

Deep learning strategies for critical heat flux detection in pool boiling

Seyed Moein Rassoulinejad-Mousavi^a, Firas Al-Hindawi^{b,c}, Tejaswi Soori^a, Arif Rokoni^a, Hyunsoo Yoon^d, Han Hu^e, Teresa Wu^{b,c}, Ying Sun^{a,*}

^a Department of Mechanical Engineering and Mechanics, Drexel University, Philadelphia, PA 19104, USA

^b School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281, USA

^c ASU-Mayo Center for Innovative Imaging, Arizona State University, Tempe, AZ 85281, USA

^d Department of Systems Science and Industrial Engineering, Binghamton University, State University of New York, Binghamton, NY 13902, USA

^e Department of Mechanical Engineering, University of Arkansas, Fayetteville, AR 72701, USA



ARTICLE INFO

Keywords:

Critical heat flux
Deep learning
Transfer learning
Convolutional neural network
Pool boiling

ABSTRACT

Image-based deep learning (DL) models are employed to enable the detection of critical heat flux (CHF) based on pool boiling experimental images. Most machine learning approaches for pool boiling to date focus on a single dataset under a certain heater surface, working fluid, and operating conditions. For new datasets collected under different conditions, a significant effort in re-training the model or developing a new model is required under the assumption the new dataset has a sufficient amount of data. This research is to explore strategies of DL adapting to new datasets with limited data available. The insights gained could help improve the practicality and reliability of DL for boiling regime studies. Specifically, convolutional neural networks (CNN) and transfer learning (TL) are studied. Using a base model trained and tested for one public dataset (DS1), CNN and TL models are trained with a small portion of a new public dataset (DS2) and tested for the rest of DS2. Results show that TL outperforms CNN by having much higher accuracy and a much lower false negative rate for scarce data (less than 5% DS2). When 1% DS2 is used for re-training in CNN versus fine-tuning in TL, the TL model can detect the CHF with an accuracy of 94.79% and a false negative rate of 0.0997, compared with the CNN model with an accuracy of 85.10% and a false negative rate of 0.3237. To further demonstrate the advantage of TL over CNN, an in-house dataset (DS3) is acquired. With less than 0.05% DS3 being used, the TL model can detect the CHF with an accuracy of 95.31% and a false negative rate of 0.0016, compared with the CNN model with an accuracy of 85.91% and a false negative rate of 0.1263. It is observed that TL has much higher robustness than CNN by having more consistent results and smaller standard deviations over multiple trials using stratified random sampling from both DS2 and DS3. Besides, the training time for TL is significantly lower than CNN when limited data used in the re-training and fine-tuning for both DS2 and DS3. These results demonstrate the ability of TL for handling data scarcity in pool boiling applications with potentials for real-time implementations.

1. Introduction

Boiling is a central phenomenon in many industrial applications including steam generation in power plant boilers and solar collectors [1], immersion cooling for high-performance electronics and data centers [2,3], integrated cooling for three-dimensional electronic packaging [4], cooling of the core and used fuel in nuclear reactors [5], among others. Most boiling applications are focused on the nucleate boiling regime with ultra-high heat transfer coefficients (HTC), whereas a catastrophic point of failure, known as the critical heat flux (CHF), limits the heat flux of nucleate boiling. Beyond CHF, boiling transits from the

nucleate boiling regime to the film boiling regime, where the heater surface is blanketed with a continuous vapor layer, leading to a drastic decrease in HTC. Failing to dissipate the heat load may lead to detrimental device failures, e.g., thermal breakdown in semiconductor junctions, the meltdown in nuclear reactors, etc. To ensure safe operations and enable the rational design of boilers, evaporators, and other devices that utilize the boiling process, it is critical to be able to detect or predict the CHF condition. However, the complex and stochastic nature of the subprocesses involved in boiling, such as bubble nucleation, growth, coalescence, and departure makes it extremely challenging to develop a comprehensive physics-based model for CHF during boiling

* Corresponding author.

E-mail address: ys347@drexel.edu (Y. Sun).

[6]. To date, theoretical models and experimental correlations for boiling CHF, e.g., the hydrodynamic model [7,8], the force-balance model [9], and wicking-based CHF models [10,11], differ from each other substantially. Recently, a continuum percolation model for the CHF detection has been proposed based on near-wall stochastic interactions of bubbles, but has only been tested for a smooth surface [12]. A general deterministic model for predicting the CHF condition that can be applied for a large range of physical parameters is not yet available.

Conventionally, boiling regimes are determined based on the experimentally measured heat flux and surface temperature. For example, the onset of nucleate boiling can be recognized as a kink in the boiling curve that represents the transition from the low-HTC convection regime to the high-HTC nucleate boiling regime [13]. The CHF is represented by a huge jump in surface temperature for boiling experiments with controlled heat flux and decreasing heat flux for experiments with controlled surface temperature [13]. High-speed visualizations of pool boiling experiments have enabled the development of regime maps based on bubble morphologies, corresponding to different boiling phenomena [14–16]. Although such analysis is shown to be successful in lab-scale experiments, the time required for the postprocessing of the images to derive the quantitative metrics prevents it from real-time prediction or online classification of boiling processes in practical applications.

Data-driven approaches provide an alternative to physics-based models by taking advantage of the available boiling data and the advances in machine learning (ML) algorithms. Using pressure, temperature, and heat flux measurements from the boiling experiments, machine learning models are developed to predict flow regimes, pressure drop, void fraction, critical heat flux, the onset of nucleate boiling, nucleation site density, bubble period, bubble growth time, and heat transfer coefficient [17–25]. Recognizing boiling images contain rich information on the dynamics of the ebullition cycle, increasing efforts on developing machine learning tools using visualization images are noted [22–25]. For example, Hobold and da Silva [22] used support vector machines (SVM) and multi-layer perceptron neural network (MLPNN), a class of artificial neural networks (ANN), to classify on-wire pool boiling experimental images (direct visualization) into three boiling regimes: natural convection, nucleate boiling, and film boiling. Classification accuracy for both SVM and MLPNN was greater than 93% even when the pool boiling heater surface was cropped out of the images (indirect visualization). In Ravichandran and Bucci [24], a feed-forward neural network (FFNN) was developed as an online, quasi-real-time analyzer of the infrared image from the heater surface to quantify bubble growth time, bubble period, and nucleation site density, showing a regression coefficient of 0.95 or higher compared to conventional image processing techniques.

The application of deep learning (DL) models such as convolutional neural networks (CNN) emerges due to their recognized performances in analyzing visual imagery [26,27]. CNN has the advantage of learning from multiple features, known as filters, which require less pre-processing compared to other image-based classifiers. CNN benefits from learnable weights and biases in layers of neurons with a massive number of features and flexible network architecture. The functional and efficient controlling parameters in the hidden layers help users to optimize network configuration by varying those hyper-parameters. In addition to the very well-known application of CNN in facial recognition systems [28], it has been adopted as a popular tool in biomedical image analysis [29–33] and in engineering applications such as combustion [34,35], aerodynamics [36,37], digital designs [38], mechanical property predictions [39], and damage and fault diagnosis and detection [40–44]. In the applications related to this research, Hobold and da Silva [23] developed CNN in conjunction with Bayesian statistics and successfully detected the transition from nucleate boiling to film boiling, again using on-wire pool boiling experimental visualizations. Efforts have also been made on using CNN to quantify heat flux values from pool boiling images where the heat flux information is embedded in the

bubble morphologies seen during the boiling phenomena. Hobold and da Silva [25] showed that correlations can be drawn between bubble morphologies seen during on-wire pool boiling experiments and experimentally measured heat flux values even when the resolution and frequency of image acquisition were reduced. This study confirms that CNN-based boiling regime classification models have the potential for quantitative characterization of the boiling process.

While promising, the above-mentioned research has mainly focused on a single dataset collected under one condition of heater surfaces, working fluids, operating parameters. A comprehensive investigation on the boiling regime would require heater surface modifications like microstructures, porous metallic surfaces, wettability alterations, and others, which will give rise to different bubble morphologies when used in a pool boiling setup. Similar behaviors are observed when working fluid is changed. To tackle this challenge, one option is to train, validate, and test a new DL model when a new dataset is being collected under different conditions. This is less than desirable because (1) it ignores the knowledge already obtained from the existing DL models; (2) it is under the assumption the new dataset has sufficient data; and (3) it is computationally expensive. Alternatively, this challenge can be constructed as a Transfer Learning (TL) problem where the existing datasets are treated as the source domain with the new dataset being considered as the target domain [45–47]. TL promotes the idea of dividing the traditional training process into two separate steps: pre-training is conducted on the source domain on all the layers from the deep model, with later layers being fine-tuned using data from the target domain. This is supported by the assumption that the knowledge gained from various source domains may help the learning task on the target domain. While the TL concept has been widely adopted in different applications such as medical imaging [45,48–51], the applicability of TL for heat exchanger design (such as boiling CHF detection) has yet been unexploited. We hypothesize that TL is a viable strategy to study boiling imagery data, especially when the target domain has limited data available.

