Applied Thermal Engineering 256 (2024) 124040

Contents lists available at ScienceDirect

Applied Thermal Engineering

journal homepage: www.elsevier.com/locate/apthermeng

Research Paper

Image-based quantification of pool boiling heat flux on varied heating surfaces: Enhancing prediction performance with automated machine learning

Ruan C. Comelli, Alexandre K. da Silva *

Department of Mechanical Engineering, Federal University of Santa Catarina, Florianópolis, SC, Brazil

ARTICLE INFO

Keywords:
Machine learning
Automated machine learning
Pool boiling
Multiple heat surfaces

ABSTRACT

The present study applied image-trained Convolutional Neural Networks (CNN) to quantify the heat flux dissipated in a pool boiling experiment. In total, four heating surfaces were used to generate over 200 thousand nucleate and film boiling images in two visualization modes: direct and indirect. In the former, the flow images included the heating surface; in the latter, the image was cropped and the heating surface was omitted. Prior to training and testing the CNN, the images were preprocessed, by grayscaling, downscaling, size uniformization and standardization. The main goals were to assess the CNN's capability to generalize to multiple operating conditions, and to quantify the performance of a CNN architecture optimized with Automated Machine Learning (AutoML) in comparison with a reference architecture. The results suggest that multi-dataset training is required to improve the CNN generalization. In other words, the CNN must be trained, even if partially, with images of the specific heating surface for which it is trying to infer the heat flux. However, the results indicate a significant performance drop for the results of multi-surface trained CNNs when compared with single-surface CNNs, suggesting limited generalization capability. Furthermore, the results obtained showed that AutoML was capable of increasing the performance of CNN models when compared with parametrically determined architectures. Also, optimized architectures tend to present a larger number of convolutional layers associated with dense blocks and, at the same time, a reduced number of trainable variables.

1. Introduction

Phase change processes are among the most efficient ways to transfer thermal energy. For instance, the heat transfer coefficient associated with pool boiling or convective phase change is orders of magnitude higher than traditional single-forced convection [1]. This occurs due to the amount of energy needed to vaporize liquid or to condensate the vapor phases. Furthermore, several commonly used systems, such as Rankine, compression-based refrigeration and heat pumps cycles, have at least two heat exchangers in which their respective working fluids are changing phase [2]. Therefore, needless to say that phase change is a very important heat transfer mechanism and, because of that, has historically received significant attention from the research community. However, to this day, most of the methods used to predict key parameters associated with phase change heat transfer (e.g., pool boiling critical heat flux) are based on empirically adjusted correlations, which often have considerable uncertainties associated to them.

However, differently from single-phase convective heat transfer, which also relies on classical and well-established heat transfer solutions, phase change processes are too complex to be solved from first principle constitutive equations while taking entirely into account their spatial and temporal dependence, in spite of the advancements in numerical modeling. Alternatively, and thanks to recent developments in the field of artificial intelligence, researchers are trying to overcome the current limitations and deal with phase change processes by employing machine learning (ML) techniques. More specifically, attempts have been made to develop real-time, visualization-based quantification and classification algorithms for phase change processes with very promising results. Among the main advantages of this technique is the non-intrusivity, needing, at most, a light pulse to capture the image to be analyzed, but no physical measurable parameters from the heat transfer surface or phase change fluid. For instance, Ref. [3] successfully used two distinct neural network models, i.e., Artificial Neural Network (ANN) and Convolutional Neural Network (CNN), to classify different images of R-134a condensing in a tube according to their flow regime.

* Corresponding author.
E-mail address: a.kupka@ufsc.br (A.K. da Silva).

https://doi.org/10.1016/j.applthermaleng.2024.124040
Received 4 March 2024; Received in revised form 24 July 2024; Accepted 24 July 2024
Available online 26 July 2024

Nomenclature

A	Area [m^2]
D^{fs}	Downscale operator [-]
HIST	Luminance distribution histogram [-]
q''	Heat flux [W/cm^2]
R	Electrical resistance [Ω]
S*	Relative cross-entropy [-]
T	Temperature [K]
$\hat{u}(\cdot)$	Uncertainty [-]
V	Voltage [V]
VAR*	Relative variance [-]
x	Feature vector/image [-]

Acronym

AutoML	Automated Machine Learning
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
D^{SW}	Refers to the small diameter wire dataset
D^{LW}	Refers to the large diameter wire dataset
D^{HR}	Refers to the horizontal ribbon dataset
D^{VR}	Refers to the vertical ribbon dataset
LED	Light Emitting Diode
MAE	Mean absolute error
MAPE	Mean absolute percentage error
ML	Machine learning
MSE	Mean squared error
RMSE	Root mean squared error
RTD	Resistance Temperature Detector

Also, Ref. [4] used a series of images to train CNNs to classify five two-phase flow patterns of R14 (tetrafluoromethane) and R50 (methane). Ref. [5] tested different machine learning algorithms focusing on the condensation of R134a in a vertical tube. The results showed that CNNs are capable of quantifying different flow parameters such as heat flux and void fraction non-intrusively through images.

As for boiling heat transfer, Ref. [6] proposed the use of a dataset composed of uncorrelated images, which were used to train machine learning algorithms to classify different pool boiling regimes. The results showed very high accuracy, i.e. above 98 % and 94 %, for direct and indirect visualization, respectively. Later, Refs. [7,8] employed CNNs to quantify the heat flux dissipated by a heating surface (a nichrome wire) subjected to pool boiling of water, also obtaining very low error metrics, which were on the same order of or better than some existing correlations. Ref. [9] used infrared images of different heat transfer surfaces along with artificial neural networks (ANNs) to estimate different parameters associated with a pool boiling phase change process. The results indicated that the proposed technique could accelerate the analysis process when compared with traditional image analysis techniques, while ensuring high performance. Furthermore, several other studies also focused on different aspects of boiling processes, while relying, for example, on infrared images, high-speed cameras and numerous computational techniques, e.g., Refs. [10–14].

However, and in spite of the interest of the scientific community in the applicability of image-trained machine learning algorithms in the context of boiling heat transfer, to the best of the authors' knowledge, few articles have specifically focused on generalization aspects. Ref. [15] trained an ANN with infrared data with the objective of determining the critical heat flux for four different surface coatings, while using only three surfaces for training, reaching a mean absolute percentage error (MAPE) of 96 %. Ref. [16] used unsupervised image-to-image translation-assisted CNN with the objective of predicting the critical heat flux in three distinct datasets of pool boiling images. More recently, Ref. [13] considered a dataset composed of pool boiling images produced from three distinct surfaces, which were used to train two machine learning algorithms aiming to identify the occurrence of the critical heat flux.

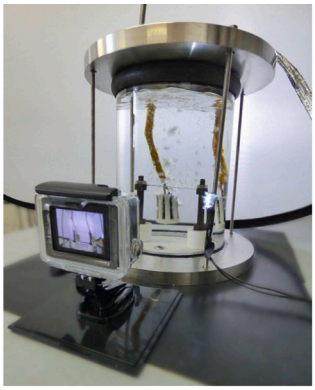
In that sense, and despite the motivating results shown by the literature, there are still limitations associated with the use of ML models in phase change systems, namely their questionable generality and relative performance. For instance, ML algorithms tend to perform poorly when tested on datasets that are different from their original training dataset. In other words, if a ML algorithm is trained with data (e.g., images) for a boiling surface under certain conditions, assuming that over time the phase change characteristics vary due to, for example, oxidation, or the relative position of the camera with respect to the boiling surface changes, the algorithm might not perform accordingly. Another aspect has to do with the optimal model architecture that would offer the lowest inference errors; often these are simply parametrically analyzed considering a few candidates, which lack a systematic exploration and potentially allow significant performance improvements.

Therefore, the present study quantifies the heat flux dissipated by different heating surfaces in pool boiling of water with CNN models trained with uncorrelated images. Two novel aspects are introduced in the present study within the non-intrusive analysis of boiling heat transfer: (i) the use of four different heating surfaces and (ii) the optimization of the CNN architecture and training hyperparameters via Automated Machine Learning (AutoML) [17,18]. The former represents possibly the first thorough analysis focusing strictly on inferring the relation between the use of different combinations of datasets for training and testing on the CNN's performance on all four surfaces. The four heating surfaces generated 98 dataset combinations: 49 cases where the heating surface is part of the image and 49 cases when the surface is suppressed (cropped out) from the image. As for the latter, the use of AutoML allows one not only to directly compare the performance of an AutoML-optimized CNN with a baseline CNN available in the literature, but also to suggest overall guidelines for architectures that perform best for this class of problems. In total, 196 cases were tested: 98 for the baseline CNN and 98 for the AutoML-optimized CNN.

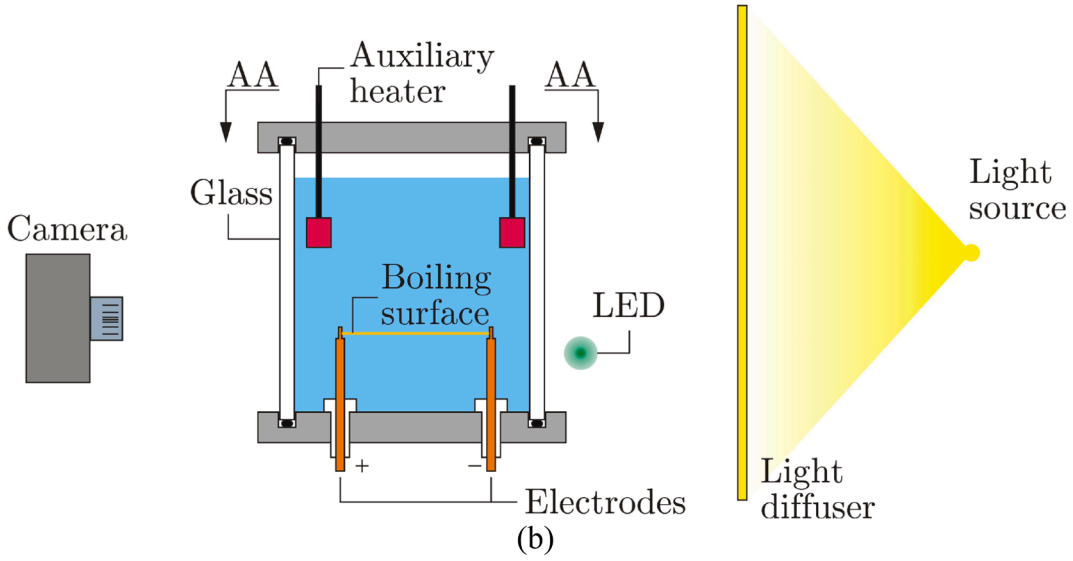
The article is organized as follows: Section 2 details the experimental setup and methods used to obtain the pool boiling images; Section 3 describes the image post-processing techniques and the training algorithm employed; Section 4 presents the validation of the training procedure; Section 5 presents the results, which are organized as single surface results (Section 5.1), multiple surfaces results and combined datasets (Section 5.2) and AutoML results (Section 5.3); the Conclusions are presented in Section 6.

