

Deep Learning Strategies for Critical Heat Flux Detection in Pool Boiling

by

Firas Al-Hindawi

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved June 2021 by the
Graduate Supervisory Committee:

Teresa Wu, Co-Chair
Hyunsoo Yoon, Co-Chair
Han Hu
Ashif Iquebal

ARIZONA STATE UNIVERSITY

August 2021

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 that the new dataset has a sufficient amount of labeled data. This research is to explore supervised, semi-supervised, and unsupervised machine learning strategies that are formulated to adapt to two scenarios. The first is when the new dataset has limited labeled data available. This scenario was addressed in chapter 2 of this thesis, where Convolutional Neural Networks (CNNs) and Transfer learning (TL) were used in tackling such situations. The second scenario is when the new dataset has no labeled data available at all. In such cases, this research presents a methodology in Chapter 3, where one of the state-of-the-art Generative Adversarial Networks (GANs) called Fixed-Point GAN is deployed in collaboration with a regular CNN model to tackle the problem. To the best of my knowledge, the approaches presented in chapters 2 and 3 are the first of their kind to utilize TL and GANs to solve the boiling heat transfer problem within the heat transfer community and are a step forward towards obtaining a one-for-all general model.

DEDICATION

First and foremost, I would like to dedicate this work to my late grandmother. Illiterate as she might have been, she knew firsthand the importance of education and made sure her children and grandchildren would never grow up without one. Nothing would have brought more joy to my life than seeing her looking at me with pride after I graduate.

I also would like to dedicate it to my mother, father, family and friends who supported me in my life journey thus far.

Finally, I like to dedicate this work to every mentor, teacher, and professor that guided and taught me with sincerity and dedication.

ACKNOWLEDGMENTS

All thanks, praise and glory be to Allah. I acknowledge all his blessings upon me and ask for his forgiveness for all my wrongs and shortcomings. Without him none of this would have been possible and to him alone is my utmost and profound gratitude.

I like to sincerely thank Professor Teresa Wu for believing in me and giving me the opportunity to join her team. I am very grateful for everything she did for me. I have learned so much in the past two years working under her supervision, and I cannot express enough how valuable this experience was for my career and education. I like to sincerely thank professor Hyunsoo Yoon as well for his guidance, assistance, and mentorship. He helped me greatly during this journey and I am truly grateful for all his help.

I also would like to thank professors Ying Sun, Han Hu, and the rest of their team for their valuable contribution to this research work.

I am extremely grateful to all my colleagues at the AMCII lab for their help and support. Special thanks to both Md Mahfuzur Rahman Siddiquee and Josh Xu for selflessly dedicating their time and experience to help solve the coding issues I faced. Their shared insights and suggestions were crucial to the success and completion of this work.

I am also greatly grateful for the Fulbright Commission in Jordan for awarding me with this scholarship to pursue a graduate degree at ASU.

Last but not least, I like to thank my parents whom without this journey would not have even began.

TABLE OF CONTENTS

	Page
LIST OF TABLES	v
LIST OF FIGURES	vi
CHAPTER	
1 INTRODUCTION	1
1.1 Heat Exchanger Problem	1
1.2 Deep Learning.....	4
1.3 Research Objective & Road Map	6
2 SUPERVISED AND SEMI-SUPERVISED MACHINE LEARNING	
APPROACH FOR CRITICAL HEAT FLUX DETECTION	8
2.1 Introduction.....	8
2.2 Methodology	13
2.2.1 Dataset preparation for CNN and TL training and testing.....	13
2.2.2 CNN-base and CNN-TL: Convolutional neural network model and Transfer learning model architecture	16
2.3 Results and Discussion	22
2.3.1 Base Experiment: CNN-base model	22
2.3.2 Comparison Experiment I: DS1 and DS2	25
2.3.3 Comparison Experiment II: DS1 and DS3.....	32
2.3.4 Discussion	34

CHAPTER	Page
2.4 Conclusion	35
3 UNSUPERVISED MACHINE LEARNING APPROACH FOR CRITICAL HEAT FLUX DETECTION	37
3.1 Introduction.....	37
3.2 Fixed-Point GAN	39
3.3 Application of Fixed-Point GAN to Critical Heat Flux Detection	41
3.2.1 CNN-0 Model Training.....	44
3.2.2 Fixed-Point GAN Training	44
3.4 Results and Discussion	45
3.4.1 CNN-0 Model and Blind Cross Domain Testing.....	45
3.4.2 Fixed-Point GAN Evaluation using FID.....	46
3.4.3 Testing Fix-Point GAN Images using CNN-0.....	49
3.4.4 Discussion	50
3.5 Conclusion	56
4 CONCLUSION AND FUTURE WORK	58
4.1 Conclusion	58
4.2 Future Work	60
4.2.1 Model Performance Improvement	60
4.2.2 Black Box Unraveling.....	61
REFERENCES	64

LIST OF TABLES

Table	Page
1. Table 1. Summary of the Three Datasets.....	16
2. Table 2. Summary of the CNN-base and CNN-TL Models in the Cross-dataset (DS1, DS2) Study.	25
3. Table 3. Metrics Generated by Using CNN-0 on DS1 and DS2	46
4. Table 4. Fid Values for Images Generated from the Validation Set Using the 30 Gan Models.....	47
5. Table 5. CNN-0 Predictions on Fake DS1 - Confusion Matrix.....	49
6. Table 6. CNN-0 Predictions on Fake DS1 - Evaluation Metrics.....	49
7. Table 7. Comparison Between Evaluation Metrics Generated Using All Methods.	50
8. Table 8. Metrics Comparison Between the 290k and the 90k Models	55

LIST OF FIGURES

Figure	Page
1. Figure 1. A Generic Steady-state Boiling Curve: Heat Flux Vs. Surface Superheat (Adopted from Hu Et Al., 2017).....	2
2. Figure 2. Research Road Map.....	7
3. Figure 3. Representative Images of Bubble Dynamics Were 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 [60], [61], and Images for DS3 Were Obtained from In-house Pool Boiling Experiments.	15
4. Figure 4. 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).....	21

Figure	Page
5. Figure 5. 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.....	24
6. Figure 6. 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.....	24
7. Figure 7. (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.....	27
8. Figure 8. 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.....	30
9. Figure 9. 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.....	31
10. Figure 10. 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.....	32

Figure	Page
11. Figure 11. Comparison Between the CNN 6 and TL 6 for (a) the Accuracy and (b) the False Negative Rate for CHF with ~0.05% DS3 for Training.....	33
12. Figure 12. Comparison Between Fixed-point GAN and Its Predecessor StarGAN on the Brain Lesion Localization Task (Siddique Et Al., 2019).....	40
13. Figure 13. Same Domain Classification Vs Cross Domain Classification.....	41
14. Figure 14. Unsupervised Learning Methodology Flow Chart.....	43
15. Figure 15. Comparison Between the Confusion Matrices Generated by Using CNN-0 on DS1 (Part a) and DS2 (Part b).....	46
16. Figure 16. Samples Generated by the 290k Model.....	48
17. Figure 17. Comparison Between the Confusion Matrices Generated Using All Methods.....	51
18. Figure 18. Real Vs Fake "ONB" Images	53
19. Figure 19. Real Vs Fake "BIC" Images	53
20. Figure 20. Real Vs Fake "CHF" Images.....	54
21. Figure 21. Confusion Matrices Comparison Between the 290k and 90k Models	55
22. Figure 22. CNN Features Extraction Using Auto Encoders.....	62
23. Figure 23. Samples of Attention Maps Generated from Both the CNN and TL Models for Each Class.....	62

CHAPTER 1

INTRODUCTION

1.1 Heat Exchanger Problem

Heat transfer refers to the mechanisms that convey energy from one location to another. Boiling heat transfer presents itself as one of the most attractive heat transfer mechanisms where a high heat transfer rates are achieved via the liquid-vapor interface. It has a wide range of applications such as cooking, water purification, power-generation plants, immersion cooling for high-performance electronics and data centers, integrated cooling for three-dimensional electronic packaging, among others. Although it is a very efficient heat transfer method, boiling heat transfer is sensitive to small changes in the power dissipated beyond the critical heat flux. This can quickly convert the desirable nucleate boiling regime into film boiling (a.k.a. boiling crisis), overheating the boiling surface [1], [2]. Figure 1 shows the steady state boiling curve and the stages the process goes through [3]. Based on these stages, boiling is primarily classified into three regimes: (1) the nucleate boiling (NB), (2) the transition boiling, and (3) the film boiling [2]. The NB regime, which is represented by the region enclosed between points A and B, is the desired region where at which the heat transfer is most efficient (very high heat transfer under near-isothermal conditions), making it the optimal choice for practical applications [2].

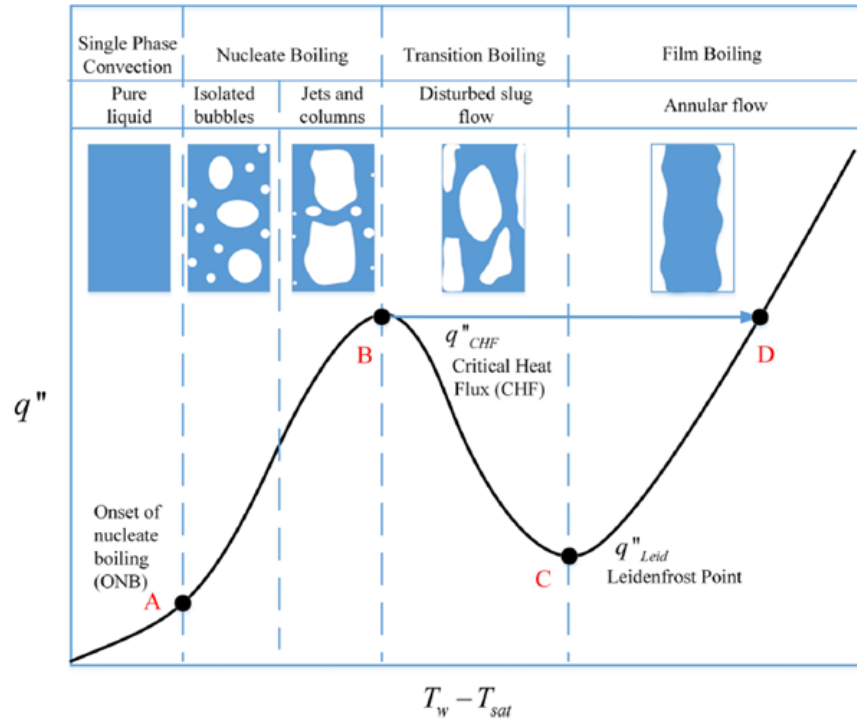


Figure 1. A Generic Steady-state Boiling Curve: Heat Flux Vs. Surface Superheat

(Adopted from Hu Et Al., 2017)

The real danger of the boiling crisis starts beyond critical heat flux (CHF) where the CHF condition starts (point B). 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 heat transfer coefficients (HTC) and a rapid increase in temperature (couple hundred degrees Celsius in a matter of minutes). 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 accurately and instantaneously [1], [2].

The importance of this problem forced researchers to focus on advanced diagnostic tools to protect the device and prevent hazardous ramifications from happening [2]. The difficulty of this problem comes from the vagueness of the mechanisms governing the process and the complexity of describing these mechanisms in a mathematical model. Although there are existing analytical and computational efforts to describe the phenomena, they are either computationally expensive or inaccurate [2]. The void fraction and other characteristics in the conventional analysis are generally time-averaged values. These characteristics and the heat flux are actually not constant over time even at the steady-state because the heat transfer mechanisms are different in different phases of the bubble ebullition cycle (nucleation, growth, departure, rewetting). In most of the boiling studies, the reported steady-state heat flux is a temporal and spatial average of the heat flux on the heating surface. There will never be a one-to-one correlation between a transient quantity and a time-averaged quantity. The conventional analysis is focused on one or a few selected characteristics of the bubble dynamics. Neglecting other parameters will make the analysis lose its generality.