Here we first develop and train a CNN model for boiling regime classification with one public boiling dataset (DS1) and use this CNN as the base model. To classify images of new boiling datasets, two sets of comparison experiments are designed where a TL model is developed based on the base model and is fine-tuned with a small proportion of another public dataset (DS2) in Comparison Experiment I and an in-house dataset (DS3) in Comparison Experiment II. A reference case is developed by combining the same small proportion of DS2 or DS3 with DS1 to retrain the CNN model. In the first experiment, the accuracy and training time of TL and the retrained CNN (reference case) are compared for various proportions of DS2, decreasing from 10%, 5%, 2%, 1.5% to 1%. Noting the dominating trends of TL over CNN as data from the target domain decreases, in the second experiment, only less than 0.05% of data from DS3 is studied for the comparison. Since the small proportion for training is randomly selected from DS2 or DS3, the statistical behavior of TL and CNN is analyzed over 15 trials per experiment.

2. Methodology

2.1. Dataset preparation for CNN and TL training and testing

Three different pool boiling experimental image datasets (DS1, DS2, and DS3) were prepared in this study where DS1 and DS2 were generated using publicly available YouTube videos [52,53], and DS3 was prepared by performing in-house pool boiling experiments. Specifically, the video from which DS1 was prepared shows a pool boiling experiment performed using a square heater made of high-temperature, thermally-conductive microporous coated copper where the surface was fabricated by sintering copper powder [54–57]. The square heater had a surface area of $\approx 100 \text{ mm}^2$ and the working fluid used was water. All experiments were performed at a steady-state under an ambient pressure of 1 atm. A T-type thermocouple was used for temperature

measurements. The resolution of the video frames was 512×480 pixels. The YouTube video from which DS2 was prepared shows a pool boiling experiment performed using a circular heater made of microporous coated copper where the surface was fabricated by sintering copper powder [58–60]. The circular heater had a diameter of ≈ 16 mm and the working fluid used was DI water. All experiments were performed at a steady-state under an ambient pressure of 0.5 atm. A T-type thermocouple was used for temperature measurements. The resolution of the video frames was 1280×720 pixels. The DS3 videos were obtained by performing pool boiling experiments in-house using a square plain copper heater with a surface area of ≈ 100 mm 2 , with water as the working fluid. Prior to the experiments, the surface was prepared by polishing with a 320 grit sandpaper and subsequently with a 600 grit sandpaper. A Phantom VEO-710 high-speed camera captured images at 3000 frames per second with 1280×800 resolution.

Images for DS1 and DS2 were prepared by downloading the videos from YouTube and extracting individual frames using a MATLAB code via the VideoReader and imwrite functions. Recognizing duplicate frames extracted from the YouTube videos, quality control was conducted to remove the repeated images by calculating the relative difference using the Structural Similarity Index (SSIM) value between two consecutive images where images with a relative difference less than 0.03% were removed [61]. This pre-processing is important to ensure DL models were not biased by identical image frames. Images for DS3 were extracted from high-speed videos captured during in-house pool boiling experiments directly without pre-processing as no-repeat frames were observed. The images were categorized into three boiling regimes: (1) Discrete bubbles (DB) where only discrete bubbles are observed before departure, (2) Bubble interference and coalescence (BIC) where frequent bubble coalescence is observed before departure, and (3) Critical heat flux (CHF) where a significant drop in the heat transfer coefficient is

observed due to a continuous vapor layer blanketing the heater surface. While the images of DS1 and DS2 have been labeled by the authors of the datasets, the images of DS3 are labeled based on the boiling curve and temperature history from our experiments. The representative bubble images in the three boiling regimes are shown in Fig. 1, DS1, DS2, and DS3 had a total of 6158, 3215, and 34,422 2D images, respectively. Table 1 shows the number of images in each regime for the three datasets. The pixel intensity values in each image were normalized to fit in the range [0,1] to ensure uniformity over multiple datasets during deep learning training.

2.2. CNN-base and CNN-TL: Convolutional neural network model and transfer learning model architecture

Once the data are ready for feeding into the network, it is important to build a robust architecture, including layers and settings parameters. After trying different architectures and manipulating the number of convolutional, max pooling, and dropout layers; an optimum configuration was selected according to the accuracy of training based on monitoring the loss (the discrepancy from actual results and predictions) and accuracy (what percent of predictions were correct) for both training, and test. Fig. 2(a) illustrates the CNN architecture adopted in

Table 1
Summary of the three Datasets.

Datasets	DB	BIC	CHF
Dataset 1 (DS1) [52]	2304	3068	786
Dataset 2 (DS2) [53]	693	1289	1233
Dataset 3 (DS3) in-house	11,488	10,890	12,044

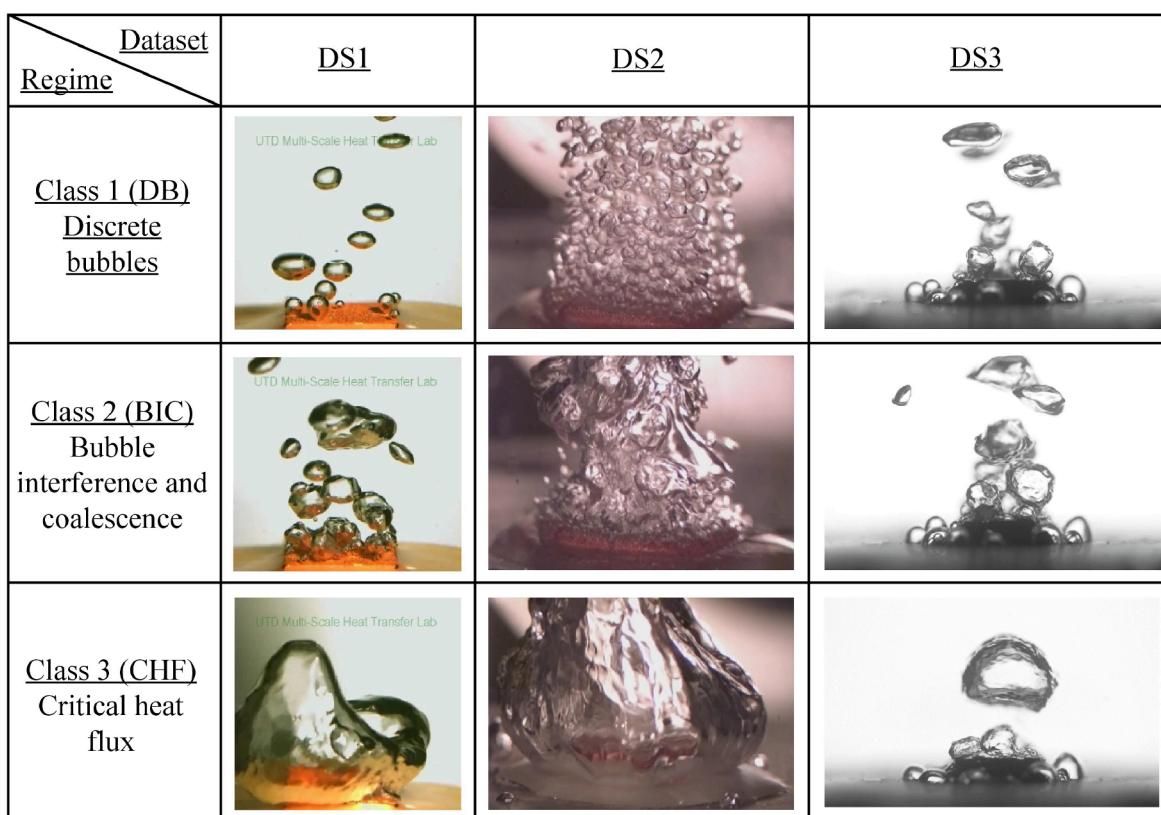
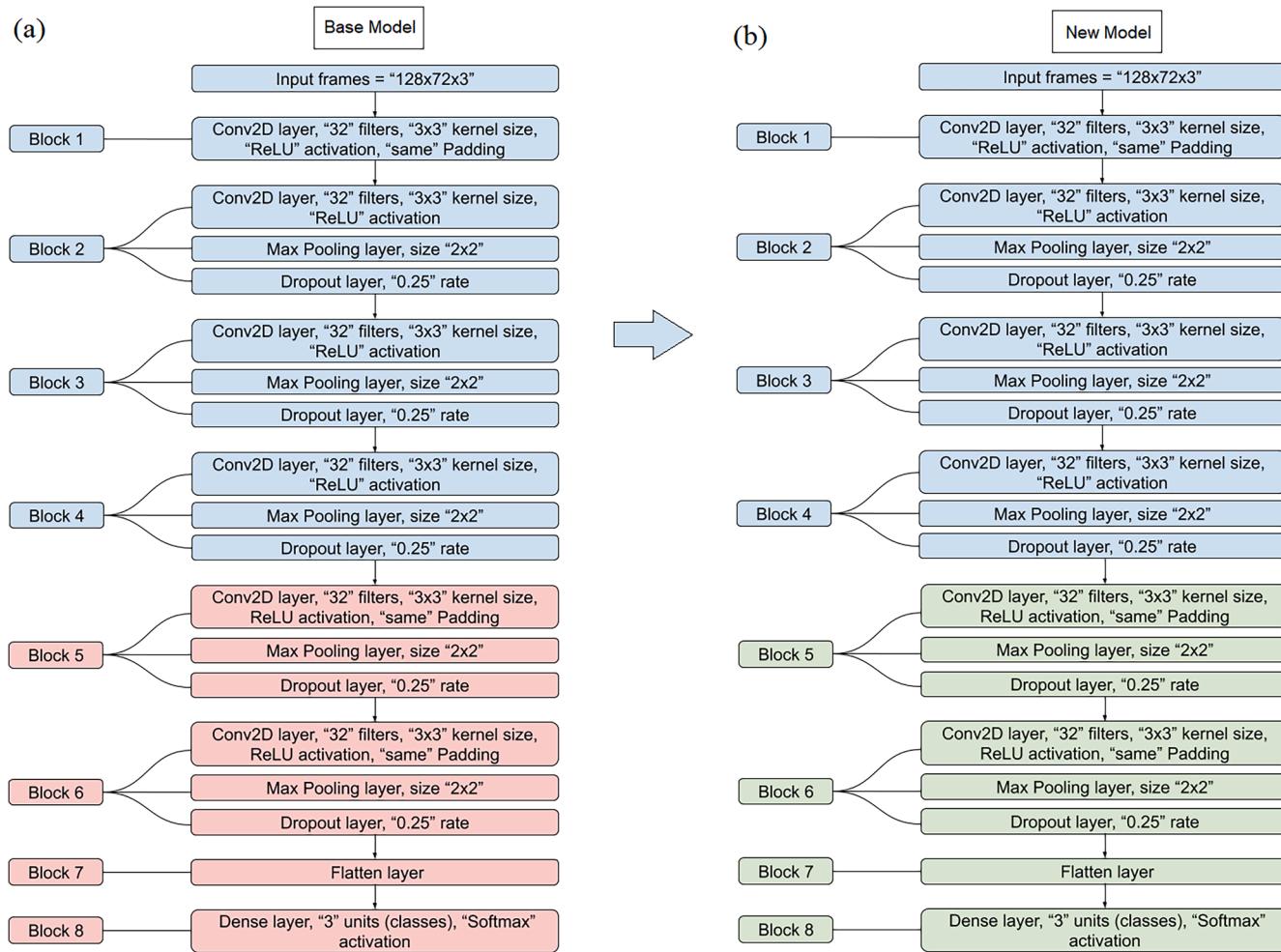


