Hybrid Learning for Leakage Detection in Sealed Detergent Containers using IR-Thermography

UNIVERSITY OF TWENTE.

Submitted by

Akash Ravi Prame

MSc Embedded Systems

Supervisors

dr.ir. M. van Keulen
Associate Professor
Faculty of EEMCS

dr. C.G. Zeinstra
Assistant Professor
Faculty of EEMCS

ir. V. Arnaoutis
Researcher
Fraunhofer Project Center (FPC)

ir. K.A. Ramaker
Lead Data Engineer
Tembogroup B.V.

Tembo

Data Management and Biometrics (DMB)
Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente
&
Dept. of Data Engineering
TDC (a company of the Tembo Group)

August, 2021

Acknowledgements

This report documents the work that I carried out during my time at Tembo group from January-August 2021. During this period, I had some of the best learning experiences of my life. I found it extremely challenging and equally rewarding at the same time. I would like to thank Kenny for making the entire process smooth and pleasurable for me. I also thank Maurice, Chris and Vasos for their guidance throughout this project and their invaluable insights that were key in shaping the outcome of this project. I would like to thank everyone at Tembo - Rick, Jensen, Edwin for sharing your knowledge at different occasions and making this journey memorable for me.

I am grateful to my parents for always believing in me and supporting me in pursuing masters. Finally, I would like to thank my girlfriend and all my friends for cheering me up during the dull days and keeping my morale high.

Contents

List of Figures

List of Tables

ABSTRACT

Quality inspection plays a critical role in the manufacturing industry. With the recent popularity of detergent pods, ensuring utmost product quality has become a necessity for detergent manufacturers. Due to the detergent pod containers being opaque, manual quality inspection becomes slow and infeasible. Therefore, there is a need for a non-destructive testing (NDT) technique to automatically detect fluid leakage inside sealed containers.

The focus of this study is to develop a method to automatically detect the presence of leakage in sealed containers. Infrared thermography (IRT) has been applied successfully by other researchers for quality inspection in cases where the test specimen is out of direct line-of-sight. Therefore, IRT has been identified as a suitable method to capture the information required for this task. Therefore, using thermal image data, we aim to build an image classification system to distinguish between instances of leakage and non-leakage.

We propose three alternate approaches for this task, namely handcrafted feature-based approach, convolutional neural network (CNN) based approach and a hybrid fusion approach combining multiple feature sources or classifiers, or both. The CNN model outperforms the baseline feature-based approach with a 4-fold accuracy of 94.48%. The two hybrid fusion schemes namely, late-fusion and early-fusion provide an improvement to the pure CNN approach with a highest overall accuracy of 95.63% obtained over a 4-fold cross validation split.

Keywords: *Leakage recognition, infrared thermography, convolutional neural networks, hybrid deep learning, feature fusion*

Chapter 1

Introduction

The process of product quality inspection is an essential part of every manufacturing process and with an increasing trend towards automation in the manufacturing industry, quality assurance has become a necessity [1]. Earlier, the inspection was done by experts manually and was highly prone to human error. Therefore, more and more industries have adopted automatic quality inspection in their production routine [2]. Several different approaches exist for automatic industrial quality inspection. In most applications, it is essential that the product under inspection is not damaged during the inspection process and this is achieved by Non-Destructive Testing (NDT).

FIGURE 1.1: Example of Leakage inside Detergent Pod Container

In recent years, a popular way of packaging liquid detergent is in the form of one-time use pods. There is a growing trend in the use of detergent pods by customers and the growth of this market is only expected to increase in the future [3]. Hence, detergent manufacturers want to ensure the quality of the pods until the last step of the production process. A common problem faced by customers buying detergent pods is that sometimes the box may contain a broken pod resulting in leakage of fluid within the container. Therefore, it is essential that the containers are devoid of leaking pods before they leave the factory. According to recent European Union regulations [4], the containers are made opaque which makes it difficult to

visually inspect the containers after the packaging process.

Defect recognition or quality inspection of manufactured products has been explored exclusively for various products [5]. The focus of this work is detecting fluid leakage in laundry detergent pod containers. It is expected that if a leaking pod is present in the container, the leaking detergent will eventually reach the bottom. Therefore, by heating the bottom surface to a certain extent, a temperature difference can be created between the liquid and the material of the container. An appropriate method to capture this would be to use thermal imaging in the infrared spectrum that can penetrate the material of the containers [6]. The task of fluid leakage detection may be considered similar to the general problem of defect detection in many ways. Leakage has texture properties that can be used to distinguish it from its surroundings. In most real-world situations, fluid leakage is out of direct line-of-sight, which makes the task of acquiring images difficult. Figure 1.2 shows examples of thermal images containing leakage and non-leakage. The leakage is highlighted by the bounding box in the image.

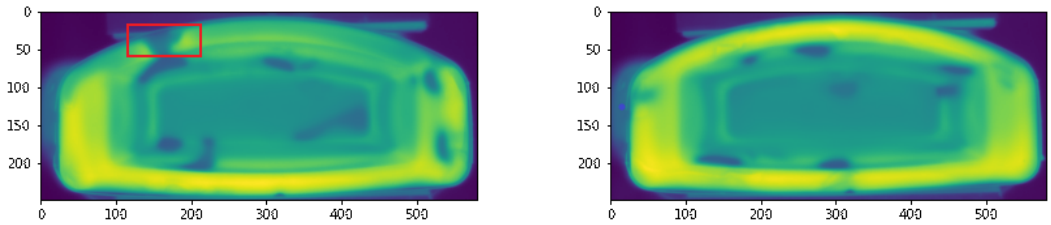


FIGURE 1.2: Infrared thermal images showing leakage (left) and non-leakage (right)

The task of recognizing leakage from thermal images can be considered as an image classification problem. Generally, image classification for industrial quality inspection has been performed in several ways. Some of the earliest methods were digital image processing (DIP) techniques based on a set of empirical rules to directly identify and measure defects [7]. More recent methods treat this as a supervised learning problem [8], where a number of features are extracted from the image and are used as input to a machine learning model such as support vector machines (SVM) [9]. The feature extraction may involve statistical methods such as

color histograms and Principal Component Analysis (PCA) to derive important features from the image ([10]) as well as image processing techniques to extract information such as shapes, textures, etc. [11]. The most recent advancements in image processing applications are deep learning based methods. In 2012, Krizhevsky et. al. ([12]) approached the 1000-Class ImageNet classification challenge with a Convolutional Neural Network (CNN) for the first time and obtained ground-breaking results compared to previous state-of-the-art methods such as Histogram of Oriented Gradients (HOG) [13]. Ever since, CNNs have dominated the field of deep learning for image and signal processing applications. The key difference in this approach is that the feature extraction step is implicit and is embedded within the neural network. They are referred to as end-to-end learning models since all the parameters are learned through training data rather than being fine-tuned by experts [2].

Apart from the methods discussed above, a few recent works have explored hybrid approaches. Hybrid learning combines two or more of the above methods to improve the overall performance. For example, the feature extraction step can be done using a CNN and a different traditional classifier such as Random Forest may be used in place of the fully connected layers of a CNN. Other methods that combine multiple feature sources even before classification have also been investigated by researchers.

1.1 Problem Statement

Production of laundry detergent pods is a growing market and there is a compelling need to ensure their quality throughout the manufacturing process. A problem faced by manufacturers is to detect fluid leakage inside sealed containers after the packaging process. It is even more challenging because recent EU regulations require detergent containers to be opaque. Therefore, the problem at hand is to automatically detect the presence of leakage in sealed detergent pod containers and the main challenge is to do so without opening the containers. Infrared thermography is chosen as the method of data acquisition. This research aims to find a suitable method to classify instances of leakage and non-leakage from thermal images.

1.2 Research Questions

For the problem stated above, a primary research question is identified and three smaller sub-questions are defined to help solve the main objective of the project.

The main research question is defined as follows:
What are the steps involved in building an image classification system to distinguish between instances of leakage and non-leakage from infrared thermal (IRT) images of detergent containers?

The sub-questions are defined as follows:

- Which feature extraction techniques may be used to extract information that distinguish between leakage and non-leakage from IRT images of detergent containers?
- How do handcrafted feature-based methods compare to convolutional neural networks in terms of classification performance?
- Do hybrid techniques that combine multiple feature sources or classifiers improve the overall recognition performance?

In addition to the above research questions, an auxiliary research question is defined as follows:

- What effect does dataset size have on the performance of the models and how to identify the amount of data required to achieve a certain level performance?

1.3 Document Outline

The rest of the document is organized as follows:

- In Chapter 2, a survey of related work in the field of automatic defect classification is presented.
- Chapter 3 provides a background to this work by briefly explaining theoretical concepts involved in data acquisition, feature extraction and classification.
- The different methods proposed in this study for the problem of leakage recognition from IRT images of detergent containers are explained in Chapter 4.
- In Chapter 5, the experimental setup is described and the results obtained from the experiments are presented in detail.
- Chapter 6 provides a conclusions with respect the research questions formulated above. A discussion on the obtained results and recommendations for future work are also given.

Chapter 2

Literature Survey

In this chapter, related work from relevant literature is discussed with respect to the research questions defined in Chapter 1.

Generally, quality inspection may involve different tasks depending on the application, such as determination of defects and conformity checking. For the purpose of this research, we consider only defect recognition. The problem of recognizing defects in manufactured products may be considered as an image classification problem where the decision of whether the product contains a certain defect is made from an image of the product captured in a controlled industrial environment. This problem has been solved in multiple ways in the past, but broadly, these methods can be divided into three categories (in chronological order): DIP based methods, feature-based supervised learning methods and deep learning methods. Figure 2.1 shows the different categories of visual defect recognition techniques. In the following sections, a few examples from each category that were found relevant to the current problem statement are discussed.

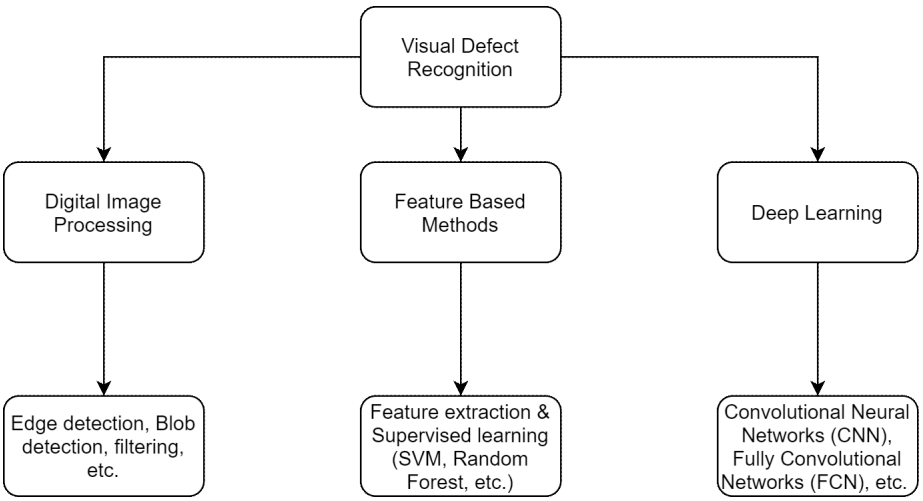


FIGURE 2.1: Classification of Visual Defect Recognition Methods

2.1 Digital Image Processing

The automatic recognition of defects using imaging techniques is a well-established topic of research. Some of the earliest applications of defect detection predominantly involved Digital Image Processing (DIP) based on a set of empirical rules to directly identify and measure defects. Li et al. (2002) [14] proposed a method for detection of surface defects in apples. For the detection and localization of defects in the input image, simple image processing techniques such as background subtraction and thresholding are used. In a similar application by Mak et al. (2009) [15], a system to identify fabric defects is proposed. It consists of a sequence of morphological operations and filtering operations to obtain a final binary image, from which the defects can easily be localized by applying a suitable threshold. The different steps involved are linear opening, linear closing and median blur filtering. Another work by Rahaman et al. (2009) [16] shows an automatic defect recognition system for ceramic tiles, where the images containing specific defects are used as reference. After capturing the images and performing image enhancement and edge detection, each image is compared with the reference image of a particular defect and a similarity score is generated to determine the specific type of defect. DIP based methods have also been used with thermal imaging. Tsanakas et al. [17] make use of image processing techniques on thermal images for diagnosis of defects in Photo-voltaic (PV) cells. The thermal images of PV modules are first converted to grayscale range. Then, a Canny edge detection is performed followed by a thresholding step to localize defective regions within the images.

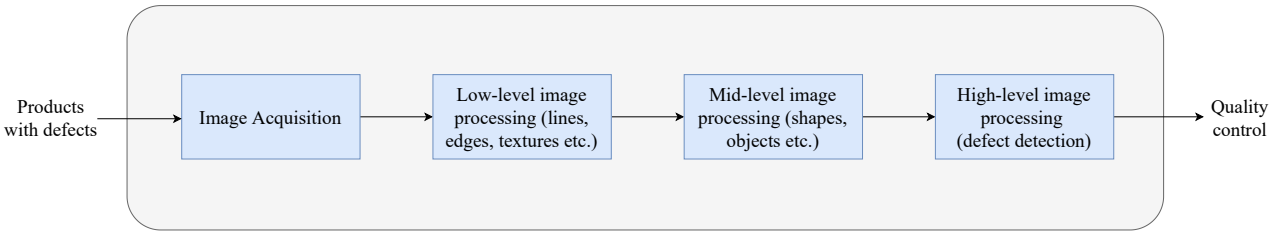


FIGURE 2.2: General Overview of Digital Image Processing Application for Defect Recognition

From the above examples, it can be seen that most of the applications involving DIP generally consist of a pre-processing stage where the image is prepared for further operations by choosing suitable pixel representations, normalizing pixel ranges and so on. It is followed by a sequence of steps such as filtering, kernel operations and morphological image processing to further enhance the image for the classification. Then, some high-level operations such as edge detection, shape detection and

blob detection depending on the application may be used. Finally the decision is made by analyzing the images and carefully choosing conditional rules defining the presence or absence of defects.

The advantages of DIP methods are: they can be implemented with a small number of images, they have a low computational requirement compared to more recent methods. They can be deployed efficiently in stable environments where the images can be acquired with high repeatability. However, they do not suit well for applications where the nature of the defect cannot be completely defined before hand. Figure 2.2 shows the outline of a typical DIP application.

2.2 Feature Based Methods

The next class of methods commonly used for defect classification are feature based methods. As introduced in Chapter 1, these methods involve deriving features from images and using them for training supervised classification models. The objective of feature extraction is to transform the input from a high-dimensional image space to a reduced feature vector X. Once the feature vector is available, the problem can be treated as a supervised learning problem, where X is mapped to the output y using a function ϕ as y = $\phi(X)$ + ϵ, where ϵ is the unaccountable error or bias. In other words, the mapping function can be described by several different supervised learning algorithms that can fit the relationship between the input and output by learning from training examples.

It can be noticed that DIP and feature based methods are similar in many ways. The major difference between the earlier methods involving DIP and feature based methods is that, the relationship between the features and the output is automatically learned rather than using a rule-based alternative as discussed in DIP.

When it comes to defect recognition, textural features are considered to provide the most valuable information from an image. In some cases, defects can also resemble specific shapes and therefore shape-based features can be used. Broadly, the features used to recognize defects can be divided into the following categories:

(i) Statistical features

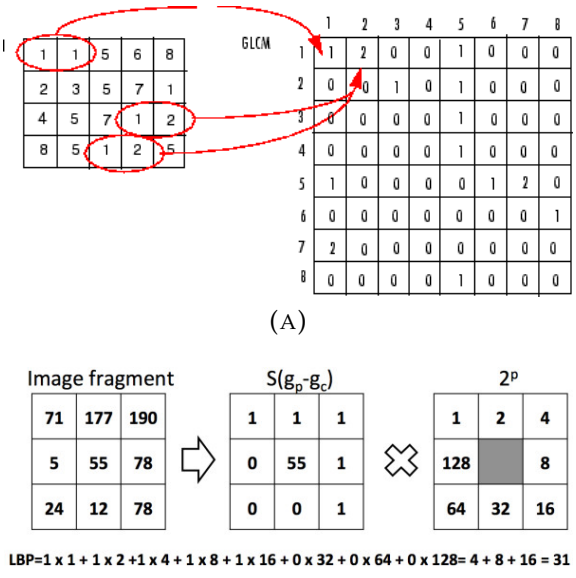
(ii) Structural or geometric features

(iii) Frequency domain features

Feature extraction techniques belonging to the above categories are discussed as follows.