Recently, a new approach has been trending in tackling this problem. Researchers started utilizing computer vision and machine learning to make sense of this phenomena. Although these attempts were highly successful, each of these attempts had certain shortcomings. Some of them were not able to rely solely on the boiling images and had to incorporate some of the physical measurements into their models. Other attempts were not

generalizable for different boiling setups and were only valid for a specific one. A new machine learning model would have to be trained from scratch on a new data set for each setup. Thus, a significant challenge remains to develop a model that could provide near real-time analysis and diagnostics of heat flux and one that could be generalized for data coming from different setups without the need of creating a model specific for each case. This research focuses on developing deep learning strategies to tackle the heat exchanger problem.

1.2 Deep Learning

Deep Neural Networks (DNNs) are a form of machine learning that uses neural networks with many hidden layers. DNNs are considered as a powerful method for solving large-scale real-world problems such as automated image classification, natural language processing, human action recognition, or physics. Although neural networks date back at least to the 1950s, the popularity soared a few years ago when deep neural networks (DNNs) outperformed other machine learning methods in speech recognition and image classification. This was because of the recent improvements DNN training methodologies (parallelization, utilization of GPUs and TPUs, etc.). These improvements enabled DNNs to harvest extremely large amounts of training data and can thus achieve record performances in many research fields [4], [5].

DNNs alter input features using several layers whose operations consist of element-wise nonlinearities and affine transformations. One well-studied DNNs is convolutional neural networks (CNN). The main idea of CNN is to base the affine transformations on convolution operators with compactly supported filters. Supervised learning aims at

learning the filters and other parameters, which are also called weights, from training data. CNNs are widely used for solving large-scale learning tasks involving data that represent a discretization of a continuous function, e.g., voice, images, and videos [6]. Sophisticated algorithms are proposed to support the training and testing of such a complex network, and the backpropagation method is widely applied to train the CNN parameters. Furthermore, to fine-tune the network to specific functions, large pools of labeled data are required for iteratively training the massive neurons and connection weights. Multiple CNN designs have been proposed in recent literature to effectively train these networks, such as AlexNet[7], VGG[8], etc. Some of these designs surpassed human-level accuracy on object recognition tasks because they emulate the human brain's natural visual perception mechanism by systematically learning features through multiple operational layers. Image-based deep learning models can play a vital role in thoroughly understanding boiling physics because boiling images are richly embedded with bubble statistics, which are quantitative measurements of the dynamic boiling phenomena [6], [9].

Despite all the recent success achieved by DNNs (e.g., CNN), the models suffer from high computational cost, slow training speed, and security vulnerability. Some of these drawbacks could be attributed to the limited understanding of the internal features learned by each convolutional layer, which made many scholars label CNN training as a black-box process. The hope is that as our knowledge about what goes on behind the scenes increases, we could identify the parts of the process causing these drawbacks. Better network interpretability will significantly increase the robustness evaluation of each network layer, as well as the network adaptability and transferability to different applications [9].

1.3 Research Objective & Road Map

The research objective is to develop a deep learning based solution that could provide near real-time analysis and diagnostics of heat flux and one that could be generalized for data coming from different setups without the need of creating a model specific for each case.

This objective will be tackled in three progressive steps:

1. A Supervised Machine Learning Approach: Demonstrate the capability of supervised CNNs in correctly classifying the state in which any boiling setup is currently at instantaneously, by simply using a model that was trained on labeled images coming from that same setup (same domain), without the need of any other physical attributes to be incorporated in the model.
2. A Semi-Supervised Machine Learning Approach: Demonstrate the capability of a semi-supervised domain adaptation technique such as transfer learning in utilizing pre-trained models (such as the one obtained in step 1) to correctly classify images from a different domain in a case where there is a scarcity of labeled data from that domain (only a limited number of labeled images are available from the other domain), while also providing superior results in terms of computation speed.
3. An Unsupervised Machine Learning Approach: Demonstrate the capability of unsupervised machine learning models such as the Fixed-Point Generative Adversarial Network (Fixed-Point GAN) in utilizing a pre-trained model (such as the one obtained in step 1) to classify images from a different domain in a case where there are no labeled images available from that domain.

As demonstrated in the research road map shown in Figure 2, steps 1 and 2 will be discussed in chapter 2 of this thesis, while chapter 3 is solely focused on step number 3.

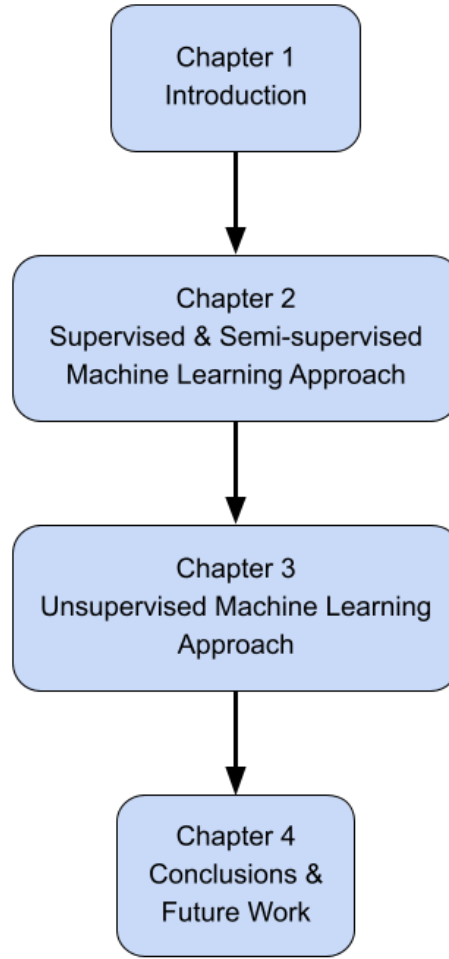


Figure 2. Research Road Map

CHAPTER 2

SUPERVISED AND SEMI-SUPERVISED MACHINE LEARNING

APPROACH FOR CRITICAL HEAT FLUX DETECTION

2.1 Introduction

Boiling is a central phenomenon in many industrial applications including steam generation in power plant boilers and solar collectors [10], immersion cooling for high-performance electronics and data centers [11][12], integrated cooling for three-dimensional electronic packaging [13], cooling of the core and used fuel in nuclear reactors [14], 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 make it extremely challenging to develop a comprehensive physics-based model for CHF during boiling [15]. To date, theoretical models and experimental correlations for boiling CHF, *e.g.*, the hydrodynamic

model [16][17], the force-balance model [18], and wicking-based CHF models [19][20], 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 [21]. 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 [22]. 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 [22]. High-speed visualizations of pool boiling experiments have enabled the development of regime maps based on bubble morphologies, corresponding to different boiling phenomena [23]–[25]. 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 [2], [26]–[33]. 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 [2], [31]–[33]. For example, Hobold and da Silva [31] 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 [33], 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 [34], [35]. 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 [36], it has been adopted as a popular tool in biomedical image analysis [37]–[41] and in engineering applications such as combustion [42], [43], aerodynamics [44], [45], digital designs [46], mechanical property predictions[47], and damage and fault diagnosis and detection [48]–[52]. In the applications related to this research, Hobold and da Silva [32] 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 [2] 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 [53]–[55]. 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 [53], [56]–[59], 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.

2.2 Methodology

2.2.1 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 [60], [61], 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 [62]–[65]. 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 [66]–[68]. The circular heater had a diameter of $\approx 16 \text{ mm}$ 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 $\approx 100 \text{ mm}^2$, 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 [69]. 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 Figure 3., DS1, DS2, and DS3 had a total of 6158, 3215, and 34422 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.





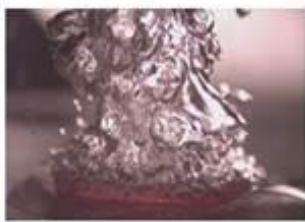




<u>Regime</u> \ <u>Dataset</u>	<u>DS1</u>	<u>DS2</u>	<u>DS3</u>
<u>Class 1 (DB)</u> <u>Discrete</u> <u>bubbles</u>			
<u>Class 2 (BIC)</u> <u>Bubble</u> <u>interference and</u> <u>coalescence</u>			
<u>Class 3 (CHF)</u> <u>Critical heat</u> <u>flux</u>			

Figure 3. Representative Images of Bubble Dynamics Were 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 [60], [61], and Images for DS3 Were Obtained from In-house Pool Boiling Experiments.

Table 1. Summary of the Three Datasets.

Datasets	DB	BIC	CHF
Dataset 1 (DS1) [60]	2304	3068	786
Dataset 2 (DS2) [61]	693	1289	1233
Dataset 3 (DS3) in-house	11488	10890	12044

2.2.2 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. Figure 4. (a) illustrates the CNN architecture adopted in this research. As seen, six convolutional blocks, each with

two dimensional convolutional layers and a rectified linear unit (ReLU) activation function, 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 convolutional layers in every block have 32 kernels except for block 4 that has 64 kernels. These kernels (also known as filters) are the vector of weight and bias, applied 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 except for block 1. 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

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

It is well recognized in the data science community that models 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 [70]. 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 [56], [70]. 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 [53], breast cancer [56]–[58], mouse brain [59], and general image classification [71]. The efforts in deep transfer learning have been categorized into instance-based, mapping-based, network-based, and adversarial-based [71]. 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 Figure 4. (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 [54], [55], [70], [72]. Instance transfer aims to reuse the samples (e.g., images), and feature transfer reuses 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 the target is kept, and deep model parameters are being transferred [53], [57], [73]. 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 Figure 4. (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 [39], 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.

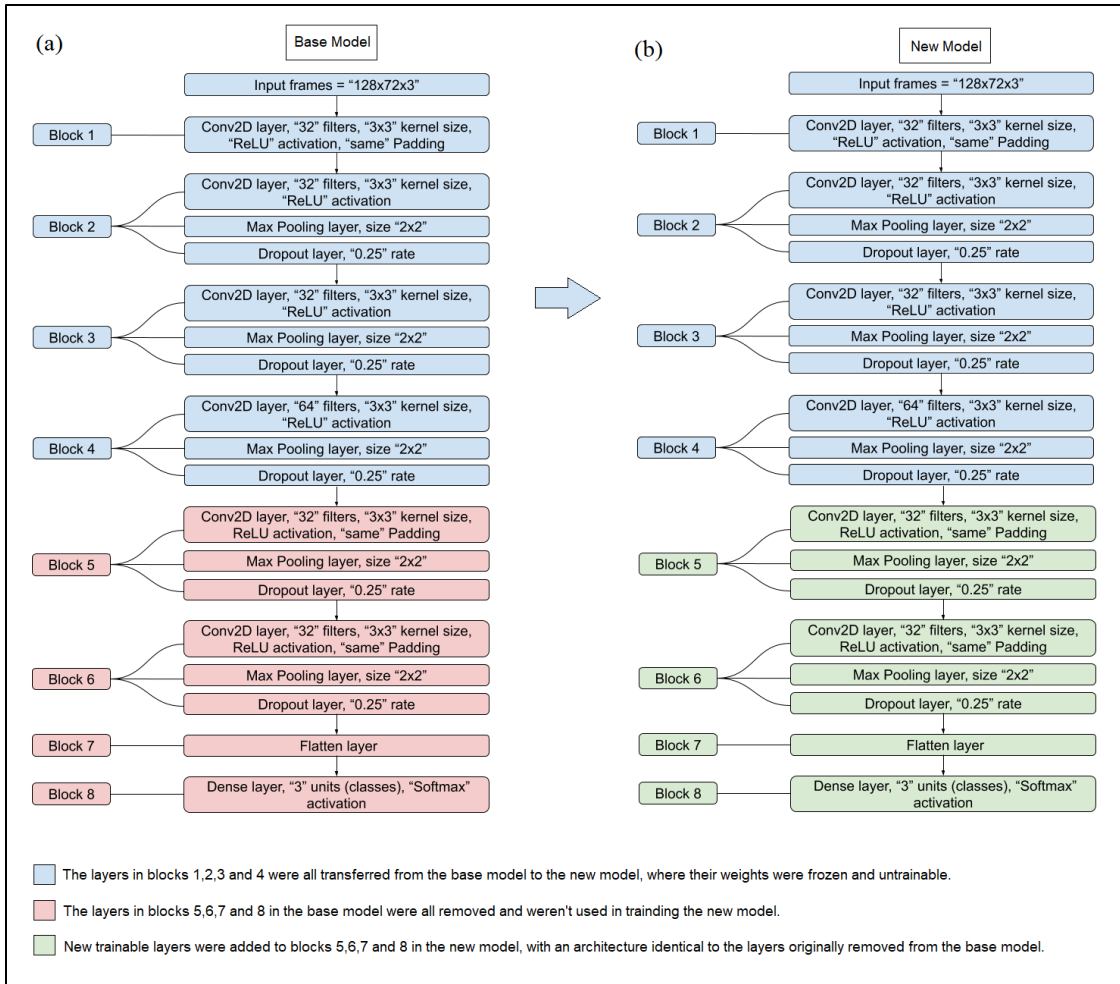


Figure 4. The TL Procedure Operates on (a) Base Model Architecture and (B) New Model Architecture. The TL Architecture Was Prepared by 1) Transferring the Layers in Blocks 1 \rightarrow 4 and Their Trained Parameters Stored after Training the Base Model (a) Purely on DS1, from the Base Model to the New 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 \rightarrow 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).