Fig. 1. Representative images of bubble dynamics observed from source videos. The three rows indicate the classes (regimes) used for ML classification. Regimes used are: Discrete bubbles (DB), Bubble interference and coalescence (BIC), Critical heat flux (CHF). Multiple images with various patterns from different experiments were shown for each specific regime. Images for DS1 and DS2 were extracted from various online sources open to public access [52,53], and images for DS3 were obtained from in-house pool boiling experiments.



- █ The layers in blocks 1,2,3 and 4 were all transferred from the base model to the new model, where their weights were frozen and untrainable.
- █ The layers in blocks 5,6,7 and 8 in the base model were all removed and weren't used in training the new model.
- █ New trainable layers were added to blocks 5,6,7 and 8 in the new model, with an architecture identical to the layers originally removed from the base model.

Fig. 2. The TL procedure operates on (a) Base Model architecture and (b) New Model architecture. The TL architecture was prepared by 1) transferring the layers in blocks 1 → 4 and their trained parameters stored after training the base model (a) purely on DS1, from the base model to the new model. These layers were then frozen (non-trainable) in the new model (b). Usually, the first layers contain the low-level features that were learned by the model. 2) layers in blocks 5 → 8 were then removed from the base model (a) along with their stored parameters and were replaced by new layers with new trainable parameters (unfrozen) in the new model (b).

this research. As seen, two convolutional blocks, each with two convolutional layers and a rectified linear unit (ReLU) activation functions, were used. ReLU is defined as the positive part of its argument and is the most used activation function, which returns 0 if it receives any negative input but returns the exact value of any positive input back. The two convolutional layers in the first and second blocks have 32 and 64 kernels, respectively. These kernels (also known as filters) are the vector of weight and bias, apply to every pixel of an input image. Max pooling layer for down-sampling the feature maps with the highest representations of an input image and the Dropout layer for preventing the overfitting (accurate only on training data) were used in each block. Then, a fully connected (dense) layer with 256 nodes with a Rectified linear unit (ReLU) activation function was used to interpret the features. ReLU is a widely used activation function to ensure model output does not go below zero

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}, \quad x = b + \sum_{j=1}^N a_j w_j \quad (1)$$

$f(x)$ is the activation function. x is the vector of inputs to the output layer. a is the input to the neuron, w is the weight associated with each neuron, and b is the bias as an additional input into the next layer. j is indexing the output units for the number of inputs, N . Finally, another dense layer with Softmax activation for the three boiling classes was added to the architecture to predict the probability distribution of a test image belonging to each of the three classes. The output of Softmax, $f(x)$, is mapped to a [0,1] range and the total sum is 1 for all classes following

$$f(x)_j = \frac{e^{x_j}}{\sum_j e^{x_j}}, \text{ for } j = 1, \dots, N. \quad (2)$$

It is well recognized in the data science community that model developed from one source of data (e.g., source domain) may perform

poorly on a separate data source (e.g., target domain), if data distributions between two domains are not similar. Transfer learning (TL) has been a viable technique to address this challenge [62]. Transfer learning can be integrated with deep learning (DL), resulting in a deep transfer learning framework that pre-trains a DL model using a large dataset and then uses the small dataset from the target domain to fine-tune the model parameters [48,62]. TL with DL models have been used for image classification in recent years and have shown promising results in broad areas such as brain image [45], breast cancer [48–50], mouse brain [51], and general image classification [63]. The efforts in deep transfer learning have been categorized into instance-based, mapping-based, network-based, and adversarial-based [63]. This research falls into the network-based category where we utilize the knowledge from the partial of network pre-trained in the source domain for the tasks from the target domain. In this research, the CNN model shown in Fig. 2(a) is used as the base model for TL. How to utilize knowledge from the sources is key to implementing TL to improve the performance of the target task. There are in general three strategies in TL, including parameter transfer, feature transfer, and instance transfer [46,47,62,64]. Instance transfer aims to reuse the samples (e.g., images), and feature transfer is to reuse the feature representations from the source to the target domain. Both do not apply here due to the interest of the study is the image. Therefore, parameter transfer is adopted, that is, the same size of input images for the sources (for pre-trained models) and target is kept, and deep model parameters are being transferred [45,49,65]. Please note each dataset consists of different characteristics of RGB colors, image intensities, and resolutions, etc. The CNN model contains the same low-level information such as edges, corners, emboss, angles like those in the desired target dataset. This information is borrowed via model parameters from the CNN model and then transferred to the new dataset to train the TL model. The TL procedure used in this study is illustrated in Fig. 2(b). The weights of layers in the first four blocks of the base model architecture were frozen and transferred to the new model with all the parameters (weights) fixed within them, while the weights of the layers in the latter four blocks were updated using the target dataset. The rationale behind this design is: there are a total of 65,891 parameters in the pre-trained deep learning model as the prior knowledge from the images regarding characteristics of boiling regimes. It is desirable that as much information from the source as possible and fine-tune them for the target task. If only the last layer parameters to be fine tuned, trainable parameters in the TL process are only 291 (0.44% of all parameters). This will be extremely challenging to adjust the model for the target images, especially when the images from source and target have significantly different characteristics. Specifically, images have different RGB colors and intensities, so that an additional layer to learn new RGB colors from the target is critical to adapt the deep learning model for the target. Also, one convolution layer is needed to help aggregate the updated outputs. Based on the empirical exploration, the model is configured as a total of 65,891 parameters out of which 37,888 (58%, layers 1–4) are transferred from the base model and are non-trainable (frozen), while 28,003 (42%, layers 5–8) of them are new trainable parameters.

In deep learning efforts, the implementation requires the following configurations: architecture, learning methods, hyperparameters, and optimizers to achieve a well-established model. Random initialization is critical to train the deep learning model. We use the standard initializers from Keras API [66], that is, Glorot normal initializer (a.k.a. Xavier normal initializer). To prevent the CNN and TL models from overfitting, although training in both models was stopped by reaching the predetermined epoch number, the model weights were not updated (saved) after every epoch unless there was an improvement in the validation loss in that epoch.

3. Results and discussion

In this section, the accuracy and generality of the trained CNN and TL models are evaluated and compared with each other. Among these

models, CNN 0 is the base model trained solely with the DS1. All transfer learning models are trained based on CNN 0 with a small portion of the DS2 and DS3. CNN models using the same architecture with the base model CNN 0 are also trained using a combined dataset that includes the DS1 along with the same small portion of the DS2 and DS3 as used for the TL models.

