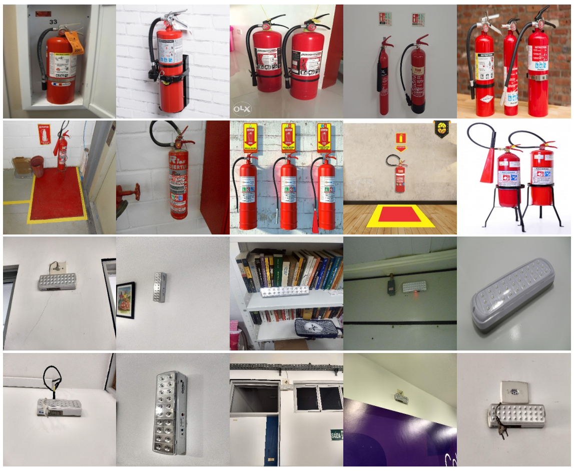


Figure 2: Database containing fire extinguishers (top couple of rows), vertical (red signs pointing out where extinguishers are) and horizontal (yellow stripes) signalization and emergency lamps (bottom two rows).

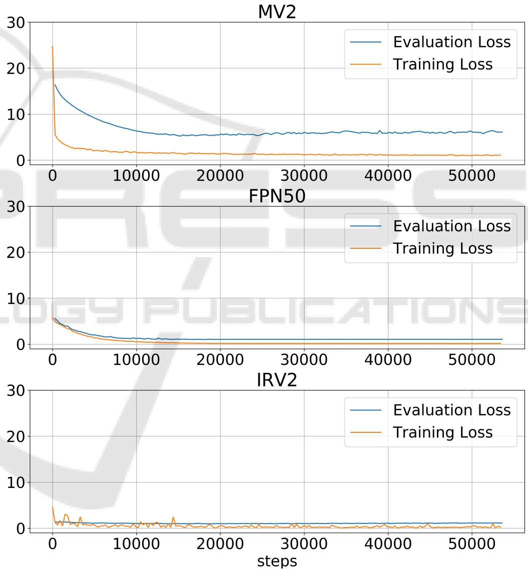


Figure 3: Training and evaluation losses for the three models.

4.2.2 Average Recall

Here it is shown the AR results using all discussed configurations : AR@1, AR@10 e AR@100. They can be seen at Figure 4. It shows that MV2 had similar results from IRV2, although it took more time to reach its value. With respect to FPN50, it showed better results considering the initial steps of training, but had worse performance when compared to other two models at the end of training. IRV2 kept better than other models during all the training, no matter what configurations was used.

4.2.3 Mean Average Precision

Regarding the mAP metric, shown in Figure 5, better results were obtained from mAP@0.5IoU configura-

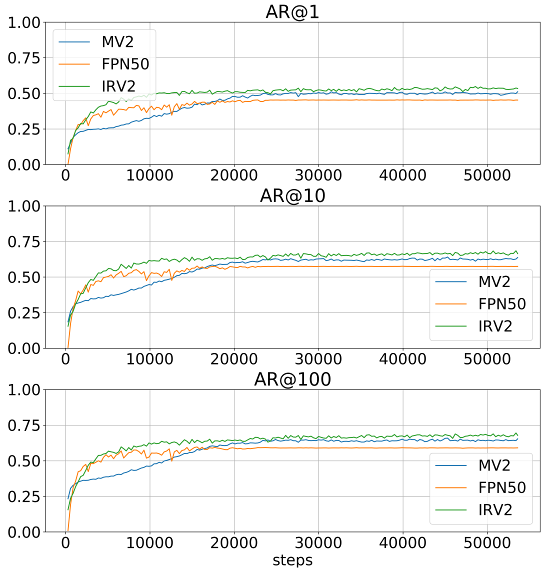


Figure 4: Average Recall for the three models.

tion. In a broader perspective, MV2 model was able to approach the IRV2 for all configurations considering the last steps of training. FPN50, once again, presented the best results at the training start, but did not manage to maintain its lead until the end. During all the training, IRV2 kept itself as the best model no matter what configuration.

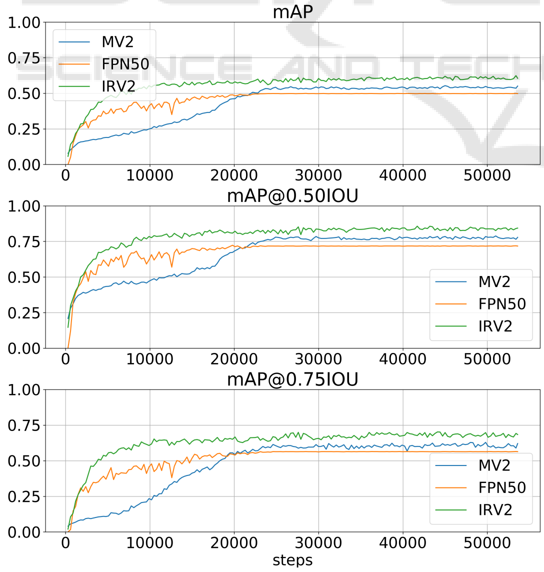


Figure 5: Mean Average Precision calculated for the three models.