2.3 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.

2.3.1 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 Figures. 5(a), 5(b), and 5(c), respectively. The first two models show high accuracy (100%), and the third model has only 40.09% accuracy. It is noted that Figures. 5(a) and 5(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 Figure 5.(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. Figures 6(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. Figure 6(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-based models on different unseen 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.

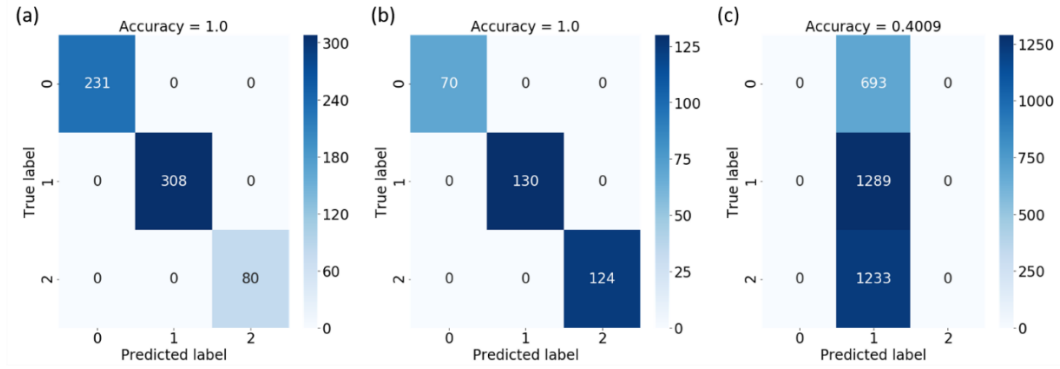


Figure 5. Confusion Matrices of (a) Testing CNN Trained with Ds1 (CNN 0) for Classifying DS1; (B) Testing CNN Trained with DS2 for Classifying DS2; And (C) Testing CNN 0 for Classifying DS2.

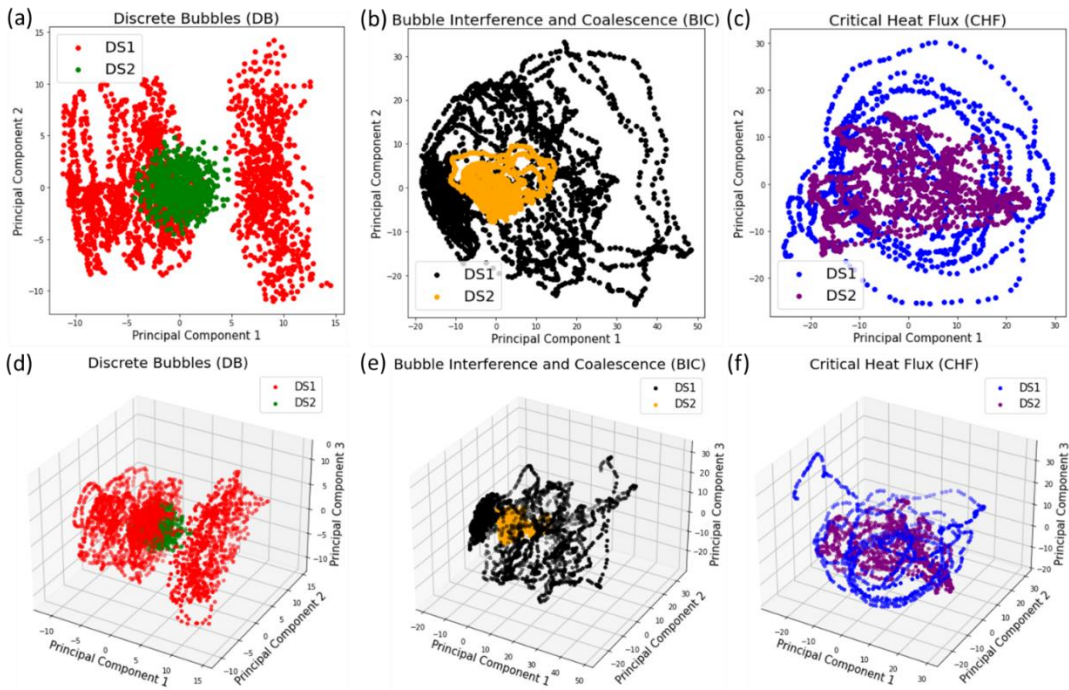


Figure 6. Principal Component Analysis of DS-1 and DS-2 Showing (a)-(c) Two-component PCA Decomposition of DB, BIC, and CHF Images, and (d)-(f) Three-component PCA Decomposition of DB, BIC, and CHF Images.

2.3.2 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.

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 + 1% DS2	TL 1	1% DS2
CNN 2	DS1 + 1.5% DS2	TL 2	1.5% DS2
CNN 3	DS1 + 2% DS2	TL 3	2% DS2
CNN 4	DS1 + 5% DS2	TL 4	5% DS2
CNN 5	DS1 + 10% DS2	TL 5	10% DS2

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

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 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.

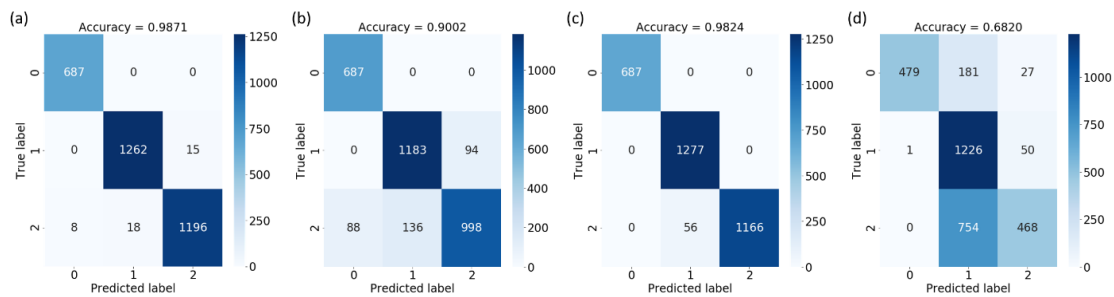


Figure 7. (a-b) Confusion Matrices of Testing TL 1 for Classifying DS2 with (a) Highest Accuracy and (b) Lowest Accuracy among 15 Trials with Randomly Selected 1% DS2 for Training; (c-d) Confusion Matrices of Testing CNN 1 for Classifying DS2 with (a) Highest Accuracy and (d) Lowest Accuracy among 15 Trials with Randomly Selected 1% DS2 for Training.

Figure 7 shows representative confusion matrices of the CNN and TL models trained with DS1 and a small proportion of DS2, with Figures 7(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 5 and CNN 5, where 1% of DS 2 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 Figure 8. (a). As shown in Figure 8. (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 $FNR = 1$. Figure 8 (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 < 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 Figure 8 (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.

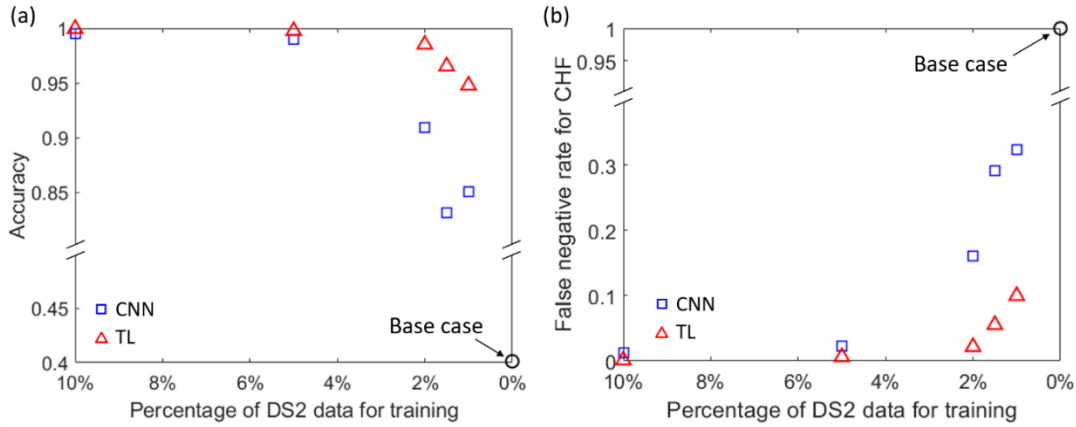


Figure 8. Comparison Between the CNN and TL for (a) the Accuracy and (B) the False Negative Rate for CHF as a Function of the Percentage of DS2 Data for Training. The Data Points Represent the Mean Values of 15 Trials per Experiment.

It is noted that the analysis in Figure 8 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, Figure 9. 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 Figure 9 (a) and (b) 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 Figure 8 and Figure 9, 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%).

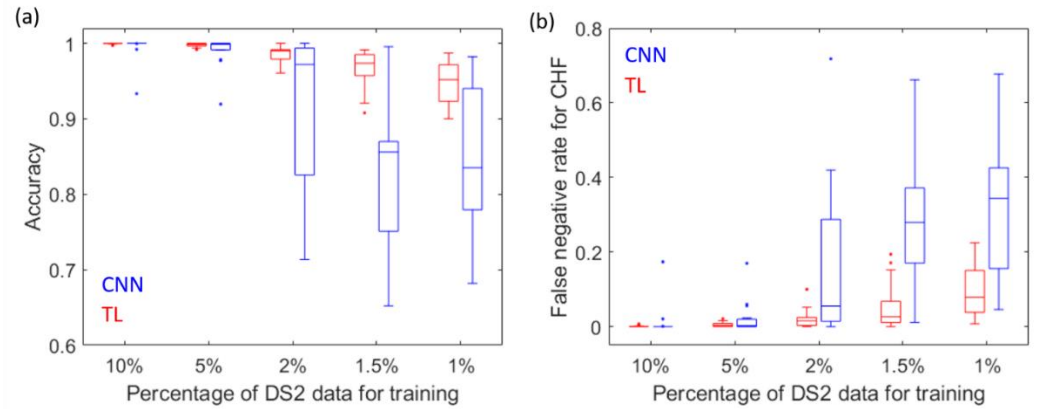


Figure 9. Box-plot of (a) the Accuracy and (b) False Negative Rate for CHF Based on 15 Trials of CNN and TL Experiments with 1%, 1.5%, 2%, 5%, and 10% DS2.

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 @3.40GHz, 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. Figure 10 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 nanoseconds per image.

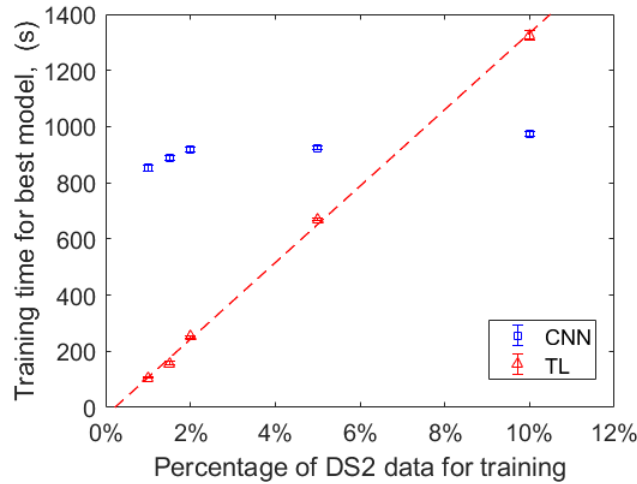


Figure 10. Comparison Between TL and CNN Models for the Computational Time Based on 15 Trials of CNN and TL Experiments with 1%, 1.5%, 2%, 5%, and 10% DS2.

