POLITECNICO DI TORINO
Repository ISTITUZIONALE

An automatic computer vision pipeline for the in-line monitoring of freeze-drying processes

Original
An automatic computer vision pipeline for the in-line monitoring of freeze-drying processes / Colucci, Domenico; Morra, Lia; Zhang, Xiaoyang; Fissore, Davide; Lamberti, Fabrizio. - In: COMPUTERS IN INDUSTRY. - ISSN 0166-3615. - STAMPA. - 115:Article 103184(2020), pp. 1-12. [10.1016/j.compind.2019.103184]

Availability:
This version is available at: 11583/2775596 since: 2021-11-24T17:30:26Z

Publisher:
Elsevier

Published
DOI:10.1016/j.compind.2019.103184

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
Elsevier postprint/Author's Accepted Manuscript

© 2020. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/.The final authenticated version is available online at: http://dx.doi.org/10.1016/j.compind.2019.103184

(Article begins on next page)

20 April 2024

Computers in Industry [ISSN: 0166-3615], 115, article 103184. DOI: 10.1016/j.compind.2019.103184

An automatic computer vision pipeline for the in-line monitoring of freeze-drying processes

Colucci Domenico[a], Morra Lia[b], Xiaoyang Zhang[b], Fissore Davide[a], Lamberti Fabrizio[b]

[a]*Dipartimento di Scienza Applicata e Tecnologia. Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy.*

[b]*Dipartimento di Automatica e Informatica. Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy.*

**Corresponding author*

Email: domenico.colucci@polito.it

Tel: 0039 011 090 4695

Abstract

Infrared imaging sensors were proposed in the past years for the real-time monitoring and control of the Vacuum Freeze Drying (VFD) process. In order to extract reliable, real-time, and quantitative features from images collected by these sensors, a robust and automated image processing pipeline is required. Two major issues that have to be addressed concern the object detection and segmentation step, which locates and segments the objects whose temperature needs to be measured, and the tracking step, which is about following the movement of objects of interest to correlate information across subsequent images. Traditional intensity-based image analysis techniques are not reliable in this specific application, since the temperature, which is the intensity of a thermal picture, varies with time, whereas deep learning-based techniques are more robust to such changes.

In this work, an object detector network based on a Faster Region Convolutional Neural Network (Faster R-CNN), and a Kernelized Correlation Filter (KCF) tracker were combined to monitor the VFD process of products contained in glass vials, which is a common scenario in the pharmaceutical domain. The object detector was trained on nine experimental acquisitions (7,981 images) and tested on nine more (7,201 images). Data augmentation methods consisting in the generation of synthetic sequences were used to effectively increase the performance while reducing the cost of data acquisition and annotation. The proposed technique achieved a recall and precision equal to 99.6%, with execution time compatible with a real-time application. The localization of the vials was precise enough to measure the mean temperature with an acceptable error.

Keywords: Vacuum Freeze Drying; Real-time monitoring; Object detector; Convolutional neural network; Data augmentation.

1 Introduction

Vacuum freeze drying (VFD) is a low temperature drying process particularly appreciated for the stabilization of liquid pharmaceutical and biopharmaceutical formulations. The main difference compared to the more traditional drying techniques is that the solvent, typically water, is mainly removed from the frozen product via sublimation, thus at a low temperature. The process entails three different stages, namely freezing, primary drying and secondary drying (see, among others, Mellor, 1978; Jennings, 1999; Oetjen and Haseley, 2004; Fissore, 2013). The liquid product is usually poured into glass vials, placed onto shelves in a drying chamber. During the freezing stage, a technical fluid, flowing into the shelves, removes heat from the solution, thus allowing for the free solvent to turn into ice. During the primary drying phase, the pressure inside the chamber is reduced and the temperature of the shelves where the product is placed is increased to allow the sublimation of the ice crystals; during the secondary drying phase, the residual bounded solvent, i.e., solvent in a liquid state absorbed to the solid structure, is desorbed by increasing the temperature.

From the physical point of view, all the three stages are characterized by a heat transfer, either from the product to the environment (freezing stage) or from the environment to the product (drying stages), intimately coupled with mass transfer processes. In the freezing stage, both nucleation and crystal growth are exothermic processes, and the evolution of the product thermal profile can be related to the crystal size distribution in the frozen product (Bald 1991; Nakagawa et al., 2007; Colucci et al., 2019a). During primary drying, given the fact that sublimation is an endothermic process, the front where sublimation occurs behaves as a heat sink, creating a local minimum in the temperature profile that moves from the top surface to the bottom of the product (Velardi and Barresi, 2008). Similar considerations stand for the secondary drying, in which the heat supplied to the product is used for water desorption (Pikal, 1990). For this reason, the measurement of the product temperature in a well-defined position (usually the bottom of the product) coupled with a mathematical model of the process allows one to infer all the variables and parameters required for the monitoring and control of the process, in particular during the primary drying stage (Fissore, 2018): (i) the heat transfer coefficient from the shelf to the product in the vial, (ii) the mass transfer coefficient from the interface of sublimation to the drying chamber, (iii) the sublimation flux and, finally, (iv) the ending point of the primary drying stage. Moreover, the use of multiple temperature measurements in different spatial positions inside the batch allows one to account for the *in-batch* variability (Bosca et al., 2013).

Traditionally a thermocouple, at lab scale, or a thermistor, preferred at an industrial scale since easier to sterilize, is inserted inside the product and used to measure the temperature at the bottom of the product, that is where the product is in contact with the vial glass. Nevertheless, these sensors present many issues. First, the presence of wires hardly fits the standards in terms of sterility and automatization typical of a pharmaceutical process (Willemer 1991; Oetjen and Haseley 2004). Even more important, the introduction of an external object inside the solution interferes with the freezing stage, making the monitored vial not representative of the rest of the batch.

With the aim to solve those issues, in recent years infrared cameras have been proposed as sensors suitable for a non-invasive monitoring of the thermal evolution of the product being freeze-dried. Emteborg et al. (2014) first presented the application of an infrared camera to the monitoring of a freeze-drying process. In their work, an infrared camera (VarioCAM, InfraTec GmbH) was mounted on the ceiling of a lab-scale freeze drier and used to monitor the thermal evolution of the product, both single dose and bulk, placed on the upper shelf of the equipment: in their setup, only the top temperature of the product could be measured. Lietta et al. (2019) removed the spatial limitation by designing a new sensor that can be placed inside the drying chamber and used to monitor the product in every shelf position, being able to track the whole axial temperature profile and, thus, also the temperature of the product at the bottom of the container. Clearly, in this setup, the temperature of the glass wall was measured, which nonetheless was proved to be very close to that of the product (Lietta et al., 2019; Colucci et al., 2019a); in any case, a simple equation could be used to estimate the product temperature from that of the glass wall (Van Bockstal et al., 2019). The information provided by this sensor has been used to develop monitoring algorithms for both the freezing (Colucci et al., 2019a) and the primary drying stage, as well as for Multivariate Statistical Process Control (MSPC) (Colucci et al., 2019b; Colucci et al., 2019c). Van Bockstal et al. (2019) presented an application of the infrared imaging technology to the optimization and monitoring of a continuous spin freeze-drying process. In their approach, the sensor was placed outside the chamber and the temperature measured was used to infer the parameters of the process. No image analysis pre-treatment was applied.

The output of an infrared camera is always a thermal image and, whenever the information required is more than the evolution of the temperature in a single pixel, an image analysis step is mandatory to extract useful features, improve the quality of the reading, and track the spatial movement of the objects during the process. When the camera provides the measurements for a monitoring algorithm or a control loop, the quality and reliability of those systems will strongly be affected by the image analysis algorithms employed.

With the increased availability of high quality, affordable industrial cameras, computer vision algorithms have been increasingly applied to the monitoring of industrial manufacturing processes (Coffie, 2018), to the inspection of pharmaceutical product lines (Bahaghighat, 2018), as well as in the food and agricultural sectors (Dias, 2018).

In many cases, industrial computer vision applications rely on ad-hoc algorithms specifically designed for the particular object and acquisition setup being monitored, with a strong focus on co-designing the acquisition and processing pipelines. Raponi et al. (2017) reviewed many applications of image processing to the monitoring and optimization of vegetable and fruit drying and found that most papers employed ad-hoc pipelines based on traditional image processing techniques. For instance, Huang et al. (2014) used a combination of segmentation and active contours to segment soybeans from hyper-spectral images, whereas Liu et al. (2016) used a combination of Canonical Discriminant Analysis and global thresholding to segment carrot slices. In both cases, the object simple shape and uniform color together with the dark uniform background made the segmentation task easy to accomplish. Colucci et al. (2019b) developed an image analysis pipeline applied to the segmentation and registration of pharmaceutical vials in thermal images, based on the camera sensor developed by Lietta et al. (2019); the proposed algorithm, while effective in extracting temperature reading for monitoring the VFD process, relied on prior knowledge about the drying chamber and the position and size of each vial on the shelves, thus requiring parameters to be adjusted for each batch being monitored. A more general approach, applicable to a wider range of setups with minimal human supervision, is needed to move beyond laboratory settings and effectively monitor an actual production process.