3.1. Base Experiment: CNN-base model

The process of classification using a trained network is that once a test image goes through a trained model, the model outputs its highest probability out of all classes as its prediction. It is thus important to examine the probability distribution for a test image to see how robust a model is classifying. Hence, some images that were neither in the training nor the validation dataset were tested (a.k.a. blind test) on the model to find the probability and accuracy of predictions. As mentioned earlier, the way that model classifies is by reporting the maximum probability of each class as its prediction. To test the generalizability of the CNN-base model, three experiments were conducted: (1) train a CNN-base model on DS1 (CNN-0) and test it on DS1; (2) train a CNN-base model on DS2 and test on DS2; (3) train a CNN-base model on DS1 (CNN-0) and test on DS2 (cross-dataset). The confusion matrices of these three tests are shown in Fig. 3(a)–(c), respectively. The first two models show high accuracy (100%), and the third model has only 40.09% accuracy. It is noted that Fig. 3(a) and 3(c) show the testing results of the same CNN-base model (CNN-0) on different datasets. The large difference in the accuracy indicates a low generality of CNN-0. To confirm that the low accuracy in Fig. 3(c) is not due to overfitting on DS-1, we have designed a 5-fold cross-validation (CV) experiment. For the rigorous design, the dissimilarities of DS1 and DS2 are examined first using principal component analysis (PCA) for DS1 and DS2. Fig. 4(a)–(c) shows the comparison between the two-component PCA decomposition of DS1 and DS2 in the (a) DB, (b) BIC, and (c) CHF regimes. Based on these plots, it is evident that the images of DS1 and DS2 have significant differences in their features. Fig. 4(d)–(f) shows the comparison of the 3-component PCA decomposition between DS-1 and DS-2. It is observed that the distribution of DS1 and DS2 are different along the third principal component as well. In sum, DS1 and DS2 have significantly different features and it should not be expected that a CNN model trained by DS1 would classify the regimes of DS2 images with high accuracy directly. After the dissimilarity between DS1 and DS2 is confirmed, a 5-fold CV is conducted, where the accuracy, precision, and F1-Score are all 100%. This result confirms that the low testing accuracy of CNN-0 on classifying DS-2 is not due to overfitting. Instead, it suggests that CNN-base models on unseen different datasets do not perform well, which is as expected. This motivates the comparison experiments on strategies of incorporating data from new datasets for improved performance.

3.2. Comparison experiment I: DS1 and DS2

Using the CNN-base model as CNN 0, five comparison experiments were conducted on two public datasets: DS1 vs. DS2. Table 2 summarizes the CNN and TL models trained with varying percentages of images from the DS2 dataset.

As seen in Table 2, the CNN model was trained using combined DS1 with different proportion data from DS2, from 10%, 5%, 2%, 1.5% to 1%. In a similar manner, for TL, after CNN is pre-trained on DS1, only layers 5–8 were fine-tuned with 10%, 5%, 2%, 1.5% to 1%, respectively. The remaining data from DS2 was reserved as test data for the blind test. Given the training data, 80% of the data was used for training vs. 20% for validation. Multifold cross-validation can help avoid biased results and evaluate the performance with robustness and generalization of the trained models. However, when it comes to CNN X vs. TL X (X = 1, 2, 3, 4, 5), multifold cross-validation becomes impractical. Since the scenario of interest in this study is only limited data available from different data

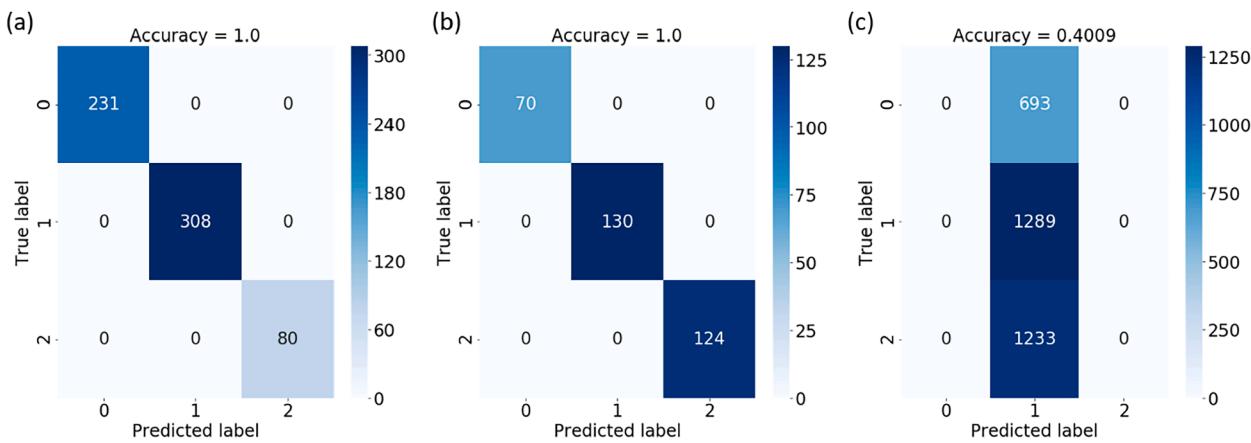


Fig. 3. Confusion matrices of (a) testing CNN trained with DS1 (CNN 0) for classifying DS1; (b) testing CNN trained with DS2 for classifying DS2; and (c) testing CNN 0 for classifying DS2.

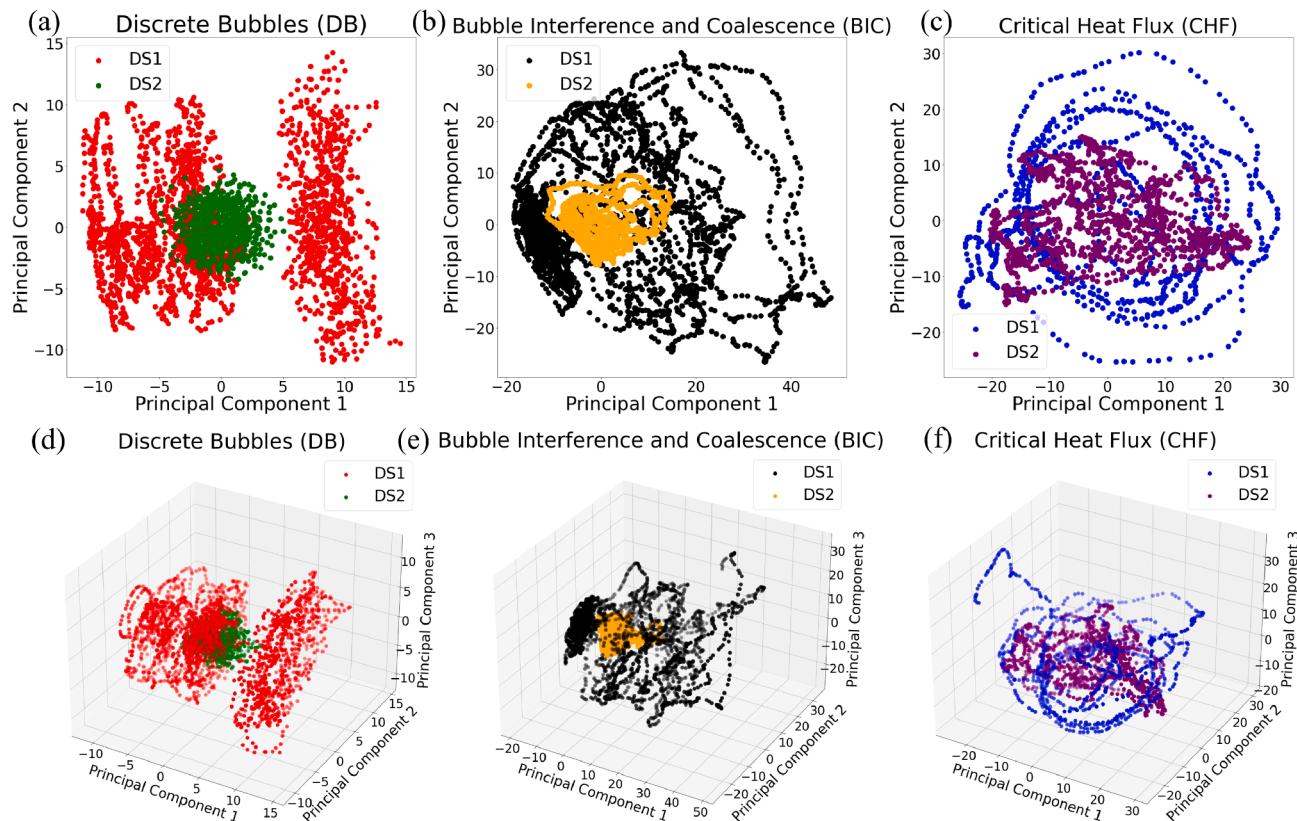


Fig. 4. Principal component analysis of DS-1 and DS-2 showing (a)-(c) two-component PCA decomposition of DB, BIC, and CHF images, and (d)-(f) three-component PCA decomposition of DB, BIC, and CHF images.