2.3.3 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 (~0.05%) from DB, BIC, and CHF, respectively. Similar to the first comparison experiment, 15 trials were conducted based on a stratified sampling strategy. Figure 11 shows the comparison of (a) the mean accuracy and (b) mean FNR for CHF over 15 trials between CNN and TL with ~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).

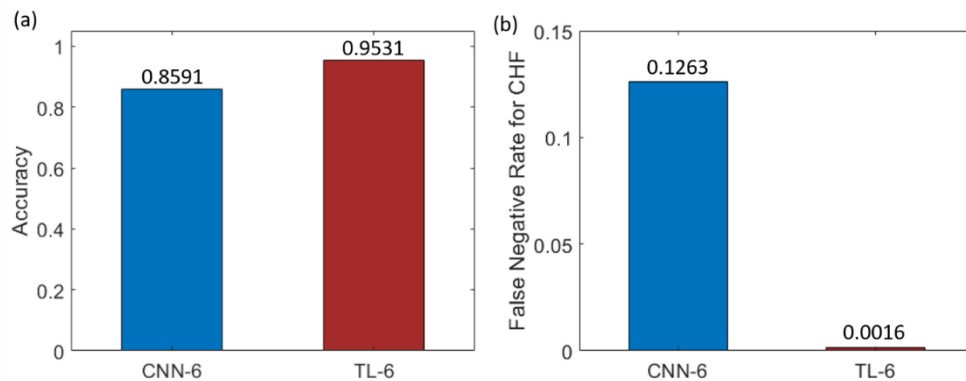


Figure 11. Comparison Between the CNN 6 and TL 6 for (a) the Accuracy and (b) the False Negative Rate for CHF with ~0.05% DS3 for Training.

2.3.4 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 chapter and will be discussed in chapter 3 of this document.

2.4 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.

CHAPTER 3

UNSUPERVISED MACHINE LEARNING APPROACH FOR CRITICAL HEAT FLUX DETECTION

3.1 Introduction

Supervised machine learning has proved itself to be successful across a wide variety of real-world applications. However, it is limited to applications where the data source the model was trained on comes from the same distribution as the data target that the model will be tested on. For the application where the target data drawn from a different distribution than the source training data, the performance of the model starts declining to the point where some models become worse than random guessing [74].

An alternative solution is to train a model on another related large-scale source domain with labels (e.g., a simulation domain) and adapt it to the unlabeled target domain (e.g., a real-world domain), also known as unsupervised domain adaptation. This adaptability is important, especially with the rise of deep neural networks that depend heavily on the availability of a large amount of labeled data for it to be effective. It becomes even more important for applications where obtaining labels for a large amount of data is very expensive or simply not possible. It allows for eliminating the costs associated with obtaining labeled data from the target domain. Thus, researchers are motivated to explore the single source homogenous unsupervised deep domain adaptation problem [74]–[76]. Most unsupervised deep domain adaptation methodologies prevail by either designing new

distance metrics to measure the discrepancy between two domains to minimize the discrepancy between the domains, or by learning domain invariant features using adversarial learning-based methods inspired by generative adversarial networks (GANs).

GAN was firstly presented in 2014. The power of GAN models comes from integrating generative modeling techniques alongside deep learning techniques, they utilize supervised learning to perform an unsupervised learning task. The GAN model is basically made of two sub models, a discriminator, and a generator. The discriminator is simply a binary classifier with the task of distinguishing source domain data from target domain data. The generator is a generative network with the task of generating images good enough to confuse the discriminator. The purpose of this min-max game is to minimize the distance between the source and target domains [77]. There has been many updates and modifications in the original GAN model ever since it was introduced by several scholars, but in general, they all share a common essence, which is basically to train two networks in parallel, a generator network (denoted as G) and a discriminator network (denoted as D). The discriminator network is a binary classifier that is trained to distinguish between real and generated images (fake images). The generator network is trained to produce images that are very similar to the real images with a goal to confuse the discriminator in its classification task. Optimality is reached when the generator is producing images with a distribution that is very similar to the real images' distribution to the point where the discriminator can no longer distinguish between the two [78]–[80].

Several GANs have been successfully introduced to suit a variety of real-world applications, but despite their success, the training process lacks a suitable stabilizing

technique. This is due to a number of different phenomena experienced such as Nash-equilibrium, internal covariate shift, mode collapse, vanishing gradient, and lack of proper evaluation metrics [78]. In this research, we are interested in exploring the applicability of Fix-Point GAN to the critical heat flux detection problem. As a continuing effort, the same dataset presented in Chapter 2 is used. Please refer to Section 2.2.1 for details on the dataset.

3.2 Fixed-Point GAN

The Fixed-Point GAN is considered state of the art in the image-to-image translation task. This newly proposed GAN dramatically reduces artifacts and faults in image-to-image translation. It has been shown to outperform the previous state-of-the-art in this task. When given images from two different domains, Fixed-Point GAN learns to translate images from one domain to the other. Figure 12 demonstrates a comparison between Fixed-Point and its predecessor starGAN. When given a data set of positive cancer x-ray images and a data set of negative images, Fixed-Point GAN shows superior capability in generating a “cancer free” image from a positive image which enables for a successful tumor localization by simply subtracting the two images [79].

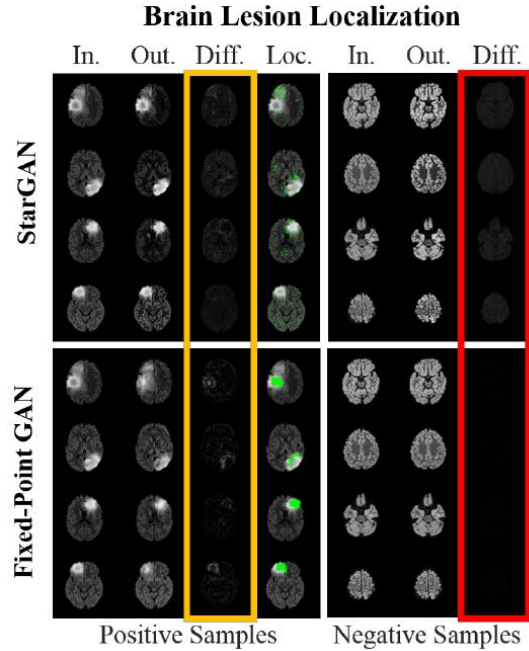


Figure 12. Comparison Between Fixed-point GAN and Its Predecessor StarGAN on the Brain Lesion Localization Task (Siddique Et Al., 2019)

Not only was Fixed-Point GAN dominant in single domain translation, but it has also shown to surpass state-of-the-art GANs in the multi domain translation task where each image has multiple labels. For example, it has shown the ability to outclass previously dominant GANs on the publicly available CelebA data set, where images have multiple domains (labeled by multiple attributes such as hair color, gender, glasses, etc.) and the task is to translate an image from one domain to another without affecting other domains (e.g., changing the hair color from blonde to black without changing the gender or hair style). Fixed-Point GAN differs from its predecessors with its ability to identify a minimal subset of pixels for domain translation or what the authors like to refer to as the “Fixed-Point translation ability.” It does so by modifying the training process to promote the fixed-

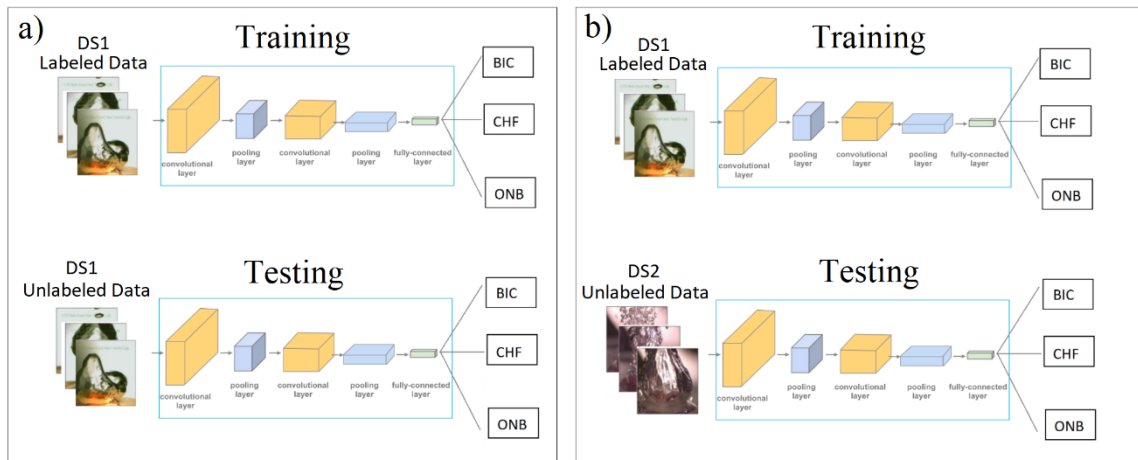


Figure 13. Same Domain Classification Vs Cross Domain Classification

point translation through supervising same-domain translation through an additional identity loss, regularizing cross-domain translation through revised adversarial, domain classification, and cycle consistency loss [79].

3.3 Application of Fixed-Point GAN to Critical Heat Flux Detection

In this section, Fixed-Point GAN is utilized to perform image-to-image translation to transform images coming from one boiling regime to look like images from a different boiling regime. The purpose behind this is to come up with a general methodology tool that will enable researchers to utilize a classifier that was previously trained on a labeled dataset to be used on any other unlabeled data set from a different domain simply by transforming the images from the new domain to the original domain that the model was previously trained on.

Figure 13 illustrates the difference between the classical same domain classification (Figure 13.a) and the cross-domain classification to be achieved (Figure 13.b). To demonstrate this, model CNN-0 from chapter 2 that was purely trained on DS1 will be used

as an example of a base model that could be generalized for usage on images from different domains. Fixed-Point GAN will be applied to generate fake DS1 images from the DS2 data set and test the generated images on the CNN-0 model. This will allow researchers in the heat transfer community to successfully use the model without the need to manually label a specific data set and train a model generated specifically for that data.

Figure 14. summarizes the methodology adopted in this chapter. The red part of the flow chart represents the CNN-0 Model training process, while the blue part represents the Fixed-Point GAN training process. The yellow part is where the results from both methods interact.