Unfortunately, in a thermal image, the intensity of the single pixels corresponds to the temperature measured in that specific position in space. If the temperature changes quickly with time, traditional intensity-based image analysis algorithms, which are not robust to these changes, may not be effective. As an example, in the early stages of primary drying when pressure is reduced, the temperature of the product increases (up to 30 degrees in some cases) in a very limited amount of time. Furthermore, the vibration of the equipment may induce the movement of the vials; hence, it is not possible to rely solely on the position of the objects to perform temperature measurements.

In recent years, deep convolutional neural networks (DCNN) have become ubiquitous in computer vision applications and are increasingly applied also in industrial settings (Coffie, 2018). Common applications include defect detection, fault prognosis and predictive maintenance (Yan, 2019), where DCNNs are used to classify images according to predefined classes and/or to predict the occurrence of possible faults. The advantage of deep learning over traditional imaging processing techniques is that the nature of the defect to be dealt with does not need to be pre-programmed, but instead can be learnt from examples, making the detector more robust to variations. One of the main limitations of deep learning in this context, however, is the need to acquire a sufficient number of samples for training, which may be difficult due to class imbalance (certain defects may be rare) or to the inherent experimental cost in acquiring training samples. These limitations can be addressed by using transfer learning and data augmentation techniques (Shao, 2019).

In machine learning, transfer learning denotes the action of transferring information from one task to another. Seminal works by Yosinki (2014), Oquab (2014) and many others have shown that features from trained convolutional neural networks can be transferred effectively across tasks and datasets. The usual transfer learning approach consists in training a base network and then copying its first *n* layers to the corresponding first *n* layers of a target network, since layers at different depths learn increasingly specific features. The remaining layers of the target network are then randomly initialized and trained towards the target task. In line with previous literature, we here use transfer learning to adapt features across tasks (from image level classification to object detection) and across domains (from RGB images to thermal images). Object detectors in the RGB domain commonly rely on pre-trained backbone networks for feature extraction (Ren, 2015; Zhao, 2019). Successful transfer from the RGB to grayscale domains has also been demonstrated in the medical domain (Morra, 2019; Shin, 2016), thermal imaging (Kwaśniewska, 2018), time-frequency imaging (Shao, 2018; Shao, 2019) and depth imaging (Carlucci, 2018). In transfer learning, one can choose to fine-tune all the layers to the new task, or freeze part of the layers, depending on the difference between the source and target tasks, and the size of the training set (Yosinski, 2014). When transferring across very different domains, it is usually preferable to fine tune all the layers (Morra, 2019), and this is precisely the strategy that we adopt in this work. The mainstream approach for using pre-trained DCNNs on grayscale images requires a mapping to make the input image compatible with RGB architectures: the most common and straightforward approach is simply to replicate the input channel to simulate an RGB image, although other strategies such as colorization have been proven successful in some cases (Carlucci, 2018).

In this paper, we propose to apply DCNNs in order to segment and track regions of interest corresponding to the product in thermal images, in order to obtain temperature measurements and other features useful for monitoring freeze-drying processes. Features learnt by DCNNs are generally robust to (reasonable) changes in illumination, contrast and so forth. Natural image classification is unaffected by changes of intensity values, as long as the local contrast does not change. Likewise, these properties are useful for our application, as the proposed technique will be insensitive to local changes in temperature. More specifically, an algorithm that combines an object detector based on a Faster Region Convolutional Neural Network (Faster R-CNN) and a Kernelized Correlation Filter (KCF) tracker has been developed and tested on a large set of experimental tests. The results showed that the proposed approach provides a reliable tool for the identifying the objects of interest in images from the thermal camera, as well as for using the obtained segmentation to measure their temperature in a non-invasive fashion.

2 Materials and methods

This section is structured as follows: first we present the algorithm designed to detect and track the vials in Sections 2.1, 2.2 and 2.3. Then, we describe the experimental setup used to acquire the dataset for training and evaluating the detector (Section 2.4). Finally, the methodology used to train and evaluate the deep learning-based object detector, including the data augmentation pipeline, is presented in Sections 2.5, 2.6 and 2.7.

2.1 Overall algorithm

The proposed algorithm combines the outputs of a detection and tracking algorithm to achieve optimal performance in localizing the objects of interest in the presence of movement. The underlying rationale is that, once the detection algorithm provides an initial set of objects, the tracker output is more stable in the presence of slow-speed movements or vibrations and may be more accurate in localization than the object detector, since it does not suffer from the downsampling introduced in the feature computation. Tracking is also much faster to compute than object detection, which may be relevant in experimental setups with high frame rates. At the same time, the output of the detector can be used to correct the tracker when a target is lost or incorrectly located. First, the number of objects is not necessarily constant, hence we need to detect when an object enters or exits the camera sensor field of view; in our experimental setup, for instance, some vials fall due to strong vibrations. Second, the performance of tracking is negatively affected by sudden changes in position, which may occur as the result of strong vibrations or fast-moving objects. These problems are particularly evident when the frame rate is very low, and in this case the output of the detector can be used to identify if and when the tracker loses its target.

The detector is based on the two-stage architecture Faster R-CNN, described in Section 2.2, whereas for tracking the KCF algorithm, described in Section 2.3, was chosen. The overall algorithm is depicted in Fig. 1.

In more detail, the algorithm is as follows. First, the image is pre-processed to remove the optical distortion known as "barrel effect", as done in Colucci et al. (2019b). Then, the detector is run every N frames (where $N = 1$ in this work). At the initial frame, the output of the detector is used to initialize the tracker, which then predicts the position of the bounding boxes at the next frame.

Every N frames, the outputs of the detector and the tracker are combined in order to correct possible tracking errors and to detect when vials enter or exit the camera field of view. Let us denote as $BB^D{}_i$ the i-th bounding box generated by the detector, and as $BB^T{}_j$ the j-th bounding box generated by the tracker. At each time step t, the tracker and the detector outputs are matched based on their overlap, determined using the Intersection over Union (IoU) criterion; specifically:

- if the number of objects identified by the detector is different than that identified by the tracker, the tracker is re-initialized using the detector bounding box;
- otherwise, the IoU for each pair of bounding boxes i,j is calculated, denoted in the following as $IoU_{i,j}$;
- for every $BB^D{}_i$, let $a = argmax_j\,(IoU_{i,j})$ be the corresponding tracker bounding box with the highest overlap; if $IoU_{i,a}$ is lower than 0.5, then the tracker bounding box $BB^T{}_a$ is deleted and substituted with the detector bounding box.

The bounding boxes at step $t+1$ are calculated using the tracker starting from the updated bounding boxes.

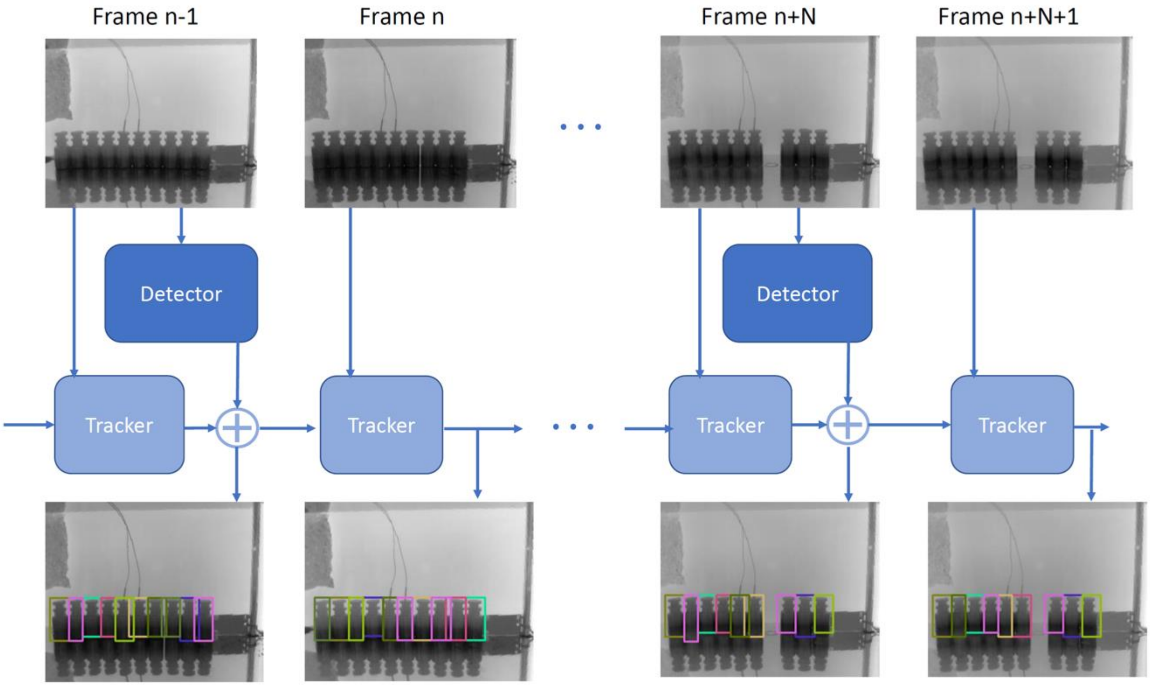

Fig. 1. Schematic representation of the overall algorithm.