2.2.1 Statistical Features

It is very common to use statistical properties such as mean intensity, standard deviation, intensity range, image moments and histogram values as features from images. An important tool in extraction of textural features is Gray Level Co-occurrence Matrix (GLCM) used to determine the spatial relationship among pixels. In other words, it is useful in calculating how often pixels of certain gray-levels appear closer to each other. From the GLCM, several features such as entropy, contrast, energy, dissimilarity and homogeneity can be extracted [18]. The co-occurrence matrix calculated for an image is unique for every distance value D and angle θ. Therefore, for different combinations of D and θ, different sets of meaningful features (i.e energy, contrast, homogeneity etc.) are obtained. This is applied by Lin et al. [19] to extract six features from the GLCM for different distances and angles. These features are used as input to an Artificial Neural Network (ANN) to detect different types of fabric defects. Mery et al. [20] proposed a method to classify the quality of corn tortillas into five classes using several statistical (GLCM) and geometric features as input to a support vector classifier. Figure 2.3 (a) illustrates how GLCM is calculated with an example.



(A)

(B)

FIGURE 2.3: Statistical Texture Extraction Methods: (a) GLCM [18], (b) LBP [21]

Local Binary Pattern (LBP) is another well-established feature extraction technique used for texture recognition. In LBP, for each pixel, a score, ranging from 0 to 255, is computed based on a comparison with each neighboring pixel. The distribution of these scores gives useful information about the texture found in the image (see 2.3 (b)). The merits of LBP are that it is rotation and illumination invariant. It is often used in combination with GLCM and other statistical and geometric features [22]. Applications of LBP texture features in quality inspection are found in [23] and [24].

Apart from the methods mentioned above, another statistical technique used for feature extraction for image classification is Principal Component Analysis (PCA). It is a popular dimensionality reduction technique in statistics and machine learning, but it is also commonly used in image recognition problems, especially in biometrics and face recognition. Bissi et al. [25] proposed a defect detection technique with PCA along with frequency domain filters and achieved an accuracy of 98.8% with a false detection rate of 0.37%. Fahimipirehgalin et al. [10] proposed an automatic leakage recognition system for chemical plants using thermographic images and videos. After background removal by subtraction of subsequent frames, PCA is applied on the frames to extract features. The classification of leakage or non-leakage is done using K-Nearest Neighbors algorithm to achieve a final accuracy of 90.9%.

2.2.2 Geometric Features

It was seen that statistical features predominantly provide information about textures and color patterns present in the image. Defect recognition applications often also require identification of specific shapes and geometrical patterns within an image. In some cases, the features also describe the geometric properties of certain shapes known to be present in the image. One such example is the task of ellipse detection, as many objects in the real world can be represented by it. Zhang et al. [26] proposed a classification system for multiple varieties of fruits. Their method consists of a segmentation stage where the object of interest is extracted from its background using morphological image operations. Then an ellipse detection step is carried out to extract features such as eccentricity, major axis length, minor axis length, perimeter and area. These features, along with other textural features (GLCM, LBP) are used to construct a feature space. Next, the feature space is subjected to PCA for dimensionality reduction and finally, the multi-class classification is performed using Support Vector Machine (SVM). Other works using ellipse features are found

in [20] and [27].

In [11], Razmjooy et al. use geometric features for the sorting of potatoes based on size and classify pixels as defective and healthy. For the classification, ANN and SVM are used with the extracted features. Before the defect inspection stage, a sequence of morphological operations such as opening and closing are used to subtract the background from the images. In [28], the authors have proposed a system for crack detection in concrete structures. First, a median subtraction is done on the input grayscale image to remove any noise present. Next, a Gaussian Low Pass Filter (LPF) is applied and the image is converted to binary by a thresholding operation. Finally, morphological operations closing and labeling are used to bring out the cracks in the image, if there are any present. The final binary image is sorted based on the number of white pixels found in each column and the first n columns are fed as input to an ANN to detect the presence of cracks.

2.2.3 Frequency Domain Features

One of the most important techniques in image processing is linear image filtering. Filtering can be used to perform a wide variety of operations such as edge detection, blurring, sharpening and so on. The common practice is to convert the image from spatial domain to frequency domain using Discrete Fourier Transform (DFT) and apply filters in the frequency domain. After filtering, Inverse Fourier Transform is applied to convert the image back to spatial domain. Transforming from spatial domain to frequency domain offers a natural way of removing noise from the image [29]. In [30], the authors perform Fourier transformation on images of machine parts to extract features such as peak frequency, central power spectrum and average power spectrum. These features are used as input to an ANN to detect surface defects. It has been observed by Nasira et al. [31] that the presence of defects in fabrics causes significant changes in the resulting Fourier spectrum and by observing the changes over different directions (horizontal and vertical), different features can be extracted and used for classification.

Instead of applying Fourier transform to entire images, the same can be done in a windowed fashion by using the convolution operation. Convolution enables extracting features at a local level, whereas DFT is a global filter. This technique is also the key principle behind CNNs. There are several known filters using different kernels meant for different purposes. A popular filter used widely for texture extraction is the Gabor filter. Several researchers have implemented the Gabor filter as a feature

extractor for both classification and semantic segmentation of defects. In [25] and [32], Gabor filter banks are used as feature extractors for classifying defects in fabrics and printed circuit boards (PCB) respectively. They act as frequency band pass filters and are good at isolating specific textures and patterns. In [33], texture features are extracted from two-dimensional Discrete Wavelet Transform (DWT) response of images of cooling radiator consisting of defects. Goyal et al. [34] used SVM classifier for detection of bearing defects in industrial rotating machinery. They used DWT to extract features from the original image, followed by selecting the strongest features by using a distance metric (Mahalanobis distance).

2.2.4 Other features

There are other feature extraction techniques that are very popular in the general context of image classification and recognition, but the three categories above were chosen as they are the most relevant in the area of defect recognition and industrial quality inspection. Some examples of popular image features are Histogram of Oriented Gradients (HOG), Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF) and Features from Accelerated Segment Test (FAST). Hossain et al. [35] used thermal cameras mounted on Unmanned Aerial Vehicles (UAV) to obtain images of underground pipelines to detect fluid leakage. The authors experiment with eight machine learning algorithms including decision trees and Random Forest with two feature descriptors - SIFT and dense SURF. The results are then compared a CNN based approach. A summary of all the feature extraction methods discussed above is shown in Table 2.1 along with the corresponding references.

TABLE 2.1: Summary of Feature Extraction Methods

Category	Method	References
Statistical	Gray Level Co-occurrence Matrix	[20], [19], [22]
	Local Binary Pattern	[22], [23], [24]
	Principal Component Analysis	[26], [25]
Geometric	Ellipse Detection	[26], [20], [27]
	Mathematical Morphology	[11], [28]
Frequency Domain	Fourier Transform	[30], [31]
	Gabor Filter	[25], [32]
	Discrete Wavelet Transform	[33]
Other	Histogram of Oriented Gradients	[36]
	Shape Invariant Feature Trasnform	[37]

2.2.5 Classification Algorithms

In the above sections, several techniques to extract useful information from digital images were discussed. The features are used to represent the high-dimensional images in a much smaller dimension. The next step in the process is classification. Classification is the supervised task of assigning a label to a given input vector based on the learned information from the training data.

In the above examples, several classification algorithms were mentioned namely: Support Vector Machines (SVM), Random Forests (RF), Artificial Neural Networks (ANN), K-Nearest Neighbors (KNN) and so on. Although one is not limited by the algorithms mentioned above for the classification, it is worth mentioning that they are found to be the most successful by other researchers in the domain. Table 2.2 highlights the references for the different classification algorithms.

TABLE 2.2: Commonly Used Supervised Learning Models for Defect Classification

Model	References
Support Vector Machine (SVM)	[26], [11], [36]
Random Forest	[36]
Artificial Neural Network (ANN)	[19], [11], [28], [30]
K-Nearest Neighbors (KNN)	[36]

2.3 Deep Learning Based Methods

The final and most recently developed group of methods used for image classification are Deep Learning (DL) based methods. It is worth highlighting the difference between the general definition of DL and its use in computer vision related literature. In general, deep learning is the branch of machine learning that makes use of neural networks with multiple hidden neuron layers used to represent complex relationships between input and output vectors [38]. In most image processing and computer vision applications, the term deep learning is ambiguously used to denote methods that use neural networks for feature extraction rather than hand-crafted features. In this report, the latter definition will be used.

The popularity of Convolutional Neural Networks (CNNs) was touched upon in Chapter 1. In recent literature, applications of CNN in several different visual tasks such as image classification, object identification and semantic segmentation are found in abundance. Recent improvements in Graphical Processing Units (GPU)

and their ease of implementation have paved the way for widespread acceptance of CNNs by a large number of researchers and practitioners. They have also been accepted as the state-of-the-art in defect recognition and quality inspection. This is majorly due to the fact that CNNs are able to extract robust spatial features automatically from training images thereby eliminating the need for domain expertise. In the last few years, the use of deep learning and CNNs on thermography based applications has seen a tremendous increase in terms of volume of articles published. Figure 2.4 shows the results from a document search in the Elsevier database using the following query: ("infrared" OR "thermal imaging") AND ("convolutional neural networks" OR "deep learning"). The increasing trend can be observed from the figure. A few such applications that were found to be relevant are discussed below.

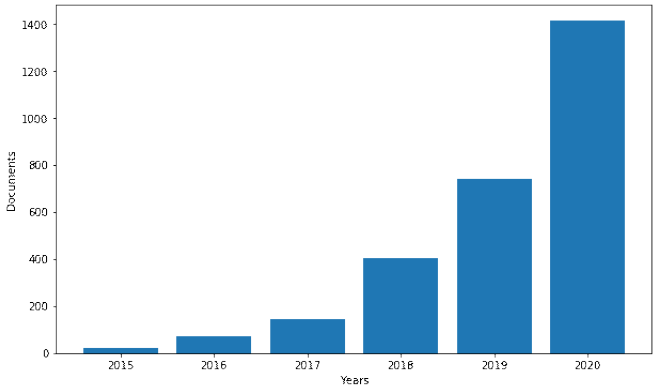


FIGURE 2.4: Growing trend in deep learning used in thermography shown through Elsevier database search results

Li et al. [39] proposed a deep learning based approach for fault diagnosis of rotating machinery using thermal imaging. The use of thermography is further motivated in this work as the defect recognition is done for largely different temperature ranges. The authors compare the performance of four DL architectures namely - Convolutional Neural Networks (CNN), Deep Belief Network (DBN), Deep Neural Network (DNN) and Stacked Auto-Encoder (SAE) on a 10-class dataset. The above architectures are used to extract features specific to each class and finally a softmax classifier is used for all the methods. The CNN approach (shown in figure 2.5) achieved superior performance with an accuracy of 99.8%. A system to classify six radiator defects including coolant leakage using IRT and CNN is proposed by Nasiri et al. [40]. The well known VGG-16 architecture is used in this study with five convolution blocks, each consisting of a number of convolution layers and maximum pooling layers. The architecture is considerably deeper than those discussed previously and consists of millions of parameters. The final classification block includes batch normalization and dropout layers to tackle overfitting. These techniques are

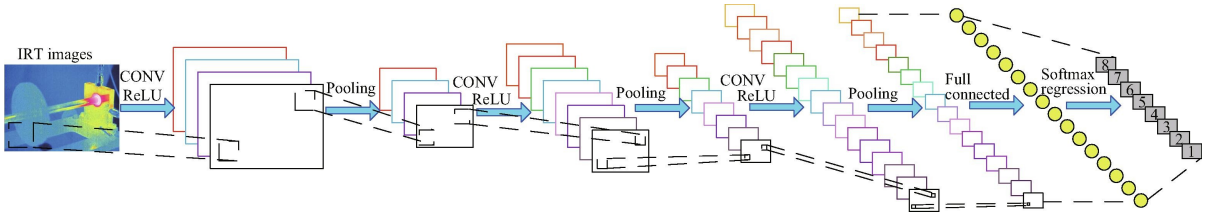


FIGURE 2.5: CNN Architecture used in [39]

commonly used in applications prone to overfitting due to shortage of training data. Data augmentation is also used in the pre-processing stage as a strategy to overcome the shortage of data. Finally, an accuracy of 96.67% was achieved in the testing set. A detailed review of several such applications of deep learning used for defect detection is shown in [41].

An important factor to be considered while choosing a suitable method is the ability to acquire a large amount of data. This is particularly more relevant for deep learning techniques where the performance is highly dependent on the amount of training data available. Shortage of training data leads to overfitting of the model to the training data. Moreover, it is not always possible to collect a huge amount of data and sometimes there may be physical factors making it impractical to acquire large amounts of data. Regularization is the process of providing the model with a higher regularization capability. Some common regularization techniques are data augmentation, weight regularization using L1 or L2 norms and adding dropout layers in neural networks.

2.4 Hybrid Methods

In recent literature, researchers have explored hybrid approaches to image classification. The hybrid methods attempt fusing together components from two or more separate techniques. Such fusion techniques have been used extensively in language processing applications where the output of various models such as auto-encoders and long short-term memory (LSTM) networks are combined to produce a concatenated vector that is finally used for classification. This strategy has also been used by researchers in the image processing domain for applications such as scene recognition and context understanding [42]. A few examples of such hybrid applications are outlined below.

Almubarak et. al [43] developed an image classification system to detect cervical cancer from histology images. Handcrafted feature extraction is performed using

two color spaces - RGB and LAB. These features include gray texture features and geometric features defined based on the characteristics of the cell. Next, they also extract spatial features from a trained CNN and concatenate the features from both the sources to perform classification using five different algorithms including SVM, logistic regression and random forest. Lahmiri [44] proposed a hemorrhage classification system from retina images for diagnosis of diabetic retinopathy. First, a deep CNN is trained and the activations from the third convolution layer are extracted for all the images in the database. Next, a Student's t-test is applied in order to select the 10 best features. The feature vector of length 10 is used as input to various supervised classifiers and the highest performance was achieved by a kernel-SVM model. Moradi et al. [45] use features from various sources (HOG, LBP, Haar features etc.) combined with activated CNN features to perform classification on CT scan images. The classifier is trained separately on each feature group and a weighted average is taken to assign the final label.

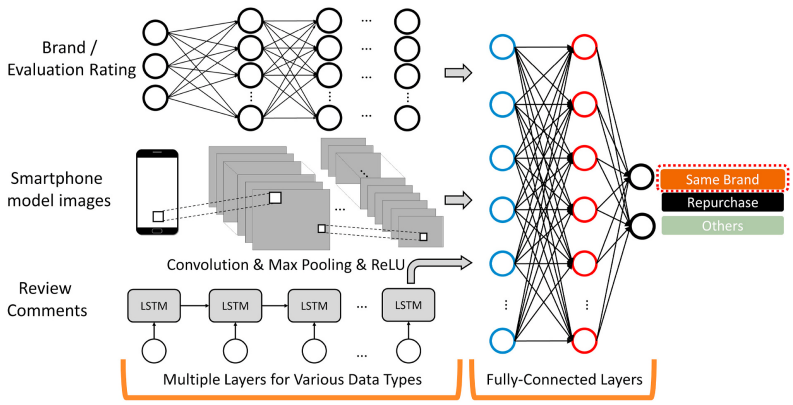


FIGURE 2.6: Hybrid deep learning model to predict smartphone repurchase behaviour from [46]

An example of a hybrid method of extracting features from multiple sources is shown in [46]. For the problem of predicting customer repurchase behaviour for a smartphone brand, the authors propose a hybrid learning model. Three sources of data are used namely - numerical customer ratings, smartphone images and review comments. A multi-layer perceptron model is used to compute features from the numerical ratings. Image features are extracted from a CNN model and LSTM model is used to process the full-text user reviews. Finally, the three feature groups are combined to produce the output using a fully-connected network.

2.5 Literature Summary

The most important findings from the literature survey with respect to the main objective of this research are listed below:

i The most common techniques used for vision-based defect recognition were categorized into three major sub-categories namely, DIP based methods, manual feature-based methods and deep learning methods. More recent methods that combine multiple models or feature sources were identified for the problem of image classification.

ii Some of the relevant works from each category were discussed in chronological order. First, a few traditional DIP-based defect recognition applications were shown. Different feature extraction techniques were discussed from image based quality inspection applications.

iii Applications based on convolutional neural networks, the current state-of-the-art in image classification, were discussed.

iv Finally, a few examples of hybrid methods from other areas of image classification were discussed.

Chapter 3

Background

In this chapter, a theoretical background is provided on the methods used in this study. First, a brief explanation of thermography as a data acquisition technique is given. Next, the algorithms used for feature extraction, feature selection and classification are explained.