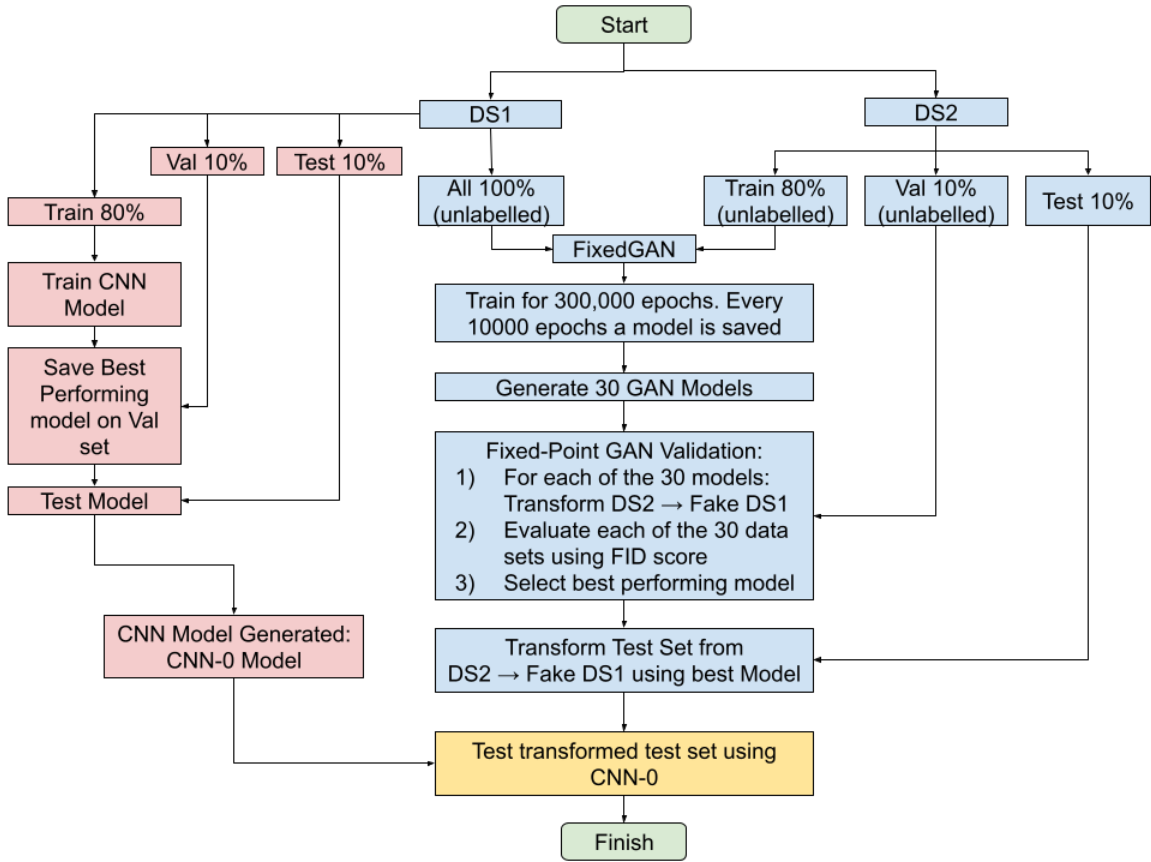


Figure 14. Unsupervised Learning Methodology Flow Chart

3.2.1 CNN-0 Model Training

Figure 13 shows a summary of the entire training methodology adopted in this chapter. The red colored part of the flow chart summarizes the training process adopted to produce the CNN-0 model, which is similar to that used in chapter 2. The data set was divided into three parts: 1) a training set (80%), 2) a validation set (10%), and 3) a test set (10%). A CNN model was trained with the same configurations used in chapter 2 and the training was stopped when the validation loss was no longer improving. The model that performed the best validation loss was selected. Finally, the selected model was blind tested on the test set where the model achieved excellent results as presented in chapter 2. The model was saved to be used in the later stage of the methodology, as viewed in Figure 14.

3.2.2 Fixed-Point GAN Training

The blue colored part of Figure 14 summarizes the procedure used in training the Fixed-Point GAN model. DS2 was divided into three parts: a) training set, b) validation set, and c) testing set. The training set from DS2 was fed into the Fixed-Point GAN model along with 100% of DS1. The GAN model was trained for 300,000 epochs, where every 10,000 epoch a model was saved. A total of 30 GAN models were saved. Unlike the supervised machine learning part, finding a proper evaluation metric that indicates when to stop the training process is not as easy since that the data is not labeled. All evaluation metrics used in the supervised learning process depend on having labels, making them useless in this case. The topic of suitable evaluation metrics for GANs is still an area under exploring within the unsupervised machine learning community. One of the most used metrics is the Fréchet Inception Distance score, or FID for short. The score reflects on the

difference between two sets of images in terms of statistical vision features. The lower the score is, the better the quality of the generated images by the GAN, with a score of 0.000 being the best obtainable score possible. Thus, the FID score was adopted in evaluating the 30 models. The validation data was fed into the 30 models to generate fake DS1 images. Thus, a total of 30 sets of generated images are to be evaluated. The FID score was used to measure the distance between each of the 30 sets and the original DS1 images. The model that scored the lowest FID score on the validation set was selected to generate fake images from the testing data set. Finally, Model CNN-0 was used to classify the newly generated fake images and the results were evaluated using traditional supervised learning metrics.

3.4 Results and Discussion

3.4.1 CNN-0 Model and Blind Cross Domain Testing

In this section, the results obtained from the CNN-0 model training will be presented. Moreover, the results will be compared with blindly testing the CNN-0 model on DS2 blindly before applying the Fixed-Point GAN model. Figure 15 below shows the confusion matrix generated by each experiment and Table 3. Table 3 shows different evaluation metrics for the same experiments. It is quite clear that the model is useless when used on images from a different domain.

a)				b)			
	BIC	CHF	ONB		BIC	CHF	ONB
BIC	308	0	0	BIC	130	0	0
CHF	1	79	0	CHF	124	0	0
ONB	0	0	231	ONB	70	0	0
CNN-0 on DS1				CNN-0 on DS2			

Figure 15. Comparison Between the Confusion Matrices Generated by Using CNN-0 on DS1 (Part a) and DS2 (Part b)

Table 3. Metrics Generated by Using CNN-0 on DS1 and DS2

Experiment	Balanced Accuracy	F1 macro	F1 micro	F1 weighted	Precision macro	Precision micro
CNN-0 on DS1	1.00	1.00	1.00	1.00	1.00	1.00
CNN-0 on DS2	0.33	0.19	0.40	0.23	0.13	0.40
Experiment	Precision weighted	Recall macro	Recall micro	Recall weighted	ROC AUC	ROC AUC
CNN-0 on DS1	1.00	1.00	1.00	1.00	1.00	1.00
CNN-0 on DS2	0.16	0.33	0.40	0.40	0.59	0.58

3.4.2 Fixed-Point GAN Evaluation using FID

Once the Fixed-Point GAN model started training, the model was saved every 10,000 epochs for a total fixed training number of epochs equal to 300,000 epochs, generating a total of 30 models. Because model evaluation in unsupervised machine learning is challenging and it is difficult to decide on when to stop the model from training, each of these 30 models were evaluated using the FID metric. The model that scored the best in the FID metric (lowest value), was selected to be used in generating fake images from the

test data set. **Error! Reference source not found.** shows the FID values generated for the fake images generated from each of the 30 models in comparison with the images in DS1.

Table 4. Fid Values for Images Generated from the Validation Set Using the 30 Gan Models

Model at epoch no.	10000	20000	30000	40000	50000	60000	70000	80000	90000	100000
FID Metric	14.43	14.68	12.08	12.20	11.00	9.64	14.30	12.40	9.98	8.51
Model at epoch no.	110000	120000	130000	140000	150000	160000	170000	180000	190000	200000
FID Metric	9.34	8.25	9.26	7.84	8.13	7.86	8.79	10.06	10.09	9.15
Model at epoch no.	210000	220000	230000	240000	250000	260000	270000	280000	290000	300000
FID Metric	7.19	7.13	7.60	7.18	8.78	9.30	9.72	7.36	7.13	7.44

As seen in **Error! Reference source not found.**, the model saved at epoch 290k was the best scoring model. The 290k model is then applied to the testing data set to generate fake DS1 images. Figure 16 shows samples generated from each class. The first row shows the original DS2 images, while the second row shows the transformed (fake) version of the same image generated using the 290k model.

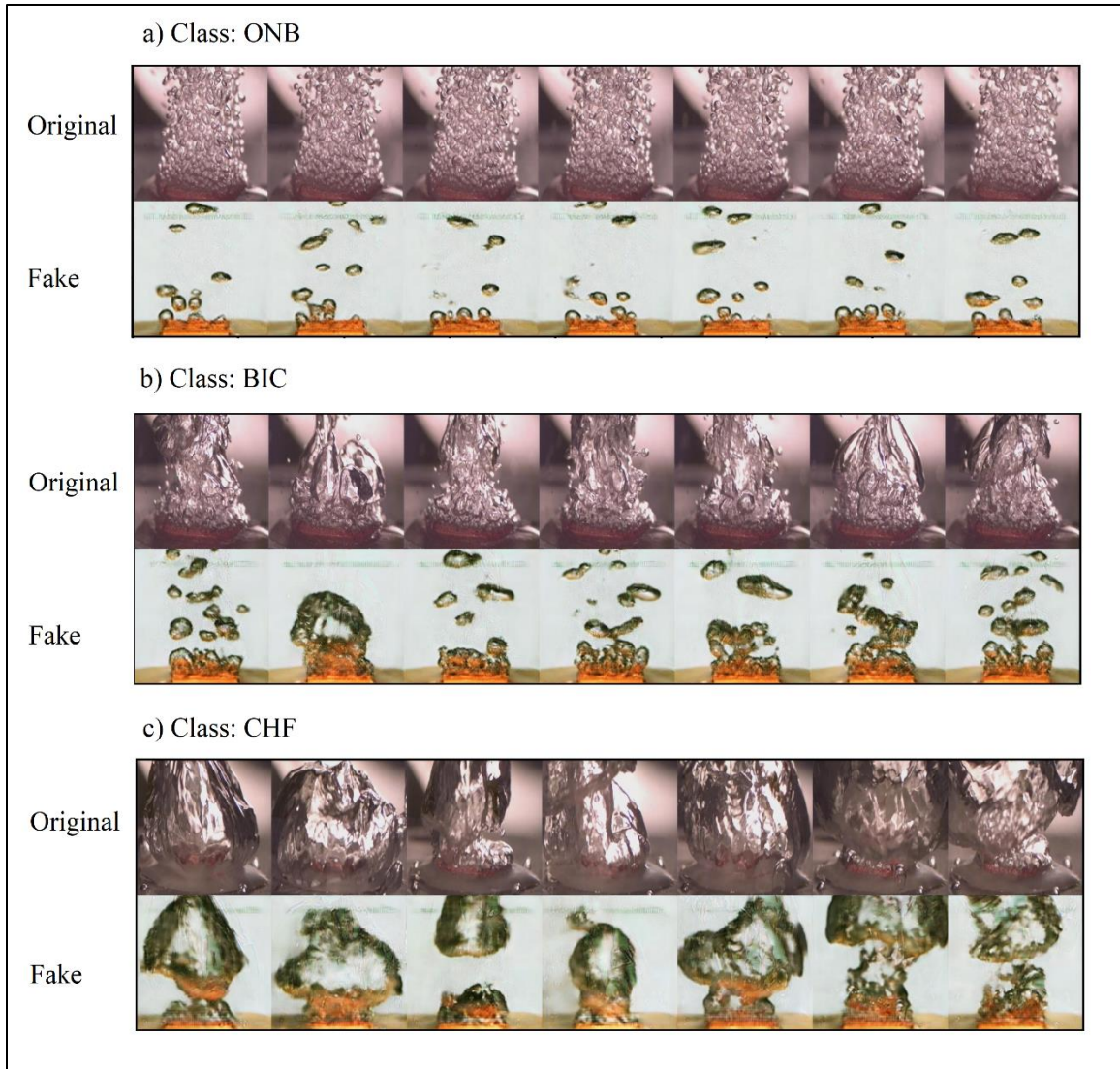


Figure 16. Samples Generated by the 290k Model.

3.4.3 Testing Fix-Point GAN Images using CNN-0

The test set images transformed by the 290k model were tested using the CNN-0 model. Table 5 summarizes the predictions made by the CNN-0 model on the fake images generated by the 290k Fixed-Point GAN model. The table shows that the model is providing incredible success in predicting the “BIC” and “ONB” classes but is struggling to provide accurate predictions for the “CHF” class. The model tends to misclassify them as “BIC” Instead.

Table 5. CNN-0 Predictions on Fake DS1 - Confusion Matrix

	BIC	CHF	ONB
BIC	130	0	0
CHF	112	12	0
ONB	0	0	70

Table 6 shows the evaluation metrics generated for the predictions made by the CNN-0 model on the fake DS1 images.

Table 6. CNN-0 Predictions on Fake DS1 - Evaluation Metrics

Experiment	Balanced Accuracy	F1 macro	F1 micro	F1 weighted	Precision macro	Precision micro
CNN-0 on Fake DS1 (DS2 transformed using Fixed-Point GAN)	0.70	0.63	0.65	0.56	0.85	0.65
Experiment	Precision weighted	Recall macro	Recall micro	Recall weighted	ROC AUC (ovr)	ROC AUC (ovo)
CNN-0 on Fake DS1 (DS2 transformed using Fixed-Point GAN)	0.81	0.70	0.65	0.65	0.82	0.85

3.4.4 Discussion

3.4.4.1 FID Results Discussion

Looking at the evaluation metrics generated for all three experiments in Table 7, it is observed that there is a significant improvement in the results when the images are first transformed from DS2 into fake DS1 and then tested by the CNN-0 model rather than blindly testing the CNN-0 model on DS2 images without any transformation. All metrics has improved significantly, but most importantly, the “balanced accuracy” metric has increased by 112%, and the “AUC (one vs. rest)” has increased by 39%. This clearly demonstrates the effectiveness of using Fixed-Point GAN in transforming the images to a domain that the CNN model has seen before and how this transformation will improve the results obtained by the same model on the same images before the transformation.