2.2 *Faster R-CNN Detector*

We selected the Faster R-CNN detector (Ren, 2015), a two-stage architecture which has achieved high accuracy in several benchmarks. Differently from image-level classification networks, object detectors work by classifying hundreds of potential object regions, which can be generated using fixed grids (one-stage detector) or by employing pre-selection mechanisms (two-stage detectors). Faster R-CNNs belong to the class of two-stage detectors, as a Region Proposal Network (RPN) first narrows down the search by learning to distinguish potential object bounding boxes from the background. A RPN takes an image (of any size) as input and outputs a set of rectangular object proposals, each with an “objectness” score. The object proposals are then classified by a second network, the classifier, into the different object classes (or background).

Both networks share the same convolutional layers for feature extraction, also known as “backbone”; the backbone is usually pre-trained on ImageNet and fine-tuned for object detection. Through layer sharing, the memory and computational requirements are largely reduced. The Faster R-CNN architecture can thus be viewed as a backbone plus two lightweight “heads”, the RPN head and the classifier head. Each head has two outputs: the classification output, and the regression output (which predicts the coordinates of the output bounding box). An overview of the entire architecture is exemplified in Fig. 2. In the following, the main architectural principles are reviewed; the interested reader is referred to the original paper for further details (Ren, 2015).

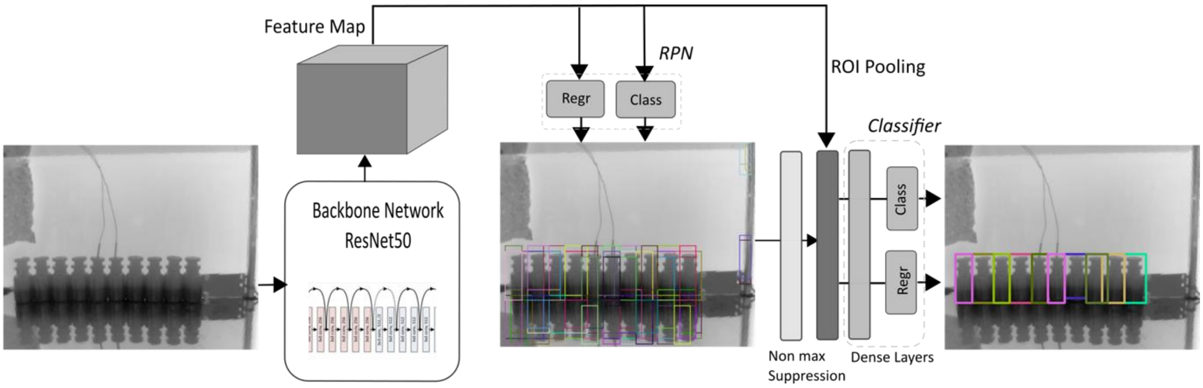

Fig. 2. Main blocks composing the Faster R-CNN architecture. The RPN generates candidate bounding boxes (only a subset is included in the figure for clarity). Note how candidates include bottles, their reflections and, less frequently, portions of the background. Then, the classifier head predicts the final location (bounding box parameters) and class of each object. Only objects classified as "vial" are shown in the final output.

To generate region proposals, the RPN slides over the input image according to a fixed grid, which depends on the implicit downsampling operated by the convolutional and pooling layers of the backbone network. We here use ResNet50 as backbone network, whose output feature map is downsampled by a factor of 16 compared to the original image (He, 2016). The selected backbone is, in our opinion, a good compromise between classification performance and inference time (Canziani, 2016), and is a well-known architecture for which pre-trained model weights are available in most deep learning libraries. There is ample empirical evidence that ImageNet accuracy is a good predictor of performance on other datasets (Kornblith, 2019) and tasks (Zhao, 2019). ResNet50 achieves better performance than VGG16 with a significantly lower parameter count; at the same time, it is faster than deeper architectures such as ResNet101 (2×) and ResNet152 (3×).

At each sliding window location, multiple region proposals are simultaneously predicted, where the number of maximum possible proposals for each location is denoted as *k*. The *k* proposals are parameterized relatively to *k* reference boxes, which are usually denoted as "anchor" boxes: the network takes as input the coordinates of each anchor box and predicts as output the coordinates of the object bounding boxes, along with the "objectness" score. A Region of Interest (ROI) pooling layer applies the RPN-generated bounding boxes to the backbone output and passes the extracted features to the classifier head (Ren, 2015). It is worth noticing that the network parameters are invariant with respect to translation, meaning that the regression and classification parameters only depend on the size, aspect ratio and content of the anchor. Since parameters are independent from the position of the anchor within the image, it is possible to share them across the image and reduce their total count.

The RPN commonly predicts multiple overlapping bounding boxes for the same object. Non-Maximum Suppression (NMS) is used to reduce the number of objects passed from the RPN to the classifier: in synthesis, for each group of overlapping bounding boxes, only the box with the highest classification score is retained, whereas the others are discarded. Overlapping bounding boxes are defined as those whose IoU exceeds a pre-defined threshold.

2.2.1 *Network configuration and training*

Hyper-parameters of the architecture were mostly kept to their default values as specified in Ren (2015). Only a small number of modifications were made.

In the default setting, nine anchor boxes are used, resulting from the combination of three scales and three aspect ratios. In our case, since vials are usually small and narrow objects, we changed the values of the scales and anchor boxes to (32, 64, 128) pixels and 1:1, 1:2, 2:1, respectively, instead of (128, 256, 512) and 1:1, 1:2, 2:1; smaller anchor boxes were proven to be capable improve the localization performance for small object detection in Faster R-CNN architectures (Eggert, 2017).

For the classifier head, we are in principle interested in only one object type, represented by the “vial” class. However, due to the low emissivity of the stainless-steel shelves, vials reflections are often visible in the camera field of view; hence, we defined an auxiliary class named “reflection”, and used it to help the network to distinguish the true objects (upright) from the unwanted reflections (which appear upside down). At test time, only detections classified as “vial” objects are passed on to the tracker, whereas those classified as “reflection” objects or “background” are excluded from further analysis.

Training of the two heads is performed jointly according to the procedure defined as Approximate Joint Training in Ren (2015). At each forward pass (where a forward pass corresponds to one image), the RPN is trained and updated. The region proposals are generated and treated like fixed, pre-computed proposals to train the detector head. This two- step solution is necessary because the interconnecting layers between the RPN and the classifier (namely, ROI Pooling and NMS) are not differentiable, hence, backpropagation is not possible. The backbone weights are shared, so they are updated at each pass.

When training the RPN and the classifier heads, each mini-batch is derived from a single image that contains many positive and negative example anchors. Anchor boxes are labeled as negative or positive depending on whether they match the ground truth annotations, based on a matching criterion which is usually the IoU.

To reduce the computational effort and remove bias towards the negative samples (which are dominating), down-sampling becomes necessary. For the RPN head, 256 examples are randomly sampled for each mini-batch; a ratio of 1:1 is maintained unless the number of positive samples are less than 128. Anchors with IoU greater than 0.7 are selected as positive examples, whereas anchors with IoU lower than 0.3 are selected as negative examples (the remainder are ignored during training). However, if no anchors satisfy the threshold on overlap, the anchor with the highest degree of overlap is chosen to ensure that at least one positive example is available for each annotated bounding box in the ground truth. For the classifier head, a mini-batch size of four is used, again balanced 1:1 between positive and negative examples. Positive examples are selected based on IoU greater than 0.5, whereas negative samples are selected among those with IoU between 0.1 and 0.5, i.e., among partially overlapping boxes: this hard-negative mining strategy ensures a faster convergence as negative boxes at the boundary of objects are usually harder to classify than those composed of sole background (Girshick, 2015).