3.1 Infrared Thermography

All objects at a temperature of above absolute zero are known to emit infrared radiation. This phenomenon is exploited by infrared thermal (IRT) cameras to capture the temperature of objects. Infrared thermography is a popular inspection technique used in various industrial applications. The biggest advantage of thermography is that it is a non-contact inspection technique and may be applied in situations where the product under inspection does not have to be in contact with the sensor. The amount of radiation emitted by an object is directly proportional to its temperature and hence, defects and other anomalies show up as temperature differences which can easily be identified using infrared thermography. There are mainly two categories of thermography - passive and active. Active thermography uses an external heat source in the inspection process for thermal excitation whereas passive thermography does not use any heat source. Figure 3.1 shows a representation of an active thermography setup.

An important factor to be considered in thermographic measurements is the emissivity of a material which is its ability to emit the incoming infrared radiation. It is a value ranging between 0 and 1 where 0 is the emissivity of a mirror surface that reflects all the radiation incident upon it and 1 refers to a black body. The thermal measuring device can be calibrated using the known emissivity of the material to obtain the exact temperature of an object. In the context of this study, active infrared thermography is used as the method of data acquisition to capture the required information to detect fluid leakage inside the containers. We primarily assume that if

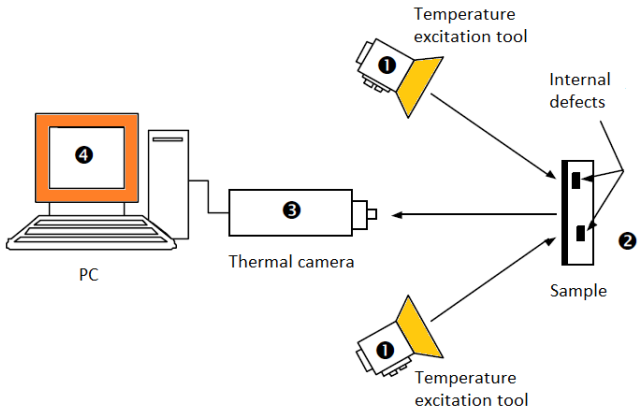


FIGURE 3.1: Components of active thermography [47]

leakage occurs inside a container, it eventually reaches the bottom surface and by heating the material of the container using an external heater, a thermal difference can be created between the detergent fluid and the material which can be captured by a thermal camera.

3.2 Feature Descriptors

In the following sections, the feature descriptors used in this work are explained. A feature descriptor is an algorithm that takes an image as input and transforms it into a feature vector. We use two feature descriptors to obtain handcrafted features that can help distinguish between leakage and non-leakage from IRT images. They are explained below.

3.2.1 Gray-Level Co-occurence Matrix (GLCM) Features

Texture extraction for image classification was first introduced by Haralick et. al [48]. These features may be calculated from the gray-level co-occurence (GLCM) matrix of an image. In Chapter 2, the concept of GLCM was briefly introduced. The co-occurence matrix provides information on how frequently pairs of pixels occur together in a specific direction θ and distance D. The GLCM is a square matrix of dimensions $N_g \times N_g$, where N_g is the highest gray value found in the image. The co-occurence matrix may be created by scanning through the input image and counting the number of times two gray values appear together in a given direction and offset distance. In the image domain, the possible values for θ are $0°$, $45°$, $90°$ and $135°$. For D, the commonly chosen value is 1 where only the immediate neighboring pixels are considered. For values greater than 1, the co-occurrence is calculated with an offset

corresponding to the value. Figure 3.2 illustrates the different values of θ on a 3×3 image fragment.

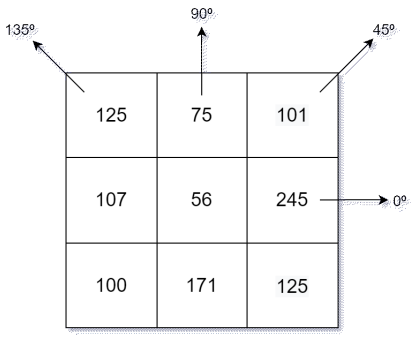


FIGURE 3.2: Possible values for angle θ in GLCM calculation

An example calculation of the GLCM for a 2-bit image is shown in figure 3.3. For every pixel in the image, only its immediate neighboring pixels to the right are considered since D=1 and θ=0°.

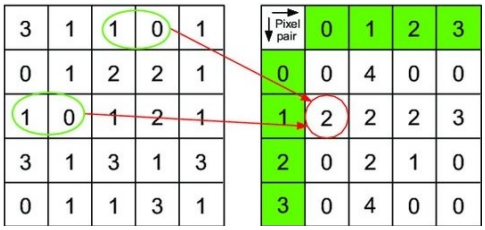


FIGURE 3.3: GLCM calculation with θ=0° and D=1 ([49])

In the original paper, the authors propose 14 texture features from the GLCM matrix, commonly known as *Haralick* features. In this work, six of these features are chosen. Consider p_{ij} as the element in the co-occurence matrix with row-index i and column index j. The expressions used to compute the six features are defined in table:

1. Angular Second Moment (ASM) = $\sum_i \sum_j p_{ij}^2$
2. Energy = $\sqrt{ASM}$
3. Contrast = $\sum_i \sum_j (i-j)^2 p_{ij}$
4. Homogeneity = $\sum_i \sum_j \frac{1}{1+(i-j)^2} p_{ij}$
5. Dissimilarity = $\sum_i \sum_j |i-j|\, p_{ij}$
6. Correlation = $\sum_i \sum_j \frac{(ij)p_{ij}-\mu_x\mu_y}{\sigma_x\sigma_y}$,
   where μ_x, μ_y, σ_x and σ_x are the means and standard deviations along the rows and columns respectively.

3.2.2 Local Binary Pattern

Local Binary Pattern (LBP) is another popular feature extraction tool introduced by Ojala et. al [50] used for various applications such as face recognition, landscape detection and generally, texture classification. The LBP algorithm is used to describe the local spatial patterns in the neighbourhood of each pixel. The original image is transformed into the LBP response by thresholding the neighboring pixels based on the center pixel. The following operation is applied to obtain the output for each pixel.

$$LBP = \sum_{n=0}^{P-1} s(g_n - g_c)2^n, \tag{3.1}$$

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{3.2}$$

where g_n is the gray value of neighboring pixel, g_c is the gray value of the center pixel and P is the number of neighboring pixels in consideration. The coordinates of each of the neighboring pixels g_n are given by $(-Rsin(2\pi n/P), Rcos(2\pi n/P))$, where R describes the radius of the circular neighborhood. In figure 3.4, a few examples for the LBP neighbourhood with different values of P and R are shown.

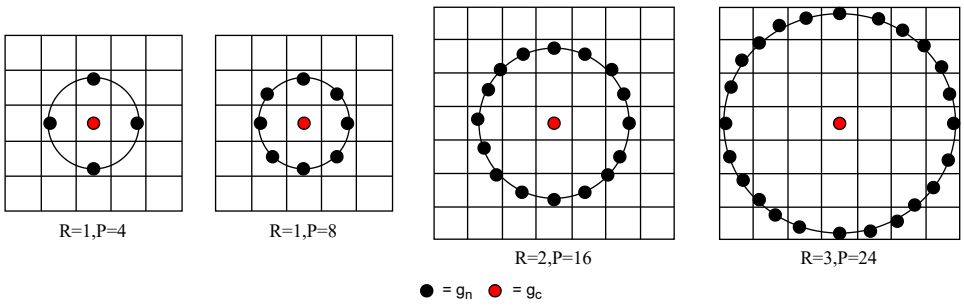


FIGURE 3.4: Circular neighbourhoods for different values of P and R

An example calculation of the LBP response was shown in figure 2.3. The range of values in the LBP response for p=8 is 0 to 255 as 2^8 unique patterns are possible. The feature vector is constructed by taking the values from the histogram of the LBP response. In this work, we use uniform LBP where only the uniform rotation-invariant patterns are considered. Ojala et. al extract certain patterns from the LBP responses that fully describe all the possible patterns in a much smaller dimension (p+2 histogram bins). A pattern is called uniform when the number of 1-0 or 0-1 transitions in the LBP response is at most two. Some common patterns identified in the original paper are shown in figure 3.5.

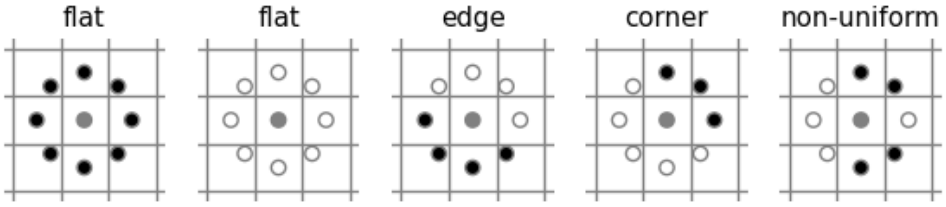


FIGURE 3.5: Rotation-invariant local binary patterns [51]

3.3 Dimensionality Reduction

An optional step after feature extraction is feature selection or dimensionality reduction. Feature selection is the process of eliminating redundant features from the original feature set. Principal component analysis (PCA), a technique famously used for feature selection is explained below.

3.3.1 Principal Component Analysis

PCA is an unsupervised statistical technique that can be used to remove redundancy in the data while retaining only useful information in a reduced dimensional space. PCA uses the covariance matrix of the dataset to find the features with most variance. The steps involved in PCA are summarized as follows [52]:

i. Consider an $N \times d$ dimensional dataset X. First, the row-wise mean ($\bar{x}$) of X is calculated as:

$$\bar{x_j} = \frac{1}{N} \sum_{i=1}^{N} X_{ij} \tag{3.3}$$

ii. Then we create the mean matrix,

$$\bar{X} = \begin{bmatrix} 1 \\ . \\ . \\ . \\ 1 \end{bmatrix} \bar{x} \tag{3.4}$$

The mean-subtracted data is given by,

$$B = X - \bar{X} \tag{3.5}$$

iii. The covariance-matrix is,

$$C = \frac{1}{N-1} B^* B \tag{3.6}$$

iv. The eigenvectors and eigenvalues of the covariance matrix can be calculated using Singular Value Decomposition (SVD). Ordering the eigenvectors based on the largest eigenvalues results in the principal components. We select k principal components resulting in a $N \times k$ dimensional dataset.

Intuitively, PCA seeks to find a set of new axes, called principal components, that capture the maximum variance of the data points. Therefore, by selecting only the first $k < d$ principal components, most of the variability in the dataset can be captured. Thus, PCA can be used to significantly reduce the dimensionality of the feature vector and remove redundancy.

3.4 Supervised Classifier

As discussed in Chapter 2, the next step after obtaining the feature vector is to classify a given input as leakage or non-leakage and this may be done using a supervised binary classifier.

3.4.1 Linear Support Vector Machine

Support Vector Machines (SVMs) were introduced by Cortes et al. [53] for two-class classification problems. Here, we stick to the linear variant of the algorithm that assumes that the two classes are linearly separable. In other words, we assume that it is possible to draw a n-1 dimensional hyperplane that separates the two classes in $\mathbb{R}^n$ space (refer figure 3.6). The equation of the hyperplane is given by $w.x + b = 0$.

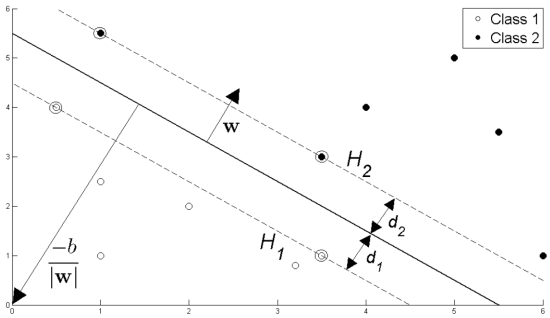


FIGURE 3.6: Hyperplane separating two classes in two-dimensional space

Support vectors are the points from each class that are closest to the hyperplane and their aim is to orient the hyperplane such that the distance between the closest

members from both classes is maximized. Suppose we have two classes y=1 and y=-1, the following can be deduced from figure 3.6:

$$x_i.w + b \geq +1 \text{ for } y_i = +1 \tag{3.7}$$

$$x_i.w + b \leq -1 \text{ for } y_i = -1 \tag{3.8}$$

Combining the above equations,

$$y_i(x_i.w + b) - 1 \geq 0 \; \forall_i \tag{3.9}$$

Referring to figure 3.6, the hyperplane is drawn exactly in between the two hyperplanes H_1 and H_2 corresponding to the support vectors of each class. The distance to the center hyperplane from H_1 and H_2, known as the margin of the SVM is equal to $d_1 = d_2$, which is the quantity to be maximized to obtain maximum separability between the classes. The equations of the planes H_1 and H_2 are given by:

$$x_i.w + b = +1 \text{ for } H_1 \tag{3.10}$$

$$x_i.w + b = -1 \text{ for } H_2 \tag{3.11}$$

If the margin is equal to $\frac{1}{|w|}$, then the problem now becomes to maximize this value or minimize $|w|$ constrained to equation 3.9. The optimization problem is formulated as follows:

$$\min \frac{1}{2} \left| w^2 \right| \text{ s.t. } y_i(x_i.w + b) - 1 \geq 0 \; \forall_i \tag{3.12}$$

The optimal values of w and b are found from the training data and for a new test example x', the label is assigned as $sgn(w.x' + b)$.

3.4.2 Ensemble Methods

Among other popular classification techniques, ensemble methods such as random forests and Adaboost are well known for image classification applications. These methods are based on decision trees, a supervised learning algorithm that can be used for both regression and classification. One of the biggest disadvantages of decision trees is that they are prone to overfitting and the ensemble models are mostly aimed at providing a better performance by combining many such weak learners.

Random forest implements a technique called bagging to reduce overfitting. The term bagging originates from two other techniques - *bootstrapping* and *aggregating*. Bootstrapping is the process of taking random sub-samples from the training data set with replacement. Multiple estimators are built from each of these bootstrapped samples and the predictions from all the estimators are aggregated. Usually, a majority vote is used to obtain the final prediction.

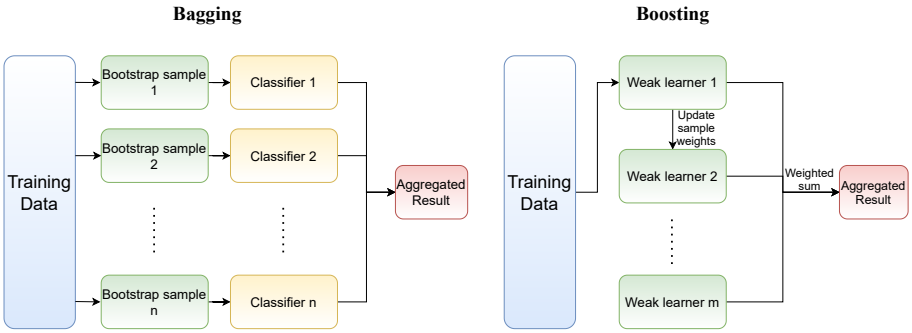


FIGURE 3.7: Process of ensemble methods - bagging and boosting

Adaboost or adaptive-boosting is another ensemble method that reduces overfitting and also improves the prediction performance significantly compared to the individual weak learners. In Adaboost, the emphasis is on reducing the prediction error by assigning sample weights to each training observation. Single-depth decision trees known as *stumps* are usually used as the individual learners. But the boosting strategy can be used on any base estimator. Consider N training samples belonging to two classes $y = +1$ and $y = -1$. Let the number of classifiers be M and the response of each classifier be G_m. Algorithm 1 explains the working of Adaboost classifier [54].

Algorithm 1 Adaboost Classification

1. Initialize sample weights $w_i = 1/N$ for i=1,2,3,...,N.
2. For all m = 1 to M:
   a) Fit an individual classifier $G_m(x)$ to the training samples using sample weights w_i.
   b) Calculate $err_m = \frac{\sum_{i=1}^{n} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{n} w_i}$
   c) Calculate the classifier importance $\alpha_m = log((1 - err_m)/err_m)$
   d) Update sample weights $w_i \leftarrow w_i.exp(\alpha_m.I(y_i \neq G_m(x_i)))$
3. Final prediction is given by $G(x) = sgn[\sum_{m=1}^{m} \alpha_m G_m(x)]$

3.5 Convolutional Neural Networks

In the previous chapters, we discussed the significance of CNNs in image classification. In this section, a brief introduction to CNNs used for the problem of image classification is given. The different aspects of building a CNN architecture are discussed from an implementation point of view.

CNNs are specialized in image and signal processing applications because they are based on the convolution operation used often in signal processing. The basic building blocks of a CNN are: convolution layers, pooling layers and fully connected layers. In the image domain, convolution is used for filtering over an image using *kernels*. Therefore, the convolution layers in a CNN consist of kernels of specific dimension that are convolved over a given image to extract different features. Figure 3.8 illustrates convolution performed on a two-dimensional image.