Table 7. Comparison Between Evaluation Metrics Generated Using All Methods.

Experiment	Balanced Accuracy	F1 macro	F1 micro	F1 weighted	Precision macro	Precision micro
Blind testing CNN-0 on DS2	0.33	0.19	0.40	0.23	0.13	0.40
CNN-0 on Fake DS1	0.70	0.63	0.65	0.56	0.85	0.65
CNN-0 on DS1	1.00	1.00	1.00	1.00	1.00	1.00
Experiment	Precision weighted	Recall macro	Recall micro	Recall weighted	ROC AUC (ovr)	ROC AUC (ovo)
Blind testing CNN-0 on DS2	0.16	0.33	0.40	0.40	0.59	0.58
CNN-0 on Fake DS1	0.81	0.70	0.65	0.65	0.82	0.85
CNN-0 on DS1	1.00	1.00	1.00	1.00	1.00	1.00

For a more in-depth analysis of the results, the confusion matrices of all three experiments are displayed in Figure 17, and the model shows 100% accuracy on both the “BIC” and the “ONB” labels. Although the model improved slightly in predicting the “CHF” label, it still

fails in predicting most of the “CHF” images correctly. It tends to predict “CHF” images as “BIC” instead.

a)	b)			c)							
	BIC	CHF	ONB		BIC	CHF	ONB		BIC	CHF	ONB
BIC	130	0	0	BIC	130	0	0	BIC	308	0	0
CHF	124	0	0	CHF	112	12	0	CHF	1	79	0
ONB	70	0	0	ONB	0	0	70	ONB	0	0	231
Blind Testing CNN-0 on DS2			CNN-0 on Fake DS1 (DS2 images transformed using Fixed-Point GAN)			CNN-0 on DS1					

Figure 17. Comparison Between the Confusion Matrices Generated Using All Methods.

To further understand the reason behind these misclassifications, a comparison between the real and the fake images for each class during different time intervals was plotted. Figure 18, Figure 19, and Figure 20 show the comparisons for the “ONB”, “BIC”, and “CHF” respectively during different stages of the regime. In Figure 18, The model seems to be able to identify that “ONB” from DS2 should be transformed to DS1 “ONB” images, but it fails however to make the latter stages of DS2 “ONB” look like the later stages of DS1 “ONB”. The classification, however still stands correct and justifies the high accuracy obtained by the model.

Similarly, in Figure 19 the model is able to identify that DS2 “BIC” must be transformed to DS1 “BIC”, but also failed to make the fake intermediate and later stages look like the real stages. Nevertheless, the classifications are correct and explain the results in the confusion matrix.

As illustrated in Figure 20, one starts to observe the reason behind the misclassifications for the “CHF” class. The images seem to be distorted and not as clear as those in the other classes. Moreover, some of the fake images such as the one in the left bottom corner (GIT_CHF1176) show traces of the old domain. The transformation seems immature in this stage which could mean that the model is under trained. It is worth noting that the best selected model using the FID score was at epoch 290,000 while the model trained for a pre-determined number of epochs of 300,000. This is an indication that more training might lead to better results. The reason why the model decided to predict the distorted images as “BIC” might be because the transformation was not mature enough for it to be detected by the model as the domain it was originally trained for, so it decided to predict it as it would with any other domain, which is to predict as the most dominant majority since it is not able to confidently assign it to any other class, which is the same thing that happened when DS2 images were blindly tested using the CNN-0 model, and the model ended up classifying all the images as the majority class “BIC.”

Although the performance of the model is the worst for the “CHF” class, the model in this class doesn’t seem to suffer from the same problem it has with the other classes. The fake, intermediate and final stages look like they were drawn from the same stage. This problem might be attributed to the fact that the number of images in the DS1 “ONB” and “BIC” classes are much larger than the number of images for the same classes in DS2. Looking at Table 1, “ONB” in DS1 is almost four times as many as the “ONB” in DS2 and “BIC” is two-three times as many as “BIC” in DS2, while the gap in number of “CHF” images between the two data sets is much smaller.

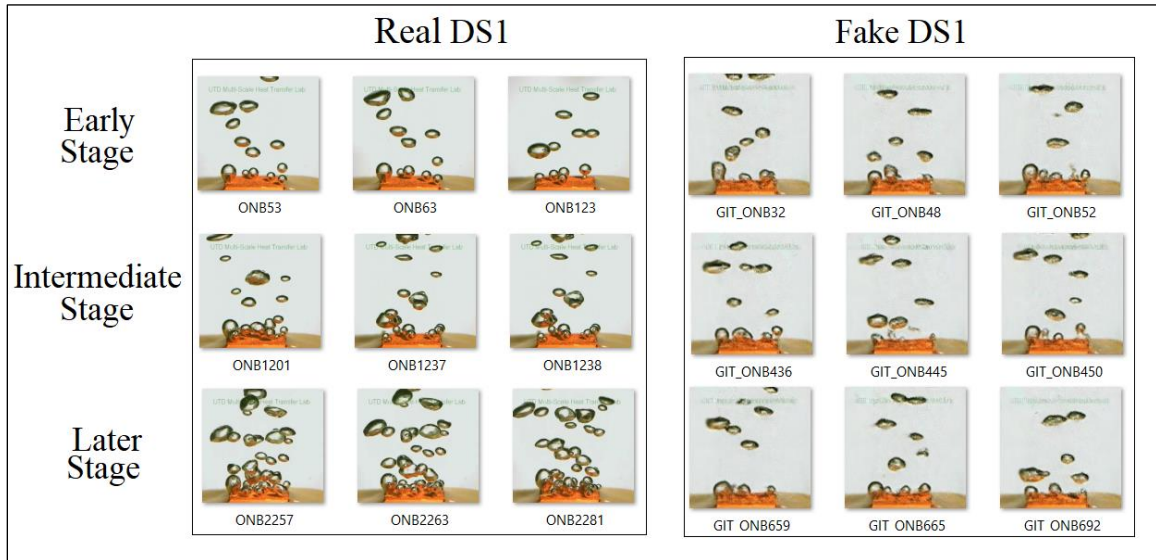


Figure 18. Real Vs Fake "ONB" Images

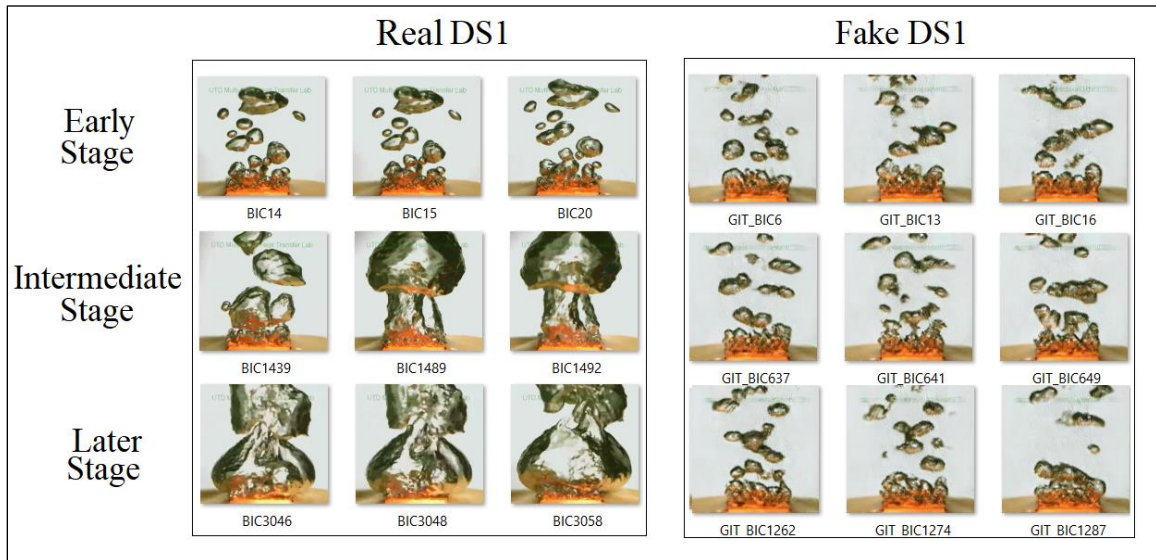


Figure 19. Real Vs Fake "BIC" Images

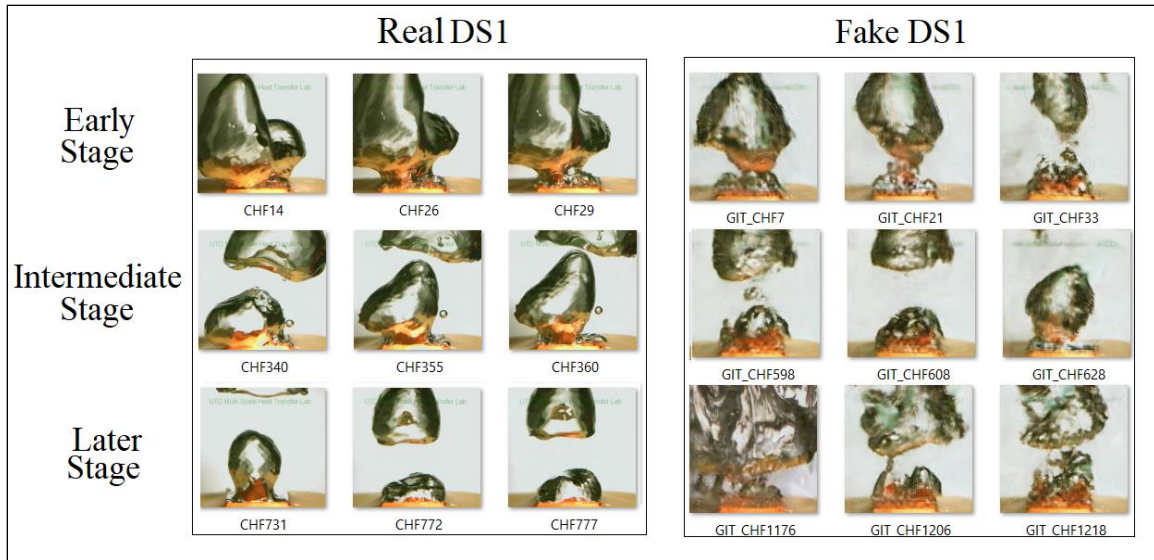


Figure 20. Real Vs Fake "CHF" Images

3.4.4.2 How good is FID in choosing the best GAN model?

Out of all 30 Fixed-Point GAN models generated, how good was the FID metric in choosing the best model? In a real life application, there is no way of knowing without having a labeled validation data set, thus it is not possible to know for sure, but for our case study, we do have the actual labels of the validation set (even though unavailability of the labels was assumed in our methodology) and we could see how far is the FID selected model from the best attainable model out of these 30. Thus, the validation process was done again, but this time using a labeled validation set and traditional evaluation metrics used for supervised learning. Out of the 30 models, the images generated by model 90k achieved the highest balanced accuracy when tested with the CNN-0 model with a value of 79% on the validation set compared to 69% achieved by model 290k that was selected by the FID metric. The 90k model was then used to generate fake images from the test set and these

images were also tested using the CNN-0 model, achieving a balanced accuracy of 81% as compared to the 70% achieved when using the 290k model as seen in Table 8.

Table 8. Metrics Comparison Between the 290k and the 90k Models

Experiment	Balanced Accuracy	F1 macro	F1 micro	F1 weighted	Precision macro	Precision micro
290k model	0.70	0.63	0.65	0.56	0.85	0.65
90k model	0.81	0.81	0.80	0.79	0.89	0.80
Experiment	Precision weighted	Recall macro	Recall micro	Recall weighted	ROC AUC (ovr)	ROC AUC (ovo)
290k model	0.81	0.70	0.65	0.65	0.82	0.85
90k model	0.87	0.81	0.80	0.80	0.94	0.94

The confusion matrices for both models are displayed in Figure 21. There is a notable improvement in the prediction accuracy of the “CHF” class. The accuracy for the “CHF” class improved from 9.7% to 53.2%, while there was a small decrease in performance in predicting the “ONB” class, and the accuracy for the “BIC” class remained the same.

a)				b)			
	BIC	CHF	ONB		BIC	CHF	ONB
BIC	130	0	0	BIC	130	0	0
CHF	112	12	0	CHF	58	66	0
ONB	0	0	70	ONB	7	0	63

CNN-0 on Fake DS1 generated using the 290k model CNN-0 on Fake DS1 generated using the 90k model

Figure 21. Confusion Matrices Comparison Between the 290k and 90k Models

As seen from the results above, although the FID metric provides a good solution, it does not guarantee the best attainable one.