Finally, the IoU threshold on the NMS is chosen experimentally to ensure optimal performance. The maximum number of candidate proposals after NMS is set to 300. It is worth noticing that this choice affects only the inference stage, and during the training stage, the NMS threshold is kept fixed at its default value. Based on ample experimental evidence of the success of transfer learning across tasks and domains (Yosinski, 2014; Zhao, 2019; Ren, 2015; Morra, 2019), the backbone is initialized starting from a pre-trained model on ImageNet (Russakovsky, 2015), whereas the two heads are randomly initialized. All layers are fine-tuned during training to account for the difference between RGB images and thermal images; we experimentally observed that fine-tuning did not lead to significant overfitting. Grayscale infrared images are converted to RGB by replicating the image on the three channels and are zero-centered by subtracting the mean value calculated over all the frames in the training set.

2.3 Tracker

A tracking algorithm exploits the strong correlation between subsequent frames to determine the position of an object at time *t*, given its position at time *t*–1. We experimented with different established tracking algorithms available in OpenCV 3.4.3 and found experimentally that KCF had the best performance (Henriques, 2014).

KCF belongs to the class of tracking-by-detection algorithms, which rely on training a classifier in an online fashion to predict the presence or absence of the target in any given image patch. At each frame, the tracker considers all possible shifts of a window of the same size as the target (dense sampling) and uses the classifier to predict the presence or absence of the target in each window. The window with the highest response, i.e., the highest probability of containing the target, determines the output position in the current frame. The classifier is continuously updated based on the appearance of the target, which may change shape or appearance during time. It is worth noticing that, in our specific case, the target is rigid, hence its shape is not expected to vary substantially (as opposed to what happens, e.g., with people tracking), but its appearance will nonetheless change due to temperature variations.

A challenging factor in the tracking-by-detection framework is the number of negative windows (e.g., windows that do not contain the target), which are virtually unlimited at each frame; to cope with the low computation demands imposed by real-time operations, many trackers resort to under-sampling, which however may negatively impact the discriminative capabilities of the classifier. The key intuition behind the KCF algorithm is that the feature maps of several translated patches can be stored effectively in the Fourier domain, thus vastly reducing memory and computational requirements for both linear and non-linear filtering. Further details can be found in the original paper by Henriques et al. (2014).

2.4 Experimental study and data set description

Freeze-drying tests were performed in a laboratory-scale equipment (LyoBeta 25™ Telstar, Spain) having a chamber volume of 0.2 m^3, a total shelf area of 0.5 m^2, and a condenser capacity of 40 kg. A total of 18 drying cycles were performed and, in this work, only the images corresponding to the primary drying phase were used. In each run, ten vials (ISO 8362-1) were filled with 5 ml of either a sucrose or a mannitol solution and placed in a row at 30 cm from the camera sensor. In all the tests, a shelf-ramped freezing protocol of seven hours in total was applied; the temperature of the cooling medium running inside the shelves was set to –50 °C, and the product cooled as quick as possible until the equilibrium was reached.

Thermal images of the primary drying phase were obtained every 300 seconds using the sensor and the imaging setup exploited in Lietta et al. (2019), as illustrated in Fig. 3A. The sensor basically consists of an IR (FLIR A35) and a RGB (HDTV 720p) camera housed, together with all the electronics required for managing data acquisition and storage as well as Wi-Fi communication with the exterior, inside a plastic case. The sensor can be placed inside the drying chamber as the case makes it resistant to the harsh conditions typical of the primary drying phase, namely low temperature, low pressure and high moisture content. The thermal images are 320×256 pixels, as shown in Fig. 3B. Chemicals were purchased from Sigma Aldrich (purity $\geq$ 99.5%) and used as received.

Since the image appearance is affected by the operating conditions, with the aim to prevent the network from learning features specific to certain operating conditions the nine experimental batches used to train the model were obtained according to a factorial design where four variables were tested: (i) the kind of vial (10R and 4R), (ii) the temperature of the fluid flowing inside the shelves (–30°C or –10°C), (iii) the pressure in the drying chamber (10 Pa or 20 Pa), and (iv) the amount of solid in the liquid solution (5% or 10% in mass). The testing dataset is composed by nine other drying experiments, five of them carried out in the same conditions (–20°C, 20 Pa, 10% b.w. solid fraction in a 10R vial), while three more batches were used to simulate possible faults that might occur. For examples, in batch #12, different faults in the filling step were simulated, batch #18 mimicked an error in the pressure control loop, and batch #19 an error in the temperature control of the shelves. In batch #15, the vibration of the equipment during the creation of the vacuum in the chamber forced one of the vials out of the camera field of view, whereas in batch #14 a vial felt off the shelf after one hour, as shown in Fig. 3B. The combination of tracking and object detection can correctly handle these abnormal behaviours.

The whole training set is made up of 7,981 images, whereas the validation set includes 7,201 images. Table 1 resumes the characteristics of the experimental tests performed, the corresponding number of images obtained for each one of them, and whether they were used to either train or validate the algorithm.

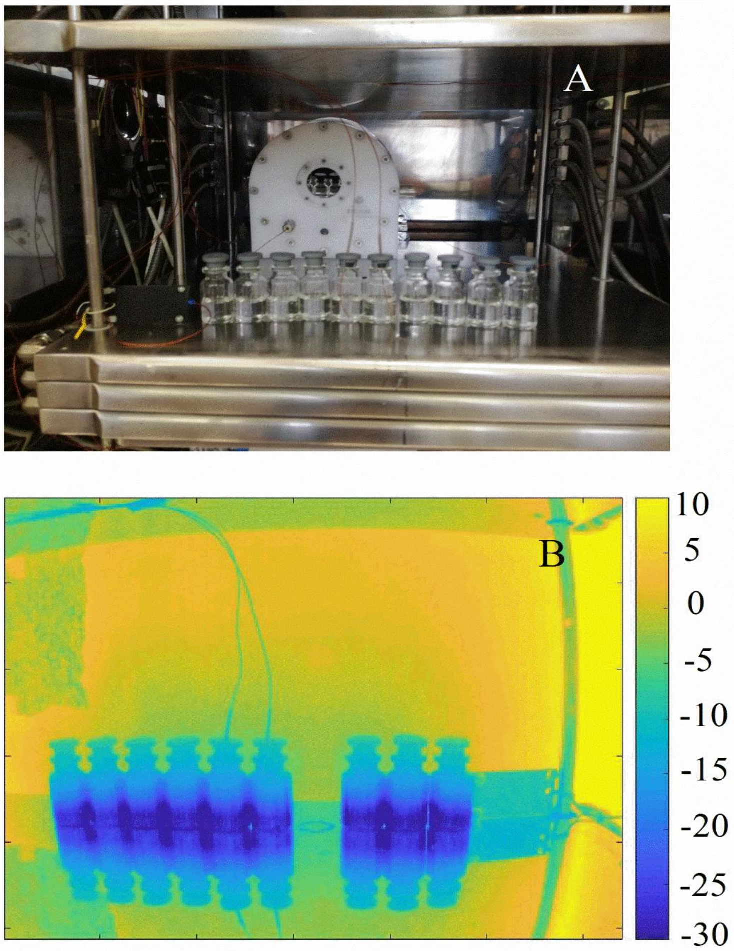

Fig. 3. Experimental setup (A) and example of thermal images obtained from the camera sensor (B).

The Visual Object Tagging Tool (VoTT)[1] was used for annotating each vial in all the frames of the training and validation sets; an example of annotation is illustrated in Fig. 4. The VoTT tools allows to partially automate the annotation process by including a tracking algorithm; the annotations are verified and manually corrected as needed.

Table 1. Details of the experimental conditions for each test, size of the resulting dataset (number of images), and applied classification as either training or validation set.

[1] https://github.com/Microsoft/VoTT#vott-visual-object-tagging-tool-15

Batch number	**Training / validation set**	**Operating conditions**				**Notes**	**Number of images**
		T, K	P_c, P_a	*Vial*	*Solid fraction*		
1	Training	-30	10	4R	5	-	1030
2	Training	-10	10	10R	5	-	751
3	Training	-30	20	10R	5	-	748
4	Training	-10	20	4R	5	-	852
5	Training	-30	10	10R	10	-	761
6	Training	-30	10	4R	10	-	766
7	Training	-10	10	4R	10	-	759
8	Training	-30	20	4R	10	-	746
9	Training	-10	20	10R	10	-	802
10	Training	-10	20	4R	5	-	766
11	Training	Various	Various	Various	Various	Data augmentation techniques	2643
12	Validation	-20	20	10R	-	4 vials NOC 2 vials piece of glass 1 vial 5% solution 1 vial water 1 vial 2.5 ml 1 vial 7.5 ml	750
13	Validation	-20	20	10R	10	-	1059
14	Validation	-20	20	10R	10	1 vial falls after 1hour from the beginning of the process	813

15	Validation	-20	20	10R	10	1 vial moves outside the field of view	782
16	Validation	-20	20	10R	10	-	733
17	Validation	-20	20	10R	10	-	787
18	Validation	-20	20	10R	10	Product boils due to pressure increase after 5 hours	747
19	Validation	-20	20	10R	10	-	767
20	Validation	-10	20	10R	10	-	763