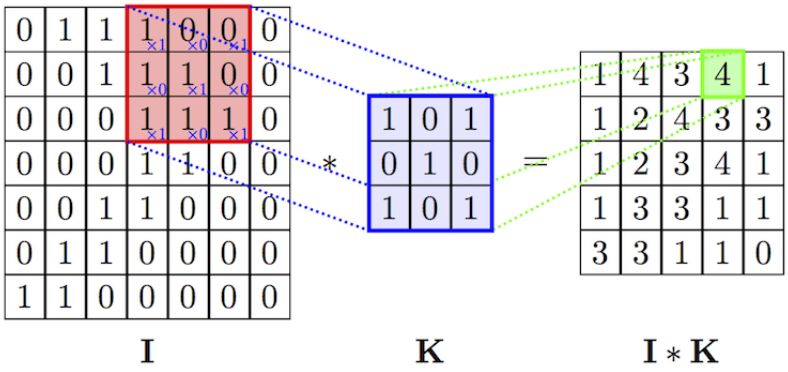


FIGURE 3.8: Example of 2D Image Convolution [55]

Convolution by different kernels are useful in extracting different features from an image. Since the values of the kernels are trainable parameters in a CNN, the most suitable kernels are automatically learned during the training process. The biggest advantage of the convolution operation is that the patterns extracted are rotation and translation invariant. Convolution layers are also known for learning hierarchical features, meaning the first few layers may extract low level patterns such as edges and lines and the deeper layers bring out the larger patterns like shapes and objects. Therefore, the filters learned from training data are extremely useful in extracting features that are unknown and are difficult to extract using known filters such as edge detectors or Gaussian filters.

An addition to the convolution layers, a non-linear activation function is used to represent the output of each layer in a specific range. It is also important that

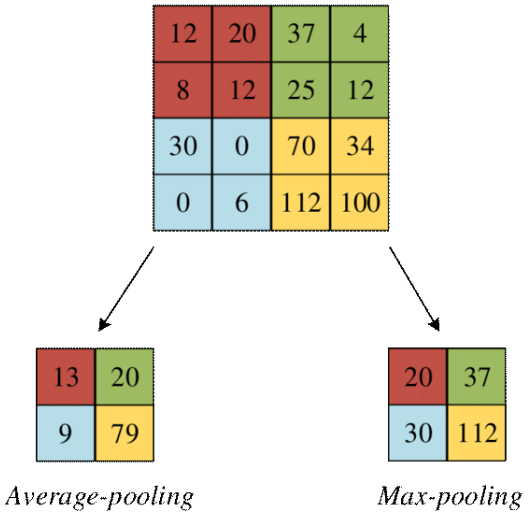


FIGURE 3.9: Max Pooling and Average Pooling [55]

the function is differentiable so that it can be optimized using back-propagation. Some of the commonly used activation functions in neural networks are sigmoid function, tanh function, Rectified Linear Unit (ReLU), softmax function and so on (figure 3.10). In most modern CNN architectures, it has become the norm to use ReLU as the activation function for convolution layers.

Lastly, for image classification, fully connected layers are required to classify the feature maps from the convolution layers into binary or multi-class output. These are similar to hidden layers in an ANN where all the neurons from one layer are connected to all the neurons of a subsequent layer. In a CNN, after the final convolution layer, the features are flattened and given as input to fully connected layers before finally arriving at the output. The activation function for the output layer is generally different from that of the convolution layers. Linear, sigmoid and softmax functions may be used to obtain a value that may be used to determine the output. For example, softmax function provides values that sum up to 1 and therefore, the values may be interpreted as probabilities of the input belonging to each class.

The fundamental parts of a CNN architecture were discussed above. In order to train a network using training data, a loss function and an optimizer must be chosen. The loss function quantifies the classification performance of the network on training data. Intuitively, it assigns a large value of loss for wrong classifications and zero or small values for correct classifications. Therefore, an optimizer is used to find a set of weights and biases for the network that attempts to minimize the loss. Some well-known loss functions used for classification are hinge loss, binary

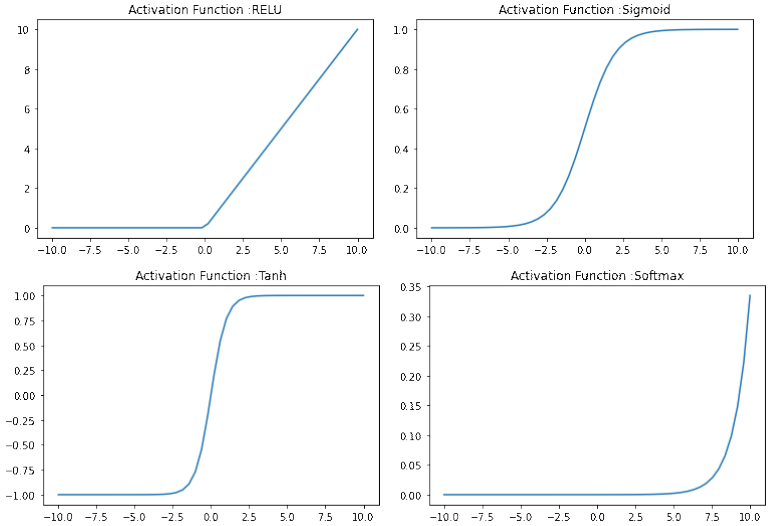


FIGURE 3.10: Non-Linear Activation Functions

and categorical cross-entropy. Most optimizers use gradient descent to find the minima of a given loss function. Stochastic Gradient Descent (SGD), Adaptive Moment (Adam) estimation and Root Mean Square Propagation (RMSProp) are examples of established optimizers for CNNs.

Chapter 4

Methodology

In this chapter, the methodology used in this study to solve the problems formulated in Chapter 1 is described. As described earlier, the main objective of the research is to build a classification system to identify leakage from IRT images of detergent containers. We divide the methods into three main sub-categories, namely handcrafted feature-based approach, deep learning approach and a hybrid approach combining the first two. The classification performances of each of these appraoches are compared in the latter parts of the report.

4.1 General Overview

The objective of any image processing system is to obtain meaningful insights from digital images by manipulating the images or extracting information from the pixels. Therefore, the basic steps involved in most image classification applications are similar. For this work, the following steps are considered to be most relevant:

1. Image acquisition
2. Pre-processing
3. Feature extraction
4. Dimensionality reduction (optional)
5. Classification

For the three approaches mentioned above, the image acquisition and pre-processing steps remain the same and they only differ majorly in the methods used for feature extraction and classification. Therefore, we first discuss the primary steps and then explain the approaches based on the methods used for feature extraction and classification. Figure 4.1 depicts the general process flow of the classification system.

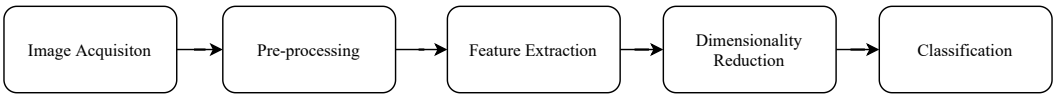


FIGURE 4.1: General process flow of image classification

4.2 Primary Steps

In this section, the initial steps of acquiring the image data and preparing it for the further steps are discussed.

4.2.1 Image Acquisition

Image acquisition is the process of capturing the images from the sensor and standardizing the image format and representation. In this case, the sensor is an IRT camera of resolution 640×480. Various pixel representations are possible for the IRT images. They may be represented directly as temperature in $^{\circ}C$, as grayscale values ranging from 0 to 255 or as false 3-channel RGB values. The temperature values may be converted grayscale as follows:

$$I_{g_{ij}} = \frac{I_{ij}}{I_{max}} * 255$$

where, $I_{g_{ij}}$, I_{ij} are grayscale and original thermal images with row and column index i and j respectively, I_{max} is the highest temperature value found in the image. In the following sections, for some methods, we use temperature values ($^{\circ}C$) and for others, grayscale values are used. This is explicitly mentioned wherever applicable.

4.2.2 Pre-processing

Pre-processing, sometimes referred to as image enhancement is the process of adjusting the images so that they are more suitable for the further analysis steps [56]. Pre-processing generally involves extracting regions of interest (ROIs), contrast enhancement and so on. The following pre-processing steps are used in this work:

Cropping

The very first pre-processing step that we perform is cropping. Due to the physical position of the camera, the image consists of some unnecessary background and Since the coordinates of the ROI are consistent in all the images, we perform cropping by index. The dimension of the image is reduced from 640×480 to 580×250 after cropping.

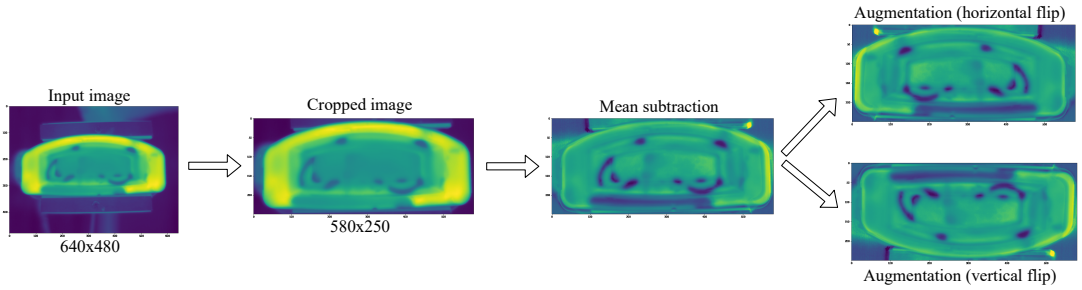


FIGURE 4.2: Various steps involved in image pre-processing

Mean Subtraction

The next step is to subtract from each pixel of every image its respective mean over the training dataset. This is done in order to center the distribution of each pixel around zero. This method is referred to as mean subtraction and is a popular pre-processing tool often used in CNN literature ([12], [57]). Suppose the training dataset consists of N images with height h and width w, then the elements of the mean image $\bar{I}$ are computed as follows:

$$\bar{I_{ij}} = \sum_{i=1}^{h} \sum_{j=1}^{w} \sum_{n=1}^{N} I_{ij_n} \tag{4.1}$$

where I_{ij_n} is the element with row-index i and column-index j from the n^{th} image of the training dataset. The mean-subtracted image,

$$I' = I - \bar{I} \tag{4.2}$$

for all images from the database. Usually, normalization is performed to fit the range of pixel values in a particular range such as [0,1] or [-1,1]. Since the subtracted images result in negative pixel values, we shift the pixels to [-1,1] range.

Data Augmentation

In this work, the data acquisition process is limited by the time taken to produce a single image. Therefore, it is practically infeasible to collect a large database of images. One way to address this limitation is to enlarge the training dataset by creating artificial copies of the existing images. This is done using simple image transforms such as rotation, flipping, scaling and translations. We perform two simple operations - flipping along horizontal axis and flipping along vertical axis. Both the transformations preserve the original dimensions of the image. Therefore, after augmentation the size of the training set is increased by a factor of 3. Figure 4.2 illustrates the different steps involved in pre-processing explained above.

4.3 Handcrafted feature-based approach

The first of the three proposed alternatives is the handcrafted feature-based approach. This is also considered the baseline method of this research. The key idea behind this approach is manual feature extraction and using the features as input for supervised learning. In order to extract numerical features from the grayscale images, we use two texture based feature descriptors - GLCM and LBP. The method of computing features from these feature descriptors was already explained in detail in Chapter 3. Apart from them, some basic intensity features are also used. Table 4.1 shows the different feature groups and the exact description of the features generated from each.

TABLE 4.1: Extraction of handcrafted features

	Feature Group	Description
1	GLCM features	Co-occurrence matrix is generated for all combinations of D=[1,3,5] and θ = [0°, 45°, 90° ,135°]. From each of the 12 resulting co-occurrence matrices, we obtain the following features: Contrast, homogeneity, dissimilarity, correlation, ASM and energy. Total features = 72
2	LBP features	Uniform LBP histogram bins for the following parameters: i. R=1, P=8 (no. of bins = P+2 = 10) ii. R=2, P=16 (no. of bins = P+2 = 18) Total features = 28
3	Intesity features	Mean intensity and standard deviation intensity

Therefore, the final feature vector is of $102 \times N$ dimension for a dataset of size N. Next we use PCA to obtain the first k principal components ($k < 102$) such that the new feature space is k-dimensional. PCA is used here for dimensionality reduction as well as selecting the features with most variability for the classification.

Finally, the $k \times N$ dimensional vector is used to train a supervised classifier that assigns a binary label to each new input, i.e. leakage or non-leakage. We use two separate classifiers - linear SVM and Adaboost and compare their recognition performance. We saw in Chapter 3 that SVM finds a hyperplane that best separates the two classes linearly in the feature space. This would be a good choice for distinguishing between leakage and non-leakage because we expect that the texture properties of both classes are different from each other. The implementation of SVM is done using the stochastic gradient descent (SGD) optimizer with hinge loss function.

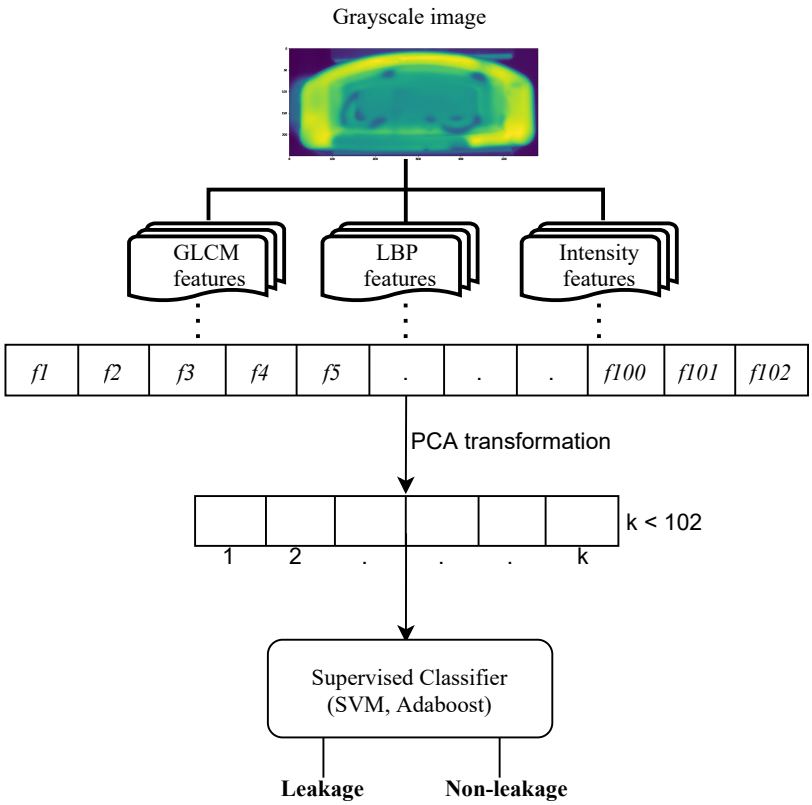


FIGURE 4.3: Overview of proposed handcrafted feature-based approach

On the other hand, the Adaboost model is used as an alternate method for classification because it provides a non-linear separation between the two classes. The boosting strategy also focuses on improving the predictions based on training samples that are often misclassified. We use single-depth decision trees (stumps) as the individual learners. Therefore, the only hyperparameter is the number of estimators M. A suitable value for M may be chosen by varying M and observing the classification error as a function of M. The overall process of the feature-based approach is summarized in figure 4.3.

4.4 Deep learning approach

The second approach mainly focuses on using convolutional neural networks for the given task. As mentioned earlier, CNNs are end-to-end learning models where feature extraction is part of the learning process and does not need to be performed manually. For the given problem of detecting leakage in IRT images, we propose a custom CNN architecture. The various parts of the architecture are described in the following sections.

4.4.1 CNN Architecture

The proposed architecture consists of two parts: feature extraction and classification. The feature extraction part of the CNN consists of convolution and pooling layers, whereas the classification layers consist of densely connected neurons leading to the output. The detailed structure of the CNN is described below.

Convolution Layers

The convolution layers produce feature maps by performing convolution over the input image using kernels of fixed size. Convolutions may be interpreted as scanning over the image for specific patterns defined by the kernels. The response of a convolution depends on the strength of match with the kernel at that particular part of the entire image. For a pixel with coordinates (i, j), the convolution response using a kernel of size $m \times n$ is defined as:

$$conv_{ij} = \sum_{x=1}^{m \times n} w_x v_x \tag{4.3}$$

where w_x is the kernel weight and v_x is the pixel intensity at position x in the $m \times n$ neighbourhood. These kernel weights are learnt during the training process and therefore, in a trained network, the kernels that provide the most distinction between the two classes are chosen.

