Building a Camera-based Smart Sensing System for Digitalized On-demand Aircraft Cabin Readiness Verification

Luis Unzueta[1][a], Sandra Garcia[2], Jorge Garcia[1], Valentin Corbin[2], Nerea Aranjuelo[1], Unai Elordi[1], Oihana Otaegui[1] and Maxime Danielli[2]

[1]*Vicomtech Foundation, Basque Research and Technology Alliance (BRTA), Mikeletegi 57,*
20009 Donostia-San Sebastián, Spain

[2]*Otonomy Aviation, 16 Avenue Pythagore, 33700 Mérignac, France*

Keywords: Intelligent Sensors, Computer Vision, Machine Learning, Deep Neural Networks, Aircraft Cabin.

Abstract: Currently, aircraft cabin operations such as the verification of taxi, take-off, and landing (TTL) cabin readiness are done manually. This results in an increased workload for the crew, operational inefficiencies, and a non-negligible risk of human errors in handling safety-related procedures. For TTL, specific cabin readiness requirements apply to the passenger, to the position of seat components and cabin luggage. The usage of cameras and vision-based object-recognition algorithms may offer a promising solution for specific functionalities such as cabin luggage detection. However, building a suitable camera-based smart sensing system for this purpose brings many challenges as it needs to be low weight, with competitive cost and robust recognition capabilities on individual seat level, complying with stringent constraints related to airworthiness certification. This position paper analyzes and discusses the main technological factors that system designers should consider for building such an intelligent system. These include the sensor setup, system training, the selection of appropriate camera sensors and lenses, AI-processors, and software tools for optimal image acquisition and image content analysis with Deep Neural Network (DNN)-based recognition methods. Preliminary tests with pre-trained generalist DNN-based object detection models are also analyzed to assist with the training and deployment of the recognition methods.

1 INTRODUCTION

In recent years artificial intelligence (AI) has strongly gained momentum, especially due to the remarkable advances obtained by one of its multiple expressions, which is machine learning, thanks to the emerging Deep Neural Networks (DNNs). Currently, DNNs constitute the basis for the most advanced computer vision and machine learning methodologies (Mahony et al., 2019).

In the aircraft cabin environment, cameras are used as of today for overall cabin monitoring purposes. Current cabin video monitoring systems are characterized by restrained video and image analysis capabilities and are not conceived for specific purposes such as taxi, take-off, and landing (TTL) cabin readiness verification. Despite the recent progress for the optimal installation of surveillance cameras to monitor different areas of aircrafts, in practice, the captured images are not being fully exploited. Moreover, different format images and cameras should need to be concealed to exploit the captured images, including AI to help the crew in handling safety procedures.

Building a camera-based intelligent system for this purpose reaching the highest *Technology Readiness Level (TRL)* (Heder, 2017), i.e., TRL9, “an actual system proven in an operational environment”, requires satisfying many challenges. With the currently available DNN-based methodologies and equipment this process would start in TRL2, i.e., “a technology concept formulated”, and the next step would be to build a TRL3 “experimental proof of concept”. This transition from TRL2 to TRL3 is not evident, and relevant technological factors must be analyzed in detail.

[a] https://orcid.org/0000-0001-5648-0910

Unzueta, L., Garcia, S., Garcia, J., Corbin, V., Aranjuelo, N., Elordi, U., Otaegui, O. and Danielli, M.
Building a Camera-based Smart Sensing System for Digitalized On-demand Aircraft Cabin Readiness Verification.
DOI: 10.5220/0010128500980105
In *Proceedings of the International Conference on Robotics, Computer Vision and Intelligent Systems (ROBOVIS 2020)*, pages 98-105
ISBN: 978-989-758-479-4