Table 2

Summary of the CNN-base and CNN-TL models in the cross-dataset (DS1, DS2) study.

CNN Model #	Training dataset	TL Model #*	Training dataset
CNN 0	DS1		
CNN 1	DS1 + 10% DS2	TL 1	10% DS2
CNN 2	DS1 + 5% DS2	TL 2	5% DS2
CNN 3	DS1 + 2% DS2	TL 3	2% DS2
CNN 4	DS1 + 1.5% DS2	TL 4	1.5% DS2
CNN 5	DS1 + 1% DS2	TL 5	1% DS2

*Note: All TL models are trained using CNN 0 as the base model.

sources (e.g., DS2, DS3), by design, we have a small number of images from the different data sources included in the training and validation. Taking 1% scenario as an example, given 100% of DS1 and 1% of DS2, we have (1) DB: 2304 (DS1), 7 (DS2); (2) BIC: 3068 (DS1), 13 (DS2); (3) CHF: 786 (DS1), 12 (DS2). If a 5-fold CV approach is to be taken, the training samples for TL (with fair comparison) are very limited (e.g., only one or two DB images from DS2 available for training). To address this challenge while still maintaining the confidence of model robustness, we decided to take a random sampling approach. In order to avoid any bias from sampling, 15 stratified random samplings were implemented, that is, 15 random splits in training, validation, and testing while keeping the same proportion of each class across splits. The model

was trained and validated using a trained dataset then blind tested on the testing dataset. The comparison metrics were collected from the 15 trials of the blind test. Please note cross-validation is well-adopted to address the overfitting issue, especially when the dataset is small. However, in this research, as the size of the dataset increases, the number of parameters in the deep learning model exponentially increases. And, the focus of the study is on a data scarcity scenario where only a small percentage from the new dataset is used for training and validation. Taking 1% as an example, a full-scale cross-validation implementation will require 65,891 parameters to be trained at least 100 times. For these reasons, stratified random sampling is adopted to address overfitting and guarantees the comparison is without bias.

Fig. 5 shows representative confusion matrices of the CNN and TL models trained with DS1 and a small proportion of DS2, with **Fig. 5(a), (b), (c), (d)** showing TL 1 with the highest accuracy, TL 1 with the lowest accuracy, CNN 1 with the highest accuracy, CNN 1 with the lowest accuracy, respectively. It is noted for TL 1 and CNN 1, where 1% of DS2 is included for training, all of the TL models retain a very high accuracy, with the highest to be 98.71% and the lowest to be 90.02%. On the contrary, CNN models have a huge spread of accuracy, with the highest at 98.24%, which is close to TL, but the lowest at 68.20%, much lower than the lowest accuracy of TL.

The accuracy of the CNN and TL models (averaged over 15 trials) are against the percentage of the DS2 data used for training in **Fig. 6(a)**. As shown in **Fig. 6(a)**, TL models have higher accuracy than CNN models for the tested entire range of used DS2 data (1–10%). At relatively higher percentages (10% DS2), the difference between TL and CNN accuracy is relatively trivial since both models give high accuracy (>99.50%). However, with the decreasing percentage of DS2 data, the accuracy of CNN drops significantly while the accuracy of TL is kept at a relatively high value. With only 1% of DS2 images used for training, the average accuracy of TL is 94.79%, which is more than 136% improvement compared to the base model with 0% DS2 images, and is well-beyond the average accuracy of CNN with 1% DS2 at 85.10%.

For the application of CHF detection, it is also important to determine how accurately the models detect CHF events. For the CHF class, the mistaken classifications include false negative, where CHF images are classified as either DB or BIC, and false positive, where DB or BIC images are classified as CHF. To mitigate the loss due to CHF, actions will be taken when CHF events are captured, e.g., reducing the heating load, activating supplementary cooling, etc. The false positive classifications will activate these actions when CHF does not occur, lowering the cooling efficiency. On the other hand, the false negative classifications will overlook CHF events, leading to overheating-induced device failures. It is evident that the false negative is more detrimental. As such, the false negative rate (FNR) for CHF becomes an important metric to evaluate the effectiveness of the machine learning models for CHF detection. The best performance is FNR = 0 and the worst is FNR = 1. **Fig. 6(b)** plots the false negative rate for CHF for both CHF-base and

CHF-TL models. The false negative for CHF of the base model is 1, demonstrating its poor performance. While the differences between the CNN and TL models for CNN are relatively trivial for models with 5% DS2 and 10% DS2, the FNR for TL models with 1%, 1.5%, and 2% DS2 is obviously smaller than the CNN models. The FNR of the CNN models increases significantly with the decreasing percentage of DS2 data while that of TL is kept at a low value (the average FNR for CHF less than 0.1 at 1% DS2). This indicates TL models yield lower FNR than CNN models for small percentages of DS2 data for training and are thus more reliable for applications of CHF detection.

As expected, with higher percentages of DS2 data used for training, both CNN and TL show better performances with increasing accuracy and decreasing false negative rate for CHF. However, as noted in **Fig. 6(a)**, CNN 2 (with 1.5% DS2) has a lower mean accuracy than CNN 1 (with 1% DS2). To examine the differences between the accuracy of CNN 1 and CNN 2 statistically, we have performed a *t*-test for the accuracy of 15 trials of CNN 1 versus 15 trials of CNN 2. The *t*-test gives a p-value of 0.89, indicating no statistically significant difference between the performances from the two models. On the other hand, the p-value for other pairs (CNN 2 vs. CNN 3, CNN 3 vs. CNN 4, and CNN 4 vs. CNN 5) is much smaller (≤ 0.05), supporting our conclusion with statistical significance. As such, the trend of decreasing accuracy with decreasing percentages of DS2 data is valid for CNN with 1.5% – 10% of DS2. The difference between the accuracy for 1% DS2 and 1.5% DS2 is not statistically significant enough to draw a conclusion.

It is noted that the analysis in **Fig. 6** is based on the accuracy and FNR for CHF averaged over 15 trials for each of the CNN and TL models. To better understand the robustness of the models, **Fig. 7** compares the statistical behavior of CNN (blue squares) and TL (red triangles) models for (a) the accuracy and (b) the FNR for CHF over varying percentages of DS2 for training. In the plots, the box shows the interquartile range (IQR), i.e., the range between the first quartile (Q1, 25th percentile) and the third quartile (Q3, 75th percentile), the line in the box represents the median, and the whiskers show the minimum (Q1 – 1.5 IQR) and the maximum (Q3 + 1.5 IQR). Data points located outside the whiskers are the outliers represented with the dots. The box plots in **Fig. 6a** and **6b** clearly show that the accuracy and the FNR for CHF of TL over 15 trials with random splits are converged into a much narrower range than CNN. This is an important advantage of TL over CNN that makes the predictions of TL models more reliable with smaller random errors. Combining the results of **Fig. 6** and **Fig. 7**, it is clear that TL outperforms CNN with higher accuracy, lower FNR, and much higher robustness at small percentages of DS2 (i.e., 1%, 1.5%, and 2%).

It is noted that all the CNN and TL experiments were conducted on 4 Nvidia GeForce GTX TITAN X graphics cards with 48 GB memory (each has 12 GB memory), Intel Xeon Gold 6128 CPU @3.40 GHz, 125 GB RAM, using Python 3.6.9, Keras 2.3.0 with TensorFlow 1.14.0 as backend. The training time of the CNN and TL models was recorded to evaluate the computational cost of the CNN and TL models. **Fig. 8**

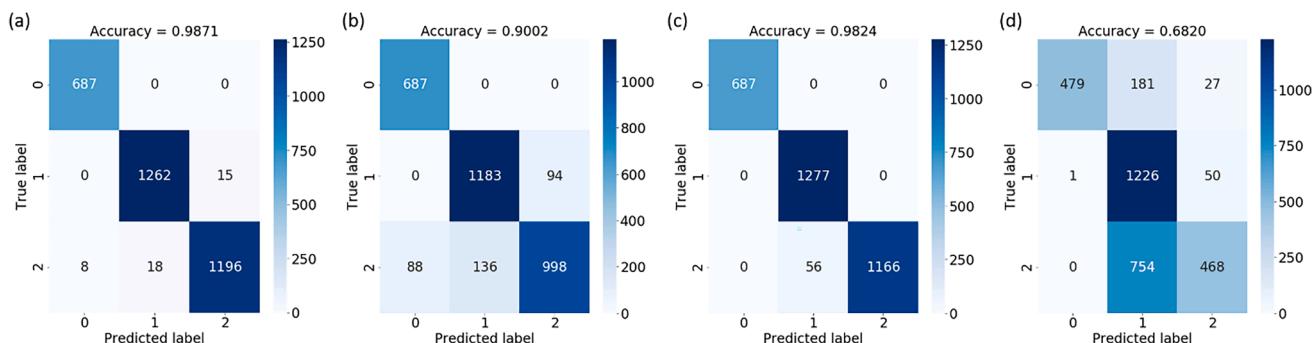


Fig. 5. (a-b) Confusion matrices of testing TL 1 for classifying DS2 with (a) highest accuracy and (b) lowest accuracy among 15 trials with randomly selected 1% DS2 for training; (c-d) Confusion matrices of testing CNN 1 for classifying DS2 with (a) highest accuracy and (d) lowest accuracy among 15 trials with randomly selected 1% DS2 for training.