Rectified Linear Unit

It is common to use a non-linear activation before successive convolutions to fit the response of each layer within a certain range. The activation function provides a non-linear mapping between two consecutive layers. We use ReLU activation for the convolution layers because it is proven to perform well for a large number of CNN applications. The popularity of ReLU activation is due to its computational advantage and robustness against the vanishing gradient problem [58]. The ReLU function is defined as follows:

$$ReLU(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \tag{4.4}$$

Pooling

Another important part of the architecture is pooling or subsampling. Pooling layers are used after convolution layers to reduce the dimensions of the feature maps by

aggregating values. In this architecture, we use maximum pooling and average pooling. The output dimension of the feature maps depends on the pooling kernel size, stride and zero padding. If the original height and width of the image are h and w, the kernel is of size $m \times n$, p is the increase in dimension due to zero padding and s is the stride of the convolution, then the altered height and width of the image are:

$$h' = \left\lfloor \frac{h - m + s + p}{s} \right\rfloor, w' = \left\lfloor \frac{w - n + s + p}{s} \right\rfloor \tag{4.5}$$

where $\lfloor . \rfloor$ denotes floor operation. All the pooling layers used in our architecture are of size 2×2 with a stride of 2 and we do not perform any zero-padding. Therefore, from 4.5, this results in a downsampling of the input array by a factor of two.

The convolution, ReLU and subsampling layers together constitute the feature extraction part of the neural network. After training, the learned weights and biases of the convolution filters help extract the features that offer the highest separation between the two classes from the images. Generally, these filters are designed such that in the early layers, low-level features such as edges and lines are extracted and the deeper layers bring out more complex features.

Fully connected layers

After the feature extraction is performed by successive convolutions and subsampling, fully-connected (FC) or dense layers are used for the final classification. Each neuron in a dense layer is connected to every other neuron from the previous layer, similar to a multi-layer perceptron (MLP). These layers map the flattened feature vector to the final binary classification output through the densely connected neuron activations. The activation of a neuron in a dense layer l is given by:

$$a_j^l = f(\sum_{i=1}^{M_j} W_{ij}^l . a_i^{l-1} + b^l) \tag{4.6}$$

where M_j is the number of input neurons connected to the j^{th} neuron of the l^{th} layer, W^l and b^l are the weight matrix and bias of the l^{th} layer, f is the non-linear activation function. The last layer consists of only one neuron in case of a binary classification and if we use a sigmoid non-linearity, the activation of this neuron corresponds to the probability of the class that an instance belongs to. The sigmoid function is given by:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.7}$$

Dropout

Dropout is a form of regularization that is used commonly to reduce overfitting in large neural networks. In a layer-wise dropout mechanism, the activations of a certain amount of neurons, defined by the dropout rate, are forced to be zero during training. This helps to reduce overfitting because the neurons are forced to fit the output without the activations of all neurons. Therefore, the overall complexity of the network is reduced and better generalization is achieved. In this architecture, we use dropout in some layers with a constant dropout-rate of 0.2. Hence, 20% of the neurons in these layers are randomly disconnected from the previous layer.

Loss function and Optimizer

In order to train the neural network using labeled images, a loss-function and an optimizer are required. The loss function is the metric based on which the weights and biases are updated. Since the output layer is activated by a sigmoid function, the most suitable loss would be a logarithmic function. Therefore, the binary cross-entropy function is chosen for this problem. It is given by:

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i.log(\hat{y}_i) + (1 - y_i)log(1 - \hat{y}_i) \tag{4.8}$$

where N is the number of training samples, y_i and $\hat{y}_i$ are the actual and predicted output of the i^{th} training sample. Various alternatives for the choice of an optimizer are available. The RMSprop optimizer has been used in many similar works and therefore we choose this for the task of minimizing the loss function.

The different components of the proposed CNN architecture to classify leakage and non-leakage from IRT images were discussed above. The detailed structure of the network is described in table 4.2. Figure 4.4 illustrates the architecture using a representative diagram.

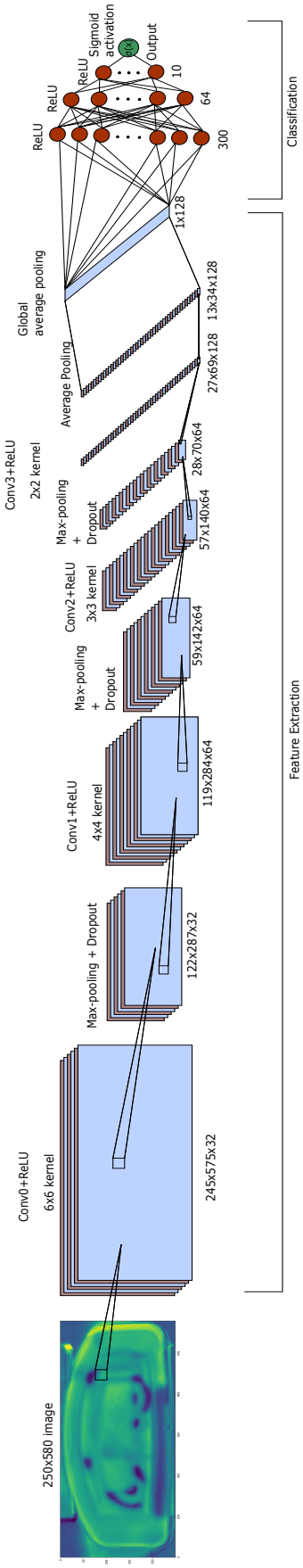


FIGURE 4.4: Proposed CNN architecture for classification of leakage from IRT images

TABLE 4.2: CNN Architecture

Layer Name	Description	Input Dimension	Output Dimension
Input	250×580 image input	-	250×580×1
Conv0 + ReLU	Conv + ReLU layer 1 with 32 6×6 kernels	250×580×1	245×575×32
Max Pooling + Dropout	Pooling layer 1 with 2×2 kernel, stride = 2	245×575×32	122×287×32
Conv1 + ReLU	Conv + ReLU layer 2 with 64 4×4 kernels	122×287×32	119×284×64
Max Pooling + Dropout	Pooling layer 2 with 2×2 kernel, stride = 2	119×284×64	59×142×64
Conv2 + ReLU	Conv + ReLU layer 3 with 64 3×3 kernels	59×142×64	57×140×64
Max Pooling + Dropout	Pooling layer 3 with 2×2 kernel, stride = 2	57×140×64	28×70×64
Conv2 + ReLU	Conv + ReLU layer 4 with 128 2×2 kernels	28×70×64	27×69×128
Average Pooling	Average pooling with 2×2 kernel, stride = 2	27×69×128	13×34×128
Global Average Pooling	Global averaging over all filters	13×34×128	1×128
Dense Layer 1	Dense layer with ReLU activation	1×128	300
Dense Layer 2	Dense layer with ReLU activation	300	64
Dense Layer 3	Dense layer with ReLU activation	64	10
Output Layer	Output layer with sigmoid activation	10	1

4.5 Hybrid approach

The last of the three approaches is a hybrid approach that combines certain aspects of both the approaches discussed in sections 4.3 and 4.4. The key difference between the two methods discussed earlier was in the techniques used for feature extraction and classification. In the handcrafted feature-based approach, manual feature extraction was performed and the classification was done using two machine learning models. In the deep learning approach, both feature extraction and classification were embedded within the network and were learnable through the training process. In the following sections, we introduce two hybrid-learning strategies that combine two or more of the techniques for feature extraction and classification.

4.5.1 Late-fusion

In this approach, the feature extraction is performed using the convolution and pooling layers of a trained CNN and in place of the fully-connected layers, a different supervised classifier is used. This is termed as late-fusion strategy because the combining of the two methods occurs only at the classification stage. The steps involved in the late-fusion method can be summarized as follows:

- The proposed CNN architecture (refer section 4.4) is trained using a training set of images.

- From the trained network, the activations at the end of the feature-extraction layers of the CNN are obtained for each of the training samples.

- The activations are considered as the input features for classification. Instead of using the fully-connected layers of the CNN for classification, two other

supervised classifiers namely - SVM and Adaboost are used for the final classification. In figure 4.5 the late-fusion scheme described above is shown.

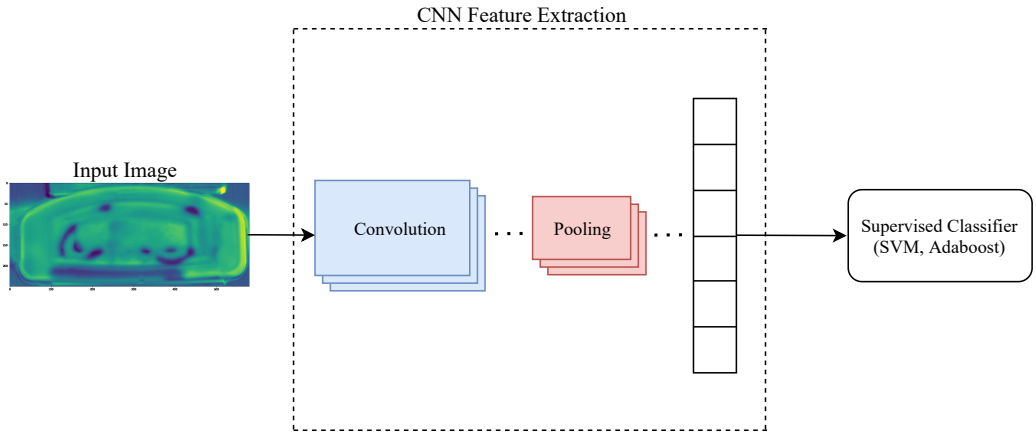


FIGURE 4.5: Hybrid Learning - Late Fusion approach

4.5.2 Early-fusion

The next approach is referred to as early-fusion. This is due to the fact that the fusion of techniques occurs earlier in the pipeline during the feature extraction stage [42]. In an early-fusion scheme, the feature extraction may be performed using different techniques or features may be obtained from different sources. For example, using text-data in combination with image features or using two separate neural network architectures to produce feature maps and merging their outputs may be considered feature-fusion. In this work, the following steps are involved in the early-fusion approach:

1. The activations from the last feature layer of the trained CNN are taken for each training sample.

2. Texture features are extracted from each image using GLCM and LBP as described in section 4.3.

3. Suppose the feature vectors obtained from step 1 and step 2 are F_1 and F_2 respectively. A new feature vector F is created by concatenating F_1 and F_2 as $F = [F_1, F_2]$.

4. F is used as input to train a supervised algorithm to obtain the final classification. Note that early-fusion strategy may also use the same fully-connected architecture of the CNN for classification since the distinction of the approach lies in the fusion of features.

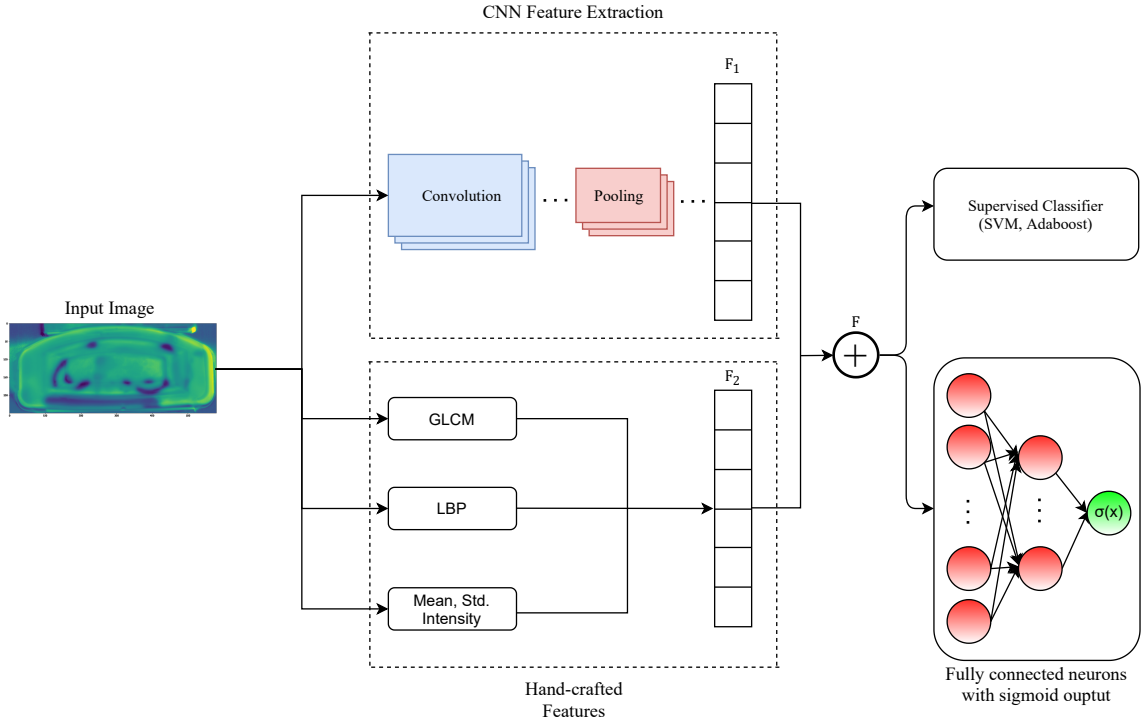


FIGURE 4.6: Hybrid Learning - Early Fusion approach

Chapter 5

Experiments and Results

In this chapter, the experimental setup used to conduct this study is first described and the results for each of the methods discussed in Chapter 4 are presented. The evaluation metrics used to evaluate the results are also defined. Finally, an additional experiment is performed in section 5.5 using the proposed CNN architecture to address the auxiliary research question defined in Chapter 1.

5.1 Experimental Setup

The experimental setup consists of the data acquisition system used to capture the required data. In the following sections, the components of the data acquisition system and the description of the dataset obtained from it are explained.

5.1.1 Data Acquisition System

An active infrared thermography setup is used to collect the data required to detect fluid leakage from detergent containers. The main requirements of such an experimental setup were stated as follows:

- The setup should be able to produce thermal images of the bottom of the containers to be able to detect fluid leakage.
- The thermal images produced through the setup must be highly reproducible.
- The heating and imaging processes must be automated so that inspection can be done for a large number of products one after the other.

Keeping the above requirements in mind, a demonstrator was set up with the following components:

1. A cobot used for pick-and-place from conveyors and carrying the containers over the setup for heating and imaging.

2. A ceramic heater that is used to heat the bottom of the containers to create the temperature difference between leaking fluid and the material of the container.

3. An infrared camera to capture images of the container immediately after heating. Figure 5.1 shows the setup with the components described above.

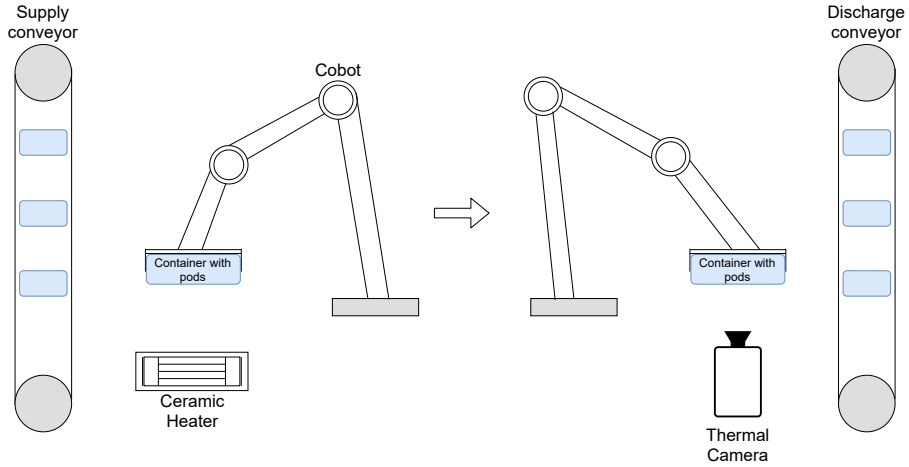


FIGURE 5.1: Thermography setup used for data acquisition

5.1.2 Dataset

An experimental dataset was created for the purpose of this study. Typically, a detergent pod consists of multiple chambers of different sizes containing different fluids. Therefore, a real instance of detergent leakage was found to be anywhere between 5ml to 25ml. Therefore, we create examples of leakages with varying amounts of fluid to represent the occurrence of leakage in a holistic way. In figure 5.2, we show the different representations of leakage included in the dataset. The leakage is highlighted by red bounding boxes in the image.