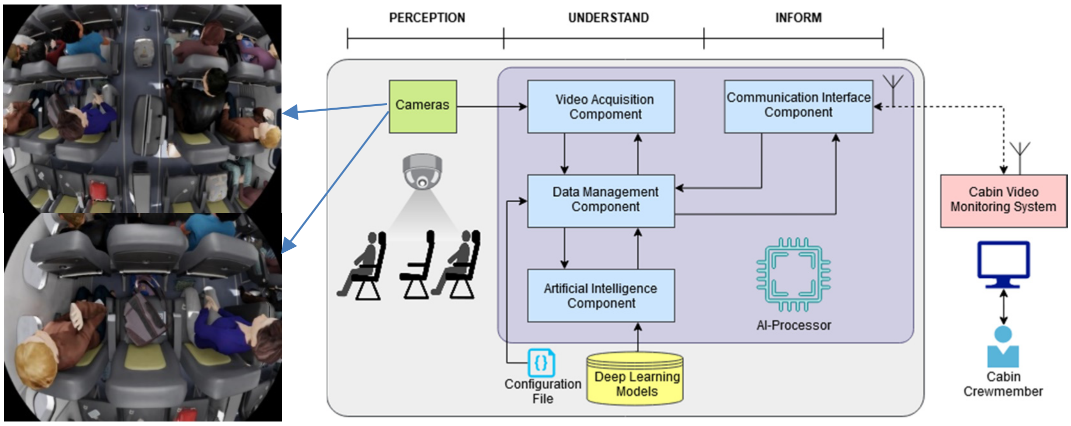
PERCEPTION
UNDERSTAND
INFORM
Cameras
Video Acquisition Component
Communication Interface Component
Data Management Component
Artificial Intelligence Component
AI-Processor
Configuration File
Deep Learning Models
Cabin Video Monitoring System
Cabin Crewmember

Figure 1: Conceptual design of a camera-based intelligent system for digitalized on-demand aircraft cabin readiness verification, and examples of the kind of images that would be captured from cameras installed over the seats and the corridor.

Figure 1 shows the conceptual design of such a system and the kind of images that would be captured from cameras over the seats and the corridor. In these examples, luggage is placed in different wrong areas for TTL, such as on the corridor's floor and seats, and passengers can also occlude totally or partially the luggage depending on their locations and poses.

The economic viability of the system requires to minimize the number of cameras to be installed. This means that each should cover the maximum possible area, e.g., two seat-rows. As it can be observed, the kind of lenses that allow this significantly distort the image content. Thus, the appearances of the visualized objects can be quite different depending on the image region where they are located and the camera position. All these factors are relevant for the design of the computer vision and machine learning algorithms (Zhao et al., 2019). Captured images would then be processed in AI-processors, which in our context, are edge computing devices that include AI-accelerator(s), i.e., a new generation of CPUs, GPUs, FPGAs, and alternative chips, specially designed for the optimal deployment of DNNs (Chen et al., 2020).

An example of DNN-based approach that could be included in the system to handle the TTL-related use-cases would be the following. For instance, a DNN-based image classifier (Wang et al., 2019) could analyze incoming images to check whether they were "grabbed in night conditions" or not, and then choose suitable detectors for each case. DNN-based object detectors would localize bounding boxes of objects of interest in the image with their corresponding class (e.g., "suitcase", "laptop", etc). Alternatively, there are also detectors that segment objects at pixel-level, but their computational cost is prohibitive for AI-processors in our context, so we would not consider them (Zhao et al., 2019). Person detectors could work in the same way as object detectors, but could also be more sophisticated approaches, such as those that localize human body keypoints for body pose estimation. However, again, the computational cost of the latter is prohibitive for AI-processors in our context, so we would only consider the former (Dang et al., 2019). These could also be trained to classify some attributes (e.g., a description of their poses, such as "seated", "crouched", etc). Finally, visual relationship detectors (Agarwal et al., 2020) would describe the relations between objects (and people) localized in the image in some TTL-related use-cases (e.g., "pouch filled, not allowing enough safe evacuation"). Thus, the system would send alerts to the crewmembers when it detects cabin luggage in wrong areas for TTL, specifying where they happened.