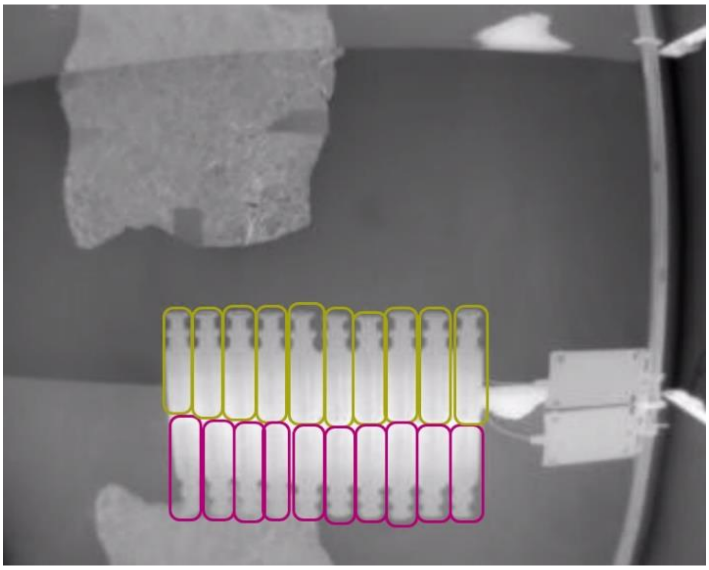

Fig. 4. Example of annotated frame: yellow boxes are vials; magenta boxes are reflections.

2.5 Data augmentation

Successfully training an object detector network requires to curate thousands of diverse images with varied background and viewpoints. However, in industrial applications it is difficult or expensive to acquire data representative of all the possible configurations. In our experiments, we found that the trained object detector may encounter difficulties when new acquisition setups are used. Such an example is given in Fig. 5A, which shows an acquisition with multiple rows of vials, whereas in most of the acquisitions only one row was present. To make the detection process robust to different acquisition setups, we used data augmentation to generate more complex background frames. Dwibedi et al (2017) showed that object detectors like Faster R-CNNs are more sensitive to local than global features, and introduced a data augmentation technique called Cut, Paste and Learn where object images are artificially inserted in background images. This simple, yet effective technique ensures that the network learns to distinguish the local properties of the object from those of the background, which in our case is often relatively constant and may be inadvertently "used" by the network to predict the presence of an object. We thus exploited a similar technique to generate synthetic sequences: we segment the vials from selected frames (Fig. 5B) and insert them in random positions in other frames (Figs. 5C and 5D). This way, 2,643 new frames were generated from all the experiments previously presented with random characteristics in terms of operating conditions and added to the training set.

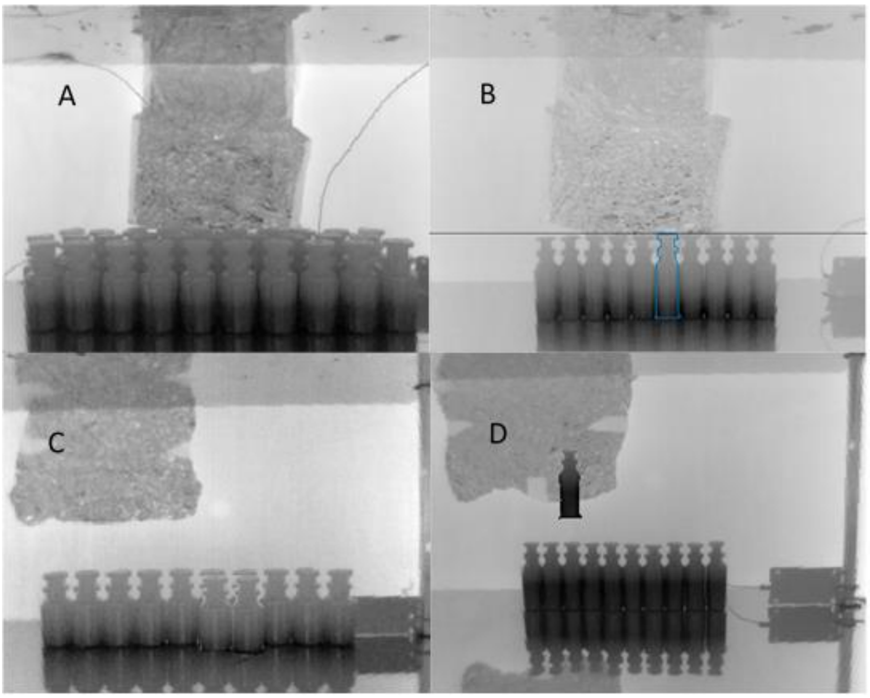

Fig. 5. Examples of experimental setup with more rows of vials (A), of manually segmented vials (B) , and of artificial frames generated through data augmentation (C-D).

2.6 Implementation and training

The network was implemented in Python 2.7 (Anaconda distribution) using Keras 2.2.4

with TensorFlow 1.10.0 backend and CUDA 8.0. Training was performed on an Amazon Web Service (AWS) EC2 px2.large instance with a NVIDIA Tesla K80 Accelerator; the instance is equipped with 1 NVIDIA GK210 GPU (12 GB memory and 2,496 CUDA cores), 4 vCPU, and 61 GB RAM. For training, we used the AWS supplied Keras Anaconda environment. For the KCF tracker, the implementation provided by the OpenCV 3.4.3 library was used.

Two experiments were performed, using only real data (Real), or real plus synthetic data (Real+Syn). In both cases, the network was trained for 24 epochs, where each epoch corresponds to cycling over the entire dataset (7,981 iterations for Exp1 and 11,707 iterations for Exp2). The Adam optimizer was used (Kingma, 2014), with initial learning rate equal to 10^{-5}; Adam adapts the individual learning rate to each parameter and was shown to converge faster than standard stochastic gradient descent, as well as less sensitive to the initial choice of the learning rate. The learning rate is reduced to 10^{-6} when the loss starts to plateau. Details of the learning rate schedule and training duration for each training experiments are reported in Table 2.

Table 2. Details of the training process of the Faster R-CNN based object detector for the two training experiments, using Real and Real+Syn datasets. The number of epochs was equal for the two experiments, where one epoch corresponds to one cycle over the entire training set. The learning rate is manually reduced when the loss starts to plateau.

Parameter	**Exp1: Real**	**Exp2: Real + Syn**
Learning rate schedule	Epoch 1 to 18: 10^{-5} Epoch 19 to 22: 10^{-6} Epoch 23 to 24: 10^{-7}	Epoch 1 to 17: 10^{-5} Epoch 18 to 24: 10^{-6}
Number of iterations	7981	11707
Number of epochs	24	24

2.7 Evaluation

For evaluating the object detector, we resorted to standard protocols employing the Average Precision (AP) metric, which is the area under the Precision-Recall curve (Ren, 2015). The recall is defined as the fraction of correctly identified objects, or the probability of correctly identifying an object. The precision represents instead the percentage of true positive detections, or the probability that the output of the detector represents a true object.

Each predicted bounding box is matched with the ground truth to determine whether it is a true positive or a false positive; conversely, each box in the ground truth is either detected or not detected depending on whether the matching is successful or not. The matching is commonly based on the IoU, setting the threshold to 0.6; usual values in the literature range from 0.5 to 0.7 (Ren, 2015). The IoU is defined as the area over the intersection of two bounding boxes over the area of the union of the bounding boxes; it varies between 0 (disjoint bounding boxes) and 1 (perfectly matching bounding boxes). In our case, since precise location is desirable, we further investigated the performance with respect to this threshold, as well as to the one used by NMS. It is worth noticing that the IoU criterion is used both for NMS (i.e., to detect overlapping bounding boxes) and during performance evaluation (i.e., to determine whether a ground truth bounding box is detected or not); the reader should bear in mind that two different thresholds are used for NMS and performance evaluation, and the results are analyzed with respect to each threshold separately. .

Additionally, we investigated the overall algorithm performance by verifying the quality of the temperature profile extracted; in other words, we aimed to establish whether imprecisions in the bounding box location affect the extraction of the temperature features, which is the ultimate goal. In the absence of a ground truth, this is achieved by comparing the extracted temperatures with the result of a semi-automatic segmentation algorithm (Colucci et al., 2019b), which was previously validated for the real-time monitoring of the same VFD process. Mean and standard deviation of the temperature are measured for the product contained in each vial to characterize the temperature profile.