3.5 Conclusion

In conclusion, the overall methodology proved to be a successful solution to the unsupervised cross domain problem facing the heat transfer community. Using a sophisticated generative adversarial network such as the Fixed-Point GAN to transform images from a new domain not seen previously by the model to the domain that the model previously trained with increases the performance substantially and could be considered as a step in the right direction for further future improvement in the solution of this problem. The balanced accuracy improved from 33% (random guessing) to 70%, while the AUC (one vs. rest) improved from 59% to 82%.

Although successful, this methodology does not come without its own challenges and obstacles. First, the model failed to correctly classify most of the “CHF” images. This problem could be attributed to the fact that the model failed in providing good enough images of the “CHF” class to fool the CNN-0 model into thinking it is from the original domain. The images generated looked distorted, and some had traces of their original domain (DS2) which forced the model to predict most of these images as the most dominant class in the imbalanced data set which is the “BIC” class. A suggested solution is to allow the model to train for a longer period of time in the hopes of improving the quality of the images generated. Moreover, the model also failed to transform images into their proper time stage in both the “ONB” and “BIC” stages. This problem was attributed to the major difference in the number of images in both of these classes between the two datasets DS1 and DS2. A suggested solution is to try data synthesizing and augmentation methods to balance the data sets. Tools such as SMOTE [81] or GANs could be used to generate

random images from each class domain to fix the imbalance issue, which is suspected to improve the model's performance.

In addition, we conclude while FID provides a good model suggestion, it was not able to suggest the best suitable model for this problem. Since GANs and their evaluation is still a relatively new topic in the machine learning community, we should expect new performance measures to emerge in the near future.

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1 Conclusion

In summary, the first phase of this work focuses on utilizing supervised and semi-supervised machine learning approaches. In this phase, 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.

The second phase of this work focuses on extending the previous work to include an unsupervised machine learning approach. In conclusion, the overall methodology proved to be a successful solution to the unsupervised cross domain problem facing the heat transfer community. Using a sophisticated generative adversarial network such as the Fixed-Point GAN to transform images from a new domain not seen previously by the model to the domain that the model previously trained with increases the performance

substantially and could be considered as a step in the right direction for further future improvement in the solution of this problem. The balanced accuracy improved from 33% (random guessing) to 70% while the AUC (one vs rest) improved from 59% to 82%. Although successful, this methodology does not come without its own challenges and obstacles. For example, the model failed to correctly classify most of the “CHF” images. In addition, further exploration of FID is needed to help develop a new evaluation metric.

4.2 Future Work

4.2.1 Model Performance Improvement

To the best of our knowledge, the proposed unsupervised learning approach is the first of its kind to utilize GANs to solve the boiling heat transfer problem within the heat transfer community. As the case in any unprecedented work, there is always a spacious room for improvement. There are different areas that could be focused on for future research work to improve the performance of this methodology. One of these areas is to focus on the hyper parameters tuning for the GAN model. Parameters such as the number of epochs to train, the architecture used, etc., could be optimized for better image generation results. Another area to focus on is providing a more sophisticated evaluation metric than the traditional FID used in this approach. As demonstrated in Chapter 3, the 290k model suggested to be used by the metric turned out to be sub-optimal. A better evaluation metric will help boost the performance of the model significantly. Moreover, the data itself could be replaced. There are many different data sources from different setups and different liquids under study. Using other data sources as the base model is a

topic to be investigated. Rather than having DS1 as the data for the base model, perhaps a different base model used from a different data set could prove to be a better model for generalization.

4.2.2 Black Box Unraveling

Although these algorithms and methodologies are producing very promising results for the heat transfer community, there still is a way to go. Heat transfer scholars to this day are still trying to unravel the ambiguity of the boiling heat transfer mechanism. Convolutional neural networks demonstrated their abilities in detecting the boiling regime with high accuracy merely by looking at the images, a task that experts in the field are unable to do without the assistance of quantitative measures such as the heat transfer coefficients and the temperature. Making sense of what enables the CNN models to correctly identify the boiling regime could help scholars make sense of this critical physical phenomena. Tools such as Auto Encoders and Attention Maps could be used for this purpose. Figure 22 and Figure 23 show initial efforts to understand what are the most important features of the image that is most important in generating these accurate predictions. Further work and analysis using such methods seem very promising in helping scientists better understand the boiling heat transfer mechanism.

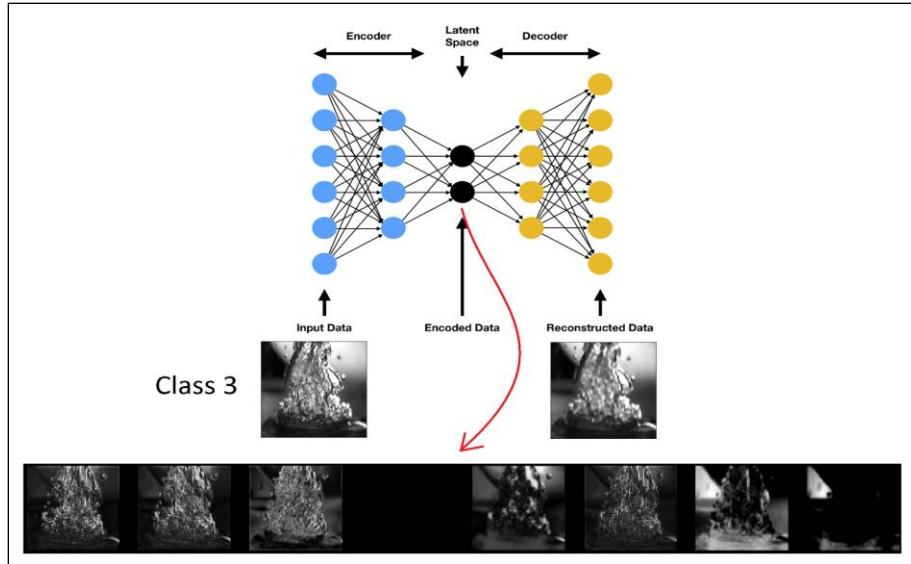


Figure 22. CNN Features Extraction Using Auto Encoders

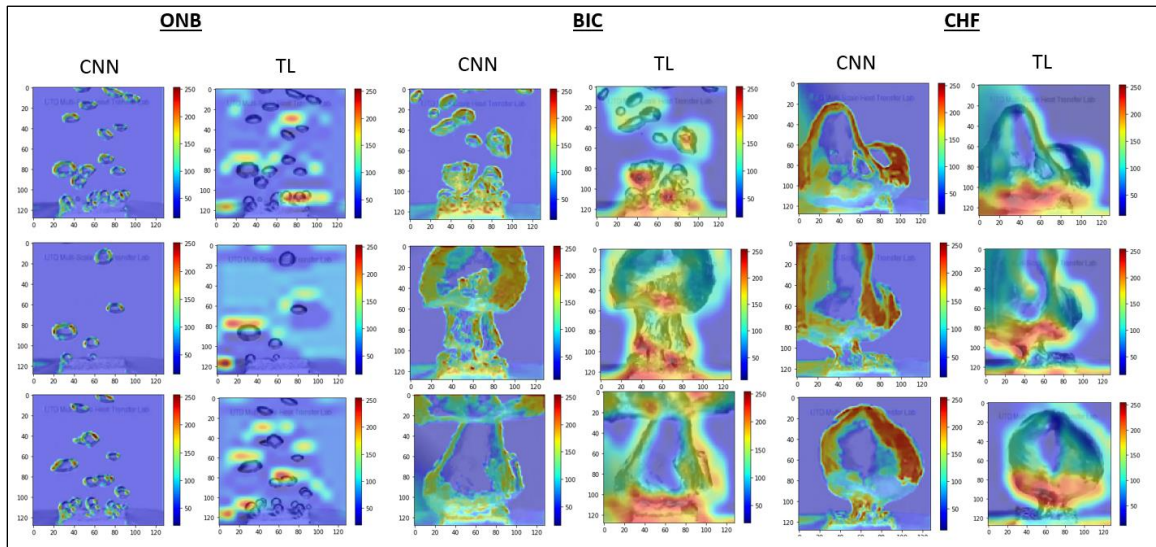


Figure 23. Samples of Attention Maps Generated from Both the CNN and TL Models for Each Class.

Another method worth exploring is utilizing the localization capabilities of Fixed-Point GAN. Siddique et al. [79] used their Fixed-Point GAN to generate negative tumor medical

images from positive tumor medical images, or what was referenced by the authors as “virtual healing.” Having two versions of the same image (one infected and one healed) and subtracting the two images from each other would result in an image showing only the location of the infected area. Similar to this approach, we could take an image from any regime and transfer it to another regime and then subtract the two to understand what it is that distinguishes one regime from the other. Correlating the features from these features/locations with the physical properties of the known boiling theories should help scientists better understand this phenomenon.

REFERENCES

- [1] S. M. Rassoulinejad-Mousavi *et al.*, “Deep learning strategies for critical heat flux detection in pool boiling,” *Applied Thermal Engineering*, vol. 190, no. October 2020, p. 116849, 2021, doi: 10.1016/j.applthermaleng.2021.116849.
- [2] G. M. Hobold and A. K. da Silva, “Visualization-based nucleate boiling heat flux quantification using machine learning,” *International Journal of Heat and Mass Transfer*, vol. 134, pp. 511–520, 2019, doi: 10.1016/j.ijheatmasstransfer.2018.12.170.
- [3] H. Hu, C. Xu, Y. Zhao, K. J. Ziegler, and J. N. Chung, “Boiling and quenching heat transfer advancement by nanoscale surface modification,” *Scientific Reports*, vol. 7, no. 1, pp. 1–16, 2017, doi: 10.1038/s41598-017-06050-0.
- [4] Y. Suh, R. Bostanabad, and Y. Won, “Deep learning predicts boiling heat transfer,” *Scientific Reports*, vol. 11, no. 1, pp. 1–10, 2021, doi: 10.1038/s41598-021-85150-4.
- [5] L. Ruthotto and E. Haber, “Deep Neural Networks Motivated by Partial Differential Equations,” *Journal of Mathematical Imaging and Vision*, vol. 62, no. 3, pp. 352–364, 2020, doi: 10.1007/s10851-019-00903-1.
- [6] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K. R. Müller, “Evaluating the visualization of what a deep neural network has learned,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 11, pp. 2660–2673, 2017, doi: 10.1109/TNNLS.2016.2599820.
- [7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks.” [Online]. Available: <http://code.google.com/p/cuda-convnet/>
- [8] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14, 2015.

- [9] Z. Qin, F. Yu, C. Liu, and X. Chen, “How convolutional neural networks see the world - A survey of convolutional neural network visualization methods,” *arXiv*, vol. 1, no. 2, pp. 149–180, 2018, doi: 10.3934/mfc.2018008.
- [10] J. Dirker *et al.*, “Thermal energy processes in direct steam generation solar systems: Boiling, condensation and energy storage—A review,” *Frontiers in Energy Research*, vol. 6, no. March, p. 147, 2019, doi: 10.3389/fenrg.2018.00147.
- [11] P. Birbarah *et al.*, “Water immersion cooling of high power density electronics,” *International Journal of Heat and Mass Transfer*, vol. 147, p. 118918, 2020.
- [12] M. S. El-Genk, “Immersion cooling nucleate boiling of high power computer chips,” *Energy conversion and management*, vol. 53, no. 1, pp. 205–218, 2012.
- [13] S. G. Kandlikar, “Review and projections of integrated cooling systems for three-dimensional integrated circuits,” *Journal of Electronic Packaging*, vol. 136, no. 2, 2014.
- [14] H. Fenech, *Heat transfer and fluid flow in nuclear systems*. Elsevier, 2013.
- [15] V. K. Dhir, “Mechanistic prediction of nucleate boiling heat transfer—achievable or a hopeless task?,” 2006.
- [16] N. Zuber, *Hydrodynamic aspects of boiling heat transfer*. United States Atomic Energy Commission, Technical Information Service, 1959.
- [17] J. H. Lienhard and V. K. Dhir, “Hydrodynamic prediction of peak pool-boiling heat fluxes from finite bodies,” 1973.
- [18] S. G. Kandlikar, “A theoretical model to predict pool boiling CHF incorporating effects of contact angle and orientation,” *J. Heat Transfer*, vol. 123, no. 6, pp. 1071–1079, 2001.