This position paper analyzes and discusses the main technological factors that system designers should consider for building such an intelligent system. The purpose of this work is not proposing a specific DNN-based approach that would allow building a camera-based system that could reach TRL9. Our motivation at this stage (TRL2) is to make system designers be aware of the relevant technological factors required for that and assist them with the TRL2 to TRL3 transition, as it involves making important decisions to build a successful solution. These include the sensor setup, system training, the selection of appropriate camera sensors and lenses, AI-processors, and software tools, for optimal image acquisition and image content analysis with DNN-based recognition methods. Preliminary tests with pre-trained generalist DNN-based object detection models are also analyzed to assist with the training and deployment of the recognition methods.

2 SENSOR SETUP AND SYSTEM TRAINING

Two critical tasks to build a suitable vision-based system for automated image content analysis are the sensor setup and the system training. The former refers to choosing the right number of cameras, their locations, orientations, and parameters such as the field of view, to guarantee the complete coverage of the areas of interest, but also considering the economic viability. Regarding the system training, current state-of-the-art DNN-based methods need to be trained with very large datasets, with over 1M labeled images. Collecting such amount of labeled data can be challenging and even prohibitive.

Training techniques that rely on the use of DNN models that have been pre-trained on generic data, such as "transfer learning" and "fine-tuning", allow mitigating the lack of labeled data. Generic data means many kinds of objects "in the wild", not highly specialized, and coming from different contexts, like those from generalist datasets such as ImageNet (Russakovsky et al., 2015), COCO (Lin et al., 2014) or Open Images (Kuznetsova et al., 2020). This way the pre-trained features would be adapted to the new data, potentially achieving meaningful improvements. Recently, Kolesnikov et al. (2020) revisited this kind of techniques and proposed a simple recipe, named "Big Transfer" (BiT), which exploits large scale pre-training to yield good performance on downstream tasks of interest. Even though best results are obtained by training the system with a large dataset specifically designed to train the DNNs with the expected kind of images, considering such pre-trained model-based training techniques is highly relevant for the development of the system.

2.1 Simulation Tool

A proper simulation tool can alleviate the effort and cost of these two tasks. Furthermore, the simulation tool can resolve legal or privacy issues, which often make the use of real data hard or impossible. However, there are no general simulation tools ready to solve them for all kinds of scenarios and use cases. Normally, 'ad-hoc' tools are built. They require integrating adequate 3D graphical assets, virtual camera models, and illumination functionalities configured employing user-friendly parameters.

For the camera setup task, the tool would allow visualizing directly the 3D scene from the virtual camera viewpoints, with camera-related effects, such as the geometric distortion introduced by the lens, while changing in real-time their parameters. For the labeled dataset generation, the tool would allow configuring scenes for a given camera setup, with a wide and balanced range of plausible situations of interest randomly generated. It would involve the required 3D graphical assets, which should be quickly captured from the virtual camera viewpoints, along with the adequate annotations and noise, for training.

Domain Adaptation techniques could be applied to guarantee successful training with data that combines synthetic and real images, as real-world images are more complex than simulated ones. These techniques try to solve the domain shift that exists between different groups of data, for example by learning domain-invariant features (Pinheiro, 2018, Chen et al., 2018) or translating the data from a domain to another one (Inoue et al., 2018).

The basis of the simulation tool could be a 3D computer graphics software such as 3DS Max or Blender or a game engine such as Unreal or Unity, which not only allow managing 3D graphical assets, virtual cameras, and illumination conditions interactively but also controlling all of these utilizing programming scripts. Each of these has its pros and cons, however, with any of them it is necessary to face the same kind of challenges:

- What kind of parameters should be used to configure the scenes?
- How should these parameters be expressed to be user-friendly?
- What kind of strategy should be applied to generate a wide and balanced range of plausible situations of interest, randomly?
- How can the captures from virtual camera viewpoints be done quickly, taking into account that the rendering time could be an important bottleneck in this process?
- What kind of strategy should be adopted to apply noise to the labeled data for the appropriate training of the system.

When designing a simulation tool that responds appropriately to all these challenges, relying on data formats such as VCD (Vicomtech, 2020), can be helpful. It is a flexible labeling structure that addresses modern requirements for ground truth description, including multi-sensor object, action, and scene-level annotations, open-source APIs to manage content, and connectivity with ontologies.