3 Results

3.1 Object detector performance

The performance of the detector (in terms of recall, precision and AP) as a function of the IoU threshold is reported in Fig. 6. Using real and synthetic data yields higher performance, and especially higher recall, than using real data alone, showing the effectiveness of data augmentation. In the remainder of the experiments, we set the threshold for the matching IoU to 0.6. At this threshold, the detector achieves a recall of 99.6%, correctly identifying all the vials, and a specificity of 99.6%, meaning that the predicted bounding boxes are almost always actual vials.

Fig. 7 shows the effect of tuning the strictness of NMS on performance. At the right end of the curve, where IoU = 1, NMS is not performed at all (except for removing exact duplicate bounding boxes).

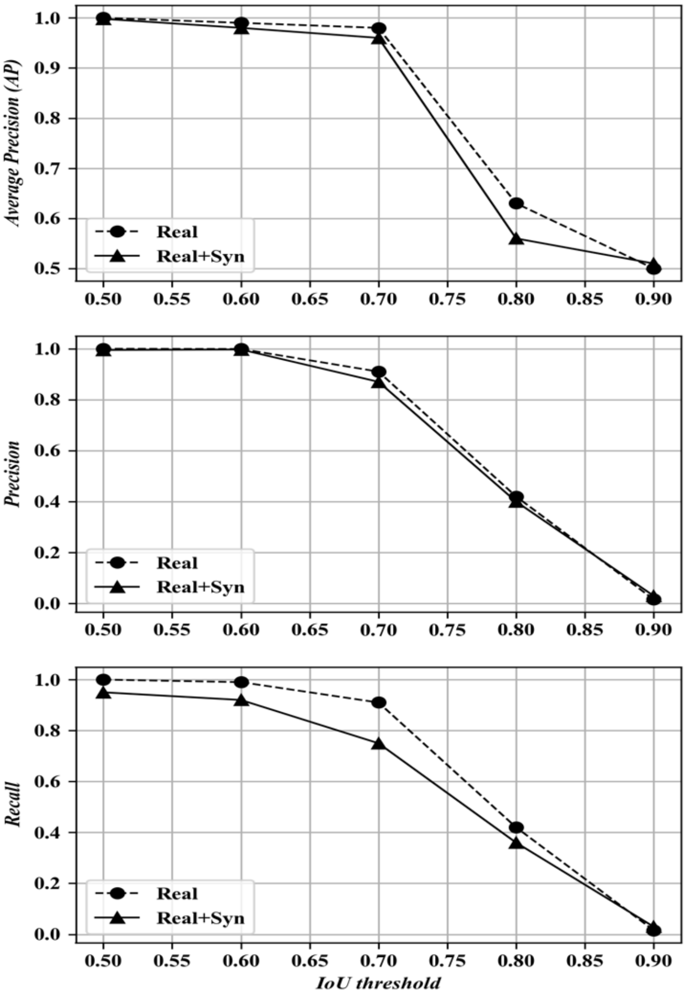

Fig. 6. Object detector performance: AP (top), precision (middle) and recall (bottom row) as a function of the IoU threshold.

As we move to the left, the IoU parameter decreases, and we begin removing detections with decreasing overlap and substituting them with the higher confidence detections. We can see that both the extremes, either lenient or strict, degrade the detector performance. By lowering the limit, NMS theoretically should, at some point, start removing overlapping true positive detections and lowering AP. In theory, a looser NMS will lead to a decline in recall, since detections with tight bounding boxes are more likely to be merged with less accurate detections, whereas a very strict NMS results in a lower precision, as many of the absorbed bounding boxes would be flagged as false positives. This behavior is, in fact, observed in both the experiments. The network trained on the Real+Syn dataset is the most robust, maintaining almost invariant the recall and then rapidly dropping when the threshold falls below 0.2. Based on these results, a threshold of 0.5 was selected as the final optimal NMS value, but for the Real+Syn experiment the range 0.2 - 0.6 would still result in optimal performance.

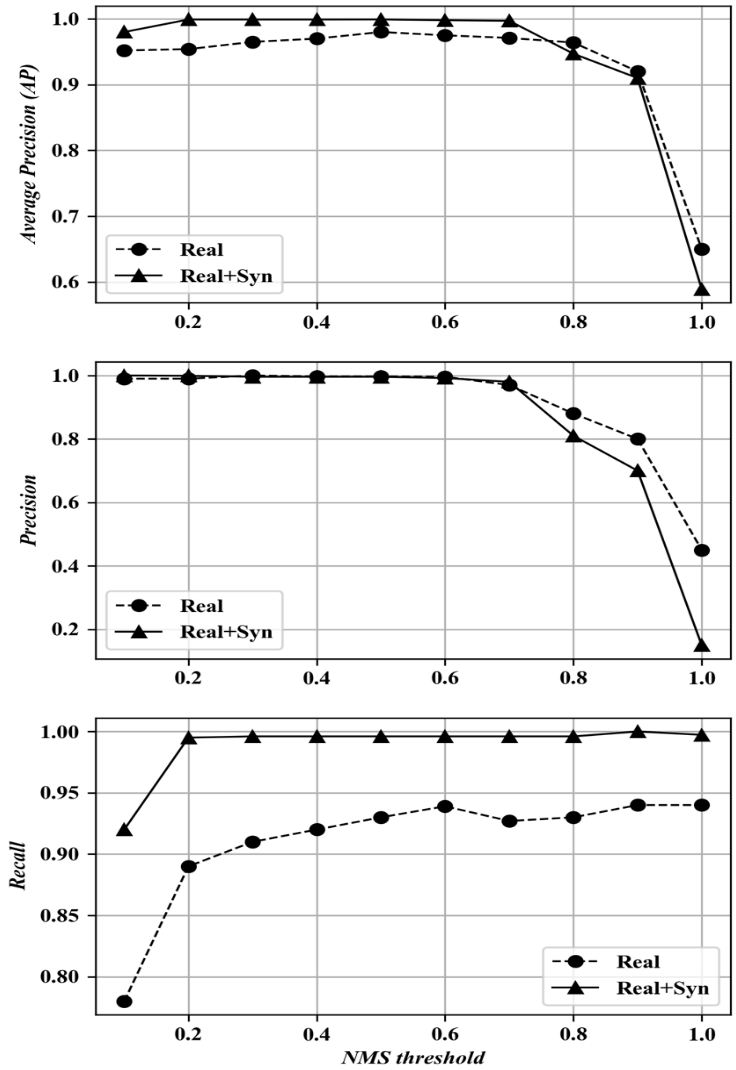

Fig. 7. Object detector performance: AP (top), precision (middle) and recall (bottom row) as a function of the IoU threshold used for NMS.

The mean IoU between the ground truth and the predicted bounding box was 0.78. The localization may be further improved: in fact, if we required an IoU of at least 0.8 to declare a true detection, the performance would drop significantly. However, as it will be shown in the following, the final localization performance is sufficient for the purpose of temperature measurement. State-of-the-art deep learning-based detectors can quickly and reliably detect objects in many scenarios but may still encounter difficulties in precisely locating very small (Eggert, 2017) or densely packed objects (Goldman, 2019; Wang, 2018). Goldman et al. (2019) observed that in densely packed scenes, another state-of-the-art deep learning architecture (RetinaNet) often returns bounding boxes that partially overlap multiple objects or detect parts of adjacent objects as separate objects. In our case, we did not observe such degradation in performance, possibly due to the different architecture; for instance, Faster R-CNN is a two-stage detector, whereas RetinaNet is a one-stage detector, which may be inherently less robust. Another possible explanation is that our dataset is simpler than that considered by Goldman et al. (2019), i.e., the number of objects per scene may be lower. Nonetheless, alternative loss functions, as in Wang et al. (2018), or more sophisticated merging schemes than NMS could be considered to further improve the localization performance.

We further analyzed the output of the detector to understand underlying causes for the occasional failures. First of all, our validation set includes sequences with common problems that may arise during the monitoring process, e.g., vials that fall from the shelves or move out of the field of view due to vibrations in the equipment (events that occurred while preparing or running the simulations are reported in the Notes column of Table 1). These occurrences prevent simple solutions, such as using fixed measurement points, or even image-based processing solutions, such as the one previously presented in (Colucci, 2019b), to work reliably without manual adjustments. However, we found that all vials were correctly detected in all the affected sequences (#12, #14, #15 and #18); an example can be seen in Fig. 8, where a vial falls off the shelf and the event is readily detected.

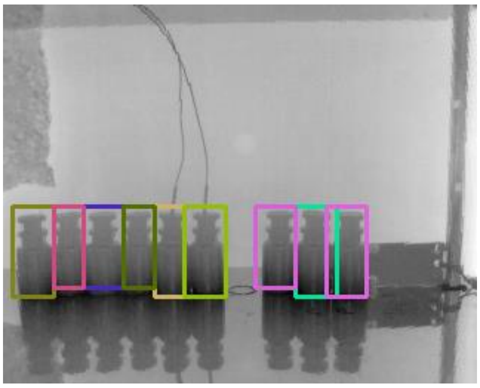

Fig. 8. Sample frame from validation sequence #14, where a vial falls off the shelf and the event is readily detected. In this case, the temperature will not be measured in subsequent frames.