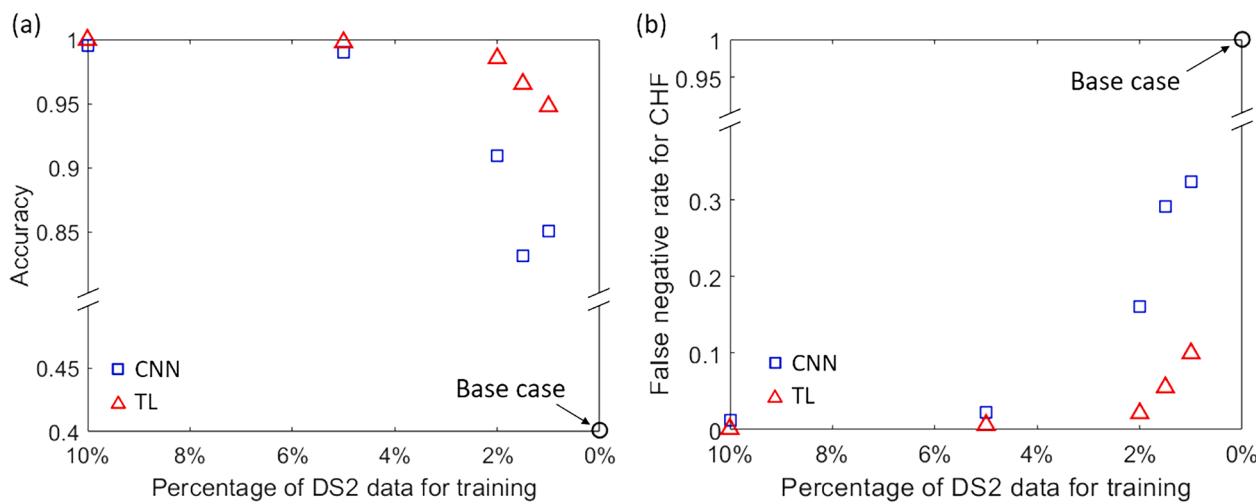


Fig. 6. Comparison between the CNN and TL for (a) the accuracy and (b) the false negative rate for CHF as a function of the percentage of DS2 data for training. The data points represent the mean values of 15 trials per experiment.

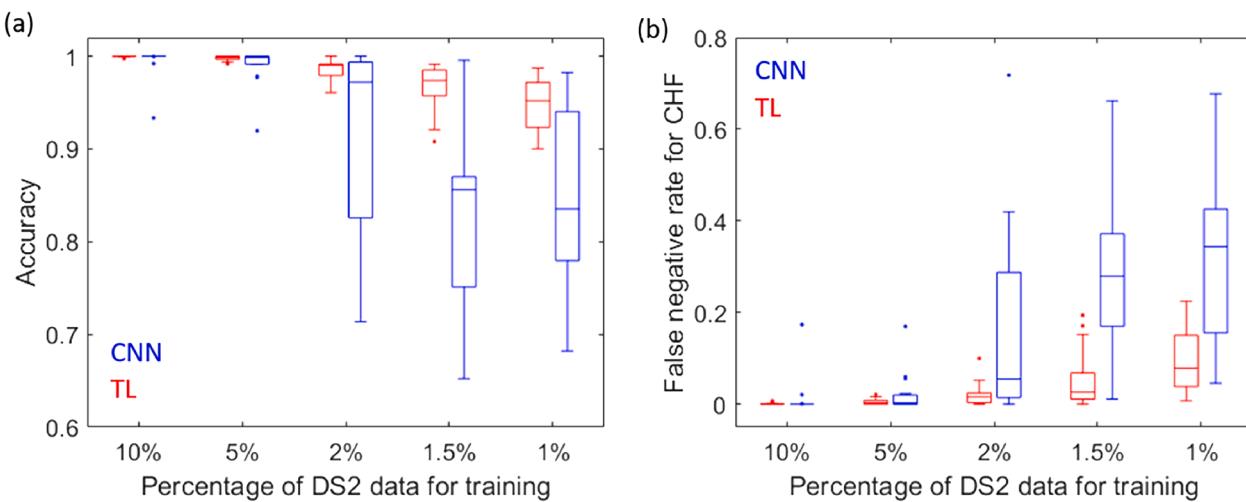


Fig. 7. Box-plot of (a) the accuracy and (b) false negative rate for CHF based on 15 trials of CNN and TL experiments with 1%, 1.5%, 2%, 5%, and 10% DS2.

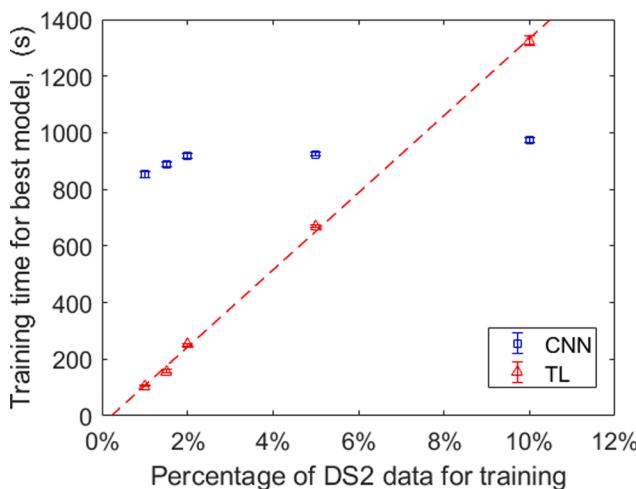


Fig. 8. Comparison between TL and CNN models for the computational time based on 15 trials of CNN and TL experiments with 1%, 1.5%, 2%, 5%, and 10% DS2.

compares the TL and CNN models for the computational time during training as a function of the percentage of the DS2 data for training. The overall computational time of TL is lower than CNN for models with 1%, 1.5%, 2%, and 5% DS2. Both the training time for the CNN model and that for the TL model scale linearly with the percentage of the DS2 data for training ($R^2 = 0.9986$ and 0.9948 , respectively). The slope for the TL model is much larger than that for the CNN model because the TL model only uses the DS2 data for training while the CNN model is trained with both the DS2 data and the DS1 data. Comparison experiment on 10% shows higher computing cost of TL vs. CNN. This is because for the 15 trials, the number of epochs reaching converged solutions for CNN is within the range of [37, 84], and for TL is within [296, 505]. Since the focus here is data scarcity and potential real-time implementation, which considers testing time, the reported time is training in 100 epochs for CNN and 1000 epochs for TL, respectively. The testing time (including the overhead time to calculate the performance metric) was 68.84 microseconds for 3186 images, and this translates to ~ 21.61 ns per image.

3.3. Comparison experiment II: DS1 and DS3

Comparison experiment II (CNN 6 vs. TL 6) was conducted on public dataset DS1 and in-house boiling experiment dataset DS3. With the

promising results of TL outperforming CNN on scarce data from the first comparison experiments, we decided to explore the lowest number of samples affordable from DS3 for training, which is five samples ($\sim 0.05\%$) from DB, BIC, and CHF, respectively. Similar to the first comparison experiment, 15 trials were conducted based on a stratified sampling strategy. Fig. 9 shows the comparison of (a) the mean accuracy and (b) mean FNR for CHF over 15 trials between CNN and TL with $\sim 0.05\%$ DS3 included during training. The results of DS3 are consistent with DS2 showing that the TL model outperforms the CNN model with higher mean accuracy (95.31% compared to 85.91%) and a lower mean false negative rate for CHF (0.1263 compared to 0.0016). The overall testing time was less than 165 microseconds for 34,750 images (with the same software and computer configuration as a comparison experiment I).

3.4. Discussion

Boiling regime experiments often generate massive imagery data under different experimental configurations. One challenge facing the application of DL is the lack of a generalized approach. Often, a DL model needs to be re-trained on any new dataset followed by validation. Ideally, with more datasets gathered, the extensive offline modeling efforts can be compensated by having a super-model generalized enough to apply to most new datasets. Unfortunately, the efforts of collecting and adding more new datasets may not positively correlate with the model performance due to the data noise, data heterogeneity, just to name a few. Inappropriate inclusion of more data may be harmful instead of helpful for predictive modeling. Transfer Learning (TL) may be a viable strategy to support the cross-dataset study.