2.2 Camera Sensor and Lens Selection

To build or choose the ideal camera for optimal image acquisition, it is necessary to define all the relevant and impacting parameters that affect the image quality. These include optical lens parameters (spatial resolution, deep of field and luminosity), sensor specifications (quantum convert, pixels and format), and the camera interface (connectors and protocols).

The objects detected by the camera need to occupy a certain number of pixels on the image to be recognized by DNN-based computer vision algorithms. This number of pixels is commonly named as object size, which depends on the distance between the optical unit and the object, as well as the spatial resolution of the camera. The latter depends on the smallest measurable gap between black and white bands expressed in frequency = line / mm.

The depth of field is the distance between the closest and farthest objects in the optical system that results in a sufficiently sharp image. A camera can only focus sharply at one specific distance. But the transition from sharp to unsharp is gradual, and the term “acceptably clear image” is generally imprecise because it is subjective, but in our context, it should be the minimum of spatial resolution before an object becomes undetectable. What defines if an image is optically sharp enough, is the circle of confusion, also known as the disk of confusion, circle of indistinctness, blur circle, or blur spot. It is an optical spot caused by a cone of light rays from a lens not coming to a perfect focus when imaging a point source. In our case, an airplane cabin is a small space where the camera should have sufficient resolution from top to bottom of the seats considering all with a fixed focal lens.

The luminosity of an object is a measurement of its intrinsic brightness and is defined as the amount of energy the object emits in a fixed time. In our case, the luminosity would come mainly from the sun and cabin lighting. The area of measurement should cover a larger area with a huge delta of illumination. We need to verify that the camera would be able to produce enough contrast, and an exploitable measurement for the computer vision and machine learning algorithms.

Starting from the lowest possible illumination within the framework of the measurement, it is necessary to define the surface of the sensor and the number of apertures of the lens, according to the quantum conversion rate, to check whether the camera would be able to measure with sufficient resolution. Knowing the value of the illumination of a surface we can measure with a resolution test if the camera can produce a sufficiently sharp image.

A minimum of spatial resolution is necessary to be able to see at a desired level of detail. This level of detail is discretized by the pixel matrix of the sensor. An aspect ratio is a proportional relationship between an image’s width and height. Essentially, it describes an image’s shape. Aspect ratios are written as a formula of width to height, like this: 3:2. Most common sensors have an aspect ratio of 16:9 or 4:3. For our application, the minimal resolution would be defined by the spatial resolution and format benchmarking according to the technical needs to operate the computer vision and machine learning algorithms. The best aspect ratio is 4:3 because the optimal deployment of DNNs in AI-processors usually is achieved for square images. To be able to process them, images should be padded with black areas to make it square or cropped if it fills the space to be analyzed.

Regarding the communication protocol, there is a large panel of choice (fiber optic with CML protocol, 3G-SDI, FireWire, USB 3.0, Ethernet-PoE, and CAN bus). There would no need for full-duplex communication because the cameras either would send or receive data but not at the same time. All protocols that can transport more than 35 Mbps of data over a plane length would fit. For instance, a good option to send the video flow to the AI-processor could be using 3G-SDI over fiber optic cables.

3 ALGORITHMS DEPLOYMENT

To fulfill the system’s requirements including those related to airworthiness certification, our criteria for the AI-processor scouting is that it should be compact, fanless, powerful enough, efficient, with the highest possible support for deploying cutting-edge DNN-based computer vision and machine learning algorithms, easily integrable in the system’s architecture, low-cost and easily replaceable by upgraded versions without affecting the software development, if required.