The components of the data acquisition system can be adjusted to get images with good contrast. For example, the heater can be mounted at different positions such that the distance between the heater and the container may be varied. Similarly, the time of heating may also be increased or decreased depending on the required temperature range. For all the images used in this study, we fix the distance and time of heating and obtain images in the same temperature range. By experimenting with these parameters, similar images can be produced even at different environmental conditions. The final dataset consists of 1305 images with an equal distribution of both the classes. A total of 653 images with leakage and 652 images without leakage

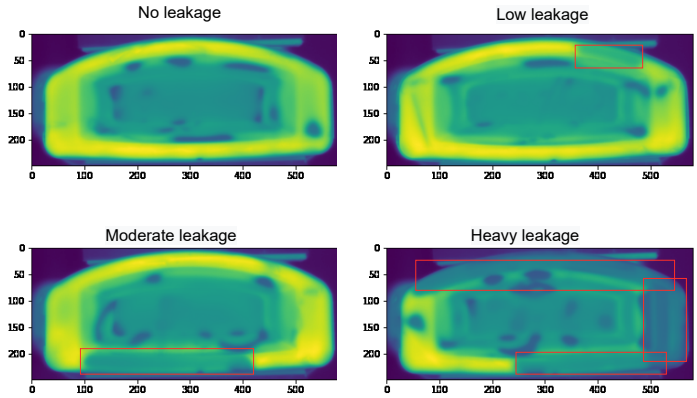


FIGURE 5.2: Various representations of leakage based on amount of fluid

were created. The images were taken in two different locations with varying ambient conditions, but the temperature ranges within the images were kept consistent by varying the time and distance of heating.

5.2 Evaluation Metrics

To evaluate the classification performance of the different methods, we mainly use the metrics - accuracy, sensitivity, specificity and f1-score. Accuracy is a direct measure of the correct predictions, i.e true positives (TP) and true negatives (TN) of the model but gives no information about the which class was misclassified more often. Therefore, to quantify the effect of false negatives (FN) and false positives (FP), we use sensitivity and specificity respectively. The above metrics are defined as follows:

1. $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
2. $Sensitivity(TPR) = \frac{TP}{TP+FN}$
3. $Specificity(TNR) = \frac{TN}{TN+FP}$
4. $F1 - Score = \frac{2TP}{2TP+FP+FN}$

Apart from the above metrics, we also use the receiver operating characteristic (ROC) curve. The ROC curve is the plot between the false positive rates (FPR) and true positive rates (TPR) as functions of threshold. The threshold decides the outcome of the classification. The area under the ROC curve (AUC) is an overall measure of the model's performance with respect to TPR and FPR.

5.3 Outline of Experiments

In all of the experiments that follow, a K-fold cross-validation (CV) split is done on the original dataset consisting of 1305 images. In K-fold CV, the dataset is split into K unique samples and the model performance may be tested on each of the K samples by using the rest of the dataset for training. This approach provides an indication on the model performance on unseen examples. We choose $K = 4$ and perform the following experiments:

- Manual feature extraction is done separately on each CV split and classification is performed using two classifiers. Section 5.4.1 shows the results obtained from this approach for all the four CV splits.
- Next, the results obtained from the CNN-architecture described in 4.4 are presented in 5.4.2.
- The trained CNN is used to extract features for the hybrid approach and the results from the late-fusion and early-fusion approaches are presented in 5.4.3.
- An additional experiment is conducted using the CNN architecture to investigate the relationship between dataset size and model performance.

5.4 Results

In the following sections, the classification results obtained for the three approaches previously described are discussed.

5.4.1 Handcrafted feature-based approach

As described in section 4.3, the feature vector is created using the two feature descriptors GLCM and LBP. Next, the 102-dimensional vector is transformed to a smaller dimensional space using PCA. To choose the number of principal components k that can sufficiently explain the variance in the dataset, we observe the performance of one of the classifiers as a function of k. Figure 5.3 shows the average accuracy obtained over the 4-fold split using the SVM classifier for different values of k.

From the above graph, it can be observed that the performance initially increases steadily until about $k = 50$ and then stagnates for the rest of the range. Therefore a value of $k = 60$ and the original feature set is reduced to 60 columns. Next, the reduced feature vector is used to train the classifiers - SVM and Adaboost. The number

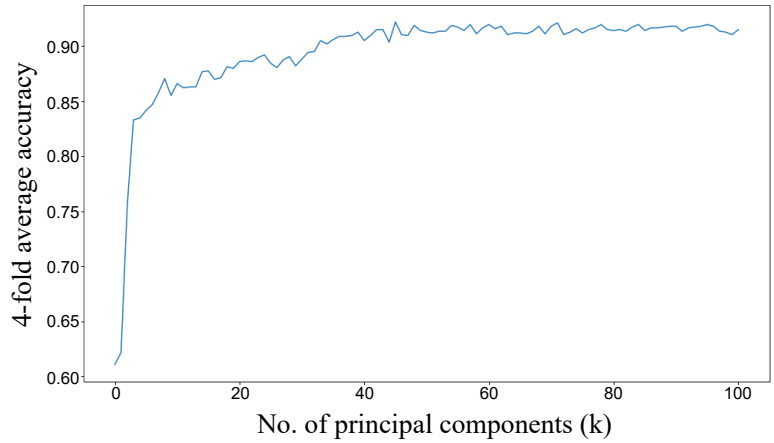


FIGURE 5.3: Accuracy as a function of number of principal components

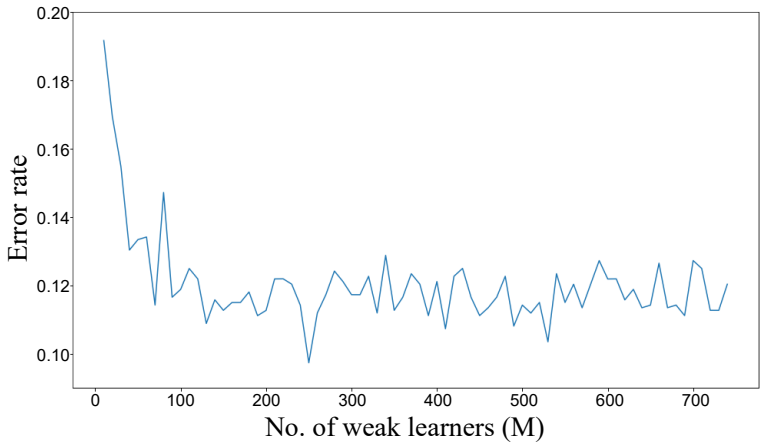


FIGURE 5.4: Error-rate as a function of number of estimators M

of individual learners M for the Adaboost algorithm is a hyperparameter to be chosen. Therefore to select a suitable parameter for M, the prediction error-rate of the model is observed over different values of M in figure 5.4. The error rate decreases as we increase M until $M = 250$ and then remains nearly constant irrespective of the increase in the number of estimators. Therefore, for all the experiments we use a value of $M = 400$.

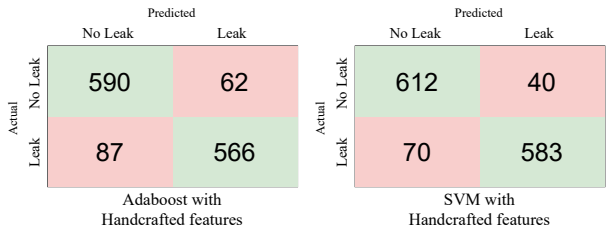
The 4-fold cross validation results for the handcrafted feature (HCF) based approach are presented. Table 5.1 shows the accuracy, sensitivity, specificity and F1-scores respectively for each fold as well as the overall scores. Since the test set in each fold is unique, the four components (TP, FP, FN and TN) of the confusion matrix from each fold may be summed. The cumulative confusion matrices are shown

TABLE 5.1: Results of SVM and Adaboost models on handcrafted features

	Model	Fold-1	Fold-2	Fold-3	Fold-4	Overall
Accuracy	HCF+Adaboost	0.8899	0.8681	0.9018	0.8834	0.8858
	HCF+SVM	0.9021	0.9141	0.9294	0.9172	**0.9157**
Sensitivity	HCF+Adaboost	0.9020	0.8848	0.9353	0.8963	0.9049
	HCF+SVM	0.9359	0.9209	0.9568	0.9427	**0.9386**
Specificity	HCF+Adaboost	0.8793	0.8509	0.8654	0.8704	0.8667
	HCF+SVM	0.8713	0.9060	0.9024	0.8935	**0.8928**
F1-Score	HCF+Adaboost	0.8846	0.8947	0.9086	0.8855	0.8879
	HCF+SVM	0.9012	0.9209	0.9309	0.9164	**0.9175**

in table 5.2. The overall scores in table 5.1 are obtained from these confusion matrices.

TABLE 5.2: Cumulative confusion matrices of SVM and Adaboost with handcrafted features



5.4.2 CNN-based approach

Before discussing the results, the process of training the neural network is first explained. The optimization algorithm takes smaller batches of training images rather than the entire dataset in each iteration before updating the weights of the network. Therefore the number of images per batch, known as the *batch-size*, must be chosen. Another hyperparameter called *epoch* decides how many times the entire training dataset propagates through the network. In one epoch, every sample from the training set will have passed through the network and contributed to the parameter updates. The optimizer takes another parameter called learning-rate that decides how large the updates to the weights are in each iteration. It is important to choose an optimal learning rate because if the updates are too small, then the loss convergence is too slow and if they are too large, then the chances of overshooting

increases. Based on initial experiments, the following values were chosen for the training parameters:

- Batch-size = 20
- Learning rate = 0.001
- No. of epochs = 150

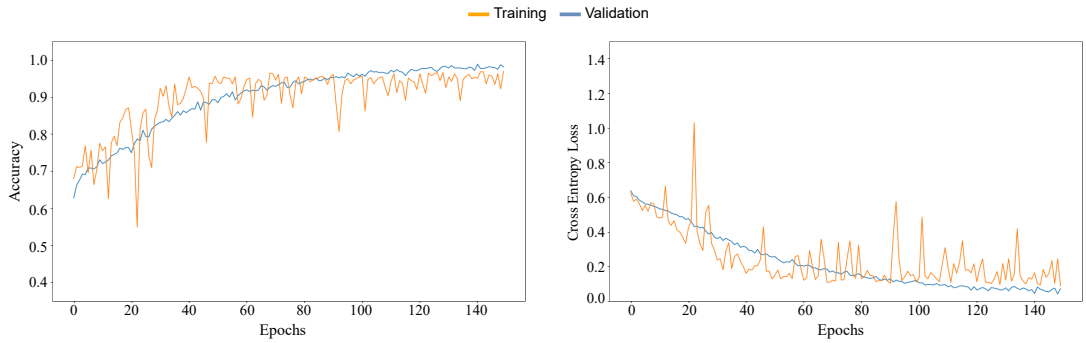


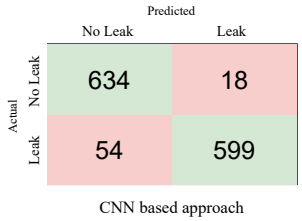
FIGURE 5.5: Training and validation accuracy and cross-entropy loss observed over 150 epochs

A fraction of the training data (30%) is used for validation of the model performance during the training process. Figure 5.5 shows the training and validation accuracy and loss obtained for 150 epochs. It can be observed that the training accuracy does not increase beyond this range and it can also be deduced that there is no significant overfit as the validation accuracy closely follows the training accuracy throughout the graph. After training, the test images are passed through the network and the sigmoid output in the range [0,1] is obtained. We assign class labels to each test sample by applying a threshold of 0.5. The 4-fold results for the CNN-based approach are given in table.

TABLE 5.3: Results of CNN-based approach

	Fold-1	**Fold-2**	**Fold-3**	**Fold-4**	**Overall**
Accuracy	0.9358	0.9417	0.9417	0.9601	0.9448
Sensitivity	0.9648	0.9820	0.9634	0.9777	0.9723
Specificity	0.9135	0.8994	0.9198	0.9388	0.9173
F1-score	0.9288	0.9452	0.9433	0.9642	0.9463

TABLE 5.4: Cumulative confusion matrix of CNN



5.4.3 Hybrid approach

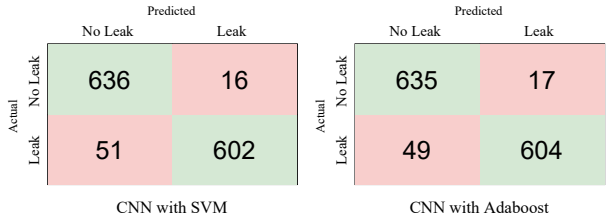
In this section, the results obtained from the two hybrid methods are presented. First, the feature vector is obtained from the last feature extraction layer of the trained CNN and two classifiers - Adaboost and SVM are used in place of the fully-connected layers. The results of this approach are shown in table 5.5.

TABLE 5.5: Results of late-fusion approach

	Model	Fold-1	Fold-2	Fold-3	Fold-4	Overall
Accuracy	CNN + SVM	0.9358	0.9509	0.9448	0.9632	0.9487
	CNN + Adaboost	0.9327	0.9540	0.9417	0.9693	**0.9494**
Sensitivity	CNN + SVM	0.9789	0.9701	0.9756	0.9777	**0.9754**
	CNN + Adaboost	0.9789	0.9760	0.9573	0.9832	0.9739
Specificity	CNN + SVM	0.9027	0.9308	0.9146	0.9459	0.9218
	CNN + Adaboost	0.8937	0.9308	0.9259	0.9524	**0.9249**
F1-Score	CNN + SVM	0.9298	0.9673	0.9467	0.9669	0.9500
	CNN + Adaboost	0.9267	0.9560	0.9429	0.9724	**0.9506**

We can deduce from the above table that both the classifiers outperform the pure CNN based results. Particularly, the Adaboost model shows higher performance in terms of overall accuracy as well as F1-score. The best scores are highlighted in the table. The cumulative confusion matrices of the late-fusion models are shown in table 5.6.

TABLE 5.6: Cumulative confusion matrix of late-fusion models



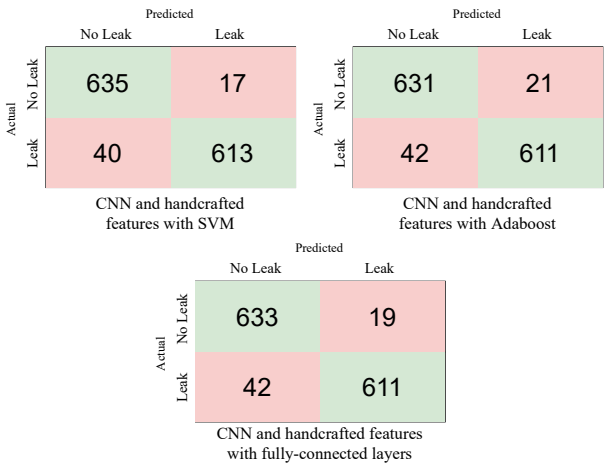
Next, feature fusion is performed by concatenating the features from the two sources - CNN and manual feature extraction. The fused feature vector F is given as input to three models - SVM, Adaboost and the fully connected neural network (FCNN). The results of the early-fusion approach are shown in table 5.7.

TABLE 5.7: Results of early-fusion approach

	Model	Fold-1	Fold-2	Fold-3	Fold-4	Overall
Accuracy	CNN + HCF + SVM	0.9415	0.9571	0.9509	0.9755	**0.9563**
	CNN + HCF + Adaboost	0.9327	0.9571	0.9448	0.9724	0.9517
	CNN + HCF + FCNN	0.9327	0.9632	0.9479	0.9693	0.9533
Sensitivity	CNN + HCF + SVM	0.9789	0.9581	0.9756	0.9832	**0.9739**
	CNN + HCF + Adaboost	0.9718	0.9641	0.9634	0.9721	0.9677
	CNN + HCF + FCNN	0.9507	0.9760	0.9756	0.9777	0.9708
Specificity	CNN + HCF + SVM	0.9135	0.9560	0.9259	0.9660	**0.9387**
	CNN + HCF + Adaboost	0.9027	0.9497	0.9259	0.9728	0.9356
	CNN + HCF + FCNN	0.9189	0.9497	0.9198	0.9592	0.9356
F1-Score	CNN + HCF + SVM	0.9360	0.9581	0.9524	0.9778	**0.9570**
	CNN + HCF + Adaboost	0.9262	0.9583	0.9461	0.9748	0.9525
	CNN + HCF + FCNN	0.9247	0.9645	0.9496	0.9722	0.9540

We can see that the addition of handcrafted features results in a further increase in the performance of the models compared to the late-fusion results. The confusion matrices of the three models are given below.

TABLE 5.8: Cumulative confusion matrix of early-fusion models



5.4.4 Visualization of results

In figure 5.6, we plot the box plots of the 4-fold accuracy and F1-scores from all the proposed methods to visualize the overall prediction performances of the models. It can be observed that the CNN outperforms the pure feature based methods by a significant margin. Both late-fusion and early-fusion models provide a small improvement in the CNN results in terms of both accuracy and F1-scores.