2. Experimental methods

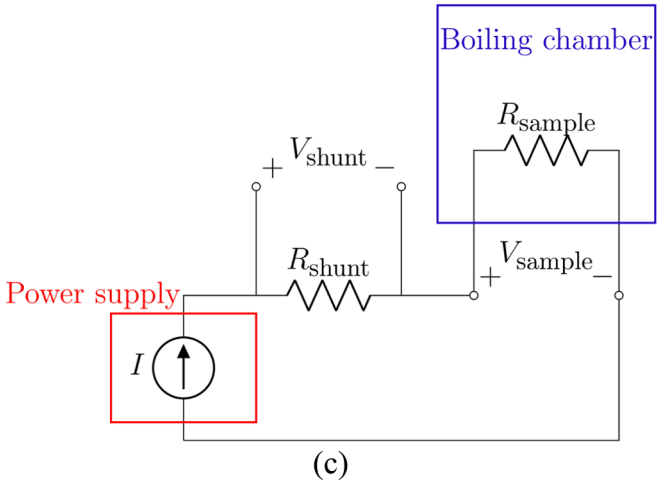
The experimental setup used to construct the datasets utilized in this work was originally developed, tested and reported in Ref. [6], and it basically mimics the well-known pool boiling experiment reported by Nukiyama (1934), apud Ref. [1]. In a few words, the experimental setup is composed of a cylindrical borosilicate glass having an internal diameter and height of 144 mm and 200 mm, respectively, which is sandwiched between two stainless steel circular disks, as can be seen in Fig. 1a. While the container is not pressurized due to an opening in the upper stainless disk, both disks have carefully machined circular grooves housing O-rings that prevent any leaking between the stainless disks and the borosilicate cylinder. The lower disk has two feed-throughs made out of Nylon, each holding a copper electrode, which are 60 mm apart



(a)



(b)



(c)

Fig. 1. (a) a picture of the experimental setup; (b) sketch of the experimental setup having the camera (left), pool boiling reservoir (middle) and the illumination system (right); (c) electric circuit that powers the heating surface.

from each other, as shown in Fig. 1b. The working fluid partially filling the reservoir is deionized water, which is kept at nearly saturation temperature by two 750 W auxiliary electric heaters; the water temperature is measured by a RTD. The heating surface responsible for boiling the water was bolted to the upper ends of the electrodes, which were threaded, by a nut. Three testing surfaces, all from Omega™, were tested in the experimental rig totaling four configurations, each detailed in Table 1. More specifically, D^{SW} and D^{LW} refer to wire heaters of small

Table 1
Geometric details and acronyms for the heating surfaces used in this work.

Surface type	Acronym	Material	Dimensions* [μm]	Position
Wire	D^{SW}	Nichrome	250 (diameter)	–
Wire	D^{LW}	Nichrome	510 (diameter)	–
Ribbon	D^{HR}	Nichrome	1590/79 (width/thickness)	Horizontal
Ribbon	D^{VR}	Nichrome	1590/79 (width/thickness)	Vertical

* All dimensions had an uncertainty of ± 0.4 μm.

and larger diameters, respectively, while D^{HR} and D^{VR} refer to two identical ribbon samples placed in a horizontal and vertical position, respectively.

The heating surface is powered by a 1,500 W power source, and the actual power dissipated by the heater is calculated by multiplying the voltage between the electrodes, which is measured, and the flowing electrical current. The current is determined by dividing the voltage drop across a shunt of known resistance, which is connected in series with the heating elements (Fig. 1c). The images used for training, validation and testing were acquired by a regular digital camera with a resolution of 1,520 × 2,704 pixels at 30 fps; this frame rate was shown to be adequate [6]. Also, and in order to help prevent the occurrence of unwanted features within the images, a backlit diffuser screen was placed behind the boiling chamber, as shown in Fig. 1b. All measurements were done with a National Instruments data acquisition system, and a custom-made code was developed in Python to read, record, store and display the data of interest [19]. At this point, it is worth mentioning that, while it might seem counter-intuitive to use a low-fps camera rather than a standard research-grade high-speed camera, the present study it is not focused on the upward vapor flow dynamics. High-speed acquisition might cause sequential images to be highly correlated, which could lead to an ill-posed scenario to machine-learning algorithms since, for example, these could be trained with a certain image and then asked to return qualitative or quantitative information of a nearly identical image taken a few milliseconds before or after. For more information please refer to Ref. [6].

The experimental procedure for each testing surface involved several steps. First, the selected heating surface was attached to the electrodes; next, the testing rig was filled with deionized water and the camera, photo diffuser and backlit screen were placed in position. Next, the auxiliary heaters were activated heating the water up to the saturation temperature. After reaching the saturation temperature, the water was allowed to boil for approximately 1 min aiming to degas it. From this point on the actual data recording process started by increasing the power dissipated by the heater surface in steps of 5 W starting at 0 W up to burnout, as shown in Fig. 2a. It is important to note that, for each power dissipation step, the data recording process included a power dissipation stabilization period followed by the activation of the LED, which timestamps the data with a voltage pulse at the instant that the power dissipation becomes stable. After that, the LED is turned off and video recording process continues under stable conditions for 1 min as can be seen in Fig. 2b; this 1-min long video is used to extract the images to train and test the machine learning models proposed in the present study. The number of videos for each heating surface varied with the critical flux. In other words, surfaces that allowed higher thermal dissipation had a larger number of 1-minute-long videos, and hence resulted in bigger datasets.

Also, because the heat flux applied to the heating surface was increased in steps of 5 W, this is actually the resolution obtained when detecting the different boiling regimes [1]. More specifically, Table 2 shows the experimental range observed for the power dissipated and the respective heat flux for the onset of nucleate boiling and the critical heat flux for all four surfaces tested. As can be seen, the experiment is only capable of determining the range where the failure/rupture of the

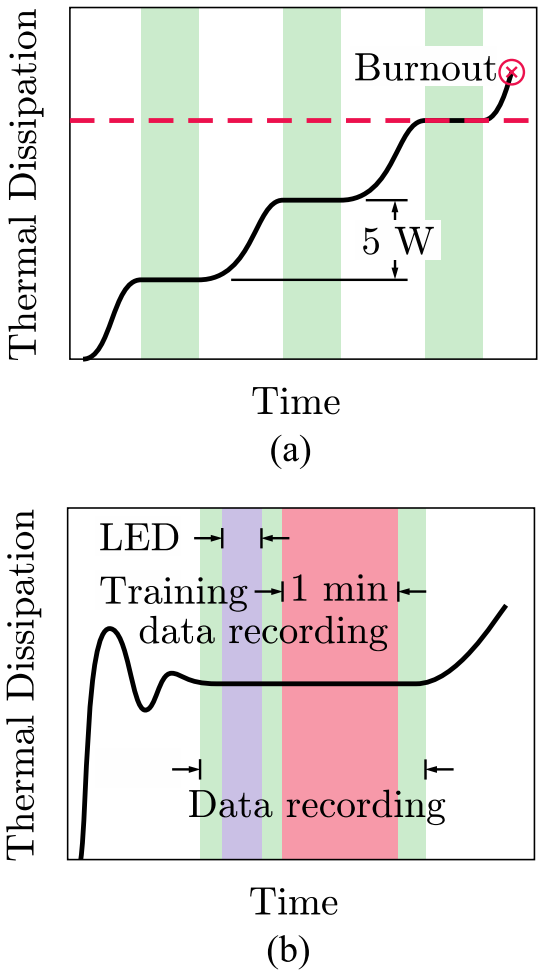


Fig. 2. Sketch representing the pool boiling process: (a) shows the heat flow increasing process in the experimental runs in steps of 5 W up to the failure of the heating surface; (b) illustrates the recording window, the LED signaling window and the 1 min-long training window, for each heat flux tested.

Table 2
Heat power dissipation and heat flux range for the surfaces tested.

Heater	Onset of Nucleate Boiling	Critical Heat Flux
D^{SW}	5–10 W (10.36 – 20.76 W/cm^2)	40–45 W (81.03 –? W/cm^2)
D^{LW}	10–15 W (9.88 – 14.38 W/cm^2)	85–90 W (86.10 –? W/cm^2)
D^{HR}	5–10 W (2.40 – 4.55 W/cm^2)	185–190 W (85.76 –? W/cm^2)
D^{VR}	10–15 W (4.78 – 6.75 W/cm^2)	170–175 W (78.60 –? W/cm^2)

heating surface occurs (i.e., critical heat flux), but its actual value is unknown; this is represented by the "?" symbol in the third column of Table 2. Furthermore, it is also worth mentioning that the actual measured heat flux varied around the corresponding nominal power input dissipated by the power supply. Therefore, the experiments cannot be modeled as a multi-classification problem, and the CNNs, which will be discussed next, are designed as regression models, containing a single, linear output.