To support the AI-processor scouting process we rely on MLPerf Inference (Reddi et al., 2019), which is a relevant benchmark suite in the machine learning community for measuring how fast machine learning systems can process inputs and produce results using a trained DNN model. It was designed with the involvement of more than 30 organizations as well as more than 200 machine learning engineers and practitioners to overcome the challenge of assessing machine learning-system performance in an architecture-neutral, representative, and reproducible manner, despite the machine learning ecosystem’s many possible combinations of hardware, machine-learning tasks, DNN models, data sets, frameworks, toolsets, libraries, architectures, and inference engines. The design and implementation of MLPerf Inference v0.5 consider image classification and object detection vision tasks with heavyweight and lightweight DNN models in different scenarios such as multi-stream processing, relevant in our context. In such a scenario, a traffic generator sends a set of inferences per query periodically (between 50 and 100 ms).

Among the performance results published in MLPerf Inference v0.5, the NVIDIA Jetson AGX Xavier shows good potential for our purpose. It is an embedded system-on-module (SoM), thus, compact, with powerful and efficient GPU/CPU and connectivity capabilities, leveraged by NVIDIA's TensorRT software tool for high-performance DNN inference.

More recently, Libutti et al. (2020) proposed some adaptations to MLPerf to handle new USB-based inference accelerators, more specifically, Intel Movidius Neural Compute Stick 2 (with an Intel Movidius Myriad X VPU) and Google Coral USB accelerator (with a Google Edge TPU). They evaluated the ability of these USB-based devices to fulfill the requirements posed by applications in terms of inference time and presented a mechanism to measure their power consumption. Their results show good potential too for our purpose.

VPUs and Edge TPUs can also be found in other kinds of hardware architectures (e.g., AI-accelerator cards) with faster transmission speeds than USB 3.0, and the software tools that power them, i.e., OpenVINO for Intel chips and TensorFlow Lite supported by the Edge TPU compiler for Google chips, like in the case of TensorRT, have remarkable resources and active communities online, as well. Hence, these are examples of potentially good options to consider for the development and deployment of the algorithms and DNN models.

3.1 Preliminary Tests with Pre-trained Generalist DNN Models

The recognition of cabin luggage in an airplane is an "open-world problem", with almost infinite different objects, textures, shapes to be recognized. This is very difficult to solve employing an automated solution. Human errors are a problem, but a machine attracting the attention of humans to false alerts (i.e., false positives) is a problem too, and not presenting alerts when necessary (i.e., false negatives) is even worse. A TRL9 system should effectively assist crewmembers with the verification of TTL cabin readiness. This means that it should have a very high accuracy, ideally perfect. In case a certain failure rate is unavoidable, the sensitivity of the system should prioritize the elimination of false negatives, even though some false positives might arise.

In the introduction we described a series of TTL-related use-case situations that could be tackled with different kind of DNN-based methods, such as image classifiers, object/person detectors and visual relationship detectors. We remark that it was only an example of a DNN-based approach to build the system, and that alternative approaches could be proposed. For instance, considering that the cameras are stationary, temporal context information could be used (Beery et al., 2020), to reduce the number of false positives. Furthermore, binary content-based image classifiers could be applied to areas of interest to classify whether TTL cabin readiness is complied or not. Incremental learning strategies (Yang et al., 2019) could also be added to readjust the system once deployed, leveraging the observed false positives and negatives to improve the accuracy through time.

Currently, the major open-source deep learning frameworks are TensorFlow and PyTorch, which count with big support not only from the companies that develop and maintain them (Google and Facebook, respectively) but also from researchers and practitioners around the world. Thus, there are many pre-trained generalist DNN TensorFlow and PyTorch models available on the Internet, especially for image classification and object/person detection tasks with generalist datasets, which can also be converted to 3rdparty inference engines for the optimal deployment of DNNs in AI-processors like those relevant in our context (e.g., for NVIDIA's TensorRT and Intel's OpenVINO).

Testing pre-trained generalist DNN models for object/person detection with images such as those shown in Figure 1, could be seen as quite simple, but it can help us clarify important aspects for the design and development of the TRL3 DNN-based approach for the system, like:

- How do image transformations like resizing and rotating affect the pre-trained generalist DNN responses?
- Considering lens distortions and that the aspect ratios of the DNN's input and the image can be quite different, is it convenient to process the full image or cropped image regions?