4.2.4 Average Recall across Scales

Results for the metric AR Across Scales are showed in Figure 6. All models presented better results for larger objects, when compared to medium and small objects. MV2 presented a better result than IRV2 for large objects. It is also shown that IRV2 presented significantly better results for small objects. FPN50 results stabilizes quicker for all configurations, and shows slightly better results for small objects.

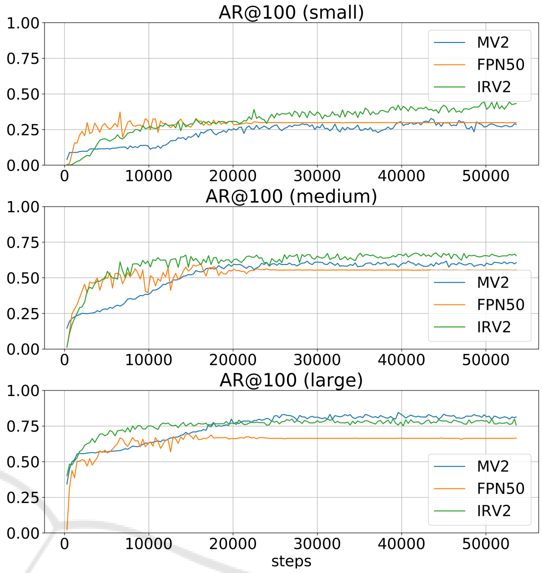


Figure 6: Average Recall Across Scales for the three models.

4.2.5 Mean Average Precision across Scales

Results showed, as illustrated in Figure 7, that MV2 did not present good results for small objects. However, for medium and large objects, results are improved for this topology, being closer to IRV2 performance. FPN50 model presented to be better than MV2, for small objects, but did not reach results as good for medium and large objects.

4.2.6 Frames per Second Ratio

As listed in Table 1, it is shown that MV2 reaches much better results than the other two models with respect to Frames per Second (FPS). MV2 is the only one that is able to run in a mobile device appropriately. FPN50 is the second fastest, which presents a good performance when running in a RTX 2080 Ti graphics card. IRV2, at last, is not able to achieve a good performance, running on average of 3 FPS.

Table 1: FPS rate for the three models in RTX 2080 Ti and mobile device.

Models	RTX 2080 Ti	Mobile Device
MV2	42	5
FPN50	12	-
IRV2	3	-

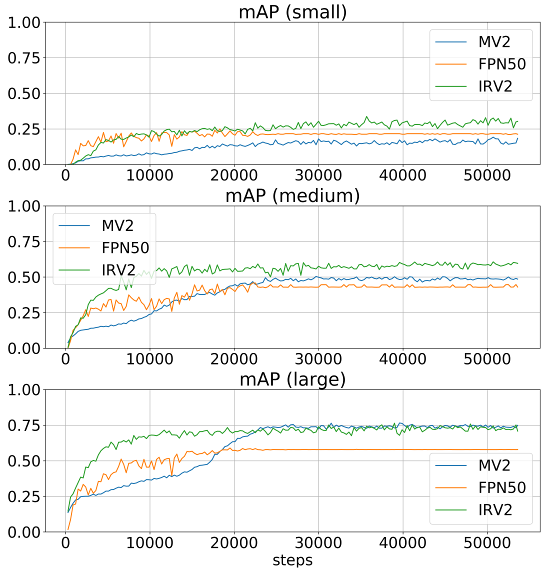


Figure 7: Mean Average Precision Across Scales for the three models.

4.2.7 Comments about Models Performance

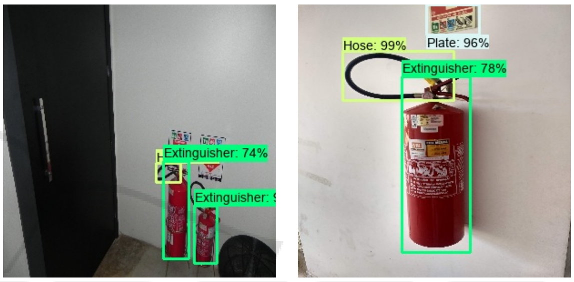
MV2 showed to be very versatile, achieving good results to medium and large objects. Along with FPN50, they were faster than IRV2. Despite achieving good results, MV2 and FPN50 did not perform as good when the input image contained objects in a more complex environment. Analyzing images presented at the Figures 8a, 9a and 10a, it's possible to notice that IRV2 managed to capture more complex patterns from the image and carry out the detection successfully even in not so well behaved images. Nonetheless, cases where images are ofter well behaved, models are able to reach similar results, as it is shown in the Figures 8b, 9b and 10b.

5 CONCLUSION AND FUTURE WORKS

Three models capable of detecting faults in fire extinguishers were detected. Used methodology may be applied to other objects within industry environment and developed models are adequate to different kinds of auditing. A specific database was built using different sources and data augmentation. It was used to train and test the models.