- [19] M. M. Rahman, E. Olceroglu, and M. McCarthy, “Role of wickability on the critical heat flux of structured superhydrophilic surfaces,” *Langmuir*, vol. 30, no. 37, pp. 11225–11234, 2014.
- [20] H. Hu, J. A. Weibel, and S. V. Garimella, “A coupled wicking and evaporation model for prediction of pool boiling critical heat flux on structured surfaces,” *International Journal of Heat and Mass Transfer*, vol. 136, pp. 373–382, 2019.
- [21] L. Zhang, J. H. Seong, and M. Bucci, “Percolative scale-free behavior in the boiling crisis,” *Physical review letters*, vol. 122, no. 13, p. 134501, 2019.
- [22] 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.
- [23] J. R. Barbosa Jr, A. H. Govan, and G. F. Hewitt, “Visualisation and modelling studies of churn flow in a vertical pipe,” *International Journal of Multiphase Flow*, vol. 27, no. 12, pp. 2105–2127, 2001.
- [24] P. J. Waltrich, G. Falcone, and J. R. Barbosa Jr, “Axial development of annular, churn and slug flows in a long vertical tube,” *International journal of multiphase flow*, vol. 57, pp. 38–48, 2013.
- [25] C. E. Brennen and C. E. Brennen, *Fundamentals of multiphase flow*. Cambridge university press, 2005.
- [26] T. Cong, R. Chen, G. Su, S. Qiu, and W. Tian, “Analysis of CHF in saturated forced convective boiling on a heated surface with impinging jets using artificial neural network and genetic algorithm,” *Nuclear engineering and design*, vol. 241, no. 9, pp. 3945–3951, 2011.
- [27] T. Cong, G. Su, S. Qiu, and W. Tian, “Applications of ANNs in flow and heat transfer problems in nuclear engineering: a review work,” *Progress in Nuclear Energy*, vol. 62, pp. 54–71, 2013.

- [28] H. Alimoradi and M. Shams, "Optimization of subcooled flow boiling in a vertical pipe by using artificial neural network and multi objective genetic algorithm," *Applied Thermal Engineering*, vol. 111, pp. 1039–1051, 2017.
- [29] Y. Liu, N. Dinh, Y. Sato, and B. Niceno, "Data-driven modeling for boiling heat transfer: using deep neural networks and high-fidelity simulation results," *Applied Thermal Engineering*, vol. 144, pp. 305–320, 2018.
- [30] M. Hassanpour, B. Vaferi, and M. E. Masoumi, "Estimation of pool boiling heat transfer coefficient of alumina water-based nanofluids by various artificial intelligence (AI) approaches," *Applied Thermal Engineering*, vol. 128, pp. 1208–1222, 2018.
- [31] G. M. Hobold and A. K. da Silva, "Machine learning classification of boiling regimes with low speed, direct and indirect visualization," *International Journal of Heat and Mass Transfer*, vol. 125, pp. 1296–1309, 2018.
- [32] G. M. Hobold and A. K. da Silva, "Automatic detection of the onset of film boiling using convolutional neural networks and Bayesian statistics," *International Journal of Heat and Mass Transfer*, vol. 134, pp. 262–270, 2019.
- [33] M. Ravichandran and M. Bucci, "Online, quasi-real-time analysis of high-resolution, infrared, boiling heat transfer investigations using artificial neural networks," *Applied Thermal Engineering*, vol. 163, p. 114357, 2019.
- [34] M. v Valueva, N. N. Nagornov, P. A. Lyakhov, G. v Valuev, and N. I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Mathematics and Computers in Simulation*, 2020.
- [35] W. H. L. Pinaya, S. Vieira, R. Garcia-Dias, and A. Mechelli, "Convolutional neural networks," in *Machine Learning*, Elsevier, 2020, pp. 173–191.
- [36] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997, doi: 10.1109/72.554195.

- [37] F. Gao *et al.*, “SD-CNN: A shallow-deep CNN for improved breast cancer diagnosis,” *Computerized Medical Imaging and Graphics*, vol. 70, pp. 53–62, 2018.
- [38] G. B. Frisoni, N. C. Fox, C. R. Jack, P. Scheltens, and P. M. Thompson, “The clinical use of structural MRI in Alzheimer disease,” *Nature Reviews Neurology*, vol. 6, no. 2, pp. 67–77, 2010.
- [39] M. Havaei *et al.*, “Brain tumor segmentation with deep neural networks,” *Medical image analysis*, vol. 35, pp. 18–31, 2017.
- [40] A. de Brebisson and G. Montana, “Deep neural networks for anatomical brain segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 20–28.
- [41] A. Esteva *et al.*, “Dermatologist-level classification of skin cancer with deep neural networks,” *nature*, vol. 542, no. 7639, pp. 115–118, 2017.
- [42] C. J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, and T. Poinso, “Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates,” *Combustion and Flame*, vol. 203, pp. 255–264, 2019.
- [43] S. S. Abdurakipov, O. A. Gobyrov, M. P. Tokarev, and V. M. Dulin, “Combustion Regime Monitoring by Flame Imaging and Machine Learning,” *Optoelectronics, Instrumentation and Data Processing*, vol. 54, no. 5, pp. 513–519, 2018.
- [44] S. Ye, Z. Zhang, X. Song, Y. Wang, Y. Chen, and C. Huang, “A flow feature detection method for modeling pressure distribution around a cylinder in non-uniform flows by using a convolutional neural network,” *Scientific Reports*, vol. 10, no. 1, pp. 1–10, 2020.
- [45] S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik, “Prediction of aerodynamic flow fields using convolutional neural networks,” *Computational Mechanics*, vol. 64, no. 2, pp. 525–545, 2019.

- [46] M. L. Dering and C. S. Tucker, "A Convolutional Neural Network Model for Predicting a Product's Function, Given Its Form," *Journal of Mechanical Design*, vol. 139, no. 11, 2017.
- [47] S. Ye, B. Li, Q. Li, H.-P. Zhao, and X.-Q. Feng, "Deep neural network method for predicting the mechanical properties of composites," *Applied Physics Letters*, vol. 115, no. 16, p. 161901, 2019.
- [48] Z. Chen, K. Gryllias, and W. Li, "Mechanical fault diagnosis using convolutional neural networks and extreme learning machine," *Mechanical Systems and Signal Processing*, vol. 133, p. 106272, 2019.
- [49] Y. Yu, C. Wang, X. Gu, and J. Li, "A novel deep learning-based method for damage identification of smart building structures," *Structural Health Monitoring*, vol. 18, no. 1, pp. 143–163, 2019.
- [50] H. Khodabandehlou, G. Pekcan, and M. S. Fadali, "Vibration-based structural condition assessment using convolution neural networks," *Structural Control and Health Monitoring*, vol. 26, no. 2, p. e2308, 2019.
- [51] Y. Cha, W. Choi, and O. Büyüköztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.
- [52] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, "Real-time motor fault detection by 1-D convolutional neural networks," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 11, pp. 7067–7075, 2016.
- [53] M. Talo, U. B. Baloglu, Ö. Yıldırım, and U. R. Acharya, "Application of deep transfer learning for automated brain abnormality classification using MR images," *Cognitive Systems Research*, vol. 54, pp. 176–188, 2019.
- [54] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

- [55] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, p. 9, 2016.
- [56] K. Wang, B. K. Patel, L. Wang, T. Wu, B. Zheng, and J. Li, "A dual-mode deep transfer learning (D2TL) system for breast cancer detection using contrast enhanced digital mammograms," *IISE Transactions on Healthcare Systems Engineering*, vol. 9, no. 4, pp. 357–370, 2019.
- [57] B. Q. Huynh, H. Li, and M. L. Giger, "Digital mammographic tumor classification using transfer learning from deep convolutional neural networks," *Journal of Medical Imaging*, vol. 3, no. 3, p. 34501, 2016.
- [58] F. Gao, H. Yoon, T. Wu, and X. Chu, "A feature transfer enabled multi-task deep learning model on medical imaging," *Expert Systems with Applications*, vol. 143, p. 112957, 2020.
- [59] T. Zeng and S. Ji, "Deep convolutional neural networks for multi-instance multi-task learning," in *2015 IEEE International Conference on Data Mining*, 2015, pp. 579–588.
- [60] S.M. You, *Pool boiling*.
- [61] H. Minseok, B. Bertina, and S. Graham, "Pool Boiling Experiment," 2014.
- [62] S. Jun, J. Kim, D. Son, H. Y. Kim, and S. M. You, "Enhancement of pool boiling heat transfer in water using sintered copper microporous coatings," *Nuclear Engineering and Technology*, vol. 48, no. 4, pp. 932–940, 2016.
- [63] S. Jun, J. C. Godinez, S. M. You, and H. Y. Kim, "Pool boiling heat transfer of a copper microporous coating in borated water," *Nuclear Engineering and Technology*, vol. 52, no. 9, pp. 1939–1944, 2020.
- [64] S. Jun, J. Kim, S. M. You, and H. Y. Kim, "Effect of heater orientation on pool boiling heat transfer from sintered copper microporous coating in saturated water," *International Journal of Heat and Mass Transfer*, vol. 103, pp. 277–284, 2016.

- [65] S. Jun, H. Wi, A. Gurung, M. Amaya, and S. M. You, “Pool boiling heat transfer enhancement of water using brazed copper microporous coatings,” *Journal of Heat Transfer*, vol. 138, no. 7, 2016.
- [66] M. Ha and S. Graham, “Pool boiling characteristics and critical heat flux mechanisms of microporous surfaces and enhancement through structural modification,” *Applied Physics Letters*, vol. 111, no. 9, p. 91601, 2017.
- [67] M. Ha and 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)*, 2016, pp. 388–394.
- [68] M. Ha and S. Graham, “Pool boiling enhancement using vapor channels in microporous surfaces,” *International Journal of Heat and Mass Transfer*, vol. 143, p. 118532, 2019.
- [69] F. Gao, T. Wu, X. Chu, H. Yoon, Y. Xu, and B. Patel, “Deep Residual Inception Encoder-Decoder Network for Medical Imaging Synthesis,” *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 1, pp. 39–49, 2020, doi: 10.1109/JBHI.2019.2912659.
- [70] F. Zhuang *et al.*, “A comprehensive survey on transfer learning,” *Proceedings of the IEEE*, 2020.
- [71] Y. Zhu, F. Zhuang, and D. Wang, “Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 5989–5996.
- [72] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” in *International conference on artificial neural networks*, 2018, pp. 270–279.
- [73] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and transferring mid-level image representations using convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.

- [74] G. Wilson and D. J. Cook, “A Survey of Unsupervised Deep Domain Adaptation”.
- [75] S. Zhao *et al.*, “A Review of Single-Source Deep Unsupervised Visual Domain Adaptation,” pp. 1–21.
- [76] B. Benjdira, Y. Bazi, A. Koubaa, and K. Ouni, “Unsupervised domain adaptation using generative adversarial networks for semantic segmentation of aerial images,” *Remote Sensing*, vol. 11, no. 11, 2019, doi: 10.3390/rs11111369.
- [77] Y. Zhang, H. Ye, and B. D. Davison, “Adversarial Reinforcement Learning for Unsupervised Domain Adaptation,” pp. 635–644.
- [78] A. Jabbar, X. Li, and B. Omar, “A Survey on Generative Adversarial Networks: Variants, Applications, and Training,” pp. 1–38, 2020, [Online]. Available: <http://arxiv.org/abs/2006.05132>
- [79] M. R. Siddiquee, Z. Zhou, N. Tajbakhsh, and R. Feng, “Learning Fixed Points in Generative Adversarial Networks : From Image-to-Image Translation to Disease Detection and Localization,” no. Iccv, 2019.
- [80] M. I. Translation, “StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation”.
- [81] N. v Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” 2002.