For these preliminary tests, we have chosen the SSD-MobileNet-v1-FPN model, which has a good trade-off between accuracy and performance among those publicly available at the TensorFlow Object Detection API webpage (Huang et al., 2017), trained with the generalist COCO dataset. Nevertheless, the object detection field is constantly progressing, and newer more accurate, efficient and smaller DNN models are being proposed, such as EfficientDet (Tan et al., 2020), released on March 18, YOLO-v4 (Bochkovskiy et al., 2020), released on April 23, and the controversial YOLO-v5 (Jocher, 2020), released on June 9, and will probably continue so during the following months and years. Similar tests could be done with these models too for the same purpose.

We have processed images such as those shown in Figure 1 in two ways: (1) the full image with padding added to maintain the aspect ratios (the DNN expects square input sizes), and (2) the image cropped in 7 regions, also with the same kind of padding for each region. In both cases, the object detector receives images at different sizes (640x640, 1280x1280 and 1920x1920 for full images; 640x640 and 960x960 for cropped images) and orientations (0º, 90º, 180º and -90º). Figure 2 and Figure 3 show some examples of the obtained detection results.

In the COCO dataset images are not as distorted as in these images, and therefore, objects/people seen from the camera over the corridor are better detected, as they are further away and less distorted. It can also be observed that the bounding boxes in some cases are correctly placed for the object's real boundaries, but the classification is incorrect. This behavior was expected since COCO contains many classes out of our context (e.g., "car", "sink", "toilet", "stop sign", "mouse", "traffic light", "toaster", etc.), and therefore the DNN has learned "noisy" visual features from them and has more chances to misclassify. Overall, the class that is better detected is that of "person", even when people are partially occluded and located in more distorted image regions.

Another interesting behavior is that some objects are correctly detected in some image orientations but not in others. Probably this happens because the data for training was not sufficiently balanced for this factor, and thus, the learned visual features for some objects work better for some orientations compared to others. It can also be observed that the chosen input size also plays an important role as, depending on it, bigger and/or smaller objects/people can be missed. The reason for other misdetections could also be because COCO does not contain so many top view perspectives of objects/people.

Regarding the convenience of processing cropped image regions instead of the full image directly, in principle, the advantage that it might have is that the input sizes of the regions can be bigger and with less padding areas when processed by the DNN, compared to when processing the full image directly. However, the analyzed preliminary tests do not reveal a clear advantage of the former for the detection accuracy. This factor would need further investigation with more data and quantitative measurements, trying to seek a good trade-off between accuracy and performance, leveraging batch processing techniques.

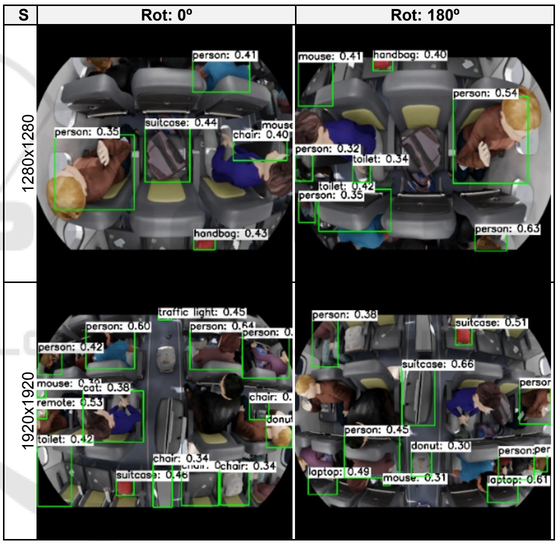

Figure 2: Tests with synthetic full images from cameras over the seats and the corridor with the COCO-trained SSD-MobileNet-v1-FPN model (Huang et al., 2017).