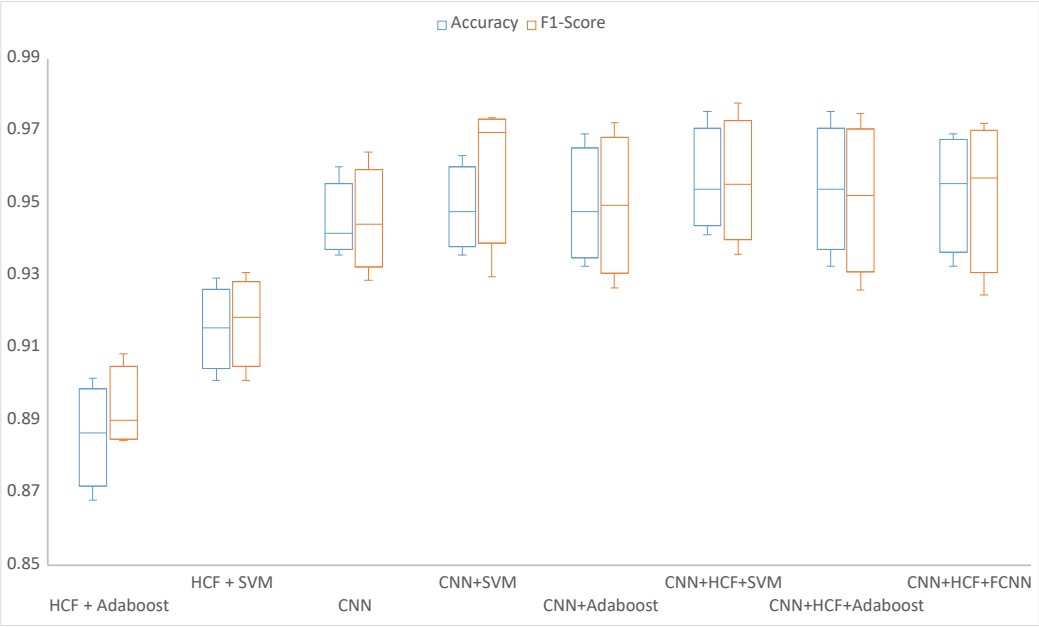


FIGURE 5.6: Box plots of accuracy and F1-scores of all the models

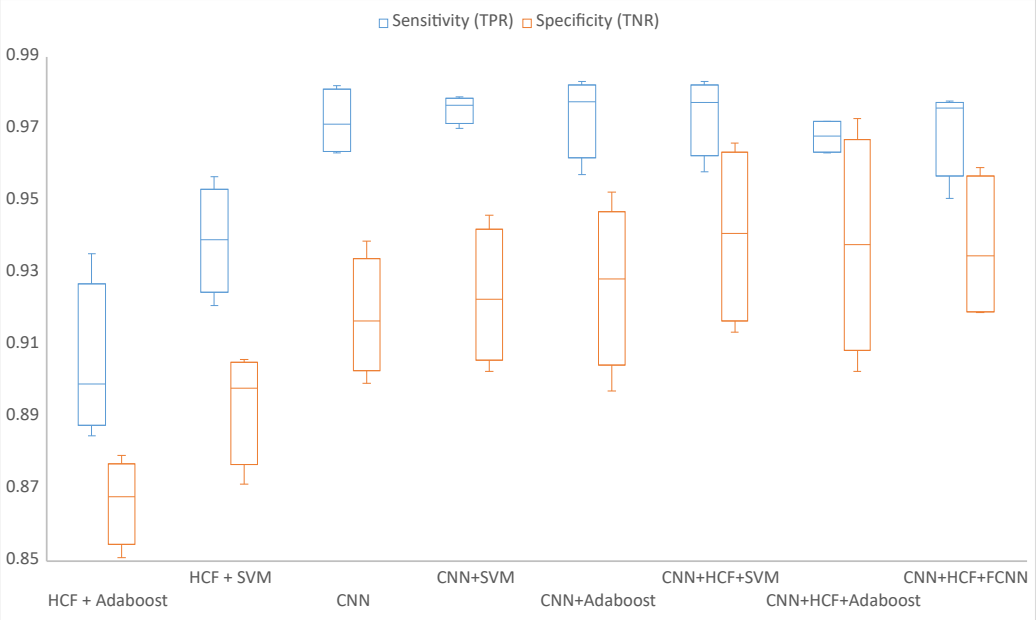


FIGURE 5.7: Box plots of sensitivity and specificity of all the models

Since accuracy and F1-score do not provide information about the ability of the models specific to the positive and negative classes, we plot the sensitivity (true positive rate) and specificity (true negative rate) of all the models in figure 5.7. Here, positive class refers to non-leakage and negative class refers to leakage. Generally, we observe that the TPRs of all the models are higher than the respective TNRs. This shows that the predictions are skewed towards the positive class. We can also observe that the TPR is not improved significantly by the fusion models, but there is a clear increase in the TNRs compared to the pure CNN method. Specifically the early fusion method with SVM model gives the best performance in terms of all the metrics in consideration.

5.4.5 ROC Results

The advantage of obtaining a score distribution as output is that the final decision may be made using a threshold. The sigmoid function outputs a probability score between 0 and 1 and generally, the threshold is set as 0.5. But if we observe the prediction and error rates as a function of the decision threshold, more insights about the model performance can be gained. As mentioned earlier, the ROC curve is the relationship between false-positive rates and true-positive rates as a function of threshold. Therefore, if we vary the threshold from 0 to 1 in small steps, the model performance can be observed over the range of thresholds.

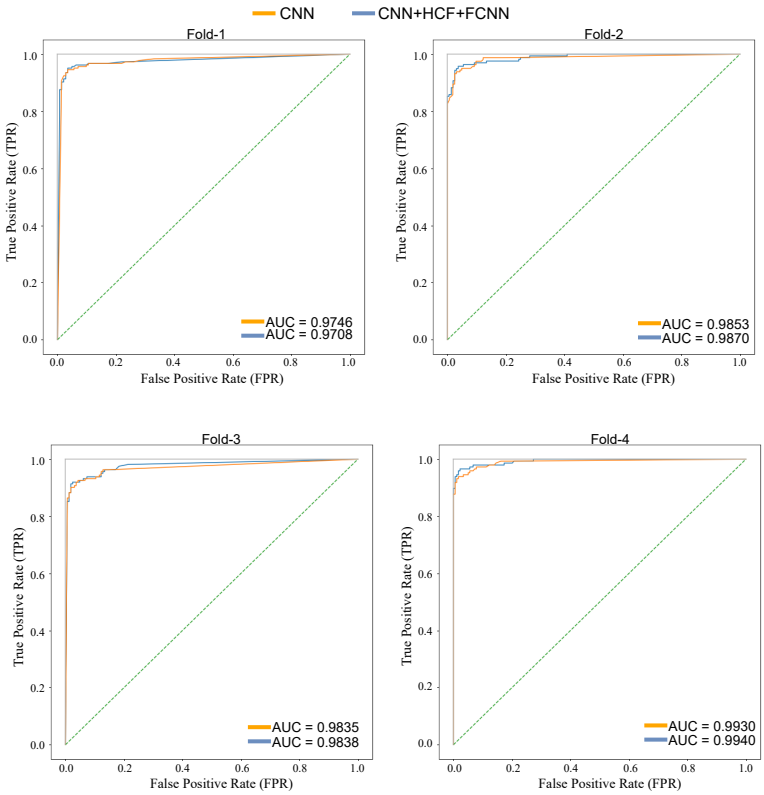


FIGURE 5.8: 4-fold ROC results for CNN and early-fusion model

In figure 5.8, we compare the ROC curves of the pure CNN approach and the early-fusion approach with fully-connected layers over the 4 folds to see the effect of adding handcrafted features before the classification. It can be observed that, in most cases, the fusion model has a better ROC curve in terms of AUC. It can also noticed that the upper-left most point on the graph is achieved by the fusion method as well. To further validate the performances, we look at the FNR at a fixed FPR of 5% for each of the models (table 5.9). It is clear that in 3 folds the fusion model achieves a smaller FNR compared to the pure CNN model.

TABLE 5.9: False Negative Rates (FNR) at False Positive Rate (FPR) of 0.05

	Fold-1	Fold-2	Fold-3	Fold-4
CNN	0.070	0.044	0.074	0.054
CNN+HCF+FCNN	0.075	0.037	0.067	0.034

5.5 Effect of dataset size on performance

The amount of data required for training neural networks is a highly relevant question in most industrial applications due to the infeasibility of collecting data. This

issue also persists in this work as the generation of images is a tedious process and the exact number of images to obtain the best performance is unknown. Therefore, in order to investigate the relationship between the size of training data and classification performance, we use the CNN architecture from section 4.4.

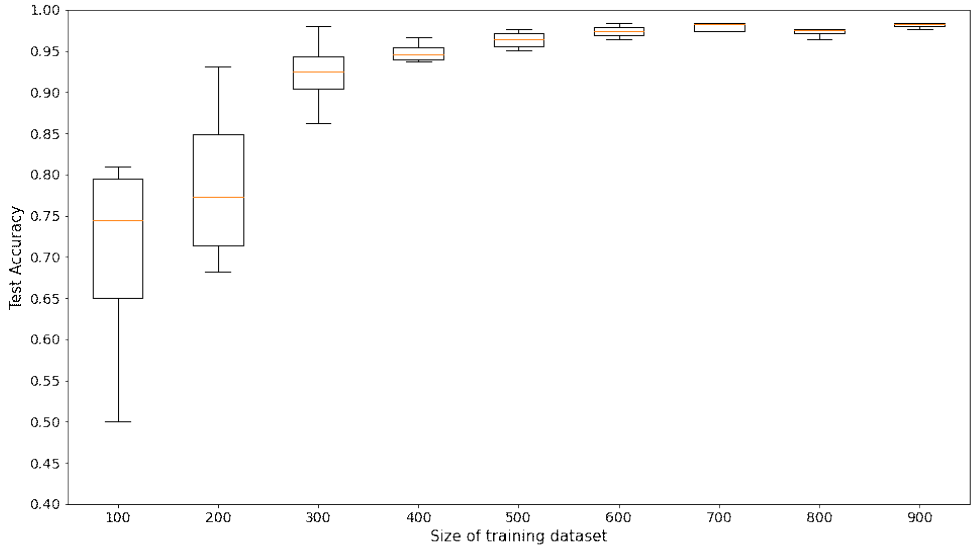


FIGURE 5.9: Testing accuracy for various values of dataset size

The performance is evaluated on a static test set of 300 images separated from the training set. The CNN architecture is trained using various values of dataset size increasing in steps of 100. To get a robust estimate of the model performance, the images are randomly sampled from the original dataset four times and their accuracy values are plotted against the size of the dataset (shown in figure 5.9). From the figure, it is clear that the performance depends on the dataset size until a certain value is reached (about 600). After this point, the graph reaches a plateau and the rate of increase of accuracy becomes very small. To model the relationship between the size of the dataset and the accuracy, we try to fit an asymptotic curve on the above data. The curve is defined by the equation given as follows:

$$f(x) = A(1 - e^{-b(x-c)}) \tag{5.1}$$

where A, b and c are parameters to be optimized. We choose the above function because it resembles the relationship observed in figure 5.9 and we know that the accuracy value cannot exceed 1. The function reaches the value A at infinity and intercepts the x-axis at c. The factor b decides the rate of increase of the exponential function. We constraint the function to exist only for values of $x > 0$ because negative values of dataset size are absurd. For $x > 0$, the function is always positive

and converges to a value $A < 1$ at infinity. Figure 5.10 shows the curve fit on the relationship between dataset size and accuracy.

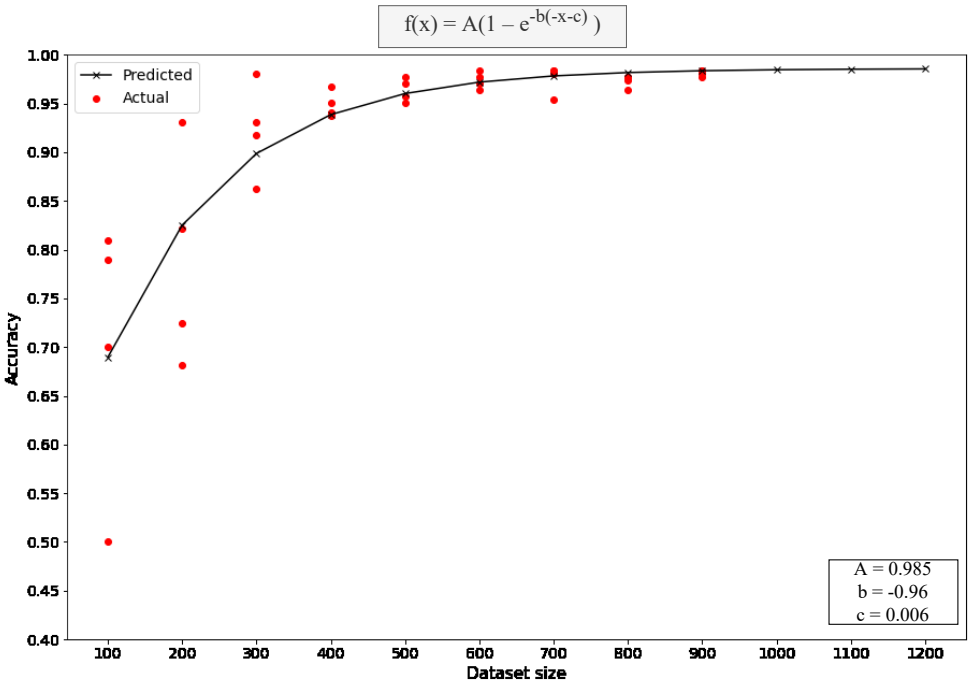


FIGURE 5.10: Curve fit on dataset size versus testing accuracy

To solve for the parameters in the above equation, we use the Levenberg-Marquardt non-linear least squares algorithm [59] from the SciPy optimization toolbox. The optimal values of A, b and c are estimated as 0.985, -0.96 and 0.006 respectively. This shows that the function is already close to reaching its maximum value. This can also be observed from figure 5.10. The purpose of performing this experiment is to provide a generic way to use the existing models to find how much significance does increasing the amount of training data have for a specific method. It can also answer questions such as how much data would be required to achieve a certain performance using a particular method.

Chapter 6

Conclusions

6.1 Research Questions

The conclusions with respect to each research question introduced in Chapter 1 are given as follows:

- **Q: What are the steps involved in building an image classification system to distinguish between instances of leakage and non-leakage from infrared thermal (IRT) images of detergent containers?**

  A: The following steps were involved in building a classifier to identify leakage from IRT images - image acquisition, pre-processing, feature extraction, dimensionality reduction and classification. Three different approaches were identified, namely handcrafted feature-based approach, deep learning approach and hybrid approach. The classification performance of each of the above methods on an experimental dataset are compared using various metrics.

- **Q: Which feature extraction techniques may be used to extract information that distinguish between leakage and non-leakage from IRT images of detergent containers?**

  A: Various methods of feature extraction were identified from relevant literature. Finally, two techniques, namely GLCM and LBP were used to extract 102 features from the images. After PCA feature selection, the reduced feature vector is used as input to two classifiers SVM and Adaboost. An overall accuracy of 91.57% and F1-score of 91.75% over 4 CV folds was achieved by the SVM classifier.

- **Q: How do handcrafted feature-based methods compare to convolutional neural networks in terms of classification performance?**

  A: A custom CNN architecture was built for classification of leakage and non-leakage from the IRT images and the choices involved in the architecture design were described. The CNN outperformed the baseline method (HCF based

approach) with an overall accuracy and F1-score of 94.48% and 94.63% respectively over the 4-fold split.

- **Q: Do hybrid techniques that combine multiple feature sources or classifiers improve the overall recognition performance?**

  A: Two hybrid learning strategies were introduced, namely late-fusion and early-fusion. Both alternatives provided a small improvement to the pure CNN-approach. The highest performance was achieved by the early-fusion approach with SVM classifier with a 4-fold accuracy and F1-score of 95.63% and 95.70% respectively. The ROC curves of the pure-CNN approach and the early-fusion approach with fully-connected layers are shown. The fusion method showed a better performance in terms of AUC and FNR at 5% FPR.

- **Q: What effect does dataset size have on the performance of the models and how to identify the amount of data required to achieve a certain level performance?**

  A: The proposed CNN model was trained using different training dataset sizes. It was found that the accuracy of the model increased with increase in dataset size of about 700, but after this point the rate of increase starts to decrease. To model this relationship and predict the performance of the model outside the known range, an asymptotic exponential function was fit to the data. The predictions from the curve fit reveal that the accuracy had almost reached its maximum and addition of training data would not significantly improve the results.