While TL concept has been widely adopted in different applications such as medical imaging, the applicability of TL for boiling regime, to the best of our knowledge, is new. This is becoming a much-needed effort because (1) there is increasing use of data (imaging, video) in this research field; and (2) notable data often are from different sources. This may be the first attempt to comprehensively evaluate strategies on the cross-dataset study, namely, CNN vs. TL, in hoping to draw insights to guide boiling regime study from a data science perspective. The model performance on detecting CHF and its nanosecond-scale testing time indicate the potential of real-time implementation. Please note the models (CNN and TL) developed in this research are all supervised learning models, that is, the images are labeled as prior. Exploring the applicability of the model on a new, totally “unseen” dataset with no labeling information will require an unsupervised learning model, which currently is not within the scope of this study.

4. Conclusion

In summary, TL is demonstrated to outperform traditional CNN in terms of detection accuracy, robustness, and computational costs, especially when the amount of data for training is limited. In the first experiment, as the training samples from DS2 decrease from 10% to 1%, the detection accuracy of CNN decreases from $99.50 \pm 1.72\%$ to $85.10 \pm 9.43\%$ while the TL model decreases from $99.96 \pm 0.08\%$ to $94.79 \pm 2.97\%$. The smaller variance (measured by standard deviation) is a strong indicator of the robustness of TL comparing to CNN. In terms of computational costs, CNN decreases from 12.62 ± 0.11 min to 10.82 ± 0.08 min while the TL model decreases from 16.76 ± 0.04 min to 1.76 ± 0.04 min. We want to note that the CNN model was conducted with 100 epochs while TL took 1000 epochs. Note choosing the optimal epoch depends on the dataset size and the characteristics of the model. We want to highlight that the numbers of input images for CNN and TL significantly differ. Taking 1% experiments as an example, in one epoch, the TL model is trained only on 1% of data from the target, the data used to train the CNN model is a combination of the 1% target data and the complete source data. As a result, the number of epochs does not directly tell the convergence for comparison. Our empirical experiments on 10% study show that the number of epochs reaching converged solutions for CNN is within [37, 84], and for TL is within [296, 505]. While it is expected the number of epochs varies depending on the training data, as a pilot testing on the feasibility of TL, we decided to take a conservative approach and took 100 epochs vs. 1000 epochs for CNN and TL, respectively for all the experiments. Still, as the samples from the new dataset used for training decrease, the training time of TL becomes much smaller than CNN. Although TL uses a larger epoch to train, the training time is significantly lower than that of CNN.

The transfer learning models are shown to be able to minimize the chance of overlooking CHF events, demonstrated with an ultra-low false negative rate for the CHF class. As such, when used for CHF detection, TL will effectively mitigate CHF-induced detrimental failures of devices that use boiling for cooling. Furthermore, the high accuracy and robustness of the TL models indicate transfer learning to be promising as the remedy for data scarcity, a very common and critical challenge for applying deep learning in solving scientific and engineering problems. Therefore, the approach and results of the present work will not only make an impact in the heat transfer community but also inspire more sophisticated deep learning approaches for engineering applications.

Declaration of Competing Interest

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

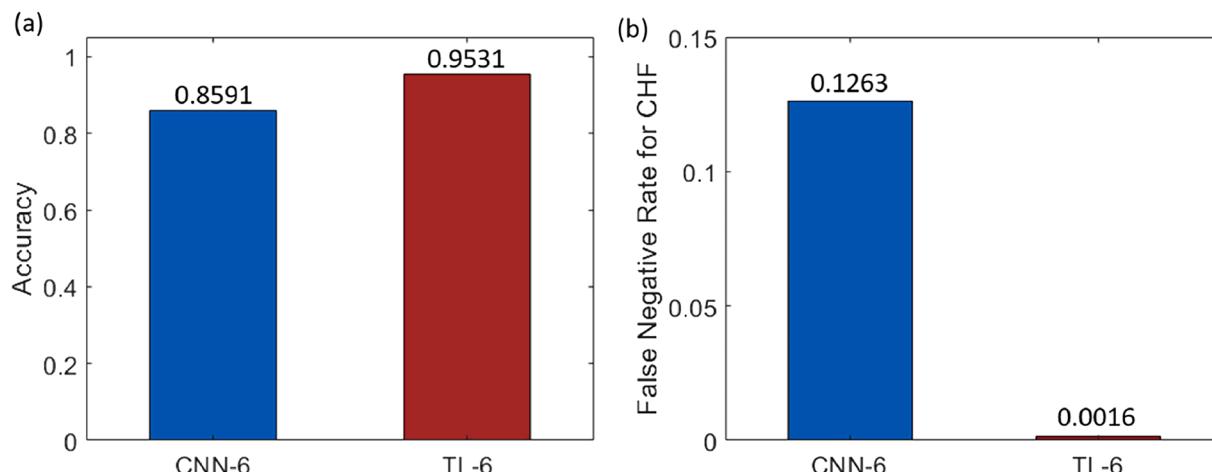


Fig. 9. Comparison between the CNN 6 and TL 6 for (a) the accuracy and (b) the false negative rate for CHF with $\sim 0.05\%$ DS3 for training.

the work reported in this paper.

Acknowledgements

Support for this work was provided in part by the US National Science Foundation under Grant Nos. CBET-1705745 and CBET-1357918.