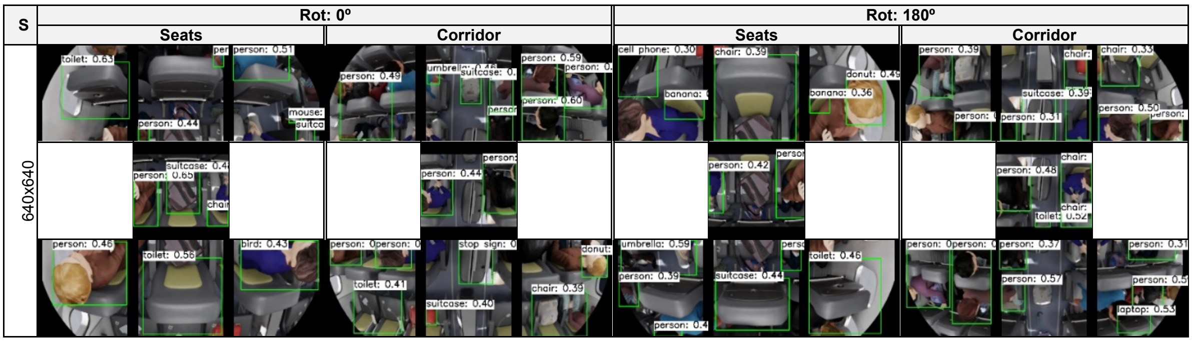

Figure 3: Tests with synthetic images cropped into 7 regions from cameras over the seats and the corridor with the COCO-trained SSD-MobileNet-v1-FPN model (Huang et al., 2017).

4 CONCLUSION AND FUTURE WORK

In this position paper, we have analyzed and discussed the main technological factors that system designers should consider for building a camera-based smart sensing system for digitalized on-demand aircraft cabin readiness verification. Our purpose is to make system designers be aware of the relevant technological factors for that and assist them with the TRL2 to TRL3 transition while designing their DNN-based approaches that would allow building camera-based systems that could reach TRL9. These include the sensor setup, system training, the selection of appropriate camera sensors and lenses, AI-processors, and software tools for optimal image acquisition and image content analysis with DNN-based recognition methods.

For the sensor setup and system training, we consider that a proper simulation tool would be helpful, along with Domain Adaptation techniques to guarantee successful training with data that combines synthetic and real images. Thus, we have reviewed which are relevant considerations to build the simulation tool and proposed relying on a metadata specification format for the description of scenes and data sequences, such as VCD (Vicomtech, 2020). We also consider that training techniques that rely on the inclusion of DNN models that have been pre-trained on generic data, like BiT (Kolesnikov et al., 2020), are beneficial to mitigate the lack of labeled data.

We have also analyzed which are the important and impacting parameters that affect the image quality, to build or choose the ideal camera for optimal image acquisition, and how they should be measured.

Regarding the selection of AI-processors and software tools for the optimal deployment of DNNs, we have established criteria to fulfill the system's requirements and analyzed some remarkable systems based on the MLPerf Inference benchmark suite (Reddi et al., 2019).

Finally, we have presented some preliminary tests with a pre-trained generalist DNN-based object/person detector with the kind of images that would be captured from cameras installed over the seats and the corridor. In summary, based on these tests, we conclude that DNN models should be trained: (1) with balanced data, using closer content to the kind of expected images for each camera placement (i.e., over the seats and the corridor) and the use cases, with all kind of orientations, (2) with the minimal required set of appropriate object classes only, to avoid learning noisy object/person visual features for detection and classification, and (3) exploring the possibility of cropping the image into regions, such as those used for the experiments, to learn their specific image characteristics. Then, trained DNN models should be deployed with appropriate image sizes for each camera placement (i.e., over the seats and the corridor) and using dynamic batch processing to make the most of the AI-processors.

Future work will involve developing a prototype with an IMX226 sensor and an FPGA card to design our camera architecture, an NVIDIA Jetson AGX Xavier SoMs, and FPGAs to process images. Besides, we will include a DNN-based detection approach tailored to the TTL-related use cases, leveraging the considerations presented here. In particular, we plan training the system with real-world and simulated data by applying domain adaptation techniques to learn domain-invariant features, including temporal context information and incremental learning strategies.

ACKNOWLEDGEMENTS

This work has received funding from the Clean Sky 2 Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No. 865162, SmaCS (https://www.smacs.eu/).