False negatives were mostly due to the presence of extreme temperature variations, similarly to what happens in the RBG domain in the case of varying illumination conditions. Fig. 9 shows three sample frames from another validation sequence, where the detection fails in the very first frame, and a false positive is observed in frame 10. Occasionally, some false positives are detected, which however appear only on one or two frames and, hence, can be discarded easily during tracking. Temperature extremes are usually found in the first or final frames, thus under-represented in the training set; we believe that these failures will be resolved in the future when more data samples will become available.

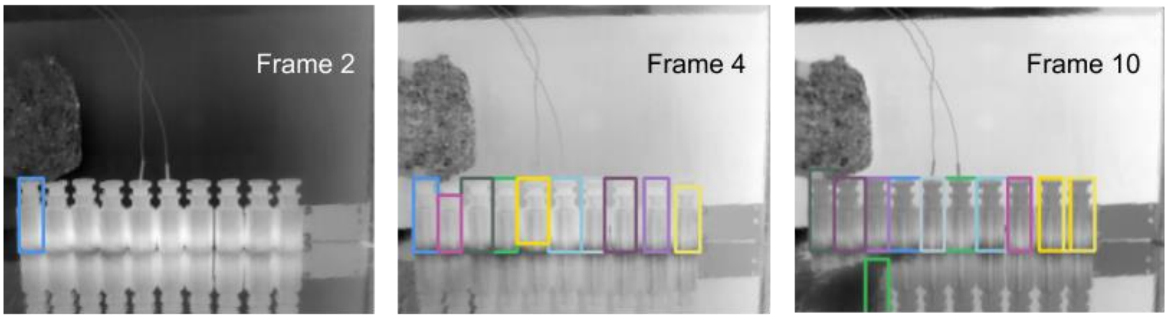

Fig. 9. Example of early frames extracted from a video in the validation set. The performance of the detector suffers in the first frame, where the temperature drops substantially, but is quickly recovered in the subsequent frames.

3.2 Overall algorithm performance

The presented algorithm aims to track and segment the regions corresponding to the product into each one of the single vials in order to obtain information about the evolution of the process.

The temperature-related features are extracted from the bottom part of the vial (40% of the total height), after excluding the external borders (left and right 10% of the width). An example of the features that could be extracted is the characterization of the thermal evolution of the process over time in terms of mean, standard deviation (std), skewness and kurtosis of the temperature distribution inside each segmented bounding box, as in Colucci et al. (2019b). Fig. 10 compares the average of the mean and standard deviation of the temperatures extracted from each one of the processed vials of batch #19. This batch was one of those where the intensity-based algorithm of Colucci et al. (2019b) performed better and, since the previous algorithm was based on the precise position and size the vials, it can be considered as a gold standard in the absence of the true measurement. The solid line is the value obtained with the algorithm presented in Colucci et al. (2019b), whereas the symbols refer to the estimation obtained with the approach discussed in this work. When the average mean temperature inside the product is compared (Fig. 10A), the two algorithms are perfectly equivalent. The average difference of the two readings over the 60 hours of the process is 0.4°C, well below the precision of the sensor, thus hardly affecting the quality of the information extracted.

In Fig. 10B, the average standard deviation measured with the approach presented in this work is always slightly underestimated. Nevertheless, the average error over the whole run is mild, i.e., 0.21°C, again hardly able to affect the quality of the monitoring or control scheme. The mild discrepancy might be due to a difference of a few pixels in the two segmented boxes.

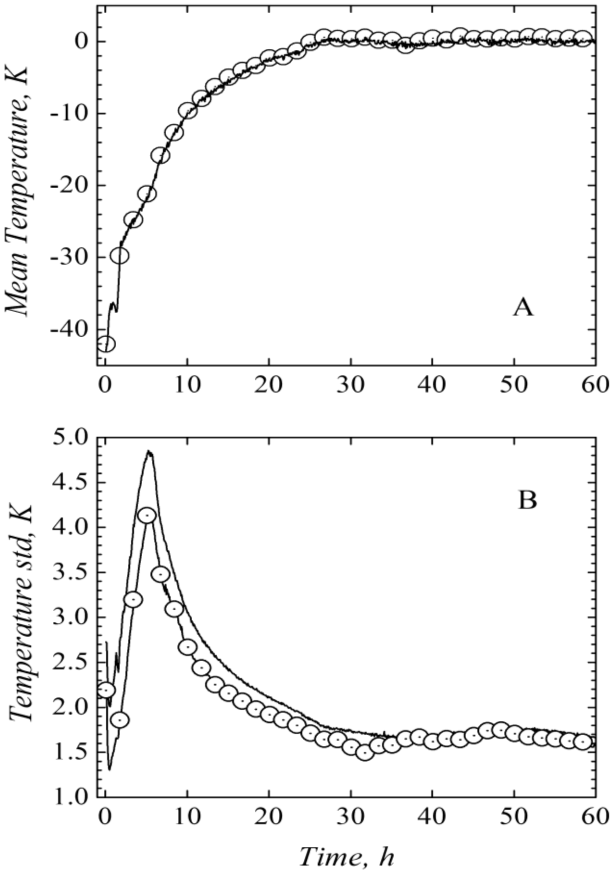

Fig. 10. Comparison of the features extracted from the computer vision-based algorithm developed in this work (solid line) and from the algorithm presented in Colucci et al. 2019 (symbols). For each of the 10 vials, the mean and standard deviation of the temperature is calculated, which are then averaged over the entire batch. Average of the ten mean temperature profiles obtained from each segmented vial (A), and average of the ten standard deviations obtained from the temperature characterization of each segmented vial (B).

4 Conclusions

In this work, the problem of object detection and tracking in the framework of the pre-treatment of the thermal images provided by an infrared camera during the real-time monitoring of a VFD process has been addressed.

Traditional intensity-based image analysis techniques are not reliable enough since the distribution of the intensity, that is the temperature measured in the single pixels, strongly varies over time, together with the gradients and edge features typically exploited by those algorithms. A computer vision algorithm including a Faster R-CNN object detector and a KCF tracker was developed, trained and tested over different sets of operating conditions. A comparison with a previous algorithm presented by Colucci et al. (2019b) showed that the two approaches are almost equivalent in terms of quality of the features extracted. The average error observed between the two measurements was in fact lower than the temperature difference detectable by the sensor.

The proposed detector achieved a good level of performance with almost perfect recall and precision (99.6%) and, when coupled with the tracking algorithm, it was able to detect 100% of the vials.

The algorithm was trained on nine videos including 7,201 images, corresponding to more than 70,000 annotated examples of vials, acquired under different experimental conditions. The cost of annotation was however reduced by using a semi-manual annotation tool which incorporates tracking. Synthetic sequences based on a cut-and-paste data augmentation technique were used to ensure robust and effective object detection. Future work will be devoted to understanding if the data required to train the system could be further reduced, e.g., by using human-in-the-loop training algorithms. Furthermore, it is reasonable to assume that fine-tuning the system to recognize other objects within the thermal domain would need less examples than in our experiments, as the network was pre-trained in the RGB domain.

Precise localization is needed to extract accurate temperature features. In our case, the localization precision is determined by the accuracy of the bounding box, assuming that the position of the object is up-right (i.e., not rotated with respect to the image borders). Further enhancements could include predicting the orientation of the bounding box (Liu, 2017) or the segmentation mask (He, 2017), which would facilitate feature extraction for objects with more irregular shapes or different orientations.

An algorithm like the one presented in this work would be very important in the application of infrared measurement to the monitoring and control of a continuous freeze-drying equipment. The proposed technique is fully automatic, can be run in real time, and is able to track objects moving in and out of the camera sensor field of view. While in our experimental setup vials movement was due to the imperfect laboratory equipment, in a real-life process the relative motion of the vials with respect to the sensor would be an avoidable feature of the system, and robust computer vision techniques are needed for effective monitoring.

The goal of the proposed technique is to identify regions of interest to extract features, such as mean, standard deviation or kurtosis, that are then feed to a process control loop. For instance, in previous work by Colucci et al. (2019b) the features were supplied to a Principal Component Analysis (PCA) algorithm used to perform MSPC. One of the aspects to be deepened concerns verifying whether the features used by the DCNN for object detection could serve also this purpose. In this way, the feature extraction step could be skipped, leveraging those already extracted during object detection, thus further optimizing the computational cost of the whole MSPC algorithm. The DCNN features are learnt from data, and, hence, could provide complementary information to standard feature extraction algorithms.