References

- [1] J. Dirker, D. Juggurnath, A. Kaya, E.A. Osowade, M. Simpson, S. Lecompte, S.M.A. Noori Rahim Abadi, V. Voulgaropoulos, A.O. Adelaja, M.Z. Dauhoo, Thermal energy processes in direct steam generation solar systems: Boiling, condensation and energy storage—A review, *Frontiers in Energy Research*, 6 (2019) 147. <https://doi.org/10.3389/fenrg.2018.00147>.
- [2] P. Birbarah, T. Gebrael, T. Foulkes, A. Stillwell, A. Moore, R. Pilawa-Podgurski, N. Miljkovic, Water immersion cooling of high power density electronics, *Int. J. Heat Mass Transf.* 147 (2020) 118918.
- [3] M.S. El-Genk, Immersion cooling nucleate boiling of high power computer chips, *Energy Convers. Manage.* 53 (2012) 205–218.
- [4] S.G. Kandlikar, Review and projections of integrated cooling systems for three-dimensional integrated circuits, *J. Electron. Packag.* 136 (2014).
- [5] H. Fenech, Heat transfer and fluid flow in nuclear systems, Elsevier, 2013.
- [6] V.K. Dhir, Mechanistic prediction of nucleate boiling heat transfer—achievable or a hopeless task? *ASME J. Heat Transf.* 128 (2006) 1–12, <https://doi.org/10.1115/1.2136366>.
- [7] J.N. Zuber, Hydrodynamic aspects of boiling heat transfer, United States Atomic Energy Commission, Technical Information Service, 1959.
- [8] J. Lienhard, V. Dhir, Hydrodynamic prediction of peak pool-boiling heat fluxes from finite bodies, *ASME J. Heat Transf.* 95 (1973) 152–158, <https://doi.org/10.1115/1.3450013>.
- [9] S.G. Kandlikar, A theoretical model to predict pool boiling CHF incorporating effects of contact angle and orientation, *J. Heat Transfer* 123 (2001) 1071–1079.
- [10] M.M. Rahman, E. Olceroglu, M. McCarthy, Role of wickability on the critical heat flux of structured superhydrophilic surfaces, *Langmuir* 30 (2014) 11225–11234.
- [11] H. Hu, J.A. Weibel, S.V. Garimella, A coupled wicking and evaporation model for prediction of pool boiling critical heat flux on structured surfaces, *Int. J. Heat Mass Transf.* 136 (2019) 373–382.
- [12] L. Zhang, J.H. Seong, M. Bucci, Percolative scale-free behavior in the boiling crisis, *Phys. Rev. Lett.* 122 (2019) 134501.
- [13] V.P. Carey, Liquid-vapor phase-change phenomena: an introduction to the thermophysics of vaporization and condensation processes in heat transfer equipment, CRC Press, 2020.
- [14] J. Barbosa Jr, A. Govan, G. Hewitt, Visualisation and modelling studies of churn flow in a vertical pipe, *Int. J. Multiph. Flow* 27 (2001) 2105–2127.
- [15] P.J. Waltrich, G. Falcone, J.R. Barbosa Jr, Axial development of annular, churn and slug flows in a long vertical tube, *Int. J. Multiph. Flow* 57 (2013) 38–48.
- [16] C.E. Brennen, C.E. Brennen, Fundamentals of multiphase flow, Cambridge University Press, 2005.
- [17] T. Cong, R. Chen, G. Su, S. Qiu, W. Tian, Analysis of CHF in saturated forced convective boiling on a heated surface with impinging jets using artificial neural network and genetic algorithm, *Nucl. Eng. Des.* 241 (2011) 3945–3951.
- [18] T. Cong, G. Su, S. Qiu, W. Tian, Applications of ANNs in flow and heat transfer problems in nuclear engineering: a review work, *Prog. Nucl. Energy* 62 (2013) 54–71.
- [19] H. Alimoradi, M. Shams, Optimization of subcooled flow boiling in a vertical pipe by using artificial neural network and multi objective genetic algorithm, *Appl. Therm. Eng.* 111 (2017) 1039–1051.
- [20] Y. Liu, N. Dinh, Y. Sato, B. Niceno, Data-driven modeling for boiling heat transfer: using deep neural networks and high-fidelity simulation results, *Appl. Therm. Eng.* 144 (2018) 305–320.
- [21] M. Hassanpour, B. Vaferi, M.E. Masoumi, Estimation of pool boiling heat transfer coefficient of alumina water-based nanofluids by various artificial intelligence (AI) approaches, *Appl. Therm. Eng.* 128 (2018) 1208–1222.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] M.V. Valueva, N.N. Nagornov, P.A. Lyakhov, G.V. Valuev, N.I. Chervyakov, Application of the residue number system to reduce hardware costs of the convolutional neural network implementation, *Math. Comput. Simul.* 177 (2020) 232–243.
- [27] W.H. Lopez Pinaya, S. Vieira, R. Garcia-Dias, A. Mechelli, Chapter 10 - Convolutional neural networks, in: A. Mechelli, S. Vieira (eds.) *Machine Learning*, Academic Press, 2020, pp. 173–191. <https://doi.org/10.1016/B978-0-12-815739-8.00010-9>.
- [28] S. Lawrence, C.L. Giles, T. Ah Chung, A.D. Back, Face recognition: a convolutional neural-network approach, *IEEE Trans. Neural Networks* 8 (1997) 98–113.
- [29] F. Gao, T. Wu, J. Li, B. Zheng, L. Ruan, D. Shang, B. Patel, SD-CNN: A shallow-deep CNN for improved breast cancer diagnosis, *Comput. Med. Imaging Graph.* 70 (2018) 53–62.
- [30] G.B. Frisoni, N.C. Fox, C.R. Jack, P. Scheltens, P.M. Thompson, The clinical use of structural MRI in Alzheimer disease, *Nature Rev. Neurol.* 6 (2010) 67–77.
- [31] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, H. Larochelle, Brain tumor segmentation with deep neural networks, *Med. Image Anal.* 35 (2017) 18–31.
- [32] A. de Brebisson, G. Montana, Deep Neural Networks for Anatomical Brain Segmentation, *arXiv* (2015). <https://arxiv.org/abs/1502.02445>.
- [33] A. Esteva, B. Kuprel, R.A. Novoa, J. Ko, S.M. Swetter, H.M. Blau, S. Thrun, Dermatologist-level classification of skin cancer with deep neural networks, *Nature* 542 (2017) 115–118.
- [34] C.J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, T. Poinsot, Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates, *Combust. Flame* 203 (2019) 255–264.
- [35] S.S. Abdurakipov, O.A. Gobyzov, M.P. Tokarev, V.M. Dulin, Combustion Regime Monitoring by Flame Imaging and Machine Learning, *Optoelectronics Instrument. Data Process.* 54 (2018) 513–519.
- [36] S. Ye, Z. Zhang, X. Song, Y. Wang, Y. Chen, C. Huang, A flow feature detection method for modeling pressure distribution around a cylinder in non-uniform flows by using a convolutional neural network, *Sci. Rep.* 10 (2020) 4459.
- [37] S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, S. Kaushik, Prediction of aerodynamic flow fields using convolutional neural networks, *Comput. Mech.* 64 (2019) 525–545.
- [38] M.L. Dering, C.S. Tucker, A Convolutional Neural Network Model for Predicting a Product's Function, Given Its Form, *J. Mech. Des.* 139 (2017).
- [39] S. Ye, B. Li, Q. Li, H.-P. Zhao, X.-Q. Feng, Deep neural network method for predicting the mechanical properties of composites, *Appl. Phys. Lett.* 115 (2019) 161901.
- [40] Z. Chen, K. Gryllias, W. Li, Mechanical fault diagnosis using Convolutional Neural Networks and Extreme Learning Machine, *Mech. Syst. Sig. Process.* 133 (2019) 106272.
- [41] Y. Yu, C. Wang, X. Gu, J. Li, A novel deep learning-based method for damage identification of smart building structures, *Struct. Health Monitor.* 18 (2018) 143–163.
- [42] H. Khodabandehlou, G. Pekcan, M.S. Fadali, Vibration-based structural condition assessment using convolution neural networks, *Struct. Control Health Monitor.* 26 (2019) e2308.
- [43] Y.-J. Cha, W. Choi, O. Büyüköztürk, Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks, *Comput.-Aided Civ. Infrastruct. Eng.* 32 (2017) 361–378.
- [44] T. Ince, S. Kiranyaz, L. Eren, M. Askar, M. Gabbouj, Real-time motor fault detection by 1-D convolutional neural networks, *IEEE Trans. Ind. Electron.* 63 (2016) 7067–7075.
- [45] M. Talo, U.B. Baloglu, Ö. Yıldırım, U.R. Acharya, Application of deep transfer learning for automated brain abnormality classification using MR images, *Cognit. Syst. Res.* 54 (2019) 176–188.
- [46] S.J. Pan, Q. Yang, A survey on transfer learning, *IEEE Trans. Knowl. Data Eng.* 22 (2009) 1345–1359.
- [47] K. Weiss, T.M. Khoshgoftaar, D. Wang, A survey of transfer learning, *J. Big data* 3 (2016) 9.
- [48] K. Wang, B.K. Patel, L. Wang, T. Wu, B. Zheng, J. Li, A dual-mode deep transfer learning (D2TL) system for breast cancer detection using contrast enhanced digital mammograms, *IIE Trans. Healthcare Syst. Eng.* 9 (2019) 357–370.
- [49] B.Q. Huynh, H. Li, M.L. Giger, Digital mammographic tumor classification using transfer learning from deep convolutional neural networks, *J. Med. Imaging* 3 (2016) 034501.
- [50] F. Gao, H. Yoon, T. Wu, X. Chu, A feature transfer enabled multi-task deep learning model on medical imaging, *Expert Syst. Appl.* 143 (2020) 112957.
- [51] T. Zeng, S. Ji, Deep convolutional neural networks for multi-instance multi-task learning, in: 2015 IEEE International Conference on Data Mining, IEEE, 2015, pp. 579–588. <https://doi.org/10.1109/ICDM.2015.92>.
- [52] S.M. You, Pool boiling, <https://msht.utdallas.edu/>.
- [53] H. Minseok, Bertina, B., Graham, S., Pool Boiling Experiment, YouTube (2014). https://www.youtube.com/watch?v=GA9MBdePwmo&ab_channel=BertinaBanks.
- [54] S. Jun, J. Kim, D. Son, H.Y. Kim, S.M. You, Enhancement of pool boiling heat transfer in water using sintered copper microporous coatings, *Nuclear Eng. Technol.* 48 (2016) 932–940.
- [55] S. Jun, H. Wi, A. Gurung, M. Amaya, S.M. You, Pool boiling heat transfer enhancement of water using brazed copper microporous coatings, *J. Heat Transfer* 138 (2016).
- [56] S. Jun, J. Kim, S.M. You, H.Y. Kim, Effect of heater orientation on pool boiling heat transfer from sintered copper microporous coating in saturated water, *Int. J. Heat Mass Transf.* 103 (2016) 277–284.
- [57] S. Jun, J.C. Godinez, S.M. You, H.Y. Kim, Pool boiling heat transfer of a copper microporous coating in borated water, *Nuclear Eng. Technol.* 52 (2020) 1939–1944.
- [58] M. Ha, S. Graham, Pool boiling characteristics and critical heat flux mechanisms of microporous surfaces and enhancement through structural modification, *Appl. Phys. Lett.* 111 (2017) 091601.
- [59] M. Ha, S. Graham, Pool boiling enhancement through hierarchical texturing of surfaces, in: 2016 15th IEEE Intersociety Conference on Thermal and

- Thermomechanical Phenomena in Electronic Systems (ITherm), IEEE, 2016, pp. 388-394. <https://doi.org/10.1109/ITHERM.2016.7517575>.
- [60] M. Ha, S. Graham, Pool boiling enhancement using vapor channels in microporous surfaces, *Int. J. Heat Mass Transf.* 143 (2019) 118532.
- [61] F. Gao, T. Wu, X. Chu, H. Yoon, Y. Xu, B. Patel, Deep Residual Inception Encoder-Decoder Network for Medical Imaging Synthesis, *IEEE J. Biomed. Health. Inf.* 24 (2019) 39–49.
- [62] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, *Proc. IEEE* (2020).
- [63] Y. Zhu, F. Zhuang, D. Wang, Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 5989–5996. <https://doi.org/10.1609/aaai.v33i01.33015989>.
- [64] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, C. Liu, A survey on deep transfer learning, in: International conference on artificial neural networks, Springer, 2018, pp. 270–279.
- [65] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 1717–1724.
- [66] X. Glorot, Y. Bengio, Understanding the difficulty of training feedforward neural networks, in: Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 249–256. <http://proceedings.mlr.press/v9/glorot10a.html>.