REFERENCES

Agarwal, A., Mangal, A. and Vipul, 2020. Visual relationship detection using scene graphs: A survey. *arXiv preprint arXiv:2005.08045*.

Beery, S., Wu, G., Rathod, V., Votel, R. and Huang, J., 2020. Context R-CNN: Long term temporal context for per-camera object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13075-13085).

Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y. M., 2020. YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Chen, Y., Li, W., Sakaridis, C., Dai, D. and Van Gool, L., 2018. Domain adaptive faster R-CNN for object detection in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3339-3348).

Chen, Y., Xie, Y., Song, L., Chen, F. and Tang, T., 2020. A Survey of accelerator architectures for deep neural networks. *Engineering, 6(3)*, pp. 264-274.

Dang, Q., Yin, J., Wang, B. and Zheng, W., 2019. Deep learning based 2d human pose estimation: A survey. *Tsinghua Science and Technology, 24*, pp. 663–676.

Heder, M., 2017. From NASA to EU: the evolution of the TRL scale in Public Sector Innovation. *The Innovation Journal: The Public Sector Innovation Journal, 22(2)*, pp. 1-23.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S. and Murphy, K., 2017. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3296-3297). https://github.com/tensorflow/models/tree/master/research/object_detection

Inoue, N., Furuta, R., Yamasaki, T. and Aizawa, K., 2018. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5001-5009).

Jocher, G., 2020. YOLOv5. https://github.com/ultralytics/yolov5

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S. and Houlsby N., 2020. Big Transfer (BiT): General visual representation learning. *arXiv preprint arXiv:1912.11370*.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T. and Ferrari, V., 2020. The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision, 128,* pp. 1956-1981.

Libutti, L. A., Igual, F. D., Pinuel, L., De Giusti, L. and Naiouf, M., 2020. Benchmarking performance and power of USB accelerators for inference with MLPerf. In *Proceedings of the Workshop on Accelerated Machine Learning (AccML)*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., Zitnick, C. L., 2014. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision, LNCS* (Vol. 8693, pp. 740-755).

Mahony, N. O., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., Riordan, D. and Walsh, J., 2019. Deep learning vs. traditional computer vision. In *Proceedings of the Science and Information Conference* (pp. 128-144).

Pinheiro, P. O., 2018. Unsupervised domain adaptation with similarity learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 8004-8013).

Reddi, Y. J., Cheng, C., Kanter, D., Mattson. P., Schmuelling, G., Wu, C.-J., Anderson, B., Breughe, M., Charlebois, M., Chou, W., Chukka, R., Coleman, C., Davis, S., Deng, P., Diamos, G., Duke, J., Fick, D., Gardner, J. S., Hubara, I., Idgunji, S., Jablin, T. B., Jiao, J., St. John, T., Kanwar, P., Lee, D., Liao, J., Lokhmotov, A., Massa, F., Meng, P., Micikevicius, P., Osborne, C., Pekhimenko, G., Rajan, A. T. R., Sequeira, D., Sirasao, A., Sun, F., Tang, H., Thomson, M., Wei, F., Wu, E., Xu, L., Yamada, K., Yu, B., Yuan, G., Zhong, A., Zhang, P. and Zhou, Y., 2019. MLPerf inference benchmark. *arXiv preprint arXiv:1911.02549*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. and Fei-Fei, L., 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision, 115,* pp. 211-252.

Tan, M., Pang, R. and Le, Q. V., 2020. EfficientDet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10781-10790).

Vicomtech, 2020. VCD - Video Content Description. https://github.com/Vicomtech/video-content-description-VCD

Wang, W., Yang, Y., Wang, X., Wang, W. and Li, J., 2019. Development of convolutional neural network and its application in image classification: A survey. *Optical Engineering*, 58(4), 040901.

Yang, Y., Zhou, D-W., Zhan, D., Xiong, H. and Jiang, Y., 2019. Adaptive deep models for incremental learning: considering capacity scalability and sustainability. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 74-82).

Zhao, Z., Zheng, P., Xu, S. and Wu, X., 2019. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems, 30(11)*, pp. 3212-3232.