In order to exemplify typical images obtained, Fig. 3 shows a sequence of images for each one of the four surfaces tested. More specifically, for each surface, three images are shown, corresponding to the three main boiling regimes observed: natural convection, partial nucleate boiling and fully developed nucleate boiling, where partial refers to the occurrence of independent bubbles and fully developed indicates the regime in which the individual bubbles start to merge. Also, the actual heat flux value of each frame is shown immediately

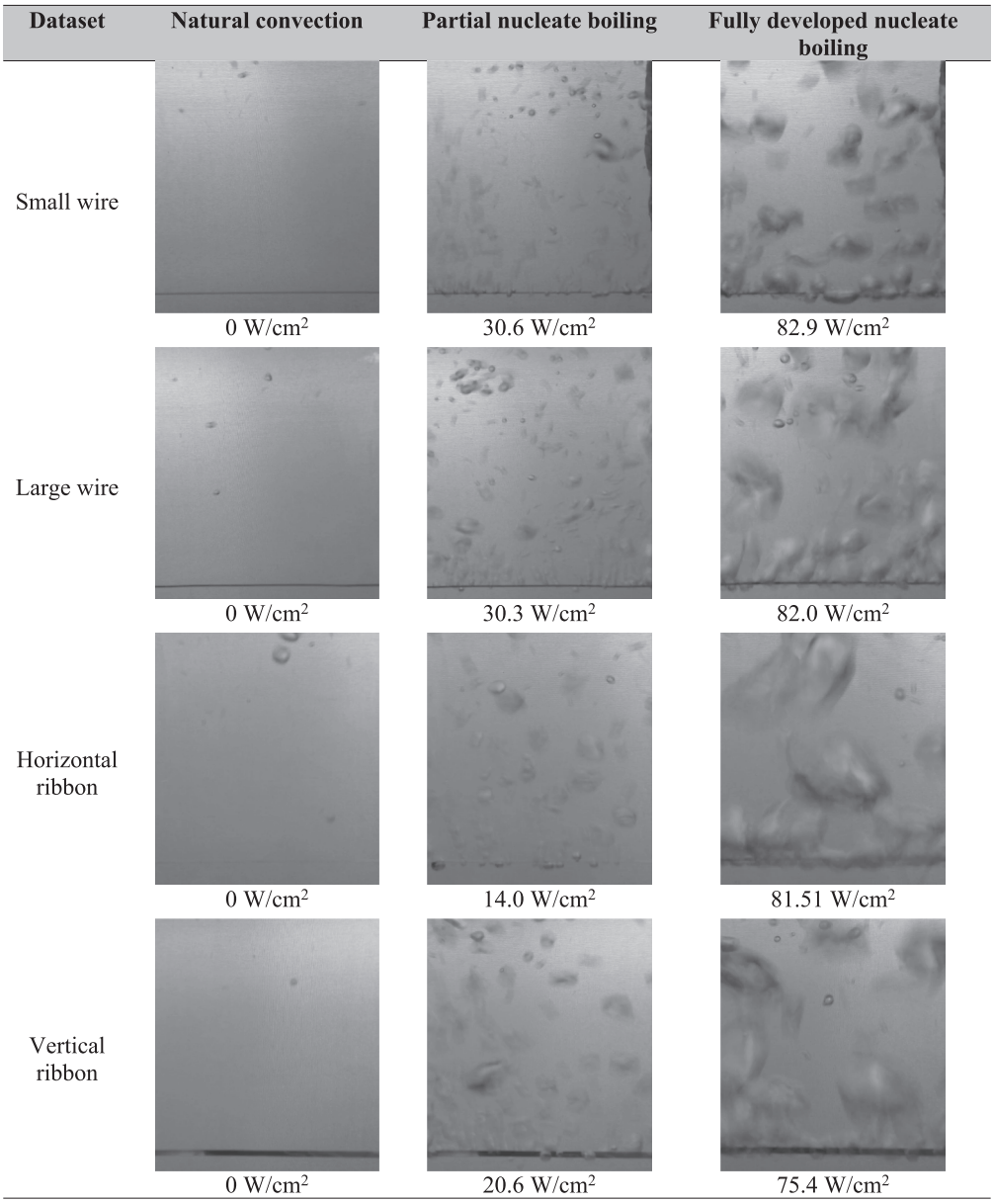


Fig. 3. Samples of non-processed images obtained for each of the four heating surfaces in different boiling regimes along with their respective measured heat flux.

underneath it, and the horizontal length of the heat transfer surface shown is approximately 5 cm. By comparing each regime for the different surfaces, one can barely visually identify the effect of the surface on the boiling process, especially when comparing both wires with the horizontal ribbon. Differently, in the vertical ribbon configuration, the heating surface is more evident. One should notice, however, that these are "untreated" images that have not been post-processed yet – this will be discussed in Section 3 below –, and have the sole objective of visually exemplifying the type of information made available to the CNN.

The experimental uncertainty is determined by the error propagation method described in Ref. [20] for the only variable considered in the present study: the dissipated heat flux. In this case, the uncertainty of the heat flux (q'') is a function of four variables: the surface area of the heater (A), the measured voltage difference between the electrodes (V_{sample}), and the uncertainty of the current, which is determined based on the measured voltage drop across the shunt (V_{shunt}) and its known resistance (R_{shunt}). Therefore, the relative heat flux uncertainty, $\hat{u}(q")/q"$, can be written as

$$\frac{\hat{u}(q")}{q"} = \sqrt{\left[\frac{\hat{u}(A)}{A}\right]^2 + \left[\frac{\hat{u}(V_{sample})}{V_{sample}}\right]^2 + \left[\frac{\hat{u}(V_{shunt})}{V_{shunt}}\right]^2 + \left[\frac{\hat{u}(R_{shunt})}{R_{shunt}}\right]^2}, \quad (1)$$

where $\hat{u}$ represents the uncertainty associated with each variable. Knowing that $\hat{u}(A)/A = \pm 1.79$ % (for D^{SW}), ± 0.88 % (for D^{LW}), and ± 0.39 % (for D^{HR} and D^{VR}), $\hat{u}(V_{sample}) = \pm 6230$ μV, $\hat{u}(V_{shunt}) = \pm 174$ μV, and $\hat{u}(R_{shunt})/R_{shunt} = \pm 0.5$ %, the expanded uncertainty with a coverage factor of 2 for D^{SW}, D^{LW}, D^{HR} and D^{VR} are ± 2.227 W/cm², ± 1.266 W/cm², ± 1.044 W/cm² and ± 0.966 W/cm², respectively. It is also worth mentioning that the RTD used to measure the bulk temperature of the water presented an uncertainty of ± 0.35 °C after calibration.

3. Image processing and CNN training

The images obtained experimentally for each heating surface went through a sequence of preprocessing steps before being used to train the

CNN described next; many of these preprocessing steps were successfully used in previous studies, e.g. [5–8,21]. The first step is splitting the videos into individual images for each of the four heating surfaces while preventing the existence of correlated images. This was done by determining the Structural Similarity Index (SSI) and observing that it was nearly constant across the datasets [22], which indicates that 30fps frame rate is low enough to prevent the occurrence of correlated images. The resulting individual images were then randomly divided into three groups: training, validation and testing using a (70 %, 15 %, 15 %) ratio [18]; notice that this standard data splitting method does not use resampling, preventing data leakage between distinct runs and simplifying the training procedure. The amount of images obtained for each subset is shown in Table 3.

Next, to prevent data corruption during the image processing stages, all images were converted to 32-bit floating number representation beforehand [23]. Then, the images were cropped such that the cropped region prevented the appearance of the unwanted features, such as the electrodes, auxiliary heaters and the LED lamp. The resulting images were then converted to grayscale, reducing their dimensionality to a third of the original RGB version [23].

Finally, the images were subjected to a downscaling process, which consists in averaging neighboring pixels, and, consequently, reducing the dimensionality of the image at the cost of also losing resolution (i.e. information) [23]. However, images with lower resolution can be analyzed quicker, reducing the time required for training and testing. In the limit where too many neighboring pixels are averaged, valuable information is lost. In that sense, it is important to quantify the amount of information lost as a function of the downscaling operator, which can be computed by the relative variance (VAR*) and the relative cross-entropy (S*). These functions can be written as

$$\mathrm{VAR}^{*}\left[\mathrm{x},\mathrm{D}^{\mathrm{fs}}(\mathrm{x})\right]=\frac{\mathrm{VAR}\left[\mathrm{D}^{\mathrm{fs}}(\mathrm{x})\right]}{\mathrm{VAR}(\mathrm{x})}=\frac{\sum_{\mathrm{j}=0}^{100}\left\{\mathrm{HIST}_{\mathrm{j}}\left[\mathrm{D}^{\mathrm{fs}}(\mathrm{x})\right]-\overline{\mathrm{HIST}}\left[\mathrm{D}^{\mathrm{fs}}(\mathrm{x})\right]\right\}^{2}}{\sum_{\mathrm{j}=0}^{100}\left[\mathrm{HIST}_{\mathrm{j}}(\mathrm{x})-\overline{\mathrm{HIST}}(\mathrm{x})\right]^{2}} \tag{2a}$$

$$\mathrm{S}^{*}\left[\mathrm{x},\mathrm{D}^{\mathrm{fs}}(\mathrm{x})\right]=\frac{\mathrm{S}\left[\mathrm{x},\mathrm{D}^{\mathrm{fs}}(\mathrm{x})\right]}{\mathrm{S}(\mathrm{x},\mathrm{x})}=\frac{\mathrm{HIST}(\mathrm{x})\ \cdot \ln \mathrm{HIST}\left[\mathrm{D}^{\mathrm{fs}}(\mathrm{x})\right]}{\mathrm{HIST}(\mathrm{x})\ \cdot \ln \mathrm{HIST}(\mathrm{x})}, \tag{2b}$$

where x is the feature vector (i.e., the image), D^{fs} is the downscale operator, and HIST is luminance distribution histogram. Much like the analysis presented by Ref. [6], the present results, shown in Fig. 4a and b, demonstrate that both the relative variance and cross-entropy are nearly stable for a downscaling of up to 5, increasing rapidly for larger values. Therefore, the downscaling factor employed in the present work is 5, which is the same used by Ref. [6,8]. Also, and because machine learning algorithms require a specific type of input (in this case, images of the same size), all images from all four datasets were further cropped to a common size. Following Ref. [6,8], the specific image dataset for each heating surface after all preprocessing steps mentioned above produced a sequence of images in which the heating surface is actually visible within the image, which is referred to as *direct* visualization mode [6]. However, because machine learning algorithms could rely on several image features other than the vapor phase itself to estimate the variable of interest, new equally sized datasets were obtained by suppressing (by cropping out) the heating element from direct visualization images, generating the so-called *indirect* visualization mode. Table 4 shows the effect of cropping, grayscaling and downscaling by a factor of 5, on the images' dimensionality (number of pixels) for both visualization modes. Notice that the preprocessing starts on the left side of Table 4, and it is possible to observe the drastic reduction in the number of pixels of the images. Also, and for sake of illustration, Table 4 shows a typical image for each of the heating surfaces for both visualization modes after all preprocessing steps. It is also wroth mentioning that the effect of downscaling on the CNN's MSE was also verified. Results indicate that the MSE is not negatively affected for a downscaling operator of approximately 5 in both visualization modes.

Table 3
Sizes of the datasets employed for training, validation and testing.

Heater	Training Set (70 %)	Validation Set (15 %)	Testing Set (15 %)	Total
D^{SW}	12,950	2,773	2,775	18,498
D^{LW}	26,416	5,661	5,662	37,739
D^{HR}	58,774	12,593	12,594	83,961
D^{VR}	51,276	10,987	10,987	73,250

Because the main goals of the present study are to verify the ability of machine learning algorithms to generalize information, which justifies the use of several heating surfaces, as well as to quantify the performance gain associated with the use of AutoML, a baseline CNN case was specified to serve as reference. In the upcoming analyses, the CNN architecture employed in a previous study reported in our group was used as baseline [8]. The main reason for implementing this architecture once again is because it performed well in a dataset very similar to ours. Therefore, it can be directly compared with the architecture proposed by the AutoML. The CNN architecture of Ref. [8] had a convolution block with 32 filters, kernel size of 5 × 5, max-pooling size of 2 × 2, ReLU activation, a regularization layer with a dropout rate of 50 %, a dense block with 200 units and a single-unit, linearly activated output, resulting in over 23 million trainable parameters for direct visualization and nearly 14 million for indirect visualization. For sake of continuity, and to fully disclose the CNN architecture presently regarded as a baseline case, its graphical representation is shown in Fig. 5. The baseline model was optimized with the Adam Optimizer [24] with a learning rate of 10^{-3} and batches of 200 samples, targeting a minimal MSE as metric. Also, it is worth mentioning that the training process presently adopted is similar to the one used in Ref. [8], and recommended by the literature [18].