Finally, the proposed methodology could be applied to other phases of the process, e.g., the freezing step, given the major importance of this step in the economy of the whole freeze-drying process. However, since the freezing step is characterized by fast kinetics, it would require a sampling rate in the order of seconds, not minutes, as in our experimental setup. Hence, further optimization of the proposed pipeline would be required to comply with real-time computational requirements. This may entail, for instance, designing and training a lightweight backbone with lower computational requirements.

References

Bald, W.B., 1991. Ice crystal growth in idealised freezing system, in Bald, W.B. (Ed.), Food Freezing, 1st edition. Springer-Verlag, London, UK, 67–80 (Chapter 5).

Bosca, S., Corbellini, S., Barresi, A. A., Fissore D., 2013. Freeze-drying monitoring using a new process analytical technology: Toward a “zero defect” process. Drying Technol. 31(15), 1744–1755. https://doi.org/10.1080/07373937.2013.807431

Bahaghighat, M., Mirfattahi, M., Akbari, L. and Babaie, M., 2018, Designing quality control system based on vision inspection in pharmaceutical product lines. In *2018 International*

Conference on Computing, Mathematics and Engineering Technologies (iCoMET) (pp. 1-4). IEEE.

Canziani, A., Paszke, A. and Culurciello, E., 2016. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*.

Carlucci, F.M., Russo, P. and Caputo, B., 2018. ^{2}CO: Deep Depth Colorization. *IEEE Robotics and Automation Letters*, *3*(3), pp.2386-2393.Coffey, V.C., 2018. Machine Vision: The Eyes of Industry 4.0. Opt. Photonics News. 29(7), 42-49.

Colucci, D., Maniaci, R., Fissore, D., 2019a. Monitoring of the freezing stage in a freeze-drying process using IR thermography. Int. J. Pharm. 566, 488-499. https://doi.org/10.1016/j.ijpharm.2019.06.005

Colucci, D., Prats-Montalbán, J. M., Fissore, D., Ferrer, A., 2019b. Application of multivariate image analysis for on-line monitoring of a freeze-drying process for pharmaceutical products in vials. Chemom. Intell. Lab. Syst. 187, 19-27.

Colucci, D., Prats-Montalbán, J. M., Fissore, D., Ferrer, A., 2019c. On-line product quality and process failure monitoring in freeze-drying of pharmaceutical products. Drying Technol. 566, 488-499. (DOI: 10.1080/07373937.2019.1614949)

Dias, P.A., Tabb, A. and Medeiros, H., 2018. Apple flower detection using deep convolutional networks. Comput. Ind. 99,17-28.

Dwibedi, D., Misra, I. and Hebert, M., 2017. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1301-1310).

Fissore, D., Pisano, R., Barresi A.A., 2018. Process analytical technology for monitoring pharmaceuticals freeze-drying – A comprehensive review. Drying Technol. 36(15), 1839-1865. https://doi.org/10.1080/07373937.2018.1440590

Eggert, C., Brehm, S., Winschel, A., Zecha, D., & Lienhart, R. 2017. A closer look: Small object detection in faster R-CNN. In *2017 IEEE international conference on multimedia and expo (ICME)* (pp. 421-426).

Emteborg, H., Zeleny, R., Charoud-Got, J., Martos, G., Luddeke, J., Schellin, H., Teipel K., 2014. Infrared thermography for monitoring of freeze-drying processes: Instrumental developements and preliminary results. J. Pharm. Sci. 103(7), 2088-2097. https://doi.org/10.1002/jps.24017

Fissore, D., 2013. Freeze drying of pharmaceuticals, in Swarbrick, J. (Ed.), Encyclopedia of Pharmaceutical Science and Technology, 4th edition, volume III. Taylor and Francis, New York, pp. 1723-1737. https://doi.org/10.1081/E-EPT4-120050278

Girshick, R., 2015. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).

Goldman, E., Herzig, R., Eisenschtat, A., Goldberger, J., & Hassner, T. 2019. Precise Detection in Densely Packed Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5227-5236).

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* (pp. 770-778).

He, K., Gkioxari, G., Dollár, P., & Girshick, R. 2017. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961-2969).

Henriques, J. F., Caseiro, R., Martins, P., Batista, J. 2014 "High-speed tracking with kernelized correlation filters." IEEE trans. pattern anal. mach. intel. 37(3), 583-596.

Huang, M., Wang, Q., Zhang, M. and Zhu, Q., 2014. Prediction of color and moisture content for vegetable soybean during drying using hyperspectral imaging technology. J. Food Eng. 128, 24-30.

Kwaśniewska, A., Rumiński, J., Czuszyński, K. and Szankin, M., 2018. Real-time Facial Features Detection from Low Resolution Thermal Images with Deep Classification Models. *Journal of Medical Imaging and Health Informatics*, *8*(5), pp.979-987.

Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. In *Proceedings of the 3^rd^ International conference on Learning Representation*, (pp. 1-15).

Kornblith, S., Shlens, J. and Le, Q.V., 2019. Do better ImageNet models transfer better?. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2661-2671).Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. In proceedings of *Advances in neural information processing systems (NIPS2012)*, pp. 1-9.

Lietta, E., Colucci, D., Distefano, G., Fissore D., 2019. On the use of infrared thermography for monitoring a vial freeze-drying process. J. Pharm. Sci. 108(1), 391-398. https://doi.org/10.1016/j.xphs.2018.07.025

Liu, C., Liu, W., Lu, X., Chen, W., Yang, J. and Zheng, L., 2016. Potential of multispectral imaging for real-time determination of color changes and moisture distribution in carrot slices during hot air dehydration. Food chem. 195, 110-116.

Liu, L., Pan, Z. and Lei, B., 2017. Learning a rotation invariant detector with rotatable bounding box. *arXiv preprint arXiv:1711.09405*.

Mellor, J.D., 1978. Fundamentals of freeze-drying. Academic Press, London.

Morra, L., Delsanto, S. and Correale, L., 2019. *Artificial Intelligence in Medical Imaging: From Theory to Clinical Practice*. CRC Press.Nakagawa, K., Hottot, A., Vessot, S., Andrieu J., 2007. Modeling of freezing step during freeze-drying of drugs in vials. AIChE J. 53(5), 1362-1372. https://doi.org/10.1002/aic.11147

Oetjen, G.W., Haseley P., 2004. Freeze-drying, 2nd edition, Wiley-VHC, Weinheim.

Oquab, M., Bottou, L., Laptev, I. and Sivic, J., 2014. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1717-1724).

Pikal, M. J., Shah, S., Roy, M. L., Putman, R., 1990. The Secondary Drying Stage of Freeze Drying: drying Kinetics as a Function of Temperature and Chamber Pressure. Int. J. Pharm. 60, 203–207. DOI: 10.1016/0378-5173(90)90074-E.

Raponi, F., Moscetti, R., Monarca, D., Colantoni, A. and Massantini, R., 2017. Monitoring and optimization of the process of drying fruits and vegetables using computer vision: a review. Sustainability 9(11), 2009.

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

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Kharpathy, A., Khosla, A., Bernstein, M., Berg, A., C., Fei-Fei, L., 2015. Imagenet large scale visual recognition challenge. *Int. J. Comp. Vis.* 115(3), 211-252.

Shao, S., McAleer, S., Yan, R. and Baldi, P., 2018. Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning. *IEEE Transactions on Industrial Informatics*, *15*(4), pp.2446-2455.

Shao, S., Wang, P. and Yan, R., 2019. Generative adversarial networks for data augmentation in machine fault diagnosis. *Computers in Industry, 106*, pp.85-93.

Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D. and Summers, R.M., 2016. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, *35*(5), pp.1285-1298.

Van Bockstal, P.J., Corver, J., De Meyer, L., Vervaet, C., De Beer, T., 2018. Thermal imaging as a noncontact inline process analytical tool for product temperature monitoring during continuous freeze-drying of unit doses. Anal. Chem. 90(22), 13591-13599. https://doi.org/10.1021/acs.analchem.8b03788

Velardi, S.A., Barresi, A.A., 2008. Development of simplified models for the freeze-drying process and investigation of the optimal operating conditions. Chem. Eng. Res. Des. 87(1), 9-22. https://doi.org/10.1016/j.cherd.2007.10.007

Wang, X., Xiao, T., Jiang, Y., Shao, S., Sun, J., & Shen, C. 2018. Repulsion loss: Detecting pedestrians in a crowd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7774-7783).

Yan R., Chen X., Wang P. and Onchis D.M., 2019. Deep learning for fault diagnosis and prognosis in manufacturing systems, *Computers in Industry* 110:1-2. https://doi.org/10.1016/j.compind.2019.05.002

Yosinski, J., Clune, J., Bengio, Y. and Lipson, H., 2014. How transferable are features in deep neural networks?. In *Advances in neural information processing systems* (pp. 3320-3328).

Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X., 2019. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*.