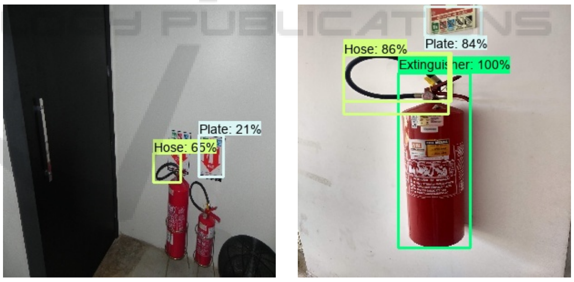
Results have shown that the MV2 allows for the execution of an auditing in real time, by using the model on a mobile device, or even on a computer if better efficiency is need. FPN50 is an in-between for the two other models, since it is able to detect small, medium and big defects in the fire extinguishers and allows for execution in real time, but cannot be executed in a mobile device. IRV2 provides the capability of detecting more complex patterns, being able to better detect the flaws in the extinguishers and significantly reducing the number of false-positives and false-negatives. On the other hand, IRV2 requires more robust computing power in order to be carried out. Using networks pre-trained on large datasets allowed for the models to converge easier when trained on smaller datasets. This kind of approach is suitable for deep learning applications.

As future works, more images will be fed into the dataset and more classes will be created. With respect to emergency lamps, more images will be collected so that their audit can be executed along with extinguish-



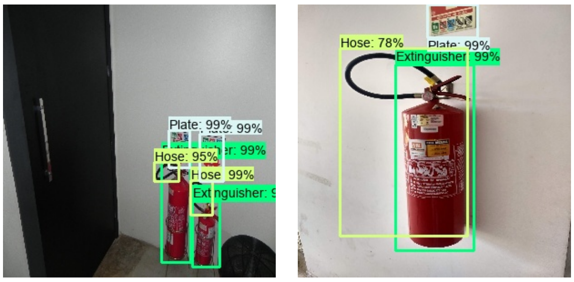
(a) Unfavorable (b) Normal

Figure 8: Performances of FPN50's topology when submitted to unfavorable and normal scenarios.



(a) Unfavorable (b) Normal

Figure 9: Performances of MV2's topology when submitted to unfavorable and normal scenarios.



(a) Unfavorable (b) Normal

Figure 10: Performances of IRV2's topology when submitted to unfavorable and normal scenarios.

ers and other objects that will be incorporated on the database.

We also intend to address other industry problems, such as the verification of the load from the extinguisher, and if its labeling panel is preserved and readable. More up-to-date topologies will also be tested, aiming to obtain better results for mobile devices. Further results will be reported eventually.

ACKNOWLEDGEMENTS

The authors would like to thank the Edge laboratory[1] at the Federal University of Alagoas[2] which funded this research through an agreement regulated by the Information Technology Law.

REFERENCES

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Springer International Publishing, 1 edition. 1

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848. 4

Choi, Y., Park, J.-H., and Jang, B. (2019). A risk estimation approach based on deep learning in shipbuilding industry. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1438–1441. IEEE. 2

Chou, Y.-C., Kuo, C.-J., Chen, T.-T., Horng, G.-J., Pai, M.-Y., Wu, M.-E., Lin, Y.-C., Hung, M.-H., Su, W.-T., Chen, Y.-C., et al. (2019). Deep-learning-based defective bean inspection with gan-structured automated labeled data augmentation in coffee industry. *Applied Sciences*, 9(19):4166. 2

Dargan, S., Kumar, M., Ayyagari, M. R., and Kumar, G. (2019). A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning. *Archives of Computational Methods in Engineering*. 1

Hocenski, Ž., Matić, T., and Vidović, I. (2016). Technology transfer of computer vision defect detection to ceramic tiles industry. In *2016 International Conference on Smart Systems and Technologies (SST)*, pages 301–305. IEEE. 1

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311. 4

Li, L., Ota, K., and Dong, M. (2018). Deep learning for smart industry: Efficient manufacture inspection system with fog computing. *IEEE Transactions on Industrial Informatics*, 14(10):4665–4673. 2

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125. 3

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer. 4

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer. 3

Rao, Q. and Frtunikj, J. (2018). Deep learning for self-driving cars: chances and challenges. In *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*, pages 35–38. 2

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520. 3

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*. 3, 4

Veeraraghavan, A., Adithya, V., Bhave, A., and Akella, S. (2017). Battery aging estimation with deep learning. In *2017 IEEE Transportation Electrification Conference (ITEC-India)*, pages 1–4. IEEE. 2

Zhang, N., Liu, Y., Zou, L., Zhao, H., Dong, W., Zhou, H., Zhou, H., and Huang, M. (2018). Automatic recognition of oil industry facilities based on deep learning. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 2519–2522. IEEE. 2

[1] http://edgebr.org/

[2] https://ufal.br/