It is also important to mention that simulations, which include preprocessing, training, validation and testing, were run locally on a laptop with a clock speed of 2.3 GHz, 32 GB of RAM, and a graphics processing unit NVIDIA GeForce RTX 3070 running Ubuntu 20.04.5. The code was written in Python 3.10 [25], while relying on following main open-source libraries: TensorFlow 2.10.1 [23], KerasTuner 1.1.3 [26] and AutoKeras 1.0.20 [17]. It is also worth mentioning that a custom-made Python source code, which is publicly available, was developed to run the simulations presented in this article; see Ref. [19].

4. Validation

In order to ensure the correctness of the entire process implemented in the present study, two keys analyses were implemented. First, the model learning curve, which highlights the influence of the size of the training set on the model's performance [27], was evaluated for the direct and indirect visualization for the large diameter wire (D^{LW}). Next, the present training and validation results, also for the large diameter wire (D^{LW}), were compared with the so-called baseline results presented by an earlier study of our group [8]. One should note that the presently tested large diameter wire, which has a diameter of 510 μm, is the closest to the diameter of the wire heater used in Ref. [8] (i.e., 451.7 μm) and because of that was used for both analyses above. Also, both analyses considered the same set of hyperparameters for the CNN model used in by Ref. [8], but distinct datasets. Furthermore, and similarly to Ref. [8], the analyses above only considered nucleate and film boiling flow regimes, i.e., q" $\geq$ 10 W/cm^2, hence ensuring a fair comparison.

The results for the learning curve are shown in frames a and b of Fig. 6, which, respectively, refer to direct and indirect visualization. The results were obtained in intervals of 1 % between 1 % and 10 %, and in intervals of 10 % between 10 % and 100 %. As can be seen, the reported mean square error (MSE) is inversely related to the size of the training set. However, no significant improvement is observed when the size of training set reaches about 80 %, indicating that 80 %-100 % of the

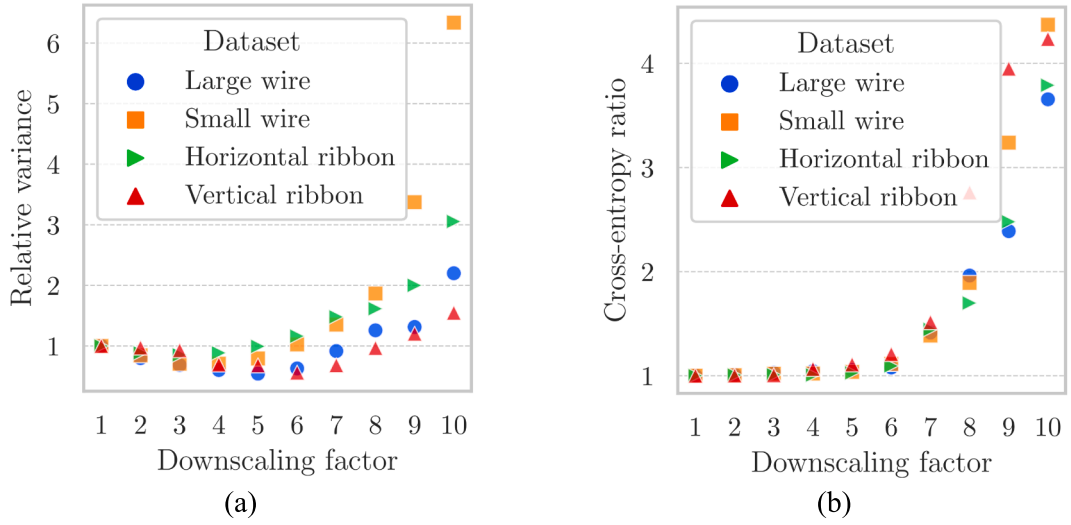
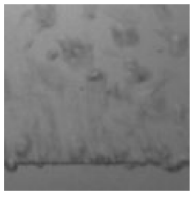
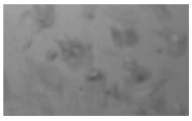
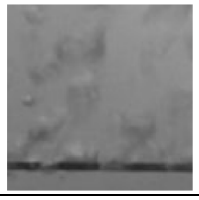


Fig. 4. Effect of downscaling on the relative variance (left frame) and on the cross-entropy (right frame).

Table 4
Effect of the image preprocessing on the dimensionally reduction (number of pixels) of the dataset of each heating surface.

Heater	Original Image	Cropping	Grayscaling	Downscaling	Visualization mode	
					Direct	Indirect
D^{SW}	1,520 × 2,704 × 3	870 × 790 × 3	870 × 790 × 1	174 × 158 × 1	120 × 120 × 1	72 × 120 × 1
						
D^{LW}	1,520 × 2,704 × 3	970 × 855 × 3	970 × 855 × 1	194 × 171 × 1	120 × 120 × 1	72 × 120 × 1
						
D^{HR}	1,520 × 2,704 × 3	850 × 790 × 3	850 × 790 × 1	170 × 158 × 1	120 × 120 × 1	72 × 120 × 1
						
D^{VR}	1,520 × 2,704 × 3	900 × 830 × 3	900 × 830 × 1	180 × 166 × 1	120 × 120 × 1	72 × 120 × 1
						

Preprocessing direction →.

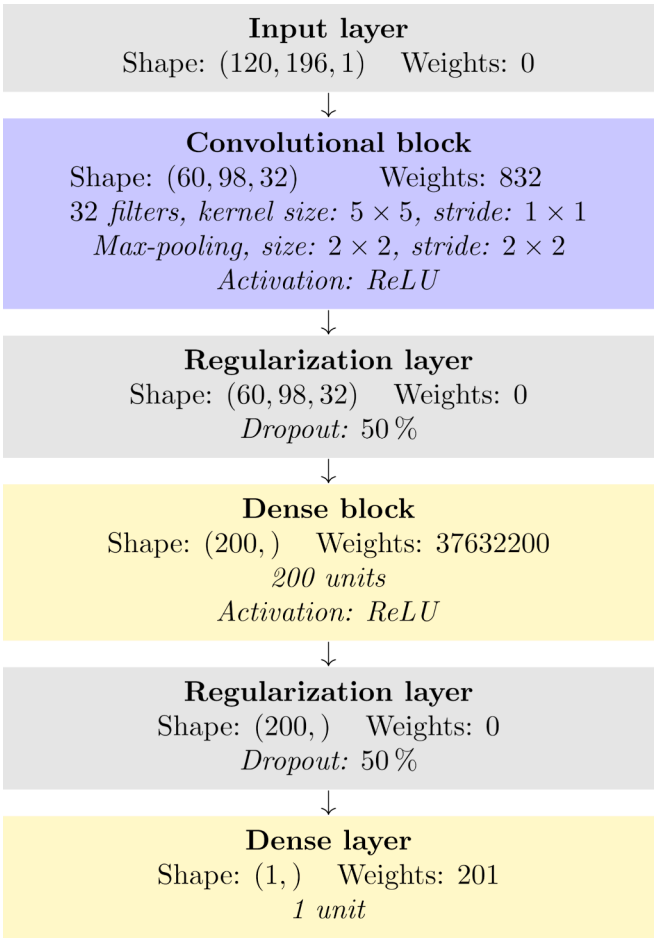


Fig. 5. Flowchart representation of the CNN architecture reported in Ref. [8] and used in the present study as a baseline.

training set is enough for the model to reach its highest performance, and that no extra data is necessary. Therefore, in the present study, 100 % of the training set was used. Furthermore, the present analysis also justifies the 70:15:15 hold-out method for splitting the datasets [18].

Next, a direct comparison between the present results and the results reported by Ref. [8] is shown in Table 5 for both visualization modes. As can be observed, both sets of results are very similar for all metrics considered, especially if one takes into consideration that the results rely on distinct experiments, and, hence, distinct sets of images. Also, similarly to the trend observed by Ref. [8], the direct visualization mode outperforms the indirect visualization mode. Therefore, based on the agreement present in Table 5, it is possible to assume that the entire process, from preprocessing to testing, is fully validated. Additionally, not only all error metrics presently calculated are similar to those reported in Ref. [8], but, also, these metrics are nearly constant between training, validation and testing, suggesting that the model is capable of generalizing well to unseen images from the same experiment. Furthermore, since the validation metrics are very close to the test metrics, one can argue that the validation set is an appropriate representation of the test set.

5. Results

In general terms, the main goal of the present work is to quantify the heat flux dissipated by the heater surface while solely considering images of the boiling process. However, because three different strategies were explored to accomplish that, the results obtained will be presented in three distinct sub-sections. The first sub-section presents the results and discussion when each CNN model is trained and tested for each type of heating surface individually (Section 5.1). The next sub-section aims to generalize the results by training and testing models with images from different heaters simultaneously (Section 5.2), i.e., mixed training and testing. Finally, the third and last sub-section uses Automated Machine Learning (AutoML) to optimize the CNN architecture and training hyperparameters (Section 5.3).

5.1. Single heating surface analysis

In the present sub-section, CNN models with the baseline architecture were trained and tested on a single heating surface each. The goal is to analyze the effect of the heating surface on the performance of the model. The results are presented in Tables 6a and b for direct and indirect visualization, respectively. Each table shows five error metrics and the respective confidence intervals for each of the four configurations considering nucleate and film boiling regimes, i.e., heat fluxes beyond 10 W/cm^2. The first general observation is that, regardless of the visualization mode (i.e., direct or indirect), the percent errors (MAPE) are significantly smaller than the expected errors provided by correlations; for instance, Ref. [1] suggested that such errors can reach ± 100 %. For example, the largest MAPE for direct visualization is in the order of 8 %, and for the indirect visualization in the order of 12 %; these results are consistent with the results presented in Ref. [8], which not only showed MAPE values within the same order, but also concluded that indirect visualization has a poorer performance when compared

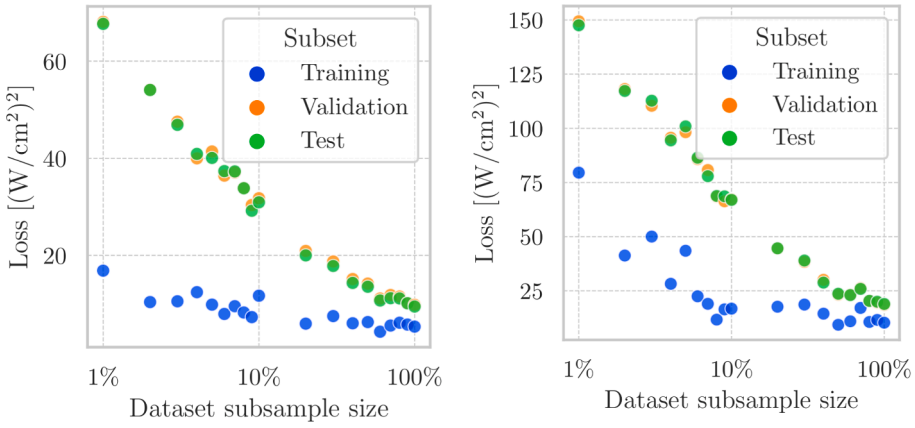


Fig. 6. Effect of the size of the training set on the model's performance in (a) direct visualization and (b) indirect visualization.

Table 5
Comparison between the error obtained in the present study and in Ref. [8] for the heat flux quantification considering the baseline CNN for nucleate and boiling regimes, i.e., q'' $\geq$ 10 W/cm^2.

	Metric	Unit	This work			Ref. [8]	
			Training	Validation	Test	Validation	Test
Direct	MSE	$(W/cm^2)^2$	$12.32^{+0.06}_{-0.06}$	$15.69^{+0.06}_{-0.06}$	$15.41^{+0.03}_{-0.03}$	13^{+1}_{-1}	13^{+1}_{-1}
	RMSE	W/cm^2	$3.507^{+0.008}_{-0.009}$	$3.959^{+0.008}_{-0.009}$	$3.925^{+0.004}_{-0.004}$	–	–
	MAE	W/cm^2	$2.720^{+0.006}_{-0.007}$	$3.053^{+0.006}_{-0.007}$	$3.025^{+0.003}_{-0.003}$	$2.8^{+0.1}_{-0.1}$	3^{+6}_{-3}
	MAPE	%	$6.97^{+0.02}_{-0.02}$	$7.62^{+0.02}_{-0.02}$	$7.57^{+0.008}_{-0.008}$	$7.4^{+0.5}_{-0.4}$	$7.6^{+0.4}_{-0.4}$
	R^2	–	$0.9688^{+0.0002}_{-0.0002}$	$0.9601^{+0.0002}_{-0.0002}$	$0.9609^{+0.00008}_{-0.00008}$	$0.983^{+0.002}_{-0.002}$	$0.983^{+0.002}_{-0.002}$
Indirect	MSE	$(W/cm^2)^2$	$29.60^{+0.10}_{-0.10}$	$33.84^{+0.05}_{-0.05}$	$33.3^{+0.1}_{-0.1}$	34^{+4}_{-4}	33^{+4}_{-4}
	RMSE	W/cm^2	$5.439^{+0.009}_{-0.009}$	$5.816^{+0.004}_{-0.004}$	$5.76^{+0.01}_{-0.01}$	–	–
	MAE	W/cm^2	$4.274^{+0.006}_{-0.006}$	$4.475^{+0.005}_{-0.005}$	$4.530^{+0.009}_{-0.009}$	4^{+10}_{-4}	4^{+10}_{-4}
	MAPE	%	$10.80^{+0.01}_{-0.02}$	$11.07^{+0.01}_{-0.01}$	$11.38^{+0.02}_{-0.02}$	$10.6^{+0.6}_{-0.6}$	$10.3^{+0.5}_{-0.5}$
	R^2	–	$0.9250^{+0.0003}_{-0.0003}$	$0.9141^{+0.0002}_{-0.0002}$	$0.9156^{+0.0004}_{-0.0004}$	$0.956^{+0.005}_{-0.005}$	$0.956^{+0.005}_{-0.005}$

with direct visualization. Also, the ratio between the MSE of Table 6b (indirect visualization) and 6a (direct visualization) varies between 1.8 and 4.5, depending on the heating surface considered, which is also compatible with the 2.5 ratio obtained by Ref. [8]. Another interesting observation that can be derived from the results presented in Table 6a is the influence of the heating surface on the model performance. For instance, smaller errors are obtained for heating elements that are more visible within the image. Therefore, since the least visible heating surfaces are the horizontal ribbon and the small diameter wire, these configurations present the largest errors. Another plausible reason for the difference in the model performance is related to the dataset size (i.e., number of images consumed during training): according to Table 3, the dataset for the large diameter wire is roughly twice as large as the small diameter wire dataset. It could also be argued that the performance for the large diameter wire would be reduced if the data were also reduced. In fact, an additional evaluation was performed for the large wire configuration using only 50 % of the original dataset shown in Table 3 for training. The resulting MSE for the training, validation and testing subsets are 9.46 $(W/cm^2)^2$, 16.37 $(W/cm^2)^2$ and 15.51 $(W/cm^2)^2$, respectively, which are very close to the metrics shown in Table 6a for the small diameter wire. Finally, and in spite of the variability in the performance metrics, the results presented in Table 6 demonstrate that the methodology employed in this study achieves considerably low errors regardless of the shape and size of the heating surface when the model is trained and tested on images of the same surface.

5.2. Multiple heating surface analysis

The objective of this sub-section is to explore the ability of the CNN models to generalize solutions to different operating conditions. In that sense, models are trained, for example, with a dataset of a specific heating surface, but are asked to predict the heat flux when analyzing an image from a different heating surface that was not used for training. Also, a new type of dataset will be introduced, which is called the combined datasets and represent by the operator $\overline{U}$. In this case, a new proportionally divided dataset is created by combining two or all four original datasets. In total, seven dataset variations were used, which are: D^{SW}, D^{LW}, D^{HR}, D^{VR}, $\overline{U}^{W}$, $\overline{U}^{R}$ and $\overline{U}^{A}$. Using this nomenclature, the dataset referred to as D^{SW}, for example, indicates that the CNN training or evaluation is performed solely with images derived from the small diameter heater, whereas the dataset referred as $\overline{U}^{W}$ indicates the combined dataset that uses 50 % of the small diameter wire dataset and 50 % of the large diameter wire dataset. Similarly, the dataset referred as $\overline{U}^{R}$ is composed of 50 % of the dataset D^{HR} and 50 % of the dataset D^{VR}. Finally, $\overline{U}^{A}$ is composed of 25 % of the dataset of each heating surface. For disclosure purposes, Table 7 shows the amount of images in each of the seven datasets considered.

For each one of the seven datasets, a CNN with the baseline architecture was trained on it and evaluated against all seven datasets. The results are presented in Fig. 7a and b for the direct and indirect visualization, respectively. Each frame is a heat map where each of the seven rows represents the training dataset used and each of the columns represents the validation dataset on which the model is evaluated. Also, each cell of the 7 × 7 matrix is populated by two values; the upper value indicates the MSE, while the value between brackets represents the MAPE; the shade of blue indicates the MSE scale located at the bottom of the figure. Several conclusions can be drawn from Fig. 7. The first one is the fact that the best performance, i.e., lowest MSE, is obtained in the main diagonal of both matrices; this is expected, since CNN models perform best when they are actually trained with the dataset derived from the surface that they will be asked to evaluate. Having said that, lower error metrics are obtained for direct visualization, which agrees with Ref. [8]. Another interesting aspect is associated with cases where models trained on combined training sets are evaluated on a heating surface that is included in their training set. For example, considering the training set $\overline{U}^{W}$, it can be seen that good performance is obtained when evaluating the heat flux of images belonging to either class of the heating surfaces, i.e., D^{SW} and D^{LW}. The same can be said for the model trained on $\overline{U}^{R}$ and evaluated on either D^{HR} or D^{VR}. Note that the metrics for these two combined training sets are very similar to cases where a single heating surface is used for training and testing (see Table 6). When the most diverse training set is used, i.e., $\overline{U}^{A}$, the results return fairly good error metrics for all seven training sets, including also the combined set, i.e., $\overline{U}^{A}$. In that sense, the use of the images of different heating surfaces in the training process helps the model become more performant when compared to a model exclusively trained with images of a single surface, hence, improving the model generalization capabilities. For instance, one can compare the performance of a CNN model trained with the combined $\overline{U}^{W}$ dataset (which contains 50 % of the images from D^{SW} and 50 % of D^{LW}) and a model trained with only 50 % of the D^{LW} dataset. In this case, the former and latter were exposed to D^{LW} subsets with the exact same size, but the former contains additional images from D^{SW}. Thus, one might be able to infer the effect of the D^{SW} dataset. This direct comparison showed that training a model with the $\overline{U}^{W}$ dataset and evaluating the heat flux on the D^{LW} surface returns a MSE that is 10 % lower than a CNN trained exclusively with a dataset that is half of the original dataset of the D^{LW} surface in the direct visualization mode. This suggests that the D^{SW} data present in the $\overline{U}^{W}$ combined dataset contributes to the generalization of the CNN. Differently, for the indirect visualization mode, a model trained on the $\overline{U}^{W}$ dataset returns MSE values that are approximately 12 % higher than a CNN trained only with 50 % of the D^{LW} dataset. Therefore, it can be argued that some level of generalization can be achieved in the direct visualization mode, but not necessarily in the indirect mode; this might be related to the actual weight of the wire on the decision making

Table 6
Effect of the heating surface on the performance of the baseline CNN for training, validation and testing; (a) direct visualization mode and (b) indirect visualization mode.

(a)			Subset		
Heater	Metric	Unit	Training	Validation	Test
Small wire (D^{SW})					
	MSE	$(W/cm^2)^2$	$11.89^{+0.05}_{-0.04}$	$18.15^{+0.05}_{-0.05}$	$19.37^{+0.03}_{-0.03}$
	RMSE	W/cm^2	$3.446^{+0.006}_{-0.006}$	$4.259^{+0.006}_{-0.006}$	$4.401^{+0.003}_{-0.003}$
	MAE	W/cm^2	$2.679^{+0.005}_{-0.005}$	$3.246^{+0.004}_{-0.004}$	$3.348^{+0.002}_{-0.003}$
	MAPE	%	$7.23^{+0.01}_{-0.01}$	$8.209^{+0.01}_{-0.01}$	$8.238^{+0.005}_{-0.005}$
	R^2	–	$0.9785^{+0.0001}_{-0.0001}$	$0.96724^{+0.00008}_{-0.00008}$	$0.96498^{+0.00005}_{-0.00005}$
Large wire (D^{LW})					
	MSE	$(W/cm^2)^2$	$5.43^{+0.04}_{-0.04}$	$9.64^{+0.03}_{-0.03}$	$9.37^{+0.03}_{-0.03}$
	RMSE	W/cm^2	$2.327^{+0.008}_{-0.009}$	$3.103^{+0.005}_{-0.005}$	$3.060^{+0.004}_{-0.004}$
	MAE	W/cm^2	$1.792^{+0.006}_{-0.007}$	$2.361^{+0.004}_{-0.004}$	$2.336^{+0.003}_{-0.003}$
	MAPE	%	$4.48^{+0.02}_{-0.02}$	$5.70^{+0.01}_{-0.01}$	$5.633^{+0.007}_{-0.006}$
	R^2	–	$0.9862^{+0.0001}_{-0.0001}$	$0.97552^{+0.00008}_{-0.00008}$	$0.97623^{+0.00008}_{-0.00008}$
Horizontal ribbon (D^{HR})					
	MSE	$(W/cm^2)^2$	$9.55^{+0.03}_{-0.03}$	$12.37^{+0.03}_{-0.03}$	$12.88^{+0.04}_{-0.04}$
	RMSE	W/cm^2	$3.089^{+0.005}_{-0.005}$	$3.517^{+0.005}_{-0.005}$	$3.589^{+0.005}_{-0.005}$
	MAE	W/cm^2	$2.359^{+0.004}_{-0.003}$	$2.660^{+0.003}_{-0.003}$	$2.715^{+0.004}_{-0.004}$
	MAPE	%	$5.69^{+0.01}_{-0.01}$	$6.267^{+0.009}_{-0.009}$	$6.415^{+0.007}_{-0.008}$
	R^2	–	$0.98056^{+0.00007}_{-0.00007}$	$0.97481^{+0.00008}_{-0.00008}$	$0.97380^{+0.00007}_{-0.00008}$
Vertical ribbon (D^{VR})					
	MSE	$(W/cm^2)^2$	$6.63^{+0.02}_{-0.02}$	$9.61^{+0.02}_{-0.02}$	$9.86^{+0.06}_{-0.05}$
	RMSE	W/cm^2	$2.574^{+0.005}_{-0.005}$	$3.100^{+0.003}_{-0.003}$	$3.137^{+0.009}_{-0.008}$
	MAE	W/cm^2	$1.995^{+0.003}_{-0.003}$	$2.394^{+0.002}_{-0.002}$	$2.384^{+0.005}_{-0.005}$
	MAPE	%	$5.220^{+0.009}_{-0.009}$	$6.140^{+0.008}_{-0.008}$	$6.13^{+0.01}_{-0.01}$
	R^2	–	$0.98279^{+0.00006}_{-0.00007}$	$0.97506^{+0.00006}_{-0.00006}$	$0.9744^{+0.0001}_{-0.0002}$
(b)			**Subset**		
Heater	Metric	Unit	Training	Validation	Test
Small wire (D^{SW})					
	MSE	$(W/cm^2)^2$	$22.54^{+0.08}_{-0.08}$	$36.06^{+0.08}_{-0.08}$	$35.47^{+0.05}_{-0.04}$
	RMSE	W/cm^2	$4.745^{+0.008}_{-0.008}$	$6.004^{+0.006}_{-0.007}$	$5.956^{+0.004}_{-0.004}$
	MAE	W/cm^2	$3.724^{+0.005}_{-0.006}$	$4.655^{+0.002}_{-0.002}$	$4.610^{+0.003}_{-0.003}$
	MAPE	%	$10.61^{+0.03}_{-0.03}$	$12.38^{+0.02}_{-0.02}$	$12.10^{+0.01}_{-0.01}$
	R^2	–	$0.9592^{+0.0001}_{-0.0001}$	$0.93481^{+0.00010}_{-0.00010}$	$0.93589^{+0.00007}_{-0.00007}$
Large wire (D^{LW})					
	MSE	$(W/cm^2)^2$	$9.41^{+0.06}_{-0.07}$	$18.08^{+0.03}_{-0.03}$	$18.14^{+0.04}_{-0.04}$
	RMSE	W/cm^2	$3.06^{+0.01}_{-0.01}$	$4.252^{+0.004}_{-0.004}$	$4.258^{+0.005}_{-0.005}$
	MAE	W/cm^2	$2.356^{+0.008}_{-0.009}$	$3.247^{+0.003}_{-0.003}$	$3.233^{+0.003}_{-0.003}$
	MAPE	%	$5.80^{+0.02}_{-0.02}$	$7.707^{+0.008}_{-0.008}$	$7.69^{+0.01}_{-0.01}$
	R^2	–	$0.9761^{+0.0002}_{-0.0002}$	$0.9541^{+0.0001}_{-0.0001}$	$0.9539^{+0.0001}_{-0.0002}$
Horizontal ribbon (D^{HR})					
	MSE	$(W/cm^2)^2$	$41.6^{+0.2}_{-0.2}$	$56.1^{+0.1}_{-0.1}$	$52.8^{+0.2}_{-0.3}$
	RMSE	W/cm^2	$6.44^{+0.01}_{-0.01}$	$7.49^{+0.01}_{-0.01}$	$7.26^{+0.02}_{-0.02}$
	MAE	W/cm^2	$4.790^{+0.009}_{-0.009}$	$5.520^{+0.008}_{-0.008}$	$5.38^{+0.01}_{-0.01}$
	MAPE	%	$10.85^{+0.03}_{-0.02}$	$12.38^{+0.03}_{-0.03}$	$11.96^{+0.03}_{-0.03}$
	R^2	–	$0.9154^{+0.0003}_{-0.0003}$	$0.8858^{+0.0003}_{-0.0003}$	$0.8924^{+0.0006}_{-0.0006}$
Vertical ribbon (D^{VR})					
	MSE	$(W/cm^2)^2$	$19.11^{+0.05}_{-0.05}$	$24.41^{+0.06}_{-0.06}$	$25.48^{+0.08}_{-0.08}$
	RMSE	W/cm^2	$4.370^{+0.006}_{-0.006}$	$4.939^{+0.006}_{-0.007}$	$5.046^{+0.008}_{-0.008}$
	MAE	W/cm^2	$3.389^{+0.005}_{-0.005}$	$3.824^{+0.006}_{-0.006}$	$3.882^{+0.007}_{-0.006}$
	MAPE	%	$8.86^{+0.01}_{-0.01}$	$9.89^{+0.02}_{-0.02}$	$10.09^{+0.02}_{-0.02}$
	R^2	–	$0.95504^{+0.0002}_{-0.0002}$	$0.9367^{+0.0002}_{-0.0002}$	$0.9338^{+0.0002}_{-0.0002}$

Table 7
Dataset combination for the multi-surface analysis.

Dataset	Training set (70 %)	Validation set (15 %)	Test set (15 %)	Total (100 %)
D^{SW}	12,950	2,773	2,775	18,498
D^{LW}	26,416	5,661	5,662	37,739
D^{HR}	58,774	12,593	12,594	83,961
D^{VR}	51,276	10,987	10,987	73,250
$\overline{U}^{W}$	19,683	4,217	4,218	28,118
$\overline{U}^{R}$	55,025	11,790	11,790	78,605
$\overline{U}^{A}$	37,354	8,003	8,004	53,361

process.

It is important to emphasize the fact that one of the goals of the present study was to infer on the CNN's ability to generalize to different heating surfaces, which, as demonstrated previously, is somewhat limited. In fact, this partially justifies the use of only four heating surfaces in the study, which are actually fairly similar to each other. For instance, previous studies have used the so-called activation maps – e.g., [5,21] – to highlight the most relevant areas of an image to image-trained machine-learning algorithms for decision making (e.g., classification or quantification). In that sense, the use of an arguably limited number of heating surfaces (i.e., 4) adds degrees of freedom to the

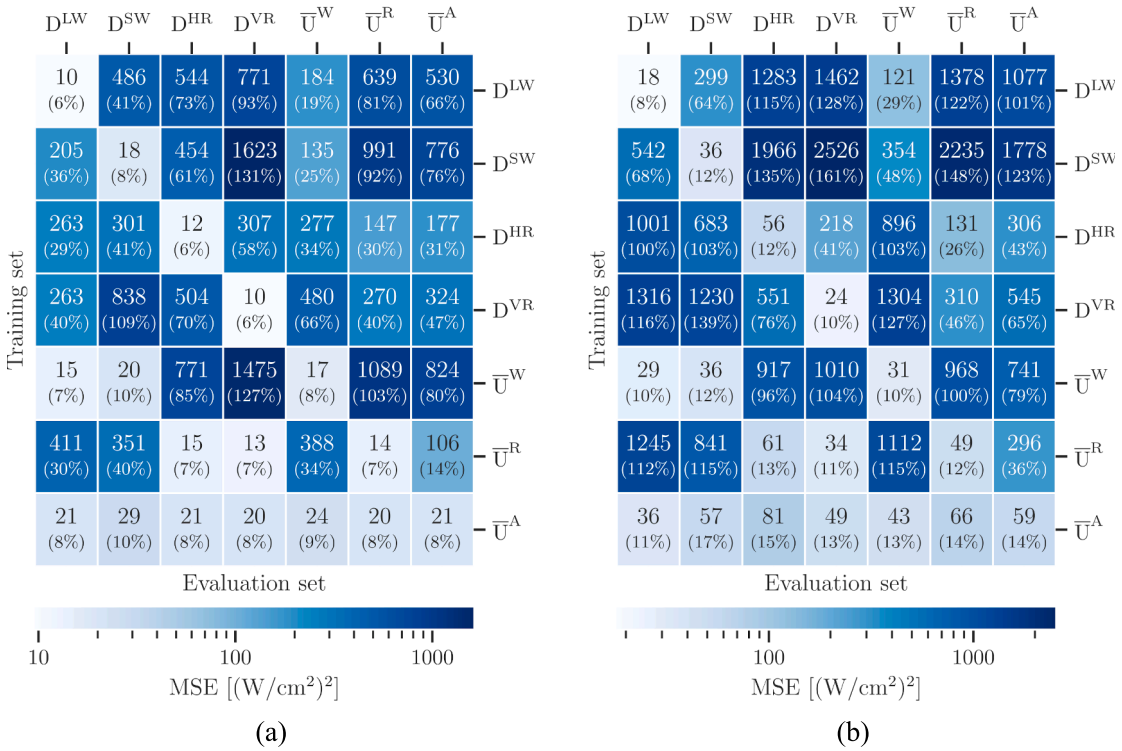


Fig. 7. Heat map displaying the MSE (upper value on each cell) and MAPE (value between brackets in each cell) for different combinations of training datasets (rows) and validation datasets (columns) considering the baseline CNN. The blue shade represents the MSE scale presented at the lower end of the figure; frames (a) and (b) represent the direct and the indirect visualization, respectively.

quantification process without overwhelming the analysis. Furthermore, the use of similar heating elements minimizes the chances of feeding unwanted information to the CNN through the training process (a geometric detail, for example), which would arguably complicate the generalization analysis and make the models more susceptible to overfitting due to the unintentional appearance of irrelevant aspects of the image.

5.3. Automated machine learning analysis

The present sub-section focuses on the use an AutoML routine [17] to optimize the architecture and hyperparameters of the CNN models. The analysis was performed for the large diameter-heating element (D^{LW}) for direct and indirect visualization and the optimization of the CNN was based on a greedy random search [17], which performed well when compared with other search algorithms tested. The optimization, which encompassed the CNN's architecture and training hyperparameters, was conducted within the pre-defined search space shown in Table 8. Approximately 100 models were needed (trained and tested) to obtain high-quality models for each visualization mode, where each model was optimized with batches of 32 samples while aiming to minimize the validation MSE. In addition, the AutoML optimization only considered CNN models with the same number of parameters as the base model, or smaller. This was done to ensure a fair comparison between the AutoML-optimized and the baseline models. The resulting best-performing hyperparameters are shown in Table 9 while the corresponding CNN architecture flowchart is shown in Fig. 8 for both visualization modes.

Table 10 shows the errors obtained with for the D^{LW} heating surface using a CNN with the architecture optimized with AutoML and described in Table 9. Generally speaking, the results for all metrics show a significant improvement when compared to the baseline results presented in Table 5 for both sets of results, i.e., present and those of Ref. [8]. For instance, the new values for the validation MSE are 4.23 $(W/cm^2)^2$ and 6.71 $(W/cm^2)^2$ for direct and indirect visualization, respectively, which

Table 8
Search space for the AutoML optimization of the CNN model.

Hyperparameter	Search Space
Architecture – Convolutional Layers	
Number of convolutional blocks	[1, 2, 3]
Number of consecutive convolutional layers per block	[1, 2]
Convolutional kernel size	[3, 5, 7]
Number of filters per layer	[16, 32, 64, 128, 256, 512]
Depthwise separable convolutions?	[NO, YES]
Apply max-pooling?	[NO, YES]
Dropout rate	[0, 25 %, 50 %]
Architecture – Reduction Layers	
Spatial reduction layer type	Flattening, Global average pooling, Global max-pooling
Architecture – Dense Layers	
Use batch normalization?	[NO, YES]
Number of dense layers	[1, 2, 3]
Number of units per layer	[16, 32, 64, 128, 256, 512, 1024]
Dropout rate	[0, 25 %, 50 %]
Optimizer	
Optimizer algorithm	[SDG, Adam, AdamW]
Learning rate	$[10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 2 \times 10^{-5}]$

are 73 % and 80 % lower than the corresponding values obtained previously. Moreover, the ratio between the MSE value for indirect and direct visualization dropped from 2.15 in Section 4 to 1.59 in the present section, which is a clear indication the AutoML-optimized models are better in identifying key, generalizable information in the images. It is also worth mentioning that the MAPE values for direct and indirect visualization are below the MAPE associated with a correlation reported by Ref. [28], which is 4.82 %; such low error metrics could not be achieved by the baseline models. Another important feature obtained with the use of AutoML is associated with the fact that this image-based, non-intrusive technique could potentially be used for real time quantification. Therefore, since the response time is a crucial parameter, it is

Table 9
Resulting best-performing CNN architecture and hyperparameters obtained through the AutoML optimization.

Hyperparameter	Best value	
	Direct	Indirect
Architecture – Convolutional Layers		
Number of convolutional blocks	2	2
Number of consecutive convolutional layers per block	2	2
Convolutional kernel size	3	3
Number of filters in layer #1	128	16
Number of filters in layer #2	64	32
Number of filters in layer #3	32	64
Number of filters in layer #4	32	16
Depthwise separable convolutions?	No	No
Apply max-pooling?	No	No
Dropout rate	50 %	50 %
Architecture – Reduction Layers		
Spatial reduction layer type	Flattening	Flattening
Architecture. Dense Layers		
Use batch normalization?	No	No
Number of dense layers	2	2
Number of units in layer #1	32	32
Number of units in layer #2	32	16
Dropout rate	0	0
Optimizer		
Optimizer algorithm	Adam	Adam
Learning rate	1×10^3	1×10^3

worth mentioning that the CNN architecture obtained with AutoML is significantly simpler and smaller than the ones traditionally tested and used as, for example, the baseline CNN, i.e., Ref. [8]. For instance, the baseline model employed for direct visualization in Section 4 had 23,041,233 trainable parameters, while the AutoML-optimized architecture had only 850,401, which represents a reduction of over 96 % in the model size. For indirect visualization, the number of trainable parameters went from 13,825,233 to 3,307,121, which represents a reduction of approximately 75 %.

By comparing the architectures of the baseline and the AutoML-optimized models, Figs. 5 and 8, respectively, important conclusions and general guidelines can be derived. More specifically, the AutoML models have a larger number of convolutional layers and dense blocks, which favors deeper rather than wider model architectures; according to Ref. [18] deeper models are better at learning abstract features, which improves their generalization capabilities. This trend becomes clear when analyzing the relation between model size (i.e., number of trainable parameters of the CNN) and performance, as shown in Fig. 9 for the direct and indirect visualization. In Fig. 9, the baseline model is represented by the orange symbol, while all cases tried by the AutoML are represented by the blue symbols, and the best architecture is represented by the green symbol. The results for both visualization modes show that the AutoML tries architectures of very different sizes, including architectures with the same size as the baseline model, which was set as upper bound for the AutoML search. However, it is visible that there are numerous configurations that are smaller than the baseline, but still are capable of outperforming it. Another interesting aspect is that there are architectures with the same size, but very different performances; this can be seen through sequences of vertically aligned blue symbols. This indicates that CNN models need to have not only their architecture optimized, but also other hyperparameters such as the learning rate and the optimizer algorithm. Finally, it is important to mention that Fig. 9 also shows the existence of trade-offs between performance and the number of trainable variables, where smaller models only represent marginal increases in the error metrics. Note that this statement is not conflicting with previous discussion since, thus far, our goal was to find the best-performing architecture, which would return the lowest error metrics while being smaller than the baseline model. However, considering one decides, for example, that a MSE of 10 $(W/cm^2)^2$ for direct visualization is acceptable – recall that $\sim$ 10 $(W/cm^2)^2$ is much larger than the 4.23 $(W/cm^2)^2$, which is the lowest value obtained –, Fig. 9a shows that at least three different model sizes satisfy this constraint: $\sim 10^5$, $\sim 10^6$ and $\sim$ 10^7. Naturally, computational demand for these configurations will change drastically, and, for the MSE of 10 $(W/cm^2)^2$, a smaller model is the best option.

Fig. 10a and b show the multiple surface heat maps obtained with the AutoML-optimized models for direct and indirect visualization, respectively; for comparison, the equivalent version of this figure for the baseline architecture was shown in Fig. 7. Generally speaking, the overall trends are similar to that observed in Fig. 7. For instance, the main diagonal shows the smallest MSE values, since testing is done with the same dataset used for training. Also, the overall performance for the direct visualization mode is better than that of the indirect visualization. Additionally, models trained with combined datasets perform well on specific datasets as long as the heating surface being tested is also part of the training dataset. Conversely, even AutoML-optimized models are not capable of performing well with respect to a specific surface if data from that heating surface was not used for training. For instance, the model trained on the dataset containing examples from all heating surfaces ($\overline{U}^A$) is the best when evaluated on all heating surfaces, achieving a validation MSE of around 9 $(W/cm^2)^2$ and 29 $(W/cm^2)^2$ in the direct and indirect visualization modes, respectively, representing decreases of $\sim$ 57 % and 51 % with respect to the baseline architecture.

6. Conclusions

The present study explores two novel aspects associated with the use of a non-intrusive technique that relies on uncorrelated images to train CNN models with the objective of quantifying key variables of two-phase processes, such as pool boiling heat transfer. The novel aspects are the use of four different heating elements and the optimization of the CNN architecture and its hyperparameters through AutoML. The image dataset for each one of the four heating surfaces (namely, the small and large diameter wires, and the vertical and horizontal ribbons) was obtained with a classic pool boiling experiment developed and reported in Ref. [8]. Each dataset went through several preprocessing steps aiming to reduce the images' dimensionality and to standardize the images' features (i.e., size adjustment through cropping). Two datasets were created for each heating surface: one where the heating surface is visible (direct visualization mode) and another one where the heating surface was cropped out (indirect visualization mode). Finally, the analyses used the CNN architecture proposed in Ref. [8] as reference or baseline.

There are five main conclusions within this study:

- CNN models that are trained with the dataset of a specific heating surface perform well when quantifying the heat flux based on images of that surface, but perform poorly when quantifying the heat flux of other surfaces whose images were not part of their training dataset; i.e., generalization is limited.
- On the other hand, if a combined dataset composed of images of different surfaces is used for training, the CNN will perform well when quantifying the heat flow of all heating surfaces considered in the training. While this trend is valid for both visualization modes, the direct visualization mode outperforms the indirect visualization mode;
- CNN architectures trained with a diverse dataset (images from different heating surfaces) might outperform a CNN architecture that is trained on the dataset for a single heating surface; in this case, both training subsets of the surface being evaluated are of the same size;
- The CNN architecture optimized with AutoML outperforms the baseline architecture used by Ref. [8], decreasing the validation error by 73 % in direct visualization and 80 % in indirect visualization;
- The CNN architecture optimized with AutoML is significantly different from the baseline architecture used by Ref. [8]. Generally

Fig. 8. Flowchart representation of the AutoML-optimized CNN architecture for the direct (left) and indirect (right) visualization modes.

speaking, the optimized CNN has more convolutional layers and dense blocks. Also, the number of trainable parameters is much smaller than that of the baseline CNN, which translates as less training and inference time as well as a smaller memory footprint. More specifically, the number of trainable parameters of the optimized CNN is roughly 96 % smaller when compared with the baseline for direct visualization, and 73 % smaller for indirect visualization;

Finally, it is worth mentioning that, while image-trained machine-learning methods have proven to be fairly performant at classifying and quantifying several parameters of flow systems, e.g. [5–8,21], it is fundamental that future studies focus on generalizing the capabilities of these algorithms, making them even more versatile, hence the importance of extending and testing them while subjected to diverse datasets. Also, studies often focus on comparing the efficiency of machine learning techniques. However, attention is needed since the models being compared might not be fully optimized. In this scenario, AutoML

Table 10
Error associated with the use an AutoML-optimized CNN to quantify pool boiling heat flux based on images of the D^{LW} heating surface.

	Metric	Unit	This work		
			Training	Validation	Test
Direct	MSE	$(W/cm^2)^2$	2.03	4.23	4.42
	RMSE	W/cm^2	1.426	2.057	2.102
	MAE	W/cm^2	1.07	1.50	1.52
	MAPE	%	2.70	3.63	3.67
	R^2	–	0.9949	0.9893	0.9888
Indirect	MSE	$(W/cm^2)^2$	1.19	6.71	7.11
	RMSE	W/cm^2	1.092	2.591	2.666
	MAE	W/cm^2	0.84	1.76	1.77
	MAPE	%	2.17	4.13	4.15
	R^2	–	0.9970	0.9830	0.9820

is a potential optimization tool for finding performant models, and, for sake of continuity, researchers are encouraged to fully disclose all hyperparameters used in their machine-learning algorithms. Also, and in order to minimize computational work, it might be interesting to explore machine learning models that were trained with raw images obtained with a known low resolution, which has been shown to possess the required information for such algorithms to return qualitative and quantitative information, e.g., [5,6,14]. Furthermore, the use unsupervised image-to-image translation, e.g., [16], might help overcome the generalization issues observed in the present study.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence

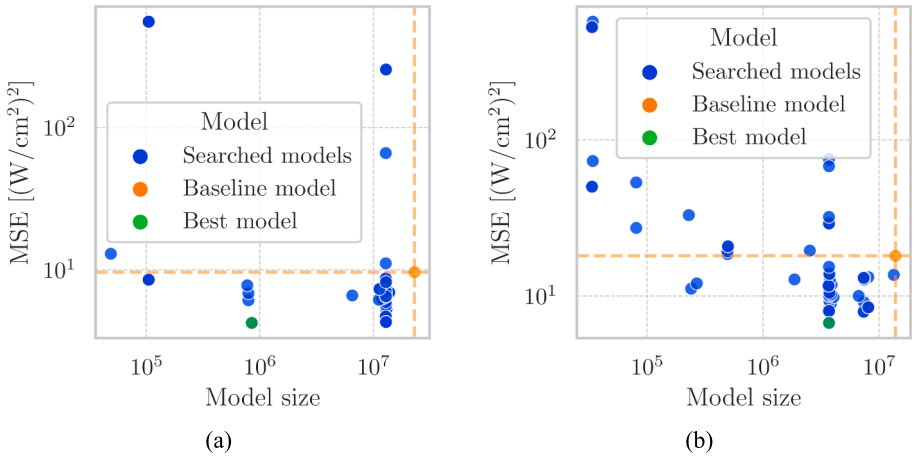


Fig. 9. Effect of the model size (i.e., number of trainable variables) on the MSE for direct (a) and indirect (b) visualization modes.

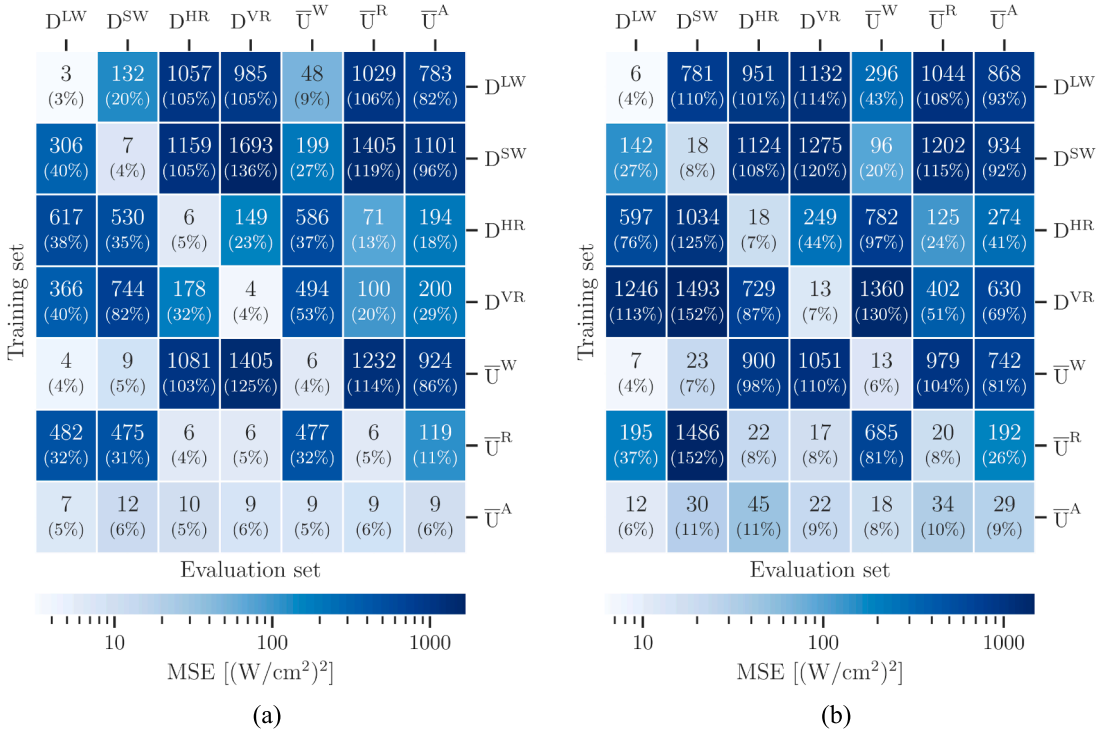


Fig. 10. Error correlation heat maps between training (rows) and validation (columns) datasets considering the AutoML-optimized CNN; frame (a) represents the direct and frame (b) the indirect visualization. The MSE is the upper value in each cell and the MAPE is the value between brackets. The blue scale represents the MSE and allows a visual comparison between all cells of the heat map.

the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

RCC and AKDS appreciate the financial support from CAPES (Brazil) and CNPq (Brazil), respectively.

References

[1] T.L. Bergman, A.S. Lavine, F.P. Incropera, D.P. Dewitt, Fundamentals of Heat and Mass Transfer, 7th ed., John Wiley & Sons Inc, New York, 2011.
[2] R.E. Sonntag, C. Borgnakke, G.J. Van Wylen, Fundamentals of Thermodynamics, 6th ed., John Wiley & Sons Inc, New York, 2003.
[3] M.K. Seal, S.M.A. Noori Rahim Abadi, M. Mehrabi, J.P. Meyer, Machine learning classification of in-tube condensation flow patterns using visualization, Int. J. Multiph. Flow 143 (2021) 103755.
[4] F. Nie, H. Wang, Q. Song, Y. Zhao, J. Shen, M. Gong, Image identification for two-phase flow patterns based on CNN algorithms, Int. J. Multiph. Flow 152 (2022) 104067.
[5] L.H. Silva Junior, J.R. Barbosa, A.K. da Silva, Multi-parameter classification and quantification of r-134a condensation using machine learning, Appl. Therm. Eng. (2023) 120880.
[6] G.M. Hobold, A.K. da Silva, Machine learning classification of boiling regimes with low speed, direct and indirect visualization, Int. J. Heat Mass Transf. 125 (2018) 1296–1309.
[7] G.M. Hobold, A.K. da Silva, Automatic detection of the onset of film boiling using convolutional neural networks and Bayesian statistics, Int. J. Heat Mass Transf. 134 (2019) 262–270.
[8] G.M. Hobold, A.K. da Silva, Visualization-based nucleate boiling heat flux quantification using machine learning, Int. J. Heat Mass Transf. 134 (2019) 511–520.
[9] M. Ravichandran, M. Bucci, Online, quasi-real-time analysis of high-resolution, infrared, boiling heat transfer investigations using artificial neural networks, Appl. Therm. Eng. 163 (2019) 114357.
[10] J.H. Seong, M. Ravichandran, G. Su, B. Phillips, M. Bucci, Automated bubble analysis of high-speed subcooled flow boiling images using U-net transfer learning and global optical flow, Int. J. Multiph. Flow 159 (2023) 104336.
[11] M. Ravichandran, A. Kossolapov, G.M. Aguiar, B. Phillips, M. Bucci, Autonomous and online detection of dry areas on a boiling surface using deep learning and infrared thermometry, Exp. Therm Fluid Sci. 145 (2023) 110879.
[12] Y. Suh, R. Bostanabad, Y. Won, Deep learning predicts boiling heat transfer, Sci. Rep. 11 (1) (2021) 5622.
[13] D. Lu, Y. Suh, Y. Won, Rapid identification of boiling crisis with event-based visual streaming analysis, Appl. Therm. Eng. 239 (2024) 122004.
[14] V.K. Scariot, G.M. Hobold, A.K. da Silva, Data-driven diagnostics of boiling heat transfer on flat heaters from non-intrusive visualization, Appl. Therm. Eng. 248 (2024) 123068.
[15] M. Ravichandran, G. Su, C. Wang, J.H. Seong, A. Kossolapov, B. Phillips, M. M. Rahman, M. Bucci, Decrypting the boiling crisis through data-driven exploration of high-resolution infrared thermometry measurements, Appl. Phys. Lett. 118 (25) (2021).
[16] F. Al-Hindawi, T. Soori, H. Hu, M.M.R. Siddiquee, H. Yoon, T. Wu, Y. Sun, A framework for generalizing critical heat flux detection models using unsupervised image-to-image translation, Expert Syst. Appl. 227 (2023) 120265.
[17] H. Jin, Q. Song, X. Hu, Auto-Keras: An Efficient Neural Architecture Search System, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Association for Computing Machinery, Anchorage, AK, USA, 2019, pp. 1946–1956.
[18] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning MIT Press, Cambridge, MA, 2016.
[19] R.C. Comelli, Boiling Learning, GitHub repository, https://github.com/ruancomelli/boiling-learning, 7036583fb510ba0e2ffc8526c719190c4a005444 (2023).
[20] J.P. Holman, Experimental Methods for Engineers, McGraw Hill, New York, 1994.
[21] L.H. Silva Junior, A.K. da Silva, Non-intrusive, real-time deep learning-based pollution analysis applied to open-channels, J. Braz. Soc. Mech. Sci. Eng. 43 (8) (2021) 388.
[22] W. Zhou, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, tensorflow.org (2015).
[24] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, arXiv, (2014).
[25] Python Software Foundation. Python Language Reference, version 3.10. Available at http://www.python.org.
[26] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi, e. al., KerasTuner, https://github.com/keras-team/keras-tuner, (2019).
[27] K.P. Murphy, Machine learning: a probabilistic perspective, The MIT Press, Cambridge, MA, 2012.
[28] V.V. Yagov, Nucleate boiling heat transfer: possibilities and limitations of theoretical analysis, Heat Mass Transf. 45 (7) (2009) 881–892.