6.2 Discussions and Recommendations

From the graphs shown in section 5.4.4, it is clear that all the models are generally able predict non-leakage better than leakage as indicated by the sensitivity (TPR). This behaviour was expected as the pods resting on the bottom of the box were also similar in appearance to an instance of leakage. Higher amounts of leakage spread over the surface and affected the temperature of the entire region, making it easy to recognize. But in the case of small amounts of fluid, it was challenging to distinguish from the pods (as seen in figure 5.2). The importance of predicting a specific class depends on the requirement of the customer. The ROC curve becomes more relevant in this case because it helps visualizes the performance of the model in terms of TPRs and FPRs over different thresholds.

As per the problem definition, it was sufficient to classify entire images as leaking or not leaking. Some researchers have performed defect classification using a sliding window. Cha et al. [60] have performed crack detection on concrete walls by dividing images of larger resolution into smaller image patches using a sliding window. A similar approach may be applied to this problem, but the major challenge would be to perform manual labeling on the image patches. The advantage of this approach is that it would be possible to obtain a more localised identification of leakage.

Hybrid methods, especially the feature-fusion models have shown promising performance with the experimental dataset. More feature groups can be added to the fused feature vector and an additional step to pick out the most important features from the fused vector may be explored. For example, GIST features have been combined successfully with CNN features and provided an improvement to the classification performance. GIST features are extracted by using Gabor filters at various scales and orientations to capture the *gist* from an image. Implementations of GIST features with CNN are shown in [61] and [62]. Another factor in fusion methods is the operation used to fuse different feature vectors. Generally, the vectors are simply concatenated with each other, but some works have benefited from using different fusion methods such as summing and max operations [42].

Transfer learning has been popularly used in defect classification problems in literature. Initial experiments on transfer learning for this problem did not yield desired results. Several network architectures that have been trained on millions of images exist that can be applied readily to any dataset. One of the challenges faced during the implementation in this case was that the thermal images were single channel images and most existing architectures are trained on RGB images and hence take 3-channel input. Therefore, the single channel images were duplicated along the three channels and given as input for training. Another disadvantage of transfer learning is that the dimensions of the input image must be resized to the size of the images that the network was trained on. Apart from the limitations mentioned above, an advantage of transfer learning is that it may be even with small datasets and the hybrid learning experiments can easily be extended by using the activations from other trained neural networks.

CNNs are predominantly used for most image based applications and are considered state-of-the-art. Recently, visual transformers have been used on image

classification problems and have been reported to outperform CNN results. Transformers use attention mechanism and apply different weights to different parts of the input based on significance to the output. This has been extensively used in NLP applications such as text classification and information retrieval. The attention mechanism has also been successfully applied to computer vision problems where the attention layers map the importance of different parts of the image. Dosovitskiy et al. [63] achieve this by dividing the image into 16×16 patches and assigning positional encoding similar to tokens in a transformer architecture.

Bibliography

[1] Silvia Satorres et al. "A machine vision system for defect characterization on transparent parts with non-plane surfaces". In: *Machine Vision and Applications* 23 (2012), pp. 1–13. DOI: 10.1007/s00138-010-0281-0.

[2] Joseph Walsh et al. "Deep Learning vs. Traditional Computer Vision". In: 2019. ISBN: 978-981-13-6209-5. DOI: 10.1007/978-3-030-17795-9_10.

[3] *Laundry Detergent Pods Market Size: Industry Report, 2019-2025*. 2019. URL: https://www.grandviewresearch.com/industry-analysis/laundry-detergent-pods-market.

[4] *Liquid Laundry Detergent Capsules Guidelines on CLP Implementation*. 2018. URL: https://www.aise.eu/documents/document/20181203162709-clp_implementation_guidelines_lldc_v2_0_261118.pdf.

[5] Osslan Vergara et al. "Automatic Product Quality Inspection Using Computer Vision Systems". In: 2014, pp. 135–156. ISBN: 978-3-319-04950-2. DOI: 10.1007/978-3-319-04951-9_7.

[6] Mina Fahimipirehgalin et al. "Visual Leakage Inspection in Chemical Process Plants Using Thermographic Videos and Motion Pattern Detection". In: *Sensors* 20.22 (2020). ISSN: 1424-8220. DOI: 10.3390/s20226659. URL: https://www.mdpi.com/1424-8220/20/22/6659.

[7] Michele De Filippo et al. "Concept of Computer Vision Based Algorithm for Detecting Thermal Anomalies in Reinforced Concrete Structures". In: *Proceedings* 27.1 (2019). ISSN: 2504-3900. DOI: 10.3390/proceedings2019027018. URL: https://www.mdpi.com/2504-3900/27/1/18.

[8] Gareth James et al. *An introduction to statistical learning : with applications in R*. Springer, 2014. ISBN: 978-1-4614-7137-0.

[9] Sergiu Deitsch et al. "Automatic Classification of Defective Photovoltaic Module Cells in Electroluminescence Images". In: *Solar Energy* 185 (2018). DOI: 10.1016/j.solener.2019.02.067.

[10] Mina Fahimipirehgalin et al. "Automatic Visual Leakage Inspection by Using Thermographic Video and Image Analysis". In: (2019), pp. 1282–1288. DOI: 10.1109/COASE.2019.8842941.

[11] Navid Razmjooy, B. Somayeh Mousavi, and F. Soleymani. "A real-time mathematical computer method for potato inspection using machine vision". In: *Computers & Mathematics with Applications* 63.1 (2012), pp. 268–279. ISSN: 0898-1221. DOI: https://doi.org/10.1016/j.camwa.2011.11.019. URL: https://www.sciencedirect.com/science/article/pii/S0898122111009850.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Neural Information Processing Systems* 25 (2012). DOI: 10.1145/3065386.

[13] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[14] Qingzhong Li, Maohua Wang, and Weikang Gu. "Computer vision based system for apple surface defect detection". In: *Computers and Electronics in Agriculture* 36 (2002), pp. 215–223. DOI: 10.1016/S0168-1699(02)00093-5.

[15] K. Mak, Polosnow Peng, and K. Yiu. "Fabric Defect Detection Using Morphological Filters". In: *Image Vision Comput.* 27 (2009), pp. 1585–1592. DOI: 10.1016/j.imavis.2009.03.007.

[16] G. Rahaman and Md Mobarak Hossain. "Automatic Defect Detection and Classification Technique from Image: A Special Case Using Ceramic Tiles". In: *International Journal of Computer Science and Information Security* 1 (2009).

[17] Ioannis (John) Tsanakas et al. "Fault Diagnosis of Photovoltaic Modules through Image Processing and Canny Edge Detection on Field Thermographic Measurements". In: *International Journal of Sustainable Energy* 34 (2015), pp. 351–372. DOI: 10.1080/14786451.2013.826223.

[18] *Texture Analysis Using the Gray-Level Co-Occurrence Matrix*. URL: https://www.mathworks.com/help/images/texture-analysis-using-the-gray-level-co-occurrence-matrix-glcm.html.

[19] I-Shou Tsai, Chung-Hua Lin, and Jeng-Jong Lin. "Applying an Artificial Neural Network to Pattern Recognition in Fabric Defects". In: *Textile Research Journal* 65.3 (1995), pp. 123–130. DOI: 10.1177/004051759506500301. URL: https://doi.org/10.1177/004051759506500301.

[20] Domingo Mery et al. "Quality classification of corn tortillas using computer vision". In: *Journal of Food Engineering* 101 (2010), pp. 357–364. DOI: 10.1016/j.jfoodeng.2010.07.018.

[21] Oscar García-Olalla et al. "Adaptive local binary pattern with oriented standard deviation (ALBPS) for texture Classification". In: *EURASIP Journal on Image and Video Processing* 2013 (2013). DOI: 10.1186/1687-5281-2013-31.

[22] Lei Zhang. "Fabric Defect Classification Based on LBP and GLCM". In: *Journal of Fiber Bioengineering and Informatics* 8 (2015). DOI: 10.3993/jfbi03201508.

[23] Olli Silvén, Matti Niskanen, and Hannu Kauppinen. "Wood Inspection With Non-Supervised Clustering". In: *Mach. Vis. Appl.* 13 (2003), pp. 275–285. DOI: 10.1007/s00138-002-0084-z.

[24] Vishwanath Sindagi and Sumit Srivastava. "OLED panel defect detection using local inlier-outlier ratios and modified LBP". In: *Proceedings of the 14th IAPR International Conference on Machine Vision Applications, MVA 2015* (2015), pp. 214–217. DOI: 10.1109/MVA.2015.7153170.

[25] Lucia Bissi et al. "Automated defect detection in uniform and structured fabrics using Gabor filters and PCA". In: *Journal of Visual Communication and Image Representation* 24 (2013), 838–845. DOI: 10.1016/j.jvcir.2013.05.011.

[26] Yu-Dong Zhang and Lenan Wu. "Classification of Fruits Using Computer Vision and a Multiclass Support Vector Machine". In: *Sensors (Basel, Switzerland)* 12 (2012), pp. 12489–505. DOI: 10.3390/s120912489.

[27] Mr Yogesh et al. "Computer vision based analysis and detection of defects in fruits causes due to nutrients deficiency". In: *Cluster Computing* 23 (2020). DOI: 10.1007/s10586-019-03029-6.

[28] Hyeong-Gyeong Moon and Jung-Hoon Kim. "Inteligent Crack Detecting Algorithm on the Concrete Crack Image Using Neural Network". In: 2011. DOI: 10.22260/ISARC2011/0279.

[29] Tamás Czimmermann et al. "Visual-Based Defect Detection and Classification Approaches for Industrial Applications—A SURVEY". In: *Sensors* 20 (2020), p. 1459. DOI: 10.3390/s20051459.

[30] Du-Ming Tsai, Jeng-Jong Chen, and Jeng-Fung Chen. "A Vision System for Surface Roughness Assessment Using Neural Networks". In: *The International Journal of Advanced Manufacturing Technology* 14 (1998), 412–422.

[31] GM Nasira and P Banumathi. "Fourier Transform and Image Processing in Automated Fabric Defect Inspection System". In: *International Journal of Computational Intelligence and Informatics* 3 (2013).

[32] Siew Mar, Prasad Yarlagadda, and C. Fookes. "Design and development of automatic visual inspection system for PCB manufacturing". In: *Robotics and Computer-integrated Manufacturing* 27 (2011), pp. 949–962. DOI: 10.1016/j.rcim.2011.03.007.

[33] Amin Taheri-Garavand et al. "An intelligent approach for cooling radiator fault diagnosis based on infrared thermal image processing technique". In: *Applied Thermal Engineering* 87 (2015), 434–443. DOI: 10.1016/j.applthermaleng.2015.05.038.

[34] Deepam Goyal et al. "Support vector machines based non-contact fault diagnosis system for bearings". In: *Journal of Intelligent Manufacturing* 31.5 (2020), pp. 1275–1289. DOI: 10.1007/s10845-019-01511-. URL: https://ideas.repec.org/a/spr/joinma/v31y2020i5d10.1007_s10845-019-01511-x.html.

[35] Kabir Hossain, Frederik Villebro, and Søren Forchhammer. "UAV Image Analysis for Leakage Detection in District Heating Systems using Machine Learning". In: *Pattern Recognition Letters* 140 (May 2020). DOI: 10.1016/j.patrec.2020.05.024.

[36] Blaise Ngendangenzwa. "Defect detection and classification on painted specular surfaces". In: Umea University, 2018.

[37] Christopher Dunderdale et al. "Photovoltaic defect classification through thermal infrared imaging using a machine learning approach". In: *Progress in Photovoltaics: Research and Applications* 28 (2019). DOI: 10.1002/pip.3191.

[38] Li Deng and Dong Yu. *Deep Learning: Methods and Applications*. Tech. rep. MSR-TR-2014-21. Microsoft, 2014.

[39] Yongbo Li et al. "Rotating machinery fault diagnosis based on convolutional neural network and infrared thermal imaging". In: *Chinese Journal of Aeronautics* 33 (2019). DOI: 10.1016/j.cja.2019.08.014.

[40] Amin Nasiri et al. "Intelligent fault diagnosis of cooling radiator based on deep learning analysis of infrared thermal images". In: *Applied Thermal Engineering* 163 (2019), p. 114410. DOI: 10.1016/j.applthermaleng.2019.114410.

[41] Jing Yang et al. "Using Deep Learning to Detect Defects in Manufacturing: A Comprehensive Survey and Current Challenges". In: *Materials* 13 (2020), p. 5755. DOI: 10.3390/ma13245755.

[42] Hilal Ergun et al. "Early and Late Level Fusion of Deep Convolutional Neural Networks for Visual Concept Recognition". In: *International Journal of Semantic Computing* 10.03 (2016), pp. 379–397. DOI: 10.1142/S1793351X16400158. URL: https://doi.org/10.1142/S1793351X16400158.

[43] Haidar Almubarak et al. "A Hybrid Deep Learning and Handcrafted Feature Approach for Cervical Cancer Digital Histology Image Classification". In: *International Journal of Healthcare Information Systems and Informatics* 14 (Apr. 2019), pp. 66–87. DOI: 10.4018/IJHISI.2019040105.

[44] S. Lahmiri. "Hybrid deep learning convolutional neural networks and optimal nonlinear support vector machine to detect presence of hemorrhage in retina". In: *Biomed. Signal Process. Control.* 60 (2020), p. 101978.

[45] Mehdi Moradi et al. "A hybrid learning approach for semantic labeling of cardiac CT slices and recognition of body position". In: *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. 2016, pp. 1418–1421. DOI: 10.1109/ISBI.2016.7493533.

[46] Jina Kim et al. "A deep hybrid learning model for customer repurchase behavior". In: *Journal of Retailing and Consumer Services* 59 (2021), p. 102381. ISSN: 0969-6989. DOI: https://doi.org/10.1016/j.jretconser.2020.102381. URL: https://www.sciencedirect.com/science/article/pii/S0969698920313898.

[47] *Introduction to active thermography*. URL: https://www.infrared-camera-blog.com/tutorials/active-thermography-introduction/.

[48] Robert M. Haralick, K. Shanmugam, and Its'Hak Dinstein. "Textural Features for Image Classification". In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-3.6 (1973), pp. 610–621. DOI: 10.1109/TSMC.1973.4309314.

[49] Manisha Verma, Balasubramanian Raman, and Subrahmanyam Murala. "Local Extrema Co-occurrence Pattern for Color and Texture Image Retrieval". In: *Neurocomputing* 165 (Mar. 2015), 255–269. DOI: 10.1016/j.neucom.2015.03.015.

[50] T. Ojala, M. Pietikainen, and T. Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7 (2002), pp. 971–987. DOI: 10.1109/TPAMI.2002.1017623.

[51] *Local Binary Pattern for texture classification*. URL: https://scikit-image.org/docs/0.10.x/auto_examples/plot_local_binary_pattern.html.

[52] Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2019. DOI: 10.1017/9781108380690.

[53] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Chem. Biol. Drug Des.* 297 (Jan. 2009), pp. 273–297. DOI: 10.1007/%2FBF00994018.

[54] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[55] Ihab S. Mohamed. "Detection and Tracking of Pallets using a Laser Rangefinder and Machine Learning Techniques". PhD thesis. 2017. DOI: 10.13140/RG.2.2.30795.69926.

[56] Rafael C. Gonzalez and Richard E. Woods. *Digital image processing*. Prentice Hall, 2008. ISBN: 9780131687288 013168728X 9780135052679 013505267X. URL: http://www.amazon.com/Digital-Image-Processing-3rd-Edition/dp/013168728X.

[57] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[58] Salman Khan et al. *A Guide to Convolutional Neural Networks for Computer Vision*. 2018.

[59] Jorge J. Moré. "The Levenberg-Marquardt algorithm: Implementation and theory". In: *Numerical Analysis*. Ed. by G. A. Watson. Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, pp. 105–116. ISBN: 978-3-540-35972-2.

[60] Young-Jin Cha, Wooram Choi, and Oral Buyukozturk. "Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks". In: *Computer-Aided Civil and Infrastructure Engineering* 32 (Mar. 2017), pp. 361–378. DOI: 10.1111/mice.12263.

[61] Khin Yadanar Win et al. "Hybrid Learning of Hand-Crafted and Deep-Activated Features Using Particle Swarm Optimization and Optimized Support Vector Machine for Tuberculosis Screening". In: *Applied Sciences* 10.17 (2020). ISSN: 2076-3417. DOI: 10.3390/app10175749. URL: https://www.mdpi.com/2076-3417/10/17/5749.

[62] Waleed Tahir, Aamir Majeed, and Tauseef Rehman. "Indoor/Outdoor Image Classification Using GIST Image Features and Neural Network Classifiers". In: *12th International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies (HONET)*. Dec. 2015. DOI: 10.1109/HONET.2015.7395428.

[63] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV].