



**MULTISPECTRAL IMAGING WITH SWIR CAMERA FOR MATERIAL
DETECTION IN DEEP LEARNING APPLICATIONS**

Selahattin Eren SÖZEN

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE
IN ELECTRICAL – ELECTRONIC ENGINEERING DEPARTMENT**

**GAZİ UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

APRIL 2025

ETHICAL STATEMENT

I at this moment declare that in this thesis study, I prepared the thesis writing rules of Gazi University Graduate School of Natural and Applied Sciences;

- All data, information, and documents presented in this thesis have been obtained within the scope of academic rules and ethical conduct,
 - All information, documents, assessments, and results have been presented by scientific ethical conduct and moral rules,
 - All material used in this thesis that is not original to this work has been exhaustively cited and referenced,
 - No change has been made in the data used,
 - The work presented in this thesis is original,
- or else, I admit all loss of rights to be incurred against me.

Selahattin Eren SÖZEN

29/04/2025

MULTISPECTRAL IMAGING WITH SWIR CAMERA FOR MATERIAL DETECTION IN DEEP LEARNING APPLICATIONS

(M.Sc. Thesis)

Selahattin Eren SÖZEN

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

April 2025

ABSTRACT

Today, the use and purpose of imaging systems have become increasingly important in recent years due to their unique capabilities. With the advancement of modern technology, specialized imaging systems tailored to various application needs have diversified, leading to a significant increase in related research and practical implementations. Various imaging techniques beyond the visible spectrum, including IR, UV, thermal, and gamma imaging, have emerged as powerful tools for capturing material characteristics that are not observable through conventional visual methods. Recently, the integration of deep learning and machine learning approaches into these imaging systems has aimed to minimize operational errors and achieve more accurate outcomes. This study focuses on the classification of liquid materials that are visually indistinguishable and highly similar in appearance. Convolutional Neural Networks (CNNs), one of the most widely used deep learning-based image analysis models, were employed for this task. The materials targeted for classification included water, propanol (alcohol), acetone, cologne, and flux. Images of these substances were captured using both Short-Wave Infrared (SWIR) and visible region (RGB) cameras, resulting in a custom dataset comprising multispectral, SWIR, and RGB images. In the five-class classification scenario, the test accuracy obtained from RGB camera images was 34.16%, while SWIR camera images yielded a test accuracy of 59.96%. When multispectral images were captured using filters with cutoff wavelengths at 1200 nm, 1300 nm, 1400 nm, 1500 nm, and 1600 nm, the classification performance significantly increased, reaching a test accuracy of 96.46%. In addition to accuracy, the F1-score was also considered, as it reflects both the precision of positive predictions and the overall balance of the model. The F1-scores were recorded as 32.72% for RGB images, 41.62% for SWIR images, and 89.09% for multispectral images. As a result, this study demonstrated that a deep learning-based classification model can successfully distinguish between visually similar liquid substances. It was shown that multispectral imaging notably enhances inter-class discriminability, providing significantly improved classification performance compared to standard RGB and SWIR imaging techniques.

Science Code : 90521

Key Words : Multispectral Imaging (MS), Short-Wave Infrared Camera (SWIR),
Deep Learning, Convolutional Neural Network (CNN)

Page Number : 79

Supervisor : Prof. Dr. Tuğba Selcen NAVRUZ

DERİN ÖĞRENME UYGULAMALARINDA MATERYAL TESPİTİ İÇİN SWIR KAMERA İLE MULTİSPEKTRAL GÖRÜNTÜLEME

(Yüksek Lisans Tezi)

Selahattin Eren SÖZEN

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Nisan 2025

ÖZET

Günümüzde görüntüleme sistemlerinin kullanımı ve amacı, sahip oldukları özel kabiliyetler sayesinde her geçen gün daha da önem kazanmaktadır. Modern teknolojiye, uygulama ihtiyaçlarına göre çeşitlenen görüntüleme sistemleri ile gerçekleştirilen çalışmalar da hızla artmakta ve çeşitlenmektedir. Görünür bölge görüntüleme dışında; kızılötesi (IR), gama, termal ve ultraviyole (UV) gibi farklı spektral bantlarda çalışan özel görüntüleme yöntemleri de bu alanda önemli yer tutmaktadır. Son yıllarda, uygulama hatalarını en aza indirmeyi hedefleyen derin öğrenme ve makine öğrenmesi yöntemlerinin, bu tür görüntüleme sistemlerine entegre edilmesiyle daha başarılı sonuçlar elde edilmeye başlanmıştır. Bu çalışmada ise, görsel olarak ayırt edilmesi güç ve birbirine oldukça benzeyen sıvı malzemelerin sınıflandırılması hedeflenmiştir. Derin öğrenme alanında yaygın olarak kullanılan görüntü tabanlı modellerden olan Evrişimli Sinir Ağı (CNN) mimarisi temel alınmıştır. Ayırt edilmesi hedeflenen malzemeler; su, propanol (alkol), aseton, kolonya ve flux olarak belirlenmiş ve bu maddelere ait görüntüler, hem Kısa Dalga Kızılötesi (SWIR) hem de görünür bölge (RGB) kameraları kullanılarak alınmıştır. Böylece, multispektral, SWIR ve RGB görüntülerden oluşan özel bir veri seti oluşturulmuştur. Beşli sınıflandırma senaryosunda, RGB kamera görüntüleri ile elde edilen test doğruluğu %34,16; SWIR görüntüleriyle %59,96 olarak belirlenmiştir. SWIR kamera ile 1200 nm, 1300 nm, 1400 nm, 1500 nm ve 1600 nm dalga boylarında alınan multispektral görüntülerle yapılan sınıflandırmada ise test doğruluğu %96,46'ya ulaşmıştır. Sadece doğruluk oranı değil, aynı zamanda sınıflar arası pozitif tahmin başarısını ve modelin genel dengesini yansıttığı için F1 skoru da değerlendirmeye dahil edilmiştir. RGB görüntüler için %32,72 olan F1 skoru, SWIR görüntülerinde %41,62 ve multispektral görüntülerde %89,09 olarak kaydedilmiştir. Sonuç olarak, bu çalışma kapsamında derin öğrenme temelli bir sınıflandırma modeli ile, gözle ayırt edilmesi güç malzemelerin tespiti hedeflenmiş; multispektral görüntüleme yönteminin, sınıflar arası ayırt ediciliği artırarak daha başarılı sonuçlar sunduğu ortaya konmuştur.

Bilim Kodu : 90521

Anahtar Kelimeler : Multispektral görüntüleme, kısa dalga kızılötesi kamera, derin öğrenme, evrişimli sinir ağı

Sayfa Adedi : 79

Danışman : Prof. Dr. Tuğba Selcen NAVRUZ

ACKNOWLEDGEMENTS

I would like to sincerely thank my advisor, Prof. Dr. Tuğba Selcen NAVRUZ, for her invaluable guidance, support, and patience throughout the course of this research.

My heartfelt thanks also go to Saime Sena SÖZEN, whose encouragement, motivation, and contributions have been deeply appreciated during the completion of this study.

I am grateful to Assoc. Prof. Dr. Muharrem Tuncay GENÇOĞLU and Prof. Dr. Muhsin Tunay GENÇOĞLU for sparking my initial interest in this field, and to Research Assistant Serhat KARAÇAM for his helpful insights and technical support throughout the process.

I would also like to acknowledge Prof. Dr. Şeref SAĞIROĞLU and Prof. Dr. Hasan Şakir BİLGE for their inspiration and encouragement.

I extend my gratitude to my mother, father, and sister, whose unwavering belief in me, along with their patience and emotional support, have been a valuable source of strength.

Special thanks are also due to Prof. Dr. Tayfun AKIN and the MIKRO-TASARIM family, including to Serhat KOÇAK, Çetin ÇEVRÜK, Murat IŞIKHAN, Pınar DÖNMEZ, Mehmet AKDEMİR, Vedat ÖZKAN, Hakan ÇEÇEN, and Zehra CAN, for friendship and ongoing support during this study.

I am especially thankful to Cengiz DEDE and Noyan DEDE, as well as to my colleagues at ONUR YÜKSEK TEKNOLOJİ, namely Alper BOTAN, Kadir TORUN, Semih AGUŞ, Serhat AGUŞ, Ezgi TABUR, Kübra Nur GÖKBULUT, Hakan COŞGUNARAS, Mehmet Burak KARABULUT, and Eren ÖZGÜ, for their valuable support, motivation, and collaboration throughout this work.

Lastly, I am truly thankful to my dear friends Aycan BEYENİR, Berk ENGİN, Oğuzhan SEZGİN, İsmail CİFCİ, and Ufuk ERCAN, for their sincere friendship and encouragement during this academic endeavor.

CONTENTS

	Page
ABSTRACT.....	iv
ÖZET	v
ACKNOWLEDGEMENTS.....	vii
CONTENTS.....	viii
LIST OF TABLES.....	x
LIST OF FIGURES	xi
SYMBOLS AND ABBREVIATIONS.....	xiii
1. INTRODUCTION.....	1
2. LITERATURE OVERVIEW	5
3. IMAGING SYSTEMS	9
3.1. Visible (RGB) Imaging Systems	10
3.2. Infrared (IR) Imaging Systems.....	10
3.2.1. Near infrared characteristic and technology.....	11
3.2.2. Short wave infrared characteristic and technology.....	11
3.2.3. Mid wave infrared characteristic and technology	13
3.2.4. Long wave infrared characteristic and technology	14
3.3. Multispectral Imaging (MS) Systems.....	15
3.4. Hyperspectral Imaging (HS) Systems	17
4. LEARNING ARCHITECTURE PERSPECTIVE	19
4.1. Deep Learning Fundamentals	20
4.2. Deep Learning Methods	22
4.2.1. Artificial neural networks.....	22
4.2.2. Activation functions	28

	Page
4.2.3. Regularization approaches.....	32
4.2.4. Loss functions	34
4.2.5. Optimization algorithms.....	36
4.3. Common Artificial Intelligence Terms	38
5. AUGMENTED DATA AND METHODS	41
6. EXPERIMENTAL EVALUATIONS	45
6.1. Data Acquisition.....	46
6.1.1. Environment variables	48
6.1.2. Input standardization	49
6.2. Architecture Selection	51
6.2.1. Model decision	51
6.2.2. Classification metrics	59
6.2.3. Performance visualization	65
6.3. Additional Applied Techniques.....	67
6.3.1. K-Fold cross validation.....	67
6.3.2. Data merge application	69
7. CONCLUSION AND FUTURE WORK	71
REFERENCES	73
CURRICULUM VITAE	79

LIST OF TABLES

Table	Page
Table 4.1. Feature differences between VGG, Inception, ResNet, and DenseNet architectures	27
Table 5.1. Amount of augmented MS, SWIR, RGB classes.....	42
Table 5.2. Data augmentation methods and parameters.....	43
Table 6.1. Technical specification comparison of different imaging modes	49
Table 6.2. Pseudo-RGB conversion of SWIR and MS data	50
Table 6.3. MS-SET3 results of different functions for 5-class classification	54
Table 6.4. General model parameter details for MS, SWIR, RGB classifications.....	55
Table 6.5. Comparison of MS, SWIR, and RGB results without data augmentation	55
Table 6.6. Augmented MS, SWIR and RGB data for 5-class classification.....	56
Table 6.7. Test accuracies in different multiclass classification scenarios	58
Table 6.8. Classification performance metrics for MS-SET1 confusion matrix	60
Table 6.9. Classification performance metrics for MS-SET3 confusion matrix	61
Table 6.10. Classification performance metrics for SWIR-SET3 confusion matrix.....	63
Table 6.11. Classification performance metrics for RGB-SET3 confusion matrix.....	64
Table 6.12. 5-fold cross validation results for MS-SET3	68
Table 6.13. Merged image dataset results	69

LIST OF FIGURES

Figure	Page
Figure 3.1. Electromagnetic spectrum views	9
Figure 3.2. Fundamentals of RGB camera operation	10
Figure 3.3. Comparison of visible light (left) and SWIR imaging (right) in low-visibility conditions. The SWIR image demonstrates to penetrate atmospheric interference [9].....	11
Figure 3.4. Packaged SWIR detector (left) and electronic boards (right) [42]	12
Figure 3.5. Temperature change over time of the SWIR camera used in this study.....	13
Figure 3.6. Comparison of images. a) MWIR image b) SWIR image [44]	14
Figure 3.7. Thermal image captured using an LWIR camera, illustrating temperature variations across different materials [46]	14
Figure 3.8. Multispectral images captured at 1200 nm, 1300 nm, 1400 nm and 1500 nm wavelengths imaging.....	15
Figure 3.9. Comparison of the MS images. a) blue, b) green, c) red visible bands, d-f) infrared bands, g) full-spectrum reflectance, and h) the original image of a plant [48]	16
Figure 4.1. Hierarchical relationship of Artificial Intelligence and subfields.....	19
Figure 4.2. Fundamental artificial neuron model.....	20
Figure 4.3. Feedforward neural network between layers [52]	21
Figure 4.4. Backpropagation neural network between layers	22
Figure 4.5. Architecture of a Convolutional Neural Network for image classification..	24
Figure 4.6. Max pooling operation for dimensionality reduction	25
Figure 4.7. Flattening a feature map into one-dimensional vector	25
Figure 4.8. DenseNet architecture with dense-blocks and transition layers [79].....	27
Figure 4.9. SoftMax activation function graphs [60]	29
Figure 4.10. Sigmoid activation function graphs [60]	29
Figure 4.11. ReLU activation function graphs [60]	30
Figure 4.12. LReLU activation function graphs [60].....	31

Figure	Page
Figure 4.13. Tanh activation function graphs [60].....	31
Figure 5.1. Distribution of original datasets for RGB, SWIR, and MS Imaging.....	42
Figure 5.2. Visual appearance of augmented test liquid data examples	44
Figure 6.1. Workflow for processing deep learning methods on experimental setup.....	45
Figure 6.2. Visual appearance of selected test liquids in RGB.....	46
Figure 6.3. Comparison of the SWIR test images.....	47
Figure 6.4. Multispectral test images at different wavelengths	47
Figure 6.5. Experimental test setup and environments. a) test setup b) RGB, SWIR, and MS imaging environments.....	48
Figure 6.6. Class prediction evaluation of input image	52
Figure 6.7. Layer structure and parameters of the DenseNet based model	57
Figure 6.8. Confusion matrix for MS-SET1 test and validation datasets	59
Figure 6.9. Confusion matrix for MS-SET3 test and validation datasets	60
Figure 6.10. Confusion matrix for MS-SET3 validation and training datasets	62
Figure 6.11. Confusion matrix for SWIR-SET3 test and validation datasets.....	63
Figure 6.12. Confusion matrix for RGB-SET3 test and validation datasets.....	64
Figure 6.13. Multispectral imaging (SET3) accuracy (left) and loss (right) outputs for 5-class DenseNet – Nadam classification at 100 epochs	65
Figure 6.14. Multispectral imaging (SET3) accuracy (left) and loss (right) outputs for 5-class DenseNet – Nadam classification at 50 epochs	66
Figure 6.15. SWIR imaging (SET3) accuracy (left) and loss (right) outputs for 5- class DenseNet – Nadam classification at 100 epochs	66
Figure 6.16. RGB imaging (SET3) accuracy (left) and loss (right) outputs for 5- class DenseNet – Nadam classification at 100 epochs	67
Figure 6.17. Structure of 5-Fold cross validation and performance averaging.....	69

SYMBOLS AND ABBREVIATIONS

The symbols and abbreviations used in this study are presented below, along with their explanations.

Symbols	Explanations
mm	Millimeter
μm	Micrometer
nm	Nanometer
w	Weights
b	Bias model
α	Learning rate
γ	Scale parameter
β	Shift parameter
p	Dropout rate
λ	Regularization
μ	Average
σ	Sigmoid
σ^2	Variance
θ	(Stochastic) gradient decent
z_i	Neuron activation
g_t	Time step gradient
n	Initial learning rate
ϵ	Numeric constant

Abbreviations	Explanations
Adagrad	Adaptive Gradient
Adam	Adaptive Moment Estimation
AI	Artificial Intelligence
ANN	Artificial Neural Networks
BCE	Binary Cross-Entropy
CCE	Categorical Cross-Entropy

Abbreviations**Explanations**

CNN	Convolutional Neural Network
DenseNet	Dense Convolutional Network
DL	Deep Learning
ELU	Exponential Linear Unit
FNN	Feedforward Neural Network
GD	Gradient Descent
HS	Hyperspectral Imaging
IR	Infrared Imaging
Lasso	L1 Regularization
LeakyReLU	Leaky Rectified Linear Unit
LSTM	Long Short-Term Memory
LWIR	Long-Wave Infrared
MAE	Mean Absolute Error
MSE	Mean Squared Error
ML	Machine Learning
MS	Multispectral Imaging
MWIR	Mid-Wave Infrared
PReLU	Parametric Rectified Linear Unit
ReLU	Rectified Linear Unit
ResNet	Residual Neural Network
RGB	Red-Green-Blue Imaging
Ridge	L2 Regularization
RMSProp	Root Mean Square Propagation
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SNR	Signal-to-Noise Ratio
SPE	Steps Per Epoch
SWIR	Short-Wave Infrared
Tanh	Hyperbolic Tangent
VGG	Visual Geometry Group
VIS	Visible
VNIR/NIR	Visible Near-Infrared

1. INTRODUCTION

In daily life, we interact with our environment through our sense organs. One of the most important of these senses is our ability to see. Our eyes perform the visual function and consist of various layers such as the iris and cornea. Most imaging systems are modeled like the eye and consist of various layers. The working range of the human eye is approximately between 380-740nm. The human eye, which has this limited working range, is not suitable for working at different wavelengths in our environment or in some cases does not have the ability to distinguish. For this reason, imaging systems capable of working at different wavelengths have been developed. Systems with this ability, with appropriate filters and processes, are used in various fields and for different purposes.

Imaging systems have so many different application areas in modern life such as industrial, military, consumer, and medical applications [1]. Imaging methods cover certain regions of the spectral band. The visible spectral band covers between about 400 nm and 800 nm. In another study [2], a short-wavelength camera operating between approximately 900 nm and 2500 nm was used. Studies have been conducted that it is a method that can be used for clinical approaches in the field of medicine. The SWIR imaging technique was used in imaging the mouse head.

Each radiation region has its unique characteristics. Camera systems have special sensor technologies that utilize from characteristics properties of radiation. For instance, thermal cameras can detect heat radiation, or VNIR (Visible Near Infrared) cameras can be operated at the low-level light. SWIR cameras capable of capturing images in foggy weather [3], detecting rot or decay in the food industry, and identifying radiation at high temperatures include VNIR and LWIR (Long-Wave Infrared) camera features. Imaging systems used for surveillance, detection, observation, security, recognition, and identification may fail to fulfill their intended purpose in some cases. For instance, visually similar materials may not be distinguishable using visible imaging systems. Additionally, IR band imaging systems may sometimes be insufficient for distinguishing certain objects. Therefore, newer and more advanced imaging methods are being explored in ongoing research [4,5].

All objects have special spectral characteristics at different wavelengths. The multispectral imaging technique is based on the principle of obtaining discrete images at different wavelengths [6]. Feature extraction can be done on multispectral images by using filters. To distinguish the materials whose visual spectral characteristics under visible light are similar [7] multispectral imaging is performed, generally using the VIS/NIR band [8]. However, in some conditions, this bandwidth is not enough for the application. The SWIR wavelength provides a larger coverage area than VIS/NIR wavelength. Hence it is more practical for object detection applications. Deep learning algorithms are applied to SWIR camera images for many applications such as classification and surveillance [9-10]. The use of CNN based architecture is especially common in deep learning applications [11-12]. There are different CNN models with updated last versions, such as Visual Geometry Group (VGG) [13], Densely Connected Convolutional Networks (Dense-Net) and Residual Networks (Res-Net) [14].

There are several parameters used in deep learning algorithms such as dropout rate, activation function, epoch number, learning rate, loss function, data augmentation, normalization, and optimizer [15]. These parameters should be fine-tuned experimentally for the architecture. Dropout rate is a method that avoids overfitting problems. SoftMax, Rectified Linear Unit (ReLU), LeakyReLU, Hyperbolic Tangent, and Sigmoid. ReLU and LeakyReLU are generally preferred activation functions in deep learning models [16] because of their ability and efficiency to avoid the vanishing gradient problem. Especially LeakyReLU is similar to ReLU but LeakyReLU solves ReLU's dying neuron issue. This situation allows the use of information-carrying neurons, especially complex datasets. The epoch number is used as the number of repetitions in the training dataset. The learning rate determines the speed of the model weights. A small learning rate provides slower reaction but more qualified learning, while a larger one enables faster but more oscillatory learning. Loss function is a technique that measures the error between the predicted values and the actual values. Cross-entropy loss is used for classification tasks, and this method was preferred in this study. Data augmentation (standard augmentation) increases data variety by implementing transformations such as rotation, zoom, and shifting [17]. Normalization is the process of scaling data. This stabilizes the model and helps it learn more efficiently. An optimizer function is an algorithm that minimizes loss value for learning.

There are various optimization approaches, such as ADAM (Adaptive Moment Estimation), Gradient Descent (GD), RMSprop (Root Mean Squared Propagation), Stochastic Gradient Descent (SGD), and AdaGrad (Adaptive Gradient Algorithm) [18]. Choice of optimizer depends on the certain preferences and situations [19]. ADAM is a widely used, popular, and effective optimizer for deep learning applications [20]. Both LeakyReLU and ADAM are well-suited for deep learning tasks. The most common performance parameters in deep learning applications are test accuracy, test loss, validation accuracy, validation loss, and F1 score [21,22]. Test accuracy shows the accuracy of the model on the test dataset, meaning how accurately the model predicts new, unseen data. High test accuracy suggests that the model performs well on new data. Test loss measures the error rate of the model on the test dataset. If test loss has a low value, this indicates good generalization. Test loss is also a metric that calculates difference between predicted values and actual classes, depending on loss function used. Validation accuracy shows the result of the validation dataset during training. It helps identify overfitting or underfitting, and validation accuracy is used for model performance. Overfitting occurs when the model fits the training data, learning noise and details specific to the training set, and underfitting is the failure of the training data's underlying patterns. Validation loss is the error rate of the model on the validation dataset. The F1 score is used to measure the model classification performance. F1 score balances the accuracy metric, which is especially helpful for imbalanced datasets [23].

Traditional methods for separating liquids typically rely on physical and chemical properties and are often performed manually in laboratory environments. Techniques such as distillation, solvent extraction, centrifugation, and filtration are commonly used but tend to be time-consuming and labor-intensive. In contrast, this study introduces a novel and contact-free approach that integrates multispectral imaging inspired by SWIR camera technology with deep learning algorithms for liquid classification. Visually similar substances such as water, flux, acetone, alcohol, and cologne were selected to challenge standard RGB imaging, which often fails to distinguish such materials. Multispectral (MS), SWIR, and RGB images were compared under identical visual conditions using filters in the 1200–1600 nm range to capture MS images. Both binary (2-class) and multiclass (5-class) classifications were performed, and results were evaluated comparatively. CNN-based deep learning models such as DenseNet, VGG16, VGG19, ResNet, and Inception were utilized alongside various optimizers including SGD, RMSProp, Nadam, and Adagrad. Experimental findings demonstrate that the proposed MS imaging approach, when combined

with deep learning, yields significantly better classification performance than RGB-based methods. This study aims to demonstrate that multispectral imaging, when combined with deep learning, offers a robust and scalable solution for the non-invasive classification of visually similar liquids, potentially reducing the need for complex laboratory procedures.

Chapter 1 introduces the purpose and scope of the study, highlighting the goal of distinguishing visually similar substances using multispectral (MS) imaging and deep learning techniques.

Chapter 2 provides a comprehensive literature review on imaging systems (RGB, SWIR, MS) and deep learning algorithms.

Chapter 3 details various imaging technologies, focusing on the characteristics and principles of visible (RGB), infrared (SWIR), and multispectral (MS) systems.

Chapter 4 presents the fundamentals of deep learning, loss functions, neural networks, optimization algorithms, activation functions, and commonly used architectures such as DenseNet, VGG16, VGG19, ResNet, and Inception.

Chapter 5 describes the dataset preparation and methodology, including how the data were collected, preprocessed, and augmented for use in training.

Chapter 6 includes experimental evaluations. Different CNN architectures and optimization algorithms (Adam, SGD, RMSProp, Nadam, Adagrad) are tested. Binary and multiclass classification results are compared across RGB, SWIR, and MS modalities, showing that multispectral imaging yields the best results.

Chapter 7 concludes this study with key findings and proposes directions for future research.

2. LITERATURE OVERVIEW

Multispectral imaging captures image data at certain wavelength bands. Each band captures distinct information, making this method suitable for applications like agriculture, environmental monitoring, and medical imaging [24,25]. Multispectral imaging offers efficient results with lower costs and reduced complexity and provides a more practical option for operational applications [26]. Infrared imaging systems have their origins in recent history. Infrared imaging systems encompass the latest technological developments, and the applications of infrared sensors have become widespread in industrial, medical, and various imaging fields [27,28,29]. However, SWIR cameras have a slightly broader range of applications in multispectral imaging due to their effective bandwidth and wide array of spectral features [1,6]. SWIR cameras operate in the 900–2500 nm range when capturing a broader spectrum, covering both the traditional SWIR range (900–1700 nm) and extending into longer wavelengths. Therefore, SWIR cameras are widely preferred for many multispectral applications, and imaging performance is further enhanced by combining the advantages of multispectral and SWIR imaging [4]. Their ability to excel in low-light and foggy conditions, as well as to detect subtle material differences, makes them indispensable in fields such as agriculture, material analysis, and environmental monitoring. Combining multispectral imaging with deep learning applications offers several powerful benefits, especially in fields requiring complex data analysis and detailed object detection [30]. Multispectral imaging, utilizing SWIR, provides a broader spectral range than standard visible light, allowing deep learning algorithms to capture subtle details across multiple SWIR wavelengths [9].

In another study [10], spectral differences between lettuce and weeds were analyzed using multispectral images. Deep learning models were trained to detect weeds by processing these images. The paper describes in detail the datasets, model architecture and training methods used in this process. A deep learning model that automatically detects water bodies using multispectral satellite images taken at different wavelengths was developed. The model aimed to distinguish water bodies in images with high accuracy using a convolutional neural network (CNN) based architectural approach [11]. The use of CNN based architecture is especially common in deep learning applications.

Deep learning, as a subfield of machine learning, utilizes multi-layered neural networks to analyze large volumes of data. Deep learning architecture learns features to get high accuracy in complex missions such as image recognition, speech processing, and language processing from the collected data [31]. A review of the literature on neural networks revealed the existence of various architectures with unique features designed for problems such as image, sound, text, and other data types. In this study, the focus was placed on CNN, which offer advantages in applications where spatial features are important, such as visual processing, image classification, and object recognition. CNN architecture was particularly preferred due to its appropriate structure and widespread use in image processing [32]. The CNN-based deep learning method produces outputs based on adjustable parameters in its architecture. Critical elements such as learning rate, epoch, and dropout directly influence the performance, generalization, and learning process for deep learning models. Consequently, the model can achieve enhanced performance, improved generalization ability, greater training efficiency, and effective data utilization [33]. In the literature, deep learning applications are used for identifying similar objects and detecting changes in images where the visible spectrum is limited, as well as for multispectral imaging applications. In a study [34], the separation of objects that are visually very similar, or indistinguishable to the human eye, using a deep learning method applied to multispectral imaging obtained with SWIR cameras is discussed. An attempt was made to leverage the bandwidth of the infrared camera and to separate the materials by incorporating deep learning into this study. This method facilitates the precise classification of materials through spectral signature analysis and can be seamlessly integrated with other imaging methods such as RGB or IR.

In the study [35], a deep learning application was conducted to the NIR band on datasets belonging to 9 different classes. Deep learning models for food detection was examined. The test accuracy and F1 score output were explained using the Dense-Net model, both with and without transfer learning. The highest accuracy of 63% was achieved by mixed datasets, but then 96.46% accuracy was achieved by using multispectral datasets collected at five distinct wavelengths. With the data augmentation method, the accuracy results increased from 63% to around 74% and the effect of data augmentation was emphasized. In the conclusion section, the importance of multispectral imaging, the applicability of deep learning methods and the conditions affecting performance were mentioned. In [36], a deep learning model was developed to classify different wound types by using RGB images. The model output was evaluated by utilizing the cross-validation method. A maximum accuracy rate of 96.4%

and an average accuracy rate of 94.28% were achieved in 2-class classification. For 3-class classification, the model achieved a maximum accuracy rate of 91.9% and an average accuracy rate of 87.7%. The test accuracy value was found to be 92.20% for 4-class classification and 84.94% for 5-class classification. These findings highlight that the model also performs well on different datasets.

Another study [37] reviewed 170 articles on the deep learning research status. Both multispectral and hyperspectral images are considered in the article. These images are combined with the deep learning method and the work in 170 different articles is examined and the best results are tried for different class numbers and classification types. Satellite imagery includes satellites such as Landsat 5 and 8, GF-1, Sentinel-1. Most of these satellites produce multispectral imagery. In particular, Sentinel-2 provides multispectral imagery with 13 different spectral bands. Situation is stated in the article that 90% accuracy was achieved in the study conducted with Sentinel-2 for 5 different classes based on CNN. In addition, 91.2% accuracy was found for 3 classes. The best results were found as 93.3% using Unet and ResNet duo for 10 classes.

In the study [38], recognition and classification operations were performed on wheat seed images obtained using three different imaging techniques. Fusion imaging achieved the highest accuracy of 92.63% in RGB-SWIR combinations. Additionally, a success rate of 91.13% was reported in tests conducted with the VGG16 model. Within the scope of the study, recognition and classification studies were carried out on the dataset created from wheat images obtained from RGB, VNIR, and SWIR imaging techniques. Using SWIR camera data, an accuracy of 95.31% was obtained. Findings indicate that the fusion technique may not always be sufficient in deep learning applications. As a potential solution, the multispectral imaging technique presents a promising new opportunity.

In a study [39] aiming to detect counterfeiting with spectroscopy-based sensors, meat pieces were mixed in 25% increments to create different levels of counterfeiting mixtures and six different samples were prepared from each level. They were analyzed using spectroscopy and multispectral imaging. The good performance of MSI-based models compared to other sensors was highlighted, with accuracy rates ranging from 87% to 100%. In another article [40], dataset features and testing techniques affecting the performance of classification algorithms were examined in detail. In the study, experiments were conducted using 32

different datasets and 9 different classification algorithms commonly used in the literature. For example, it was shown how effective the physical class change was for the Biomechanical Properties of Orthopedic Patients class.

In this study utilized deep learning algorithms to analyze five distinct materials. Five different classes were created, and data for these classes were collected in a laboratory environment. Datasets were obtained using SWIR cameras and filters of different wavelengths. Then, these data were processed using deep learning methods, and the results were recorded. An ideal architecture was determined by analyzing these results. 2-class, and 5-class classifications were examined. Study began with data obtained from RGB, SWIR, and multispectral images for 5-class classification. The highest 5-class test accuracy of 96.46% was obtained using multispectral imaging, compared to 59.96% with SWIR and 44.16% with RGB. Metrics such as test loss, validation accuracy, F1 score, and validation loss, which influence the deep learning method, were also examined. In this study, class differences in 5-class multispectral imaging were also analyzed, and the results were validated accordingly. A slight increase in test accuracy was observed as the number of classes was reduced.

3. IMAGING SYSTEMS

In this section, visible, infrared, multispectral, and hyperspectral imaging techniques are explained. SWIR and multispectral imaging techniques, which are the focus of this study, will be discussed in detail. Imaging systems are technological tools designed to capture, analyze, and process visual data from specific objects, areas, or environments. These systems typically utilize sensors or cameras to detect light or other electromagnetic waves and convert this information into a digital format for further analysis. In this section, the types of imaging systems, sensors, and optics are explained. The study also includes an explanation of the data collection methodology. The Electromagnetic spectrum encompasses electromagnetic radiation range, categorized by energy, wavelength, and frequency. It begins with radio waves and ends with gamma rays, encompassing diverse energy levels. Electromagnetic spectrum is given in Figure 3.1.

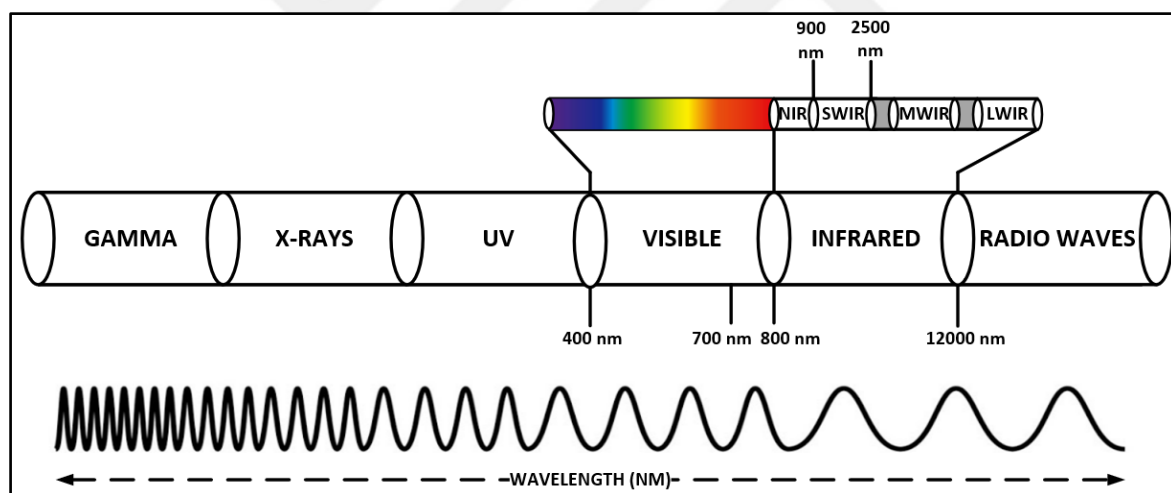


Figure 3.1. Electromagnetic spectrum views

There are grey areas in the infrared region represent atmospheric attenuation due to gases, water vapor and other atmospheric situations. The boundaries in spectral analysis are not rigidly defined and should be considered flexible. For instance, 1000nm can belong to both the NIR band and the SWIR band. Likewise, the visible spectrum does not have a sharp cutoff, and it cannot be definitively said to end at 700nm [38]. The four primary types of imaging systems are RGB imaging, multispectral imaging, IR imaging, and hyperspectral imaging.

3.1. Visible (RGB) Imaging Systems

The RGB imaging method, widely used in daily life, is an imaging technology that operates within the visible light spectrum. The RGB imaging method captures data from the three main colors perceivable by the human eye (red, green, blue) and combines these color channels to create a full-color image. The RGB spectrum spans approximately 400 nm to 700 nm, corresponding to the range perceivable by the human eye. The working principle of the RGB camera is illustrated in Figure 3.2. An image captured in analog form is converted into digital format and processed further to produce an output [39,40].

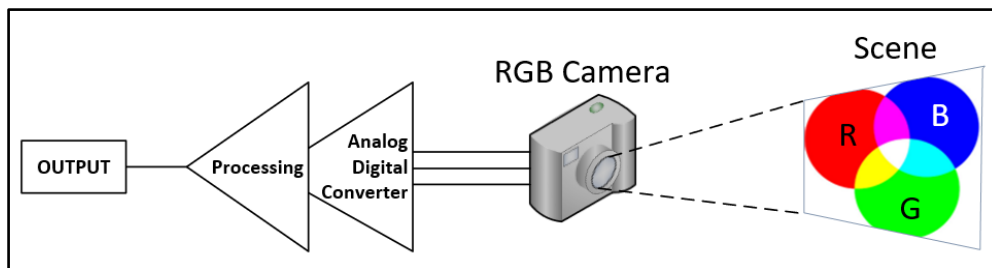


Figure 3.2. Fundamentals of RGB camera operation

RGB imaging systems, which is frequently used in our daily lives, offers many advantages but also has certain limitations. It is limited in its ability to capture wavelengths outside the visible light spectrum. Therefore, alternative imaging methods are employed to overcome these limitations.

3.2. Infrared (IR) Imaging Systems

Infrared (IR) imaging systems have a technology designed to detect infrared light and convert it into an image format. Infrared light, which cannot be seen by human eye, occupies the region of the electromagnetic spectrum beyond visible light, ranging approximately from 700 nanometers to 12 millimeters. Infrared camera systems are classified into four distinct bands such as VNIR/NIR (Visible Near Infrared), MWIR (Mid-Wave Infrared), LWIR (Long-Wave Infrared), and SWIR (Short-Wave Infrared). In this section, the characteristics of the SWIR band and the technology in SWIR band cameras are detailed.

3.2.1. Near infrared characteristic and technology

The NIR (Near-Infrared) band which lies within the electromagnetic spectrum ranging between 700 nm and 1000 nm exists just beyond visible light and before the SWIR (Short-Wave Infrared) region. Absorption and reflection properties in this band are sensitive for distinguishing material characteristics. Infrared light penetrates deeper than visible light and is more effective for environmental conditions.

3.2.2. Short wave infrared characteristic and technology

The SWIR (Short-Wave Infrared) band occupies a region of the electromagnetic spectrum, ranging approximately from 900 nm to 2500 nm, beyond the NIR (Near-Infrared) region and before the MWIR (Mid-Wave Infrared) region. Positioned with wavelengths that reflect light similar to visible light, SWIR provides unique imaging capabilities. Its absorption and reflection properties are effective for identifying subtle material differences and detecting parameters like water content, chemical composition, and surface properties.

SWIR is capable of penetrating through atmospheric elements such as haze, smoke, and fog, making it suitable for challenging environmental conditions. Images captured by SWIR and visible cameras under foggy weather conditions are presented, as shown in Figure 3.3.



Figure 3.3. Comparison of visible light (left) and SWIR imaging (right) in low-visibility conditions. The SWIR image demonstrates to penetrate atmospheric interference [9]

Figure 3.4 illustrates views of an InGaAs (Indium Gallium Arsenide) SWIR camera system. SWIR imaging method is used in applications for agriculture, surveillance, material analysis,

security, and industrial due to its ability to detect features beyond the visible and NIR bands. In this study, multispectral imaging leveraging SWIR technology was applied.

The production of SWIR cameras involves various processes, including ROIC (Read-Out Integrated Circuit) fabrication, flip-chip bonding, wire bonding, packaging, and electronic integration [43].

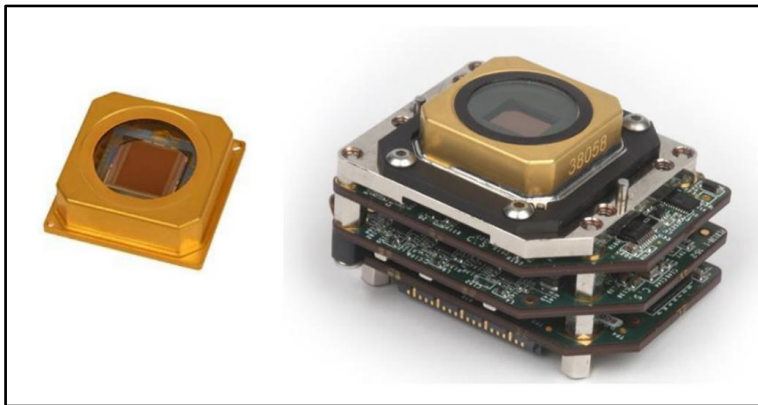


Figure 3.4. Packaged SWIR detector (left) and electronic boards (right) [42]

The ROIC is a key component that reads electrical signals from the photodetector array and processes them for output. Fabrication involves advanced semiconductor manufacturing techniques to ensure low noise and high performance in signal readout. Flip-Chip bonding is a process where the photodetector array is flipped and directly mounted onto the ROIC using precise alignment. This method minimizes signal loss and improves the electrical connection between the sensor and the circuit. In cases where flip-chip bonding is not used, wire bonding connects the sensor and ROIC through ultra-thin wires. It is a reliable and cost-effective technique for creating electrical connections.

Packaging which may also include the integration of optical filters or windows, depending on the application is a process of enclosing the SWIR sensor in a protective housing to shield it from environmental factors like moisture, dust, and mechanical stress. Electronic Integration involves connecting the SWIR sensor module to additional electronic components, such as proximity card, analog-to-digital converters (ADC), interface card and power supply systems, to enable seamless functionality in devices or systems.

Calibration is the process of ensuring that the sensor operates accurately within its specified wavelength range and performance parameters. Calibration involves adjusting the sensor's response to light intensity, wavelength, and other environmental factors to optimize accuracy and consistency. Calibration for a SWIR camera involves addressing key aspects such as correcting dead pixels, accounting for the quantum efficiency effect, and ensuring proper sensor cooling.

Temperature significantly impacts the performance of SWIR cameras in several ways. At higher temperatures, the dark current in SWIR detectors increases, which can reduce the Signal to Noise Ratio (SNR). Temperature changes may cause slight shifts in the spectral response or lead to a degradation in quantum efficiency. Maintaining stable temperature conditions is therefore important for optimal SWIR camera performance. Figure 3.5. indicates temperature change of the SWIR camera used in this study over time.

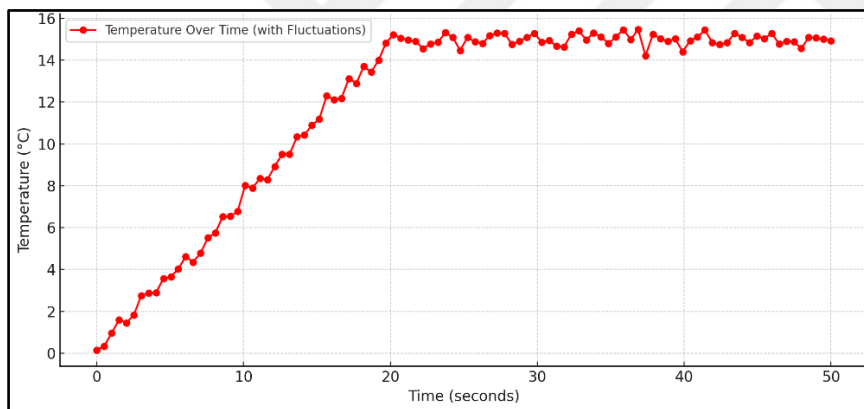


Figure 3.5. Temperature change over time of the SWIR camera used in this study

3.2.3. Mid wave infrared characteristic and technology

The MWIR (Mid-Wave Infrared) band occupies a region of the electromagnetic spectrum between 3 μm and 5 μm (3000 nm to 5000 nm). MWIR imaging is focusing on thermal emissions from objects, making it distinct from reflective SWIR imaging and longer wavelength LWIR thermal imaging. Materials like HgCdTe (MCT) and InSb (Indium Antimonide), which are used in MWIR detectors, involve more complex and expensive manufacturing processes. On the other hand, the InGaAs (Indium Gallium Arsenide) material used in SWIR sensors benefits from more widespread and mature production

technologies, providing a significant cost advantage. Figure 3.6 presents a visual comparison of images captured by MWIR and SWIR cameras.

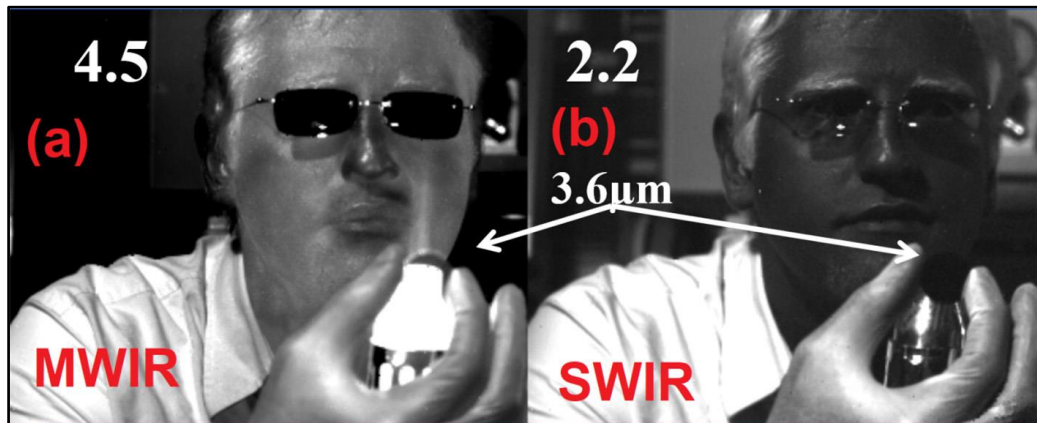


Figure 3.6. Comparison of images. a) MWIR image b) SWIR image [44]

3.2.4. Long wave infrared characteristic and technology

The LWIR (Long Wave Infrared) band covers a wavelength range between 8 μm and 12 μm (8000 nm to 12000 nm) in the electromagnetic spectrum, and is primarily associated with thermal radiation rather than reflected light. LWIR imaging focuses on capturing the heat naturally emitted by objects, which makes ideal for applications requiring temperature measurement and thermal imaging, for low-light conditions [45]. In Figure 3.7, images of different materials captured with an LWIR camera are presented.



Figure 3.7. Thermal image captured using an LWIR camera, illustrating temperature variations across different materials [46]

3.3. Multispectral Imaging (MS) Systems

Multispectral imaging method captures image data at certain wavelengths. Unlike traditional cameras that record images in three color channels, multispectral systems utilize multiple narrow spectral bands, often extending into non-visible regions. These systems are designed to extract detailed spectral information about objects or scenes, making them highly valuable for applications requiring precise material or feature differentiation. Multispectral imaging can be used effectively because it operates at discrete wavelengths within specific intervals [47]. This enables the differentiation of materials and objects based on their unique spectral signatures. It is widely applied in agriculture, defense, security, industrial quality control, remote sensing, and medical diagnostics, where accurate material characterization is critical. For instance, in agriculture, infrared bands can detect plant stress, water content, or chlorophyll concentration, which are invisible to the naked eye. Similarly, in material science, spectral data in non-visible ranges can help distinguish between materials based on their reflectance properties. Multispectral images of five different materials (water, propanol, cologne, acetone, flux) used in this thesis are presented in Figure 3.8.

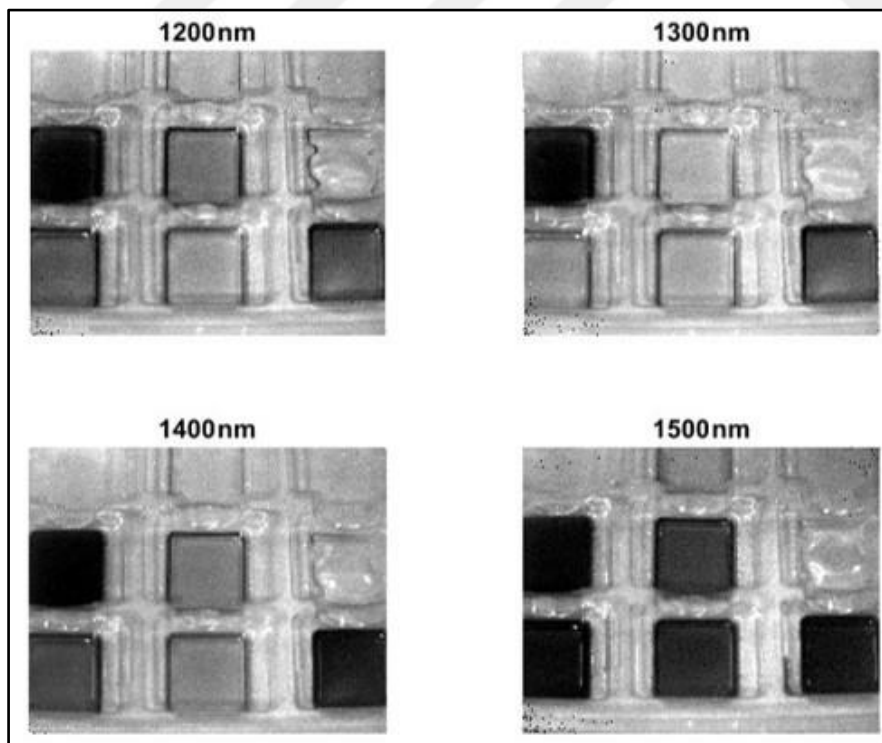


Figure 3.8. Multispectral images captured at 1200 nm, 1300 nm, 1400 nm and 1500 nm wavelengths imaging

Materials exhibit different responses at various wavelengths. These variations in spectral responses highlight the unique characteristics of each material, which can be leveraged for precise identification and analysis. This multispectral approach is particularly useful for distinguishing between materials based on their spectral signatures. A multispectral image comparison of a plant, as shown in Figure 3.9, demonstrates its utility in analyzing images captured within the 400 nm to 1000 nm range. Multispectral imaging is not limited to visible light; it also extends into infrared regions, enhancing its range of applications. By incorporating wavelengths such as NIR and SWIR, multispectral imaging systems become invaluable for remote sensing, healthcare diagnostics, and industrial quality control.



Figure 3.9. Comparison of the MS images. a) blue, b) green, c) red visible bands, d-f) infrared bands, g) full-spectrum reflectance, and h) the original image of a plant [48]

The ability to analyze data across diverse spectral bands makes multispectral imaging systems versatile tools in science, industry, and beyond. Due to their cost-effectiveness, faster data acquisition, and ease of integration, we prefer multispectral imaging for applications where detailed spectral resolution is not critical but efficiency and practicality are essential [46]. In multispectral imaging, light interacts with materials through reflection, absorption, and transmission, providing unique spectral signatures for each material.

Reflection occurs when light bounces off a material's surface, with the amount and type of reflection depending on the material's properties and the wavelength. Absorption happens when materials absorb specific wavelengths of light, converting them into energy such as heat. Transmission means that light passing through a material. Transparent materials like water or thin films may transmit some visible light while blocking other wavelengths, such as near-infrared. By analyzing these interactions, multispectral imaging generates unique spectral signatures that allow for precise material identification [49].

3.4. Hyperspectral Imaging (HS) Systems

Hyperspectral imaging systems capture image data across hundreds of narrow wavelength bands along the electromagnetic spectrum. Unlike traditional multispectral systems, hyperspectral systems utilize a much larger number of spectral bands, providing more comprehensive and detailed information about an object or scene. These systems analyze the spectral signature of each pixel, enabling highly precise material identification and feature detection.

Hyperspectral imaging operates in continuous and narrow wavelength ranges, allowing even the smallest spectral differences between materials and objects to be identified. This makes it critical in fields such as agriculture, environmental monitoring, defense, medical diagnostics, and remote sensing. Wide spectral coverage enables precise analysis regardless of environmental conditions.

Multispectral imaging has several advantages over hyperspectral imaging, particularly in terms of practicality and cost-efficiency. Another advantage is the faster data acquisition. Since multispectral imaging involves capturing fewer bands, it operates more quickly, enabling real-time imaging or faster processing in time-sensitive applications such as industrial inspections or remote sensing. Portability makes them ideal for fieldwork or applications requiring lightweight and easy-to-use equipment [50].



4. LEARNING ARCHITECTURES PERSPECTIVE

In this section, techniques for machine learning and fundamentals of deep learning, which are part of artificial intelligence, are discussed. Deep learning is subset of machine learning, like the way humans learn from data. Artificial intelligence (AI) includes machine and deep learning that excels in analyzing large, complex datasets to identify structures and make estimations. Deep learning models include multiple layers, known as artificial neurons, that process data. Each layer learns the specific features from the input data, gradually building a more abstract understanding as the data moves [48]. In the human brain, neurons communicate through electrical impulses. Similarly, in artificial neural networks, neurons are connected by weighted links, and these weights determine how signals are passed and combined. The conceptual relationship between deep learning and the human brain has been a guiding principle in the development of neural network architectures. Hierarchical relationship between Artificial Intelligence, Deep Learning, Machine Learning, and Neural Networks is shared in Figure 4.1. Deep learning models are composed of multiple hidden layers [51]. Early layers capture low-level features like edges, middle layers recognize parts of objects, and deeper layers identify entire objects.

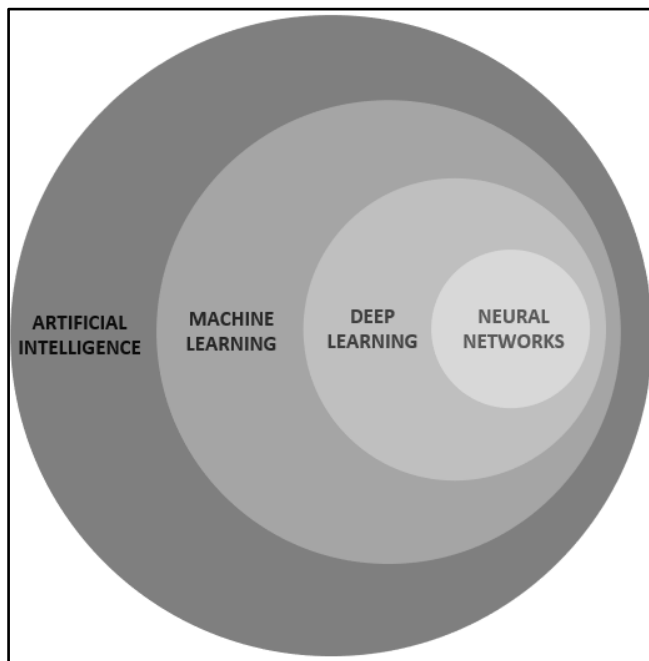


Figure 4.1. Hierarchical relationship of artificial intelligence and subfields

4.1. Deep Learning Fundamentals

Deep learning is a machine learning approach that mimics human brain functions through artificial neural network models. It consists of several fundamental building blocks and principles. These layers progressively extract higher-level features, making it particularly effective for complex missions such as language processing, image recognition, and speech analysis. Artificial neural networks are inspired by biological nerve cells. They are models developed within this structure and are based on mathematical calculations. An artificial neural network is basically formed of neurons, layers and weighted connections [18]. The structure of the artificial neuron is shown in Figure 4.2.

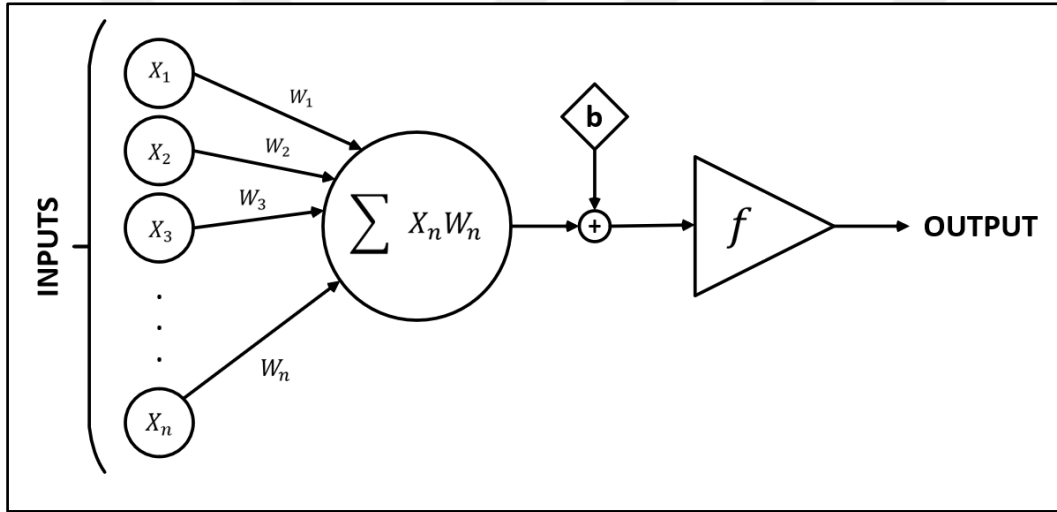


Figure 4.2. Fundamental artificial neuron model

Each artificial neuron receives input. It multiplies these inputs with certain weights. Because the relevant neurons contain different information. Then it passes through a summation function. After the summation function, the total information content obtained is summed with a bias value [18]. The total information content collected with bias is transmitted to the activation function. It is ensured that the information to be obtained at the output is consistent with each other. "X" represents the inputs of the artificial neuron. The parameter indicated as "W" shows the effect of the inputs on the model. Each input has a different effect. These inputs are collected together with their effects. The bias value indicated by "b" is transmitted to the activation function. The information content passed through the activation function is obtained as output data. Output of a neuron is calculated by the following formula.

$$z = \sum_{k=0}^n (X_n * W_n) + b \quad (4.1)$$

Neural networks are important for deep learning, consisting of layers of connected neurons that process input data and extract meaningful features. Structure of a neural network is composed of three main layers which are input, output, and hidden. The input layer is the first layer of the network, where raw data, such as text, images or signals, is fed into model. The hidden layers are responsible for feature extraction by applying transformations like convolutions or non-linear activations to learn complex patterns in the data [61].

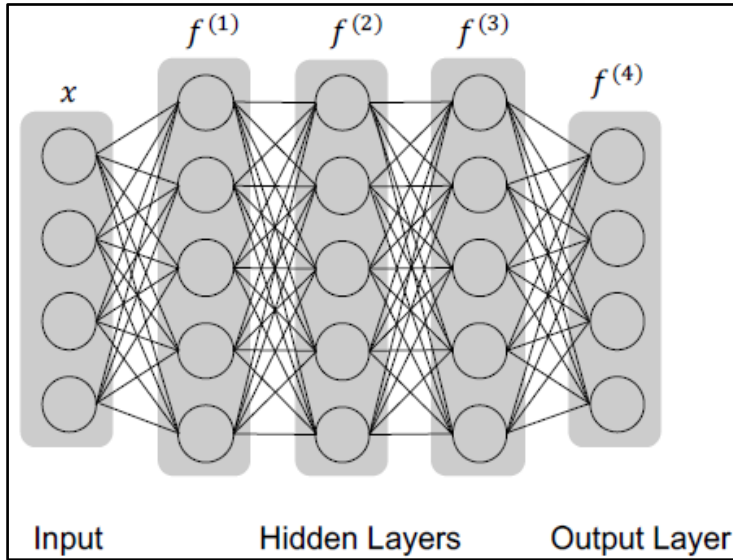


Figure 4.3. Feedforward neural network between layers [52]

The forward propagation equation is used to calculate the output of the neural network by processing it layer by layer. In this process, the input data is transmitted through the network and the weighted sum of the inputs is calculated in each layer. A bias term is added to this sum and the result is passed through an activation function to determine the output of that layer. This output is transmitted to the next layer.

The process continues through all hidden layers and finally reaches the output layer, creating the model's prediction. Forward propagation allows the network to make predictions based on the given input data. Backpropagation is an algorithm used to train artificial neural networks by minimizing the difference between predicted and actual outputs.

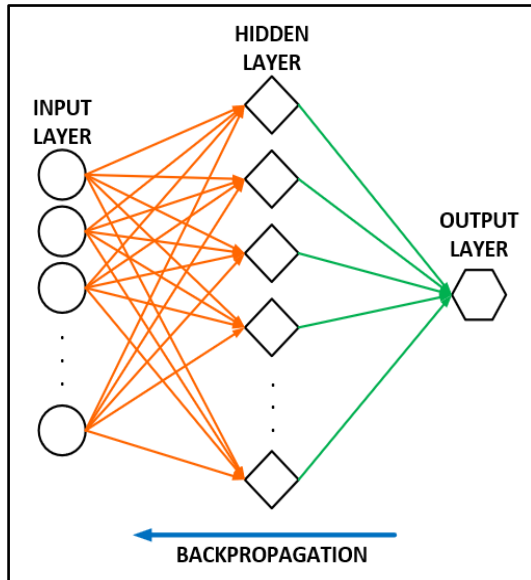


Figure 4.4. Backpropagation neural network between layers

Backpropagation begins with the initialization of weights and biases, which are typically set to random values in the network. In the next step, a forward pass is performed, and the loss is calculated using a chosen loss function to measure the difference [66]. In the image, the arrows represent the forward flow of information (orange) and the back propagation of the error signal (blue).

4.2. Deep Learning Methods

Deep learning represents a specialized field within machine learning that focuses on training models with multiple layers to automatically extract features from raw data and make predictions. This section explains that fundamental concepts of deep learning, providing an understanding of activation functions, regularization methods, loss functions, optimization strategies, and the structure of artificial neural networks. These concepts form the foundation for building and fine-tuning effective deep learning models [60].

4.2.1 Artificial neural networks

Artificial Neural Networks (ANNs) models are inspired by structure of biological neural networks like human brain. It is a key concept in artificial intelligence, particularly in deep and machine learning methods, designed to identify patterns and relationships in data to make decisions or predictions. Neural networks form base of deep learning. Consistency of

neurons that processes input data and transform it into meaningful representations. Each neuron applies a mathematical operation, such as a weighted sum of activation function, to learn features from the data. Structure of ANNs basically includes input layer, output layer, and hidden layer.

The input layer receives raw data, where each neuron represents a feature. The hidden layers process this data, identifying patterns and relationships using weighted sums, biases, and activation functions, with their number and size impacting performance. The output layer generates final predictions, with neurons representing classes for classification or continuous values for regression tasks. The key components of neural networks include neurons, which are basic computational units that process and transfer data.

Weights (w) and biases (b) are parameters adjusted during training to minimize prediction errors. Weights determine the impact of each input in predicting the neuron's output and are learned during the process. The bias is an additional parameter in artificial neural networks, used to adjust the output along with the weighted sum provided as input to the neuron [18]. Activation functions introduce non-linearity to learn complex relationships. The loss function parameters of error between the predicted outputs and actual targets, guiding the training process. Optimization algorithm adjusts weights and biases to minimize the loss function [18].

Long Short-Term Memory (LSTM)

LSTMs effectively handle the vanishing gradient problem, making them ideal for processing sequential data over extended periods [71]. Cell state shows the long-term memory of the network, acting as a pipeline that transfers information across sequential steps while retaining important data and filtering out irrelevant information [18].

Recurrent Neural Network (RNN)

RNNs are class of neural networks structured to process ordered data using a hidden state that accumulates contextual knowledge from previous steps. They are commonly used for tasks such as time series prediction, speech recognition, and natural language processing.

Convolutional Neural Networks (CNN)

CNN is a specialized form of artificial neural network designed to process structured data, particularly images, videos, and other spatially-organized data. CNNs are used in tasks like image recognition, object detection, and computer vision. CNNs are built on notion of convolution, which involves applying filters to input data to extract important features such as edges, textures, and patterns. Features are learned automatically during training, making CNNs highly effective for processing visual data. Architecture of a CNN typically consists of several types of layers, including convolutional layers for feature extraction, flatten, pooling layers for dimensionality reduction, and fully connected layers for making classifications or predictions [73]. Basic structure of a CNN includes several layers. The input layer accepts raw data, such as an image represented as matrix of pixel metrics. The convolutional layer extracts local features from input using filters that sense specific patterns like edges or textures by sliding over the data. The image classification structure of the CNN architecture is shown in Figure 4.5.

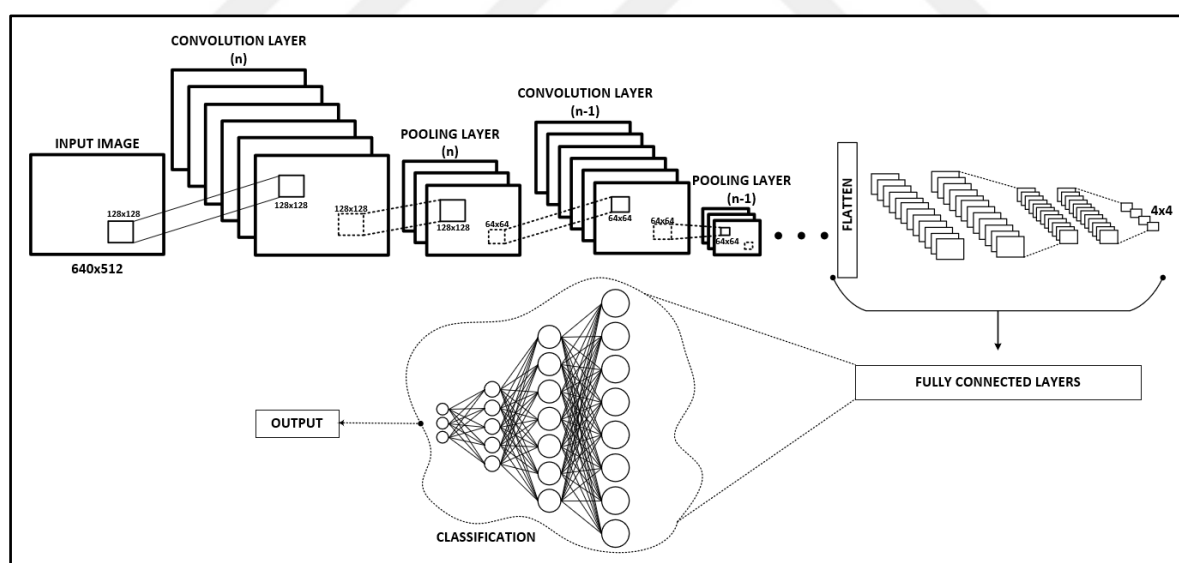


Figure 4.5. Architecture of a Convolutional Neural Network for image classification

Activation function applies a non-linear transformation to the feature extraction maps. Pooling layer decreases spatial dimensions of the feature maps, protecting important features while lowering computational complexity, using methods like average output and max pooling. The fully connected layer links all neurons to produce the final output, often used for predictions such as class probabilities. Output layer generates the network's predictions,

with neurons representing target classes or regression outputs [73]. Process begins with an input image that is passed through convolutional layers to extract features like edges and textures. Pooling layers follow, reducing dimensions of the activation pattern while preserving major features and lowering complexity. Max pooling method is shown in the Figure 4.6.

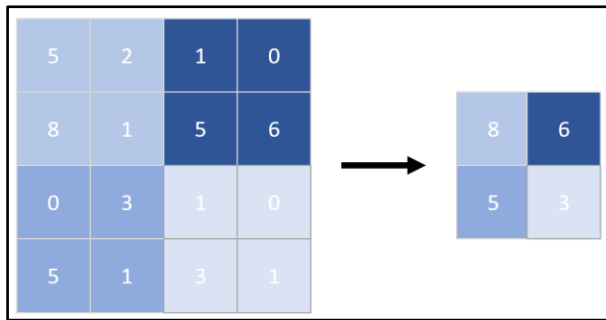


Figure 4.6. Max pooling operation for dimensionality reduction

The image illustrates a 2x2 Max Pooling method generally used in CNNs. In this process, input feature map is divided into 2x2 partitions, and maximum value from each region is selected to form a smaller output matrix. This reduces the spatial dimensions, lowers computation, and helps retain the most important features. The feature maps are then reduced into a one-dimensional vector and passed into fully connected layers. Flattening is a process of converting the multi-dimensional feature map into a one-dimensional vector. This step is essential before feeding the data into fully connected layers for classification. Flattening for feature map is shared in Figure 4.7.

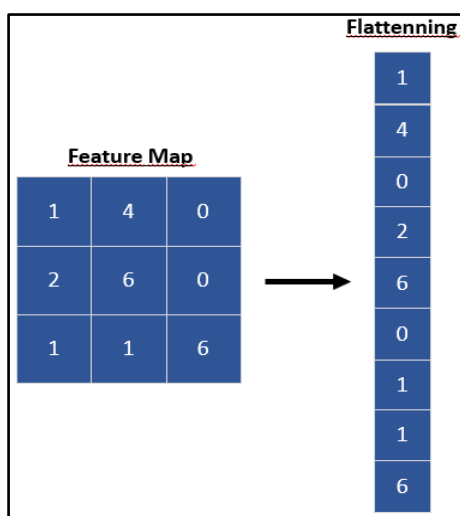


Figure 4.7. Flattening a feature map into one-dimensional vector

Finally, the output layer provides the predictions, assigning the input to a specific class. This architecture highlights the hierarchical feature extraction and decision-making capabilities of CNNs. ResNet, VGG16, VGG19, Inception and DenseNet architectures are explored to evaluate their performance in feature extraction and classification tasks.

- Residual Neural Network (ResNet)

ResNet is designed to improve the output of deep learning models. ResNet addresses vanishing gradient problem that happens in deep structure. Structure of ResNet includes the residual block and the deep architecture. Residual block consists of a series of layers, typically convolutional layers, along with connection for the layers [76].

- Visual Geometry Group (VGG)

VGG19 and VGG16 are developed by the VGG group [74]. VGG16 includes of 16 weight layers, including 3 fully connected and 13 convolutional layers, while VGG19 is a deeper version with 19 weight layers, including 3 fully connected and 16 convolutional layers [77].

- Inception

Inception was designed to improve computational efficiency and enhance performance in image classification and object detection tasks. Inception architecture is especially known for its innovative use of modules, which activate network to capture features at multiple scales within a single layer [78].

- Dense Convolutional Network (DenseNet)

DenseNet was designed to address some of limitations of traditional deep networks, such as inefficiency in parameter usage and difficulty in training very deep models. It achieves this through dense connectivity, where each layer is connected to subsequent layers, enhancing feature reuse and improving gradient flow [79]. The relationship between the dense blocks and the layers within the DenseNet architecture is shared in Figure 4.8.

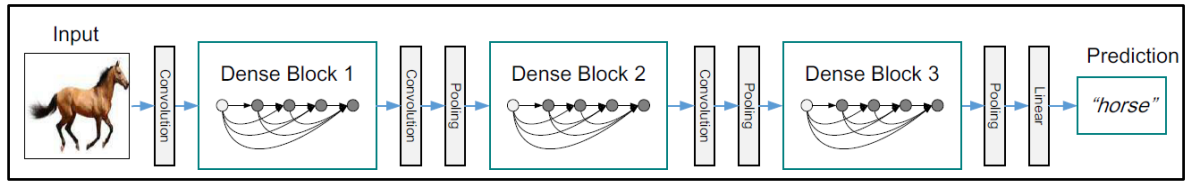


Figure 4.8. DenseNet architecture with dense-blocks and transition layers [79]

Dense block is the core component of DenseNet, where layers are connected to other layers in a feed forward structure. Within a dense block, features are concatenated rather than added, as seen in ResNet, which allows the network to preserve the original features. Between dense blocks, transition layers are used to reduce size of feature maps by using 1x1 convolutions and average pooling [79]. The growth rate determines how many new features each layer contributes; a smaller growth rate keeps the model compact, while a larger growth rate increases its representational power [80]. Design principles and differences outlined in Table 4.1.

Table 4.1. Feature differences between VGG, Inception, ResNet, and DenseNet architectures

Feature	VGG	Inception	ResNet	DenseNet
Depth	Medium	Medium to deep	Very deep	Medium to very deep
Parameters	High	Moderate	Moderate to high	Low (fewer than ResNet)
Connectivity	Sequential	Parallel convolutions	Residual connections	Dense connections
Feature Reuse	Limited	Limited	Partial	Extensive
Gradient Flow	Weak	Moderate	Strong	Very strong
Efficiency	Low	Moderate to high	High	High

DenseNet is highly efficient as it requires fewer parameters than ResNet, VGG, or Inception while maintaining high accuracy. Its dense connections enable feature reuse across layers, improving generalization and efficiency.

The dense connectivity also enhances gradient flow during backpropagation, making DenseNet highly trainable, even for very deep networks.

4.2.2 Activation functions

Activation functions are essential components and mathematical functions in neural networks that introduce nonlinearity, allows model to learn complex structures within the data. Each neuron implements an activation function after computing weighted sum of its inputs, determining whether the neuron should be activated. SoftMax, Hyperbolic Tangent (Tanh), Sigmoid, Rectified Linear Unit (ReLU), and Leaky ReLU are among the most generally used activation functions.

Activation functions identify output of artificial neuron, allowing network to learn nonlinear relationships and are applied at each layer of neural network. Artificial neural networks form the basic architectural structure that processes input data and produces output results. Regularization techniques are used in the training process to reduce the over-learning problem of the model. Loss functions help reduce errors with the backpropagation algorithm by evaluating the accuracy of the predictions made by model. Optimization algorithms improve the prediction performance by updating the weights and bias values of the model. In the following sections which are given below, detailed information about each of these components will be provided.

SoftMax

SoftMax function is preferred in output layer for multi-class classification missions. This method converts raw model outputs into probabilities, ensuring that sum of all probabilities equals 1. Each probability corresponds to a specific class [62].

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}} \quad (4.2)$$

$\sigma(z_i)$ represents probability assigned to class and z_i is the raw score in Eq. 4.2. $\sum_{j=1}^n e^{z_j}$ is the sum of exponentials of all logits, which normalizes the probabilities. Graph of SoftMax function is given at Figure 4.9.

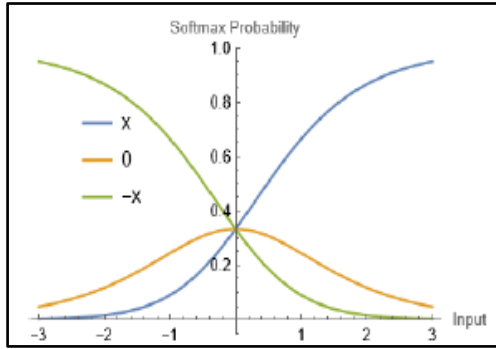


Figure 4.9. SoftMax activation function graphs [60]

Sigmoid

Sigmoid function is used in neural networks for 2-class classification as an activation function. Function limits any real-valued input into a range between 0 and 1, making Sigmoid useful for representing probabilities [63].

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4.3)$$

Input value is the x and e is base of the algorithm in Eq. 4.3. Outputs are easy to interpret as probabilities in binary classification. Graph of Sigmoid function is given at Figure 4.10.

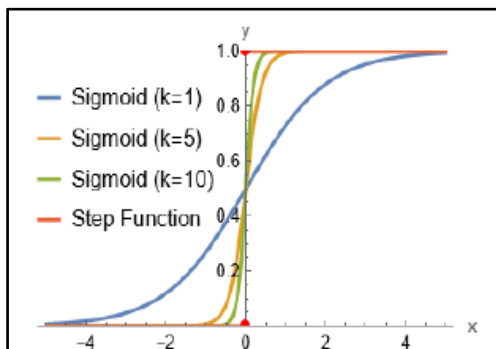


Figure 4.10. Sigmoid activation function graphs [60]

Rectified Linear Unit (ReLU)

ReLU is an activation function in neural networks application for deep learning models. ReLU introduces non-linearity for model, enabling model to learn and complexity

connections in the data. ReLU is computationally simple and effective, making it a popular choice in modern architectures.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \quad (4.4)$$

x is input value for the function in Eq. 4.4. If input is positive, output is the input itself. If input is zero or negative, output is zero [65]. Graph of ReLU function is given at Figure 4.11.

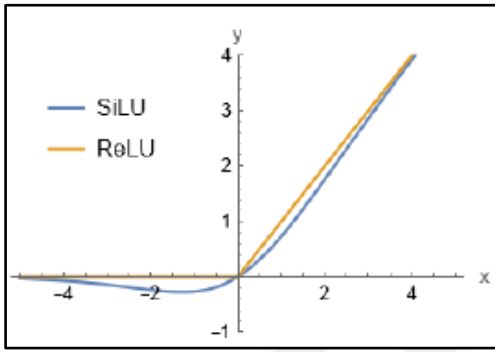


Figure 4.11. ReLU activation function graphs [60]

Leaky Rectified Linear Unit (Leaky ReLU)

Leaky ReLU is a type of ReLU activation function designed to address its key limitation: the "dead neuron" problem. Unlike ReLU, which outputs zero for all negative input values, Leaky ReLU allows that non-zero gradients for negative inputs.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases} \quad (4.5)$$

x is the input value for the function and α is a positive constant that controls the slope for negative inputs in Eq. 4.5.

Leaky ReLU is widely used in deep learning tasks, especially in networks where ReLU struggles with dead neurons [60]. Graph of LReLU function is given at Figure 4.12. Leaky ReLU is used in neural network architectures and image processing tasks as an activation function due to its ability to prevent dead neurons, enhance feature extraction.

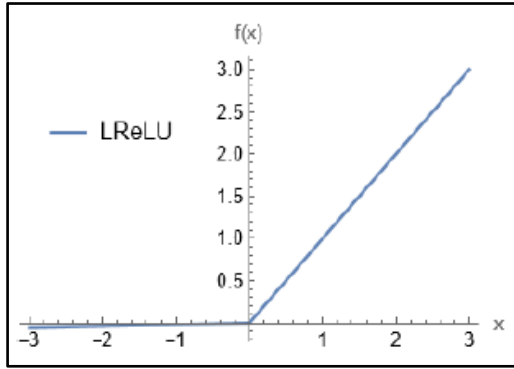


Figure 4.12. LReLU activation function graphs [60]

By retaining information from negative inputs, it enables models to process complex patterns more effectively and achieve better performance, particularly on challenging datasets and advanced architectures.

Hyperbolic tangent (Tanh)

Tanh function is non-linear solution that limits to input into range between -1 and 1, enabling useful for applications where outputs need to capture both positive and negative relationships. Graph of Tanh function is given at Figure 4.13.

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4.6)$$

e is the base of the logarithm and x being the input value in Eq. 4.6.

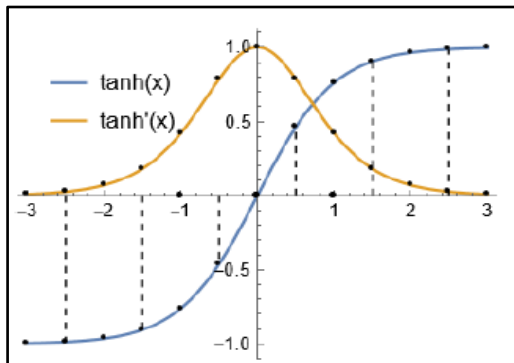


Figure 4.13. Tanh activation function graphs [60]

4.2.3 Regularization approaches

Regularization techniques are methods used in deep and machine learnings to prevent overfitting on training data. Overfit problem happens when a model performs exceptionally good for training data but struggles to generalize to unseen data. Regularization reduces model capacity to memorize the training data, encouraging it to learn general patterns instead. Regularization techniques contain L1 and L2 Regularizations, Dropout, Early Stopping and Batch Normalization [60].

L1 regularization (Lasso)

L1 regularization prevents overfitting and enhance feature selection. Method works by adding punishment term to the loss function. Method stimulates the model to reduce the weights of less critical features to zero, effectively performing feature selection [68].

$$\text{Loss} = \text{Original Loss} + \lambda \sum |w| \quad (4.8)$$

w is the weights for model and λ is the regularization parameter controlling penalty factor. During optimization, the algorithm minimizes this regularized loss function, which discourages large weight values.

L2 regularization (Ridge)

L2 regularization prevents overfitting by adding a punishment term to the loss function. Penalty is commensurate to the sum of the squared values of the model's weights. Unlike L1 Regularization, which can shrink weights to zero, L2 Regularization reduces their magnitude without making them exactly zero [60].

$$\text{Loss} = \text{Original Loss} + \lambda \sum w^2 \quad (4.9)$$

w is the weights for model and λ is the regularization parameter controlling the penalty strength. During training, the algorithm minimizes this regularized loss function, discouraging large weight values and distributing the importance across all features.

Dropout

Dropout is a regularization method used for deep learning applications to prohibit overfitting by randomly "dropping out" a part of neurons during training. This forces the network to learn more strong and generalized representations by preventing it from relying too heavily on specific neurons [66, 69].

$$z_i^{\text{dropped}} = r_i * z_i \quad (4.10)$$

z_i represents neuron i , and r_i is a random binary mask applied with a probability $(1-p)$ and p is the dropout rate.

Batch normalization

Batch normalization improves the training of neural networks by normalizing the inputs to each layer. It stabilizes and accelerates the training process by reducing the internal shift, which refers to changes in the dispersion of inputs to a layer as the model learns for deep learning algorithms [67,70].

$$\mu = \frac{1}{m} \sum_{i=1}^m x_i, \quad \sigma^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu)^2 \quad (4.11)$$

Mean (μ) and variance (σ^2) calculation are shared in Eq. 4.11. Then, the inputs are normalized for the mean and standard deviation.

$$x_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \quad (4.12)$$

ϵ is a constant added for numerical balance. The learnable parameters γ (scale) and β (shift) are applied to allow the neurons to learn the optimal distribution of the normalized data.

$$y_i = \gamma * x_i + \beta \quad (4.13)$$

Early stopping

Early stopping is a regularization technique used to prohibit overfitting by observing the model's generalization ability using validation data while training progresses. If the validation loss stops improving or begins to worsen for a specified number of consecutive epochs, training is stopped.

4.2.4 Loss functions

A loss function is a mathematical equation to measure difference between the predicted output of a model and the actual target value. Model training minimizes the loss function, thereby improving the accuracy and performance of the model. Loss functions have an important in performance measurement by quantifying how well or poorly a model is performing [71].

Regression loss functions

Regression loss functions are mathematical equations to measure failure between predicted output and target value in regression tasks. Regression involves predicting continuous numerical values, and choice of loss function specification how the model evaluates and improves its predictions [71].

Regression loss function is detailed under two headings: Mean Squared Error and Mean Absolute Error.

- Mean Squared Error (MSE)

MSE is used for loss function in regression missions for deep learning algorithms. MSE provides a quantitative metric to evaluate how well a model's predictions align with the true outputs [71].

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (z_i - \hat{z}_i)^2 \quad (4.14)$$

- Mean Absolute Error (MAE)

MAE is used for regression missions to measure average absolute difference between predicted values. Unlike MSE, MAE treats each errors equally by taking absolute value of differences [71].

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |z_i - \hat{z}_i| \quad (4.15)$$

Classification loss functions

A classification loss function is a mathematical equation used in machine learning applications to evaluate how well a classification model predicts the correct class labels. Method quantifies error between the model's predicted probabilities or labels and actual target labels, guiding the model to improve its performance during training [71]. There are 2 types such as Categorical Cross-Entropy and Binary Cross-Entropy.

- Binary Cross-Entropy (BCE)

BCE, which is Log Loss, is a loss function used for 2-class classifications. BCE measures variation between the predicted probability and actual binary label for each data point. BCE is used when the target variable has only two possible probabilities, such as 0 or 1 [71].

$$\text{BCE LOSS} = -\frac{1}{n} \sum_{i=1}^n [z_i * \log(\hat{z}_i) + (1 - z_i) * \log(1 - \hat{z}_i)] \quad (4.16)$$

z_i is the actual label for the i -th sample and $\hat{z}_i$ is the predicted probability of the positive class.

- Categorical Cross-Entropy (CCE)

CCE, known as SoftMax loss, is a loss function used in multi-class classification tasks where the target output is a single class out of multiple possible classes. CCE calculates difference

between probability distribution of true label and predicted distribution probability from the model [71].

$$\text{CCE LOSS} = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k z_{i*j} * \log(\hat{z}_{i*j}) \quad (4.17)$$

n is patterns number and k is number of classes. z_{i*j} is true label for j -th class of the i -th sample and $\hat{z}_{i*j}$ is predicted probability for the j -th class of i -th sample. Suppose a dataset has three classes, and the true label for a given sample is represented as $[0, 1, 0]$. The predicted probabilities for the sample are $[0.2, 0.7, 0.1]$.

$$\text{CCE LOSS} = -\log(0.7) \approx 0.357 \quad (4.18)$$

The loss decreases as predicted probability for the true class approaches 1, meaning the model is making more accurate predictions.

4.2.5 Optimization algorithms

Optimization algorithms are mathematical methods used in machine and deep learning applications to minimize loss function by calibrating the model's parameters, such as weights and biases. Training models by finding the set of parameters that result in the predictions for the given data [18].

Stochastic Gradient Descent (SGD)

SGD is an optimization equation to minimize the loss function by adjusting the model's parameters iteratively. Unlike traditional GD, which computes gradients using the all dataset, SGD updates parameters based on a single data sample, making faster and more stable for large datasets [18].

$$\theta = \theta - n * \nabla L(\theta) \quad (4.20)$$

θ are the model parameters and n is learning rate. $\nabla L(\theta)$ is gradient of loss function with dependent to θ . The gradient is computed using a single randomly selected sample.

Adaptive Gradient (Adagrad)

Adagrad is an optimization designed to adjust the learning metric for parameters based on magnitude of gradients [18].

$$G_t = G_{t-1} + g_t^2, \quad \theta_t = \theta_{t-1} - \frac{n}{\sqrt{G_t + \epsilon}} * g_t \quad (4.21)$$

g_t is gradient of the parameter at time step t and G_t is an accumulated sum of squared gradients for the parameter. θ_t is parameter at time step t and n is initial learning rate. ϵ is a tiny constant to prohibit division by zero.

Root Mean Square Propagation (RMSProp)

RMSProp is used for addresses issues like exploding or vanishing gradients by adapting the learning rate for parameter based on the magnitude of gradients [18].

$$E[g^2]_t = \beta * E[g^2]_{t-1} + (1 - \beta) * g_t^2, \quad \theta_t = \theta_{t-1} - \frac{n}{\sqrt{E[g^2]_t + \epsilon}} * g_t \quad (4.22)$$

g_t is gradient at time step t and β is loss rate for the moving average. n is learning rate and ϵ is tiny constant to prohibit division by zero.

Nesterov- accelerated Adaptive Moment Estimation (Nadam)

Nadam (Nesterov accelerated Adaptive Moment Estimation) builds upon Adam. This results in a more responsive adjustment to the gradients, especially in scenarios with rapidly changing loss surfaces. Nadam often improves convergence speed and generalization performance compared to Adam by anticipating direction of the gradient more effectively. Adam combines the strengths of SGD with momentum and RMSProp to provide better

performance. Adam adaptively adjusts learning rate for parameters, making it robust and effective for various tasks.

4.3. Common Artificial Intelligence Terms

This section covers the concepts of the epoch, steps per epoch, batch size, target size, test and validation accuracies, test and validation losses, overfitting, underfitting, and F1 score.

Epoch

Epoch indicates that number of cycles in deep learning method. Epoch identifies how many times the model processes the full dataset. Training for multiple epochs allows the model to learn progressively by updating its weights iteratively [60].

Learning rate

Learning rate controls step size at which a neural network updates its weights during the training process. This method controls how quickly or slowly a model learns.

Batch size

Batch size is quantity of training samples processed by the model. Small batch sizes can lead to more generalized learning due to increased gradient variability, while larger batch sizes offer faster computation but may require more memory [60].

Step Per Epoch (SPE)

SPE define number of batches the model will process during a single epoch. This parameter directly affects the total number of iterations required to complete one pass over the entire dataset.

$$\text{SPE} = \frac{\text{Number of Training Samples}}{\text{Batch Size}} \quad (4.23)$$

Test accuracy

Test accuracy is rate of correctly predicted patterns out of total samples in the test dataset. Results show that how well model generalizes to unseen data [60].

$$TA = \frac{\text{True Predictions}}{\text{Total Test Examples}} \quad (4.24)$$

Test loss

Test loss is evaluation of the model's error on test dataset. A lower test loss generally indicates better model performance, but this technique must align with other metrics like test accuracy to ensure proper evaluation [60].

Validation accuracy

Validation accuracy measures performance on validation dataset for the model. It helps track whether the model is overfitting or underfitting as this provides understandings into how well model generalizes to unseen data [60].

Validation loss

Validation loss indicates error on the validation dataset. It is a critical metric used to identify overfitting; when validation loss increases while training loss decreases, model may be overfitting to training data [60].

F1 score

F1 score is harmonic mean of precision and recall, making balanced evaluation and imbalanced outputs.

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4.25)$$

F1 score shows that model performance in terms of precision and recall [60]. Precision calculated proportion true instances out of all predictions made as positive. Recall shows that predicted positive samples out of positive samples in the dataset.

Overfitting/Underfitting

Overfitting happens when a model learns not only underlying patterns but also noise and irrelevant patterns. As a result, model performs well on the training data on unseen or validation data [60]. Underfitting happens when a model is simplistic or lacks the capacity to learn underlying patterns in the data [60].

Target size

Target size refers to the dimensions (height and width) to which input images are resized. [60]. This resizing ensures that all input images have a consistent shape, making them compatible with neural network architecture.

5. AUGMENTED DATA AND METHODS

Data Augmentation is a common technique used for artificially increase the dataset in machine learning and deep learning projects. It is used to increase the generalization ability of the model during training and to prevent overfitting. Image data augmentation is a standard method used to provide more successful results for models trained with a limited number of images. This record is created by artificially applying various transformations on existing data. Thus, the model is made more robust to different conditions and has a higher generalization ability. The most commonly used image verification methods are given below.

- **Rotation:** The image is viewed from certain angles and data is created from different perspectives.
- **Flipping:** The image can be mirrored vertically or vertically. Vertical reflection (horizontal translation) is especially used very often.
- **Cropping & Scaling:** Cropping is used in training by cutting a certain part of the image. This method allows the model to learn different regions of the object. The size data of the image is resized, but the original proportions are preserved.
- **Shifting:** The image is shifted a certain amount on the horizontal or vertical axis.

Various data augmentation techniques such as flipping, scaling, rotation, cropping, and shifting were applied to increase the diversity of the training dataset in this study. These transformations were performed randomly to ensure that each image underwent a unique and unpredictable modification during training. By applying these augmentations in a stochastic manner, the model was exposed to a wide range of variations in image orientation, size, and position. This approach allowed the model to better generalize by preventing it from memorizing specific patterns and helped it to become more robust when encountering previously unseen data. The random application of these techniques also simulates natural variations that may occur in real-world scenarios, further enhancing the model's adaptability. Prior to data augmentation, the dataset consisted of 100 images per class, totaling 500 images, all collected under controlled laboratory conditions. To improve model performance and increase generalization capability, data augmentation were applied individually to each imaging modality (RGB, SWIR, and MS). This process resulted in

larger datasets across different configurations: SET2, SET3, and SET4, with the number of samples per class progressively increasing in each set.

Table 5.1. Amount of augmented for MS, SWIR, RGB classes

Classes	SET1 (Original Data)	SET2 (Light Aug.)	SET3 (Medium Aug.)	SET4 (Heavy Aug.)
Alcohol	100	170	240	310
Acetone	100	170	240	310
Flux	100	170	240	310
Water	100	170	240	310
Cologne	100	170	240	310
TOTAL	500	850	1200	1550

The base dataset (SET1) consisted of 100 original images per class. For the augmented sets (SET2 to SET4), various augmentation techniques such as flipping, rotation, shifting, zooming, and shearing were applied randomly and independently to each image in the RGB, SWIR, and MS datasets. This ensured the creation of diverse and balanced datasets across all imaging modalities. For multispectral (MS) imaging specifically, each class included five spectral bands, with 20 original images per band, totaling 100 images per class. Figure 5.1. illustrates the data distribution strategy applied to RGB, SWIR, and multispectral imaging datasets.

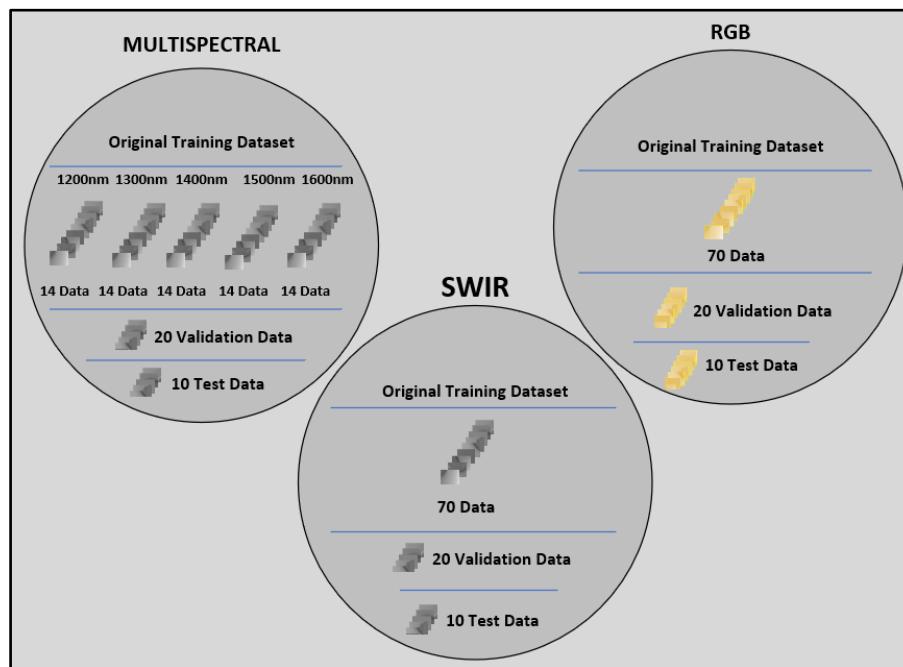


Figure 5.1. Distribution of original datasets for RGB, SWIR, and MS Imaging

During augmentation, transformations were applied individually to each spectral band image to preserve spectral integrity while increasing data variety. Consequently, the number of images grew proportionally across all bands in the MS modality. Moreover, the augmentation intensity was progressively increased in parallel with dataset size. For example, the rotation range was set to 30° in SET2, 40° in SET3, and 50° in SET4. Similar incremental adjustments were made for shifting, shearing, and zooming. This strategy enabled the model to learn from a broader set of variations, improved its generalization ability, and helped mitigate overfitting. For each imaging modality, the original dataset was divided into three subsets such as 10% for testing, 20% for validation, and 70% for training. In the case of multispectral data, the training set includes images captured at five distinct wavelengths (1200nm to 1600nm), with 14 images per band. This standardized partitioning approach ensures consistent evaluation and supports robust model development across different imaging types. Experimental evaluations conducted using the DenseNet architecture demonstrated that data augmentation significantly enhanced classification accuracy across all imaging modalities. Among the tested configurations, SET3 yielded the highest accuracy, particularly in the multispectral (MS) modality. These findings highlight the effectiveness of multispectral imaging in differentiating visually similar liquid samples and underscore the importance of both dataset diversity and augmentation intensity in developing robust deep learning models. The performance improvements were especially notable in scenarios with limited real-world data. Data augmentation parameters are shown in Table 5.2.

Table 5.2. Data augmentation methods and parameters

Augmentation Method	Applied Minimum Value
Rotation Range	30
Rescale	1./255
Width & Height Shift Range	0.3
Shear & Zoom Range	0.4
Horizontal Flip	True

The image under shows examples of data augmentation applied to liquid samples. Various transformations such as rotation, scaling, shifting, zooming, shearing, and horizontal flipping were used to simulate different imaging conditions. Data augmentation methods, including controlled geometric transformations, were applied to expand the training dataset. Each image was altered randomly using predefined parameters to improve the model's

learning capabilities. Data augmentation is a powerful technique to enhance model generalization; however, excessive or unrealistic augmentation may introduce noise and distort the true characteristics of the data, potentially leading to decreased model performance. Therefore, augmentation must be applied in a balanced and controlled manner. In this study, the size of the validation and test datasets was relatively limited, with a total of 50 test samples for each imaging method. This increases the sensitivity of performance metrics to individual misclassifications. For example, a single incorrect prediction in a test set of 50 samples leads to a 2% decrease in accuracy. Consequently, variations in evaluation metrics are more pronounced and should be interpreted with caution, considering the constraints of the dataset size.

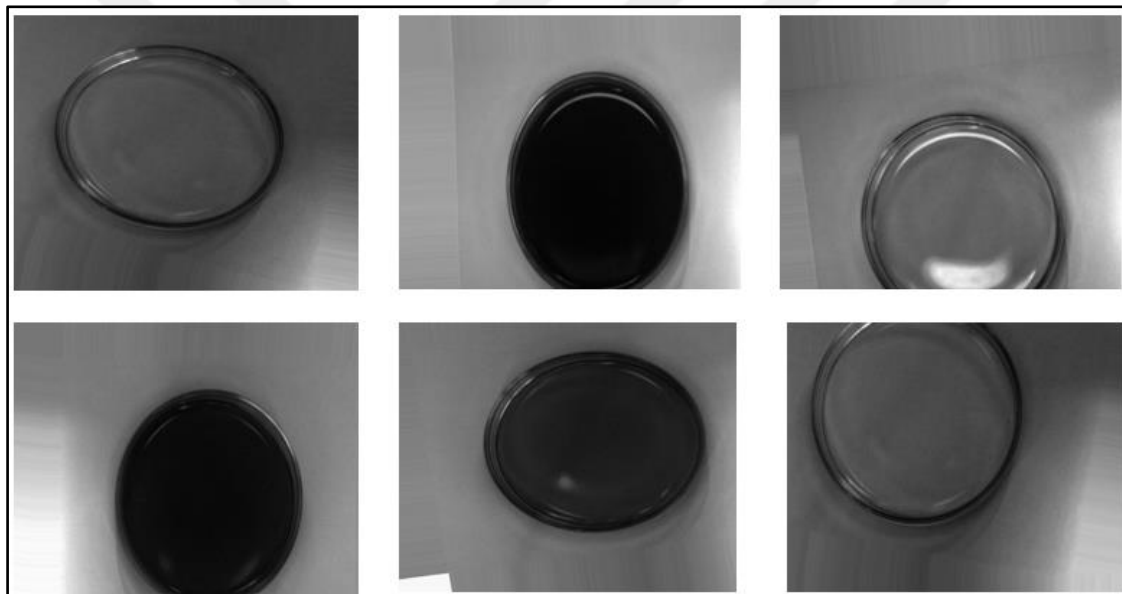


Figure 5.2. Visual appearance of augmented test liquid data examples

6. EXPERIMENTAL EVALUATIONS

In this section, the process of collecting data during the thesis study and the evaluation of the developed architecture are detailed. The methodology used for acquiring Short-Wave Infrared (SWIR) and Multispectral (MS) images is outlined, and the preferred deep learning architecture is validated through obtained results. Various artificial neural network architectures were examined, and the one yielding the most optimal results was selected as the reference model. In this context, emphasis is placed on the CNN architecture, which is widely recognized and utilized in image and video processing. The workflow diagram is presented in Figure 6.1.

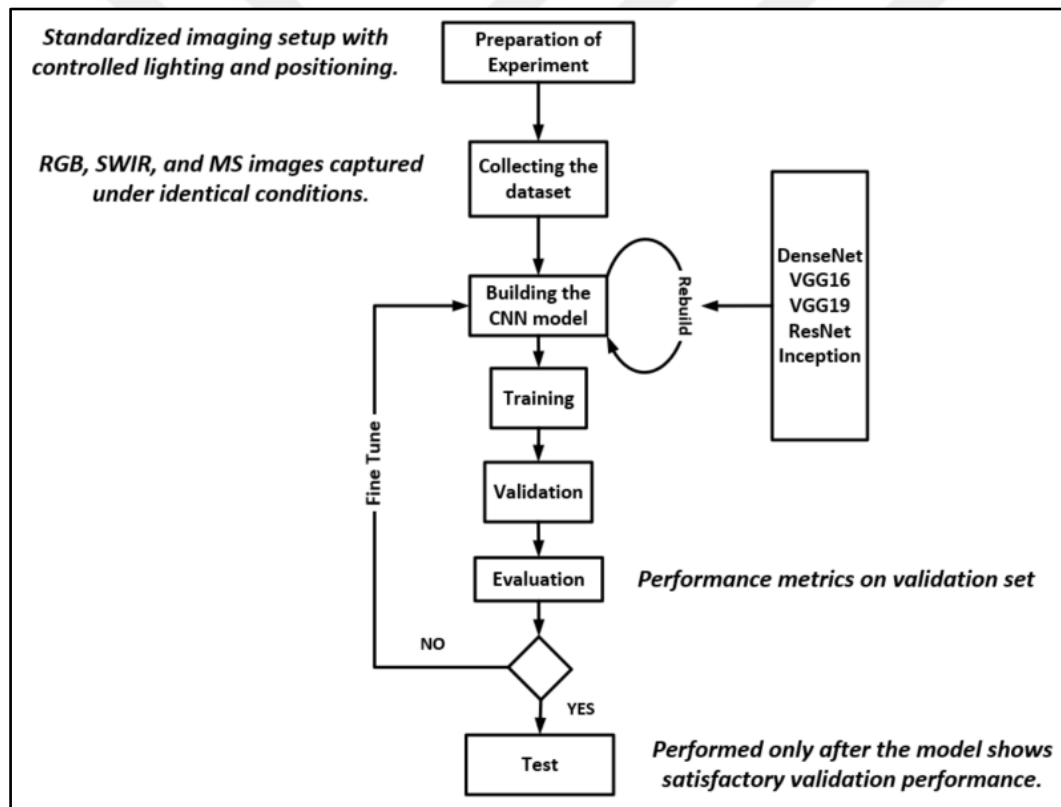


Figure 6.1. Workflow for processing deep learning methods on experimental setup

According to the diagram, the tests began with the preparation of the image acquisition mechanism in the laboratory environment and proceeded with the acquisition of datasets for RGB, SWIR, and Multispectral (MS) images. RGB images were captured for materials that are visually similar and cannot be distinguished through simple observation, such as propanol, cologne, water, acetone, and flux. Following this, SWIR images were captured, and MS images were obtained using a SWIR camera with five different filters having cut-

off frequencies of 1600 nm, 1500 nm, 1400 nm, 1300 nm, and 1200 nm. After completing the data collection process, deep learning architectures were explored. Since the goal was to achieve separation among the five classes using a deep learning approach, CNN architecture, which is widely used and suitable for this purpose, was chosen. Experiments with other architectures were also conducted, and the best results were obtained with CNN. Tests were conducted for binary classification and 5-class classification. Metrics; test accuracy, validation accuracy, test loss, validation loss, and F1 scores were recorded and analyzed. The importance of IR camera calibrations has been mentioned in previous topics. Now, the SWIR camera configuration used for Multispectral imaging will be explained.

6.1. Data Acquisition

In this thesis, RGB, SWIR, and MS images were collected from five different liquids selected to facilitate material detection. These liquids were chosen because they cannot be distinguished by either the naked eye or an RGB camera. An image of the liquids captured with an RGB camera is presented in Figure 6.2.



Figure 6.2. Visual appearance of selected test liquids in RGB

RGB images of five different materials were captured in a controlled chamber. Similarly, images of the same materials were taken using a SWIR camera. These images are presented in Figure 6.3.

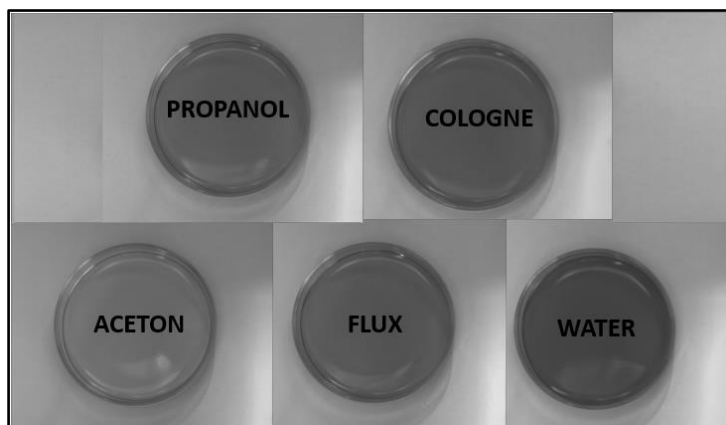


Figure 6.3. Comparison of the SWIR test images

Multispectral images of the materials were captured under consistent environmental conditions using filters with cut-off frequencies of 1.6 μm , 1.5 μm , 1.4 μm , 1.3 μm , and 1.2 μm . Sample of acquired multispectral (MS) images belonging to alcohol is presented in Figure 6.4.

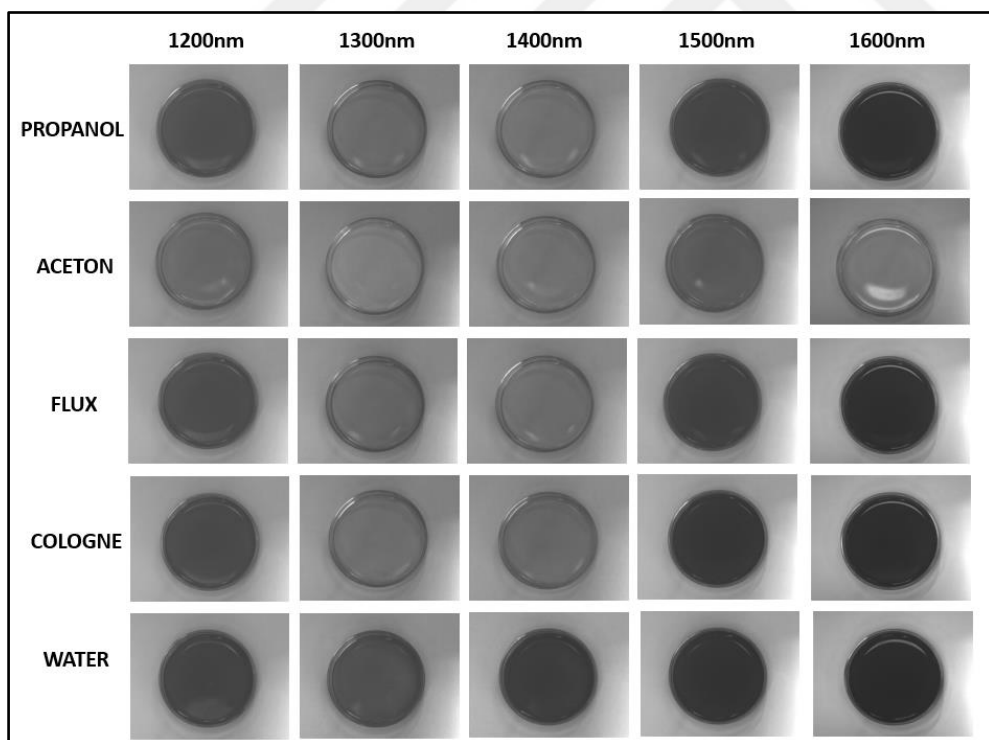


Figure 6.4. Multispectral test images at different wavelengths

These multispectral images highlight the spectral differences of the materials across various wavelengths, enabling detailed analysis of their unique properties. This approach is particularly useful for distinguishing materials that appear identical in traditional RGB

images and SWIR images. For each class, separate training, validation, and test folders were created. Within these folders, five subfolders corresponding to the five different materials were organized, and the collected data was stored accordingly. Each material, regardless of the imaging method, includes 100 images.

6.1.1. Environment variables

Environmental effects such as angle and lighting were carefully controlled to ensure consistency. A fixed backlight was left inside the black box. This fixed light source was used in all imaging methods. The camera angles were adjusted to be perpendicular to the data used in RGB, SWIR and MS imaging methods. The appearance of the test set in the laboratory environment, along with the RGB, SWIR, and multispectral imaging setups used during dataset acquisition, is presented in Figure 6.5.

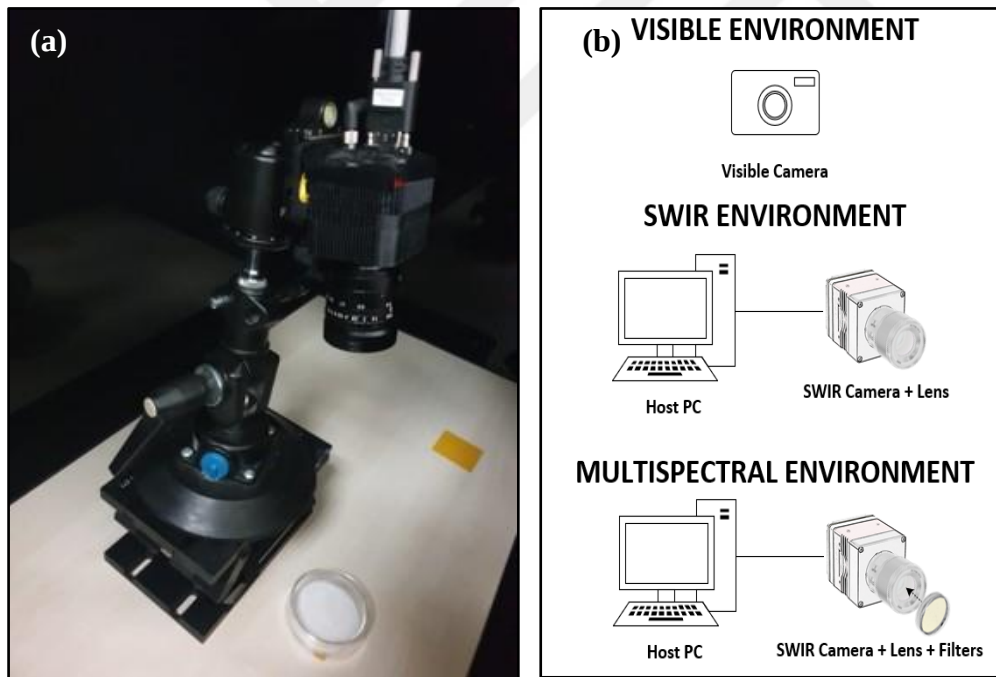


Figure 6.5. Experimental test setup and environments. a) test setup b) RGB, SWIR, and MS imaging environments

The ambient temperature is approximately 25 degrees Celsius as it is a laboratory environment. The ambient temperature is approximately 25 degrees Celsius since it is a laboratory environment. Temperature change is important for the calibration of cameras. Especially for SWIR cameras, temperature change is more important because it affects the calibration status. In RGB imaging, an RGB camera was used with backlighting in a black

box. After the SWIR camera was fixed, data was collected using a lens with transparent glass. In the MS imaging method, the same structure was not disrupted, and images were taken by fixing filters with cut-off frequencies of 1200nm, 1300nm, 1400nm, 1500nm and 1600nm in front of the lens.

6.1.2. Input standardization

In order to achieve the objectives of this thesis, two different camera types—RGB and SWIR—were used. The data collected from these cameras were utilized within the deep learning architecture. To minimize the effects of variable differences, the characteristics of the acquired data were standardized as much as possible. In this context, resolution and bit depth were carefully considered and are explained in this section. A consistent input size provides a significant advantage for architectures such as convolutional neural networks. Therefore, the resolution for both camera types was set to 640×512 pixels, with 640 pixels in width and 512 pixels in height. Adjusting for bit depth is essential for effective feature extraction. RGB images include of three main color channels such as red, green, blue, and each with 8-bit depth, resulting in a total of 24 bits per pixel.

Table 6.1. Technical specification comparison of different imaging modes

Specification	RGB Image	SWIR Image	MS Image
Format	PNG	PNG	PNG
Width	640	640	640
Height	512	512	512
Mode (Color Depth)	RGB 3 channels	L - Grayscale 1 channel	L - Grayscale 1 channel
Minimum Pixel Value	R: 0 G: 0 B: 0	0 (Black)	0 (Black)
Maximum Pixel Value	R: 255 G: 255 B: 255	255 (White)	255 (White)
Transformation Method	RGB data	Pseudo-RGB data	Pseudo-RGB data

On the other hand, SWIR cameras, which were used for both SWIR and multispectral (MS) imaging in this study, produce single-channel outputs in float32 format with a 32-bit depth. Importantly, the SWIR camera used in this study provided output data in a normalized

format, with pixel values already scaled between 0 and 1. This ensured compatibility across different imaging modalities and supported stable and efficient model training. Due to the bit depth difference, each pixel value was converted from the range of 0-255 to the range of 0-1. This process allows the model to learn more stably and quickly because standardized inputs are used. The dataset was partitioned into 10% for testing, 20% for validation, and 70% for training. Distribution was chosen to allow sufficient data for model training while maintaining separate subsets for hyperparameter tuning and unbiased performance evaluation. The 70-20-10 split was chosen to ensure that the model is trained on a sufficiently large portion of the data (70%), while reserving adequate and balanced portions for validation (20%) and testing (10%). This distribution helps achieve a good trade-off between learning capacity and reliable performance assessment. A smaller training share might limit the model's ability to generalize, especially when the dataset is not very large. SWIR and MS images, originally single-channel grayscale, were transformed into pseudo-RGB format by duplicating the grayscale values across the three-color channels to ensure compatibility with CNN models expecting RGB input. The relevant conversion is shown in the table below.

Table 6.2. Pseudo-RGB conversion of SWIR and MS data

Original Grayscale	Red Channel	Green Channel	Blue Channel
0 (Black)	0	0	0
128 (Gray)	128	128	128
255 (White)	255	255	255

SWIR and MS images contain spectral information beyond the visible range, offering richer content compared to standard RGB data. In this study, single-channel SWIR and MS images were transformed into pseudo-RGB format by replicating the same grayscale information across all three channels to ensure compatibility with convolutional neural network architectures. This conversion process does not result in any loss of information, as it serves only to achieve structural compatibility. In contrast, converting RGB data into SWIR or MS format would require artificially estimating spectral components that are inherently absent in RGB images. Such estimations may introduce significant information loss and degrade classification performance. Therefore, the chosen direction of conversion in this study provides a more accurate and reliable approach by preserving the integrity of the original data while ensuring compatibility with deep learning models.

In order to ensure compatibility of single-channel images with deep learning models, each pixel's intensity value is replicated across the red, green, and blue channels to create a pseudo-RGB representation. This process is commonly referred to in the literature as “channel replication” or “grayscale-to-RGB conversion.” The method does not alter the underlying image content; instead, it restructures the data format to meet the input requirements of convolutional neural networks that are typically designed for three-channel (RGB) inputs. As a result, grayscale data can be effectively utilized in RGB-based deep learning architectures without loss of information.

6.2. Architecture Selection

Selection of the deep learning architecture and application, primary focus was placed on CNN architectures. During the development of these architectures, experiments were conducted on the main model and activation functions. The results were compared, and the architectural structure that yielded the best performance was selected as the criterion. After selecting the ideal architecture, the underlying architecture was applied to RGB, SWIR and MS imaging methods and the differences were examined.

6.2.1. Model decision

Architectural experiments were carried out using five different classes of multispectral images. When constructing the CNN model, various activation functions can be used, each with its own advantages depending on the learning dynamics of the data. For the output layer, the SoftMax activation function was selected, as it is well-suited for multi-class classification by generating a probability distribution over the possible classes. In this study, input images are processed through pretrained CNN architectures. Extracted features are forwarded to fully connected layers, and the SoftMax classifier determines the most probable class. In one example, the model successfully classified the input as “Alcohol.” Several well-established CNN architectures were explored to enhance classification performance. VGG16 and VGG19 offer a straightforward, deep structure with stacked 3×3 convolutional filters. While easy to implement, these models require a significant number of parameters and are computationally demanding. ResNet, on the other hand, introduces residual connections, which help preserve gradient flow and allow training of very deep networks by learning

identity mappings. This technique addresses the vanishing gradient issue and supports the development of highly expressive models.

DenseNet adopts a different strategy by creating direct connections from each layer to each subsequent layers. This dense connectivity leads to efficient feature reuse, improved gradient propagation, and often better generalization with fewer parameters. Inception, also known as GoogLeNet, takes a modular approach by applying different types of convolution filters in parallel. This allows the model to analyze visual features at multiple spatial scales, offering a balance between computational cost and accuracy. Each of these architectures brings unique strengths, and the selection depends on factors such as task complexity, dataset size, and available hardware resources. The full classification pipeline, including feature extraction and prediction steps, is illustrated in Figure 6.6.

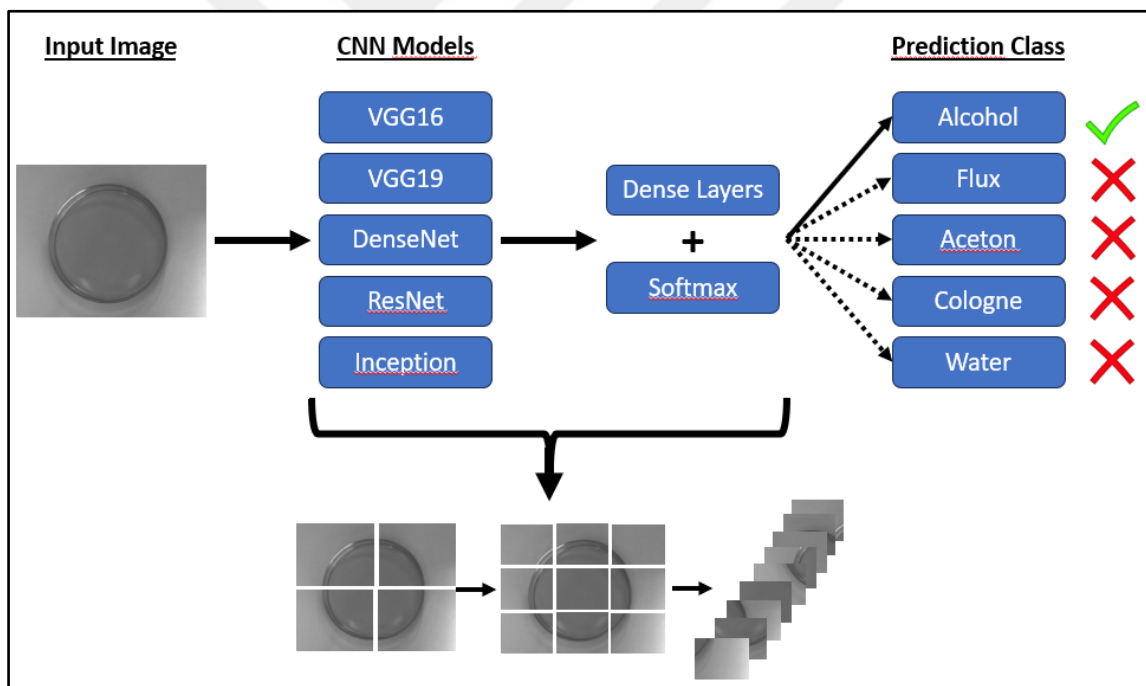


Figure 6.6. Class prediction evaluation of input image

In deep learning, activation functions and optimization algorithms serve fundamentally different roles, both of which are essential for the learning process. Activation functions are implemented to the output of each neuron in a neural network to introduce non-linearity. Without them, the network would behave like a simple linear model, regardless of how many layers it has. These functions help the network learn complex patterns by transforming the weighted sum of inputs into a non-linear output. General activation functions include ReLU,

Leaky ReLU, Sigmoid, SoftMax, and Tanh. Each function has its own characteristics and is chosen based on the layer's purpose within the network. On the other hand, optimization algorithms, often referred to as “optimizers,” are responsible for updating the model's weights during training. They work by minimizing the loss function using gradients calculated through backpropagation. Optimizers determine how quickly and effectively the model learns from the data. Popular examples include SGD, Adam, RMSprop, Nadam, and Adagrad. These algorithms adjust parameters such as learning rates and momentum to ensure faster convergence and better accuracy.

While activation functions influence how the network processes information at each layer, optimization algorithms guide the network in improving its performance over time. Both are crucial: activation functions define how neurons activate, while optimizers drive the learning process by fine-tuning the model's weights. Understanding and selecting the appropriate activation function and optimizer combination is key to building a well-performing neural network. Different combinations of activation functions were tested. Results obtained using LeakyReLU with Adam, SGD, RMSProp, Nadam, and Adagrad optimizers were compared for the 5-class classification. The results are shared in Table 6.2.

Table 6.2. Activation function combinations and performance comparison for 5-class MS SET3 imaging classification

Combination/Results	Test Accuracy (%)	Validation Accuracy (%)	Test Loss	Validation Loss
LeakyReLU – Nadam	96.46	89.09	0.36	0.87
LeakyReLU – SGD	15.99	17.59	2.32	2.17
LeakyReLU – RMSProp	85.33	69.01	0.64	0.97
LeakyReLU – Adagrad	34.66	20.80	2.02	2.17

The Nadam optimizer, when combined with the LeakyReLU activation function, produced the most successful outcomes in this study. It delivered the highest accuracy and the lowest loss values across both test and validation sets. In particular, Nadam reached a test accuracy of 96.46% and a validation accuracy of 89.09%, outperforming other tested optimizers in terms of stability and generalization capability. Although RMSProp yielded reasonable performance with a test accuracy of 85.33%, its validation accuracy dropped to 69%, suggesting that its generalization ability was more limited. In contrast, both SGD and Adagrad resulted in considerably lower accuracy and higher loss values. Among them, SGD showed the weakest results, with a test accuracy of only 15.99%, indicating its unsuitability

for this classification task. Compared to Adam, which also showed promising performance in earlier experiments, Nadam slightly surpassed it in validation accuracy. This confirms Nadam as the most reliable optimizer tested in this context. The LeakyReLU–Nadam pairing proved to be the most effective setup, offering a strong balance between learning efficiency and generalization.

Various well-known CNN architectures, including VGG16, VGG19, ResNet, Inception, and DenseNet, were integrated within the model’s design. In particular, DenseNet, Inception, and VGG networks are widely acknowledged for their effectiveness in image classification and feature extraction tasks. The evaluation results of these architectural configurations, applied to a five-class classification scenario using the MS-SET3 dataset, are summarized in Table 6.3.

Table 6.3. MS-SET3 results of different functions for 5-class classification

Hidden Layer Function	Optimizer	Model Design	Test Accuracy (%)	Test Loss	Validation Accuracy (%)	Validation Loss	F1 Score
LeakyReLU	Nadam	VGG16	94.38	0.51	87.40	0.79	92.12
LeakyReLU	Nadam	VGG19	94.82	0.50	88.26	0.76	93.00
LeakyReLU	Nadam	DenseNet	96.46	0.36	89.09	0.87	92.24
LeakyReLU	Nadam	ResNet	38.66	1.45	25.86	1.56	37.41
LeakyReLU	Nadam	Inception	94.73	0.49	83.46	0.51	93.38

“Test Accuracy” shows that percentage of true predictions made by model on the test dataset. Higher results are preferred but overfit problem should be observed. If the algorithm overfits, early stopping can be applied. “Test Loss” is the error of calculated test dataset and lower values indicate a model that performs better on new data.

“Validation Accuracy” measures the percentage of true predictions on the real world which means that model’s performance on data it has not seen before. “Validation Loss” is the error calculated on the validation dataset during training. F1 score considers both precision and recall, indicating how well the model balances between making correct predictions. General model parameters are shown in Table 6.4.

Table 6.4. General model parameter details for MS, SWIR, RGB classifications

GENERAL MODEL PARAMETERS	
Model	LeakyReLU
Architectural Function	DenseNet
Activation Function (Output)	SoftMax
Loss Function	Categorical Cross-Entropy
Normalization	Batch Normalization
Optimization	Nadam
Dropout	0.5
Input Shape	224 * 224
Learning Rate	1e-4
L1 & L2 Regularization	Yes
Flatten	Yes

The table outlines the model parameters selected for classifying image data obtained through multispectral (MS), short-wave infrared (SWIR), and RGB imaging techniques. The overall architecture is built on the DenseNet framework, which employs dense layer connectivity to strengthen feature transmission and mitigate gradient vanishing issues. LeakyReLU was chosen for the hidden layers to support effective information flow, while SoftMax function was implemented in the output layer to enable multi-class decision making. In order to enhance the model's ability to generalize and reduce overfitting, various regularization strategies were integrated, including dropout, combined L1 and L2 penalties, and batch normalization. The optimization process relied on the Nadam algorithm which contributed to stable and efficient training. The input dimensions for the model were fixed at 224×224 pixels. Before feeding into the fully connected layers, a flattening operation was performed to convert the extracted spatial features into a suitable vector format. These configuration choices were made to ensure strong and consistent performance across different imaging types. Furthermore, the impact of this deep learning-based approach was examined with respect to each imaging modality. Experimental results, as presented in Table 6.5, highlight the effectiveness of the model, especially in the multispectral domain.

Table 6.5. Comparison of MS, SWIR, and RGB results without data augmentation

Class SET	Imaging Method	Test Accuracy (%)	Test Loss	Validation Accuracy (%)	Validation Loss	F1 Score
SET1	RGB	17.52	3.64	21.19	9.82	24.41
SET1	SWIR	38.21	1.82	28.72	2.98	40.26
SET1	MS	94.23	0.42	85.76	0.83	86.68

The table presents the classification results for SET1, which was evaluated without applying any data augmentation techniques. Among the imaging methods, MS (Multispectral) shows significantly better performance in both test and validation metrics, with a test accuracy of 94.23%, validation accuracy of 85.76% and an F1 score of 86.68%. In contrast, RGB and SWIR methods exhibit much lower performance, especially in validation accuracy and F1 scores. These results highlight the advantage of MS imaging in distinguishing between the classes when no data augmentation is applied. The results of the images reproduced by applying the data augmentation technique are shared in Table 6.6.

Table 6.6. Augmented MS, SWIR and RGB data for 5-class classification

Class SET	Imaging Method	Test Accuracy (%)	Test Loss	Validation Accuracy (%)	Validation Loss	F1 Score
SET2	RGB	22.36	2.98	22.98	8.62	30.26
SET2	SWIR	40.58	1.76	28.93	2.95	40.36
SET2	MS	95.68	0.42	86.20	0.99	88.15
SET3	RGB	34.16	2.11	23.72	8.16	35.70
SET3	SWIR	48.96	1.69	31.62	2.93	45.89
SET3	MS	96.46	0.36	89.09	0.87	92.24
SET4	RGB	27.37	1.89	31.19	8.21	29.36
SET4	SWIR	45.99	1.74	35.87	2.37	46.44
SET4	MS	93.96	0.46	87.12	0.79	88.15

The findings indicated that SWIR imaging alone did not yield sufficiently high classification accuracy, likely due to limited spectral resolution. In contrast, multispectral (MS) imaging offered a more detailed spectral representation, enabling the network to distinguish between materials that appear nearly identical in conventional image data. Through the use of MS filters capturing responses at distinct wavelengths, even subtle spectral differences became apparent, significantly enhancing classification performance. The overall structure and layer distribution of the proposed CNN model are presented in Figure 6.7. The CNN architecture used in this study was designed to classify five different liquid classes; hence, the output layer includes five neurons. Each input image of size 224×224 is passed through a DenseNet121-based architecture, resulting in a 7×7 output feature map before reaching the final dense layers. The fully connected layers include 256 neurons, and Leaky ReLU activation function is applied to enhance non-linearity and information flow.

Layer (type)	Output Shape	Param #
densenet121 (Functional)	(None, 7, 7, 1024)	7037504
batch_normalization (Batch Normalization)	(None, 7, 7, 1024)	4096
flatten (Flatten)	(None, 50176)	0
dense (Dense)	(None, 256)	12845312
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 5)	1285

Figure 6.7. Layer structure and parameters of the DenseNet based model

Backbone of the model, DenseNet121, consists of 121 layers and employs dense connectivity, allowing each layer to receive feature maps from all preceding layers. This structure improves feature distribution and reduces the vanishing gradient problem. For evaluation, a model inference approach was applied based on the pre-trained weights of DenseNet121, fine-tuned to the dataset.

In a study [34], to further investigate the spectral contributions of individual bands in the multispectral imaging setup, a series of experiments were conducted. In these experiments, the model was trained using data augmented from a single spectral band at a time. Each “band” corresponds to a specific wavelength region, capturing distinct spectral features. The results demonstrate that Band 4, Band 14, and Band 17 achieved the highest classification performance among all bands. For example, Band 4 yielded an F1 score of 0.80, while Band 14 and Band 17 reached 0.81 and 0.82, respectively. These results indicate that certain wavelengths carry more discriminative information, playing a key role in distinguishing visually similar liquid materials.

In another article [39] examines the effectiveness of visible and multispectral imaging (MSI) techniques in identifying varying levels of meat adulteration, particularly in mixtures containing pork and chicken. The experiments were conducted on both fresh and frozen-thawed samples, with classification accuracy evaluated across different mixing ratios. The results demonstrate that the visible imaging system exhibited performance limitations, with the lowest classification accuracy recorded as 58.33%. In contrast, the MSI approach showed significantly higher robustness, achieving a minimum accuracy of 87.50% and, in many cases, delivering near-perfect classification outcomes. This performance difference

highlights the superior capability of MSI in capturing spectral features that are not discernible in the visible spectrum.

Building upon this insight, the final model configuration using LeakyReLU activation and the Nadam optimizer achieved a test accuracy of 96.46% and an F1 score of 92.24% in this study. These results confirm that multispectral imaging, particularly when enhanced by well-selected spectral bands and a robust CNN architecture like DenseNet121. An inverse relationship is observed between the number of classes and test accuracy according to Table 6.7. and decreasing from 92.20% to 68.69% as the class count increases from 4 to 6. This decline in accuracy highlights the increasing complexity of multiclass classification tasks, especially when visually similar classes are involved. The use of multispectral (MS) imaging in our research significantly mitigated this issue. The MS modality achieved the highest test accuracy of 96.46%, even with five classes, which is notably higher than those reported in the ensemble CNN-based wound classification study.

Table 6.7. Test accuracies in different multiclass classification scenarios

Work	Material	Num of Classes	Classes	Test Accuracy (%)
Ref. [35]	Solid (Injuries)	5-class	BURN GRANULATING NECROTIC DEBRIDED SLOUGH SURGICAL WOUND	84.94
This study	Liquid	5-class	ALCOHOL ACETONE FLUX COLOGNE WATER	96.46
Ref. [39]	Solid (Meat)	3-class	BEEF MIXED MUTTON	93.33

The reported test accuracy for this method is 84.94%. In contrast, the second study focuses on the classification of visually similar liquid substances, namely alcohol, acetone, flux, cologne, and water. Utilizing an ensemble deep CNN-based classification approach, this study achieved a test accuracy of 96.46%. These results show that the examined method in the current study outperforms the earlier work, despite the challenge posed by the visual

similarity of the liquid classes. The high classification accuracy indicates the model's strong discriminative capability and its potential effectiveness in similar multi-class classification tasks.

6.2.2. Classification metrics

The distribution in the number of classes was tried to be kept equal. Ensuring equality in the distribution of data between classes is important for the performance of machine and deep learning models. An imbalance in the classes in the dataset may cause model to perform poorly in terms of the amount of data in the class. According to these results, the accuracy rates within each class were compared, except for the general accuracy result. Since these results show the rate at which each class is matched, accuracy values are taken as reference. The low margin between them shows that the matching is done properly and correctly. Confusion matrix is evaluated within the scope of 2-class classification because confusion matrix is a tool that visualizes the performance of a classification model by analyzing the predicted labels with the target labels. Therefore, Figure 6.8. and Figure 6.9. present the confusion matrices generated for the MS-SET1 (without data augmentation) and MS-SET3 (with data augmentation) test and validation datasets.

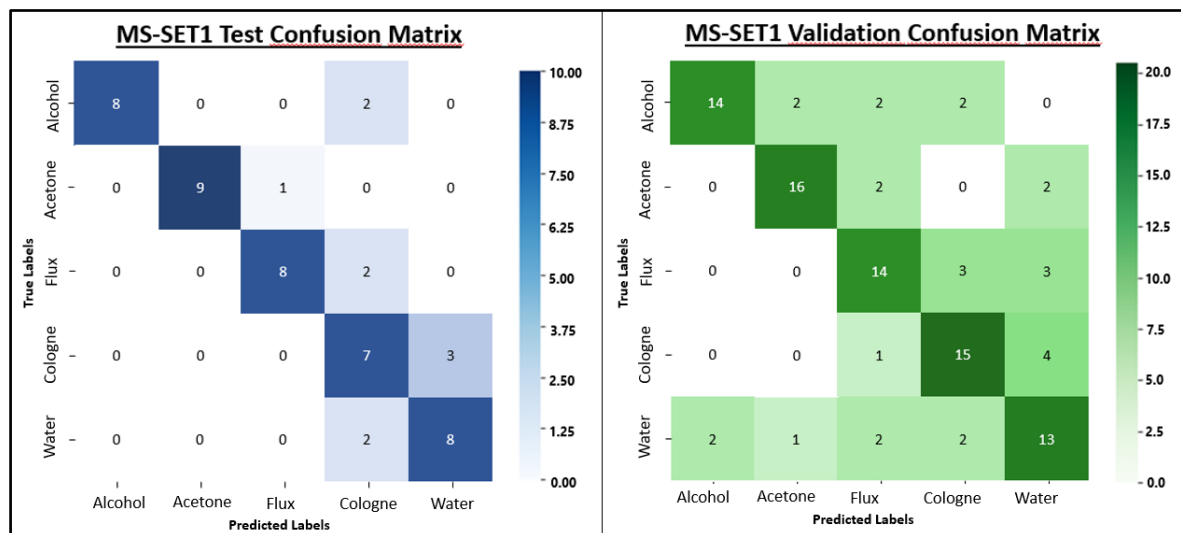


Figure 6.8. Confusion matrix for MS-SET1 test and validation datasets

The information content of the classes for MS-SET1 is detailed in the table below. Accordingly, "Test Precision", "Validation Precision", "Test Recall", "Validation Recall", "Test F1 Score", and "Validation F1 Score" are calculated separately for each class. These

metrics allow for a more detailed estimation of model's output on a per-class basis, beyond overall accuracy. The differences between test and validation scores also provide findings on the model's generalization capability and potential class-specific learning challenges. The values presented in the table are calculated based on the corresponding confusion matrices and then rounded for clarity. Decimal values were not explicitly written; instead, approximate figures were shared to enhance readability and facilitate interpretation.

Table 6.8. Classification performance metrics for MS-SET1 confusion matrix

Classes	Test Precision (%)	Test Recall (%)	Test F1 Score (%)	Validation Precision (%)	Validation Recall (%)	Validation F1 Score (%)
Alcohol	100	80	88	87	70	77
Acetone	100	90	94	84	80	82
Flux	88	80	84	66	70	68
Cologne	53	70	60	68	75	71
Water	72	80	76	59	65	61

When comparing these results, it is evident that although overall classification performance improved with data augmentation, the misclassifications tend to occur in similar classes across both datasets. This indicates that the remaining classification errors are likely due to inherent visual similarities between certain classes rather than insufficient training data. As a result, even after augmentation, the model continues to confuse visually similar samples, such as cologne and water, highlighting the challenge of inter-class similarity in spectral image classification.

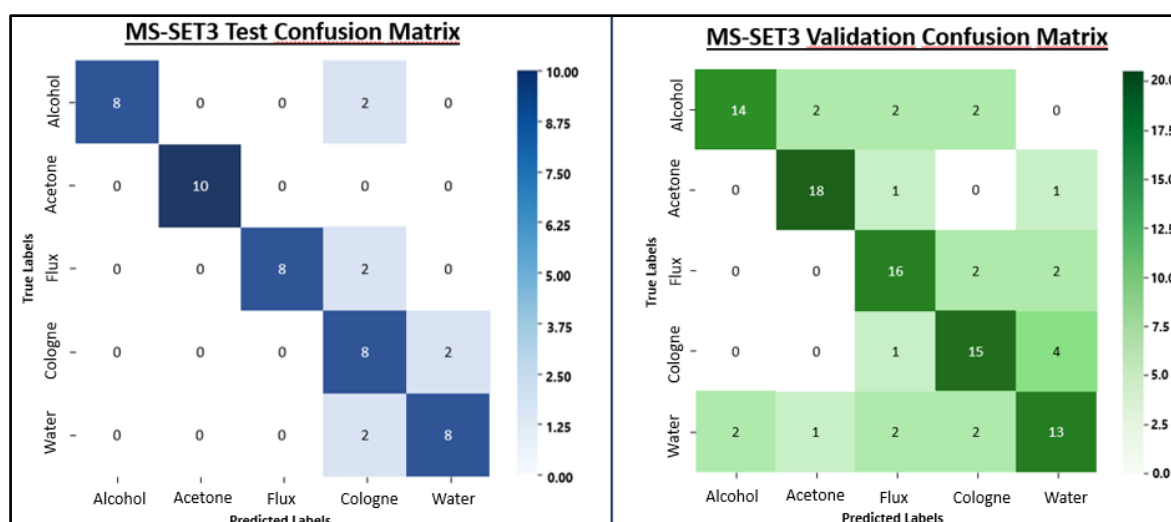


Figure 6.9. Confusion matrix for MS-SET3 test and validation datasets

The figure displays confusion matrices for the validation and test sets of a classification model. The validation confusion matrix (right) shows that the model performs well overall, but still presents several misclassifications, particularly between a few closely related classes. The test confusion matrix (left) reflects a similar pattern, with relatively good classification accuracy but with observable confusion between certain class pairs. The information content of the classes for MS-SET3 is detailed in the table below. Accordingly, "Test Precision", "Validation Precision", "Test Recall", "Validation Recall", "Test F1 Score", and "Validation F1 Score" are calculated separately for each class.

Table 6.9. Classification performance metrics for MS-SET3 confusion matrix

Classes	Test Precision (%)	Test Recall (%)	Test F1 Score (%)	Validation Precision (%)	Validation Recall (%)	Validation F1 Score (%)
Alcohol	100	80	89	88	70	78
Acetone	100	100	100	86	90	88
Flux	100	80	89	73	80	76
Cologne	57	80	67	68	75	71
Water	80	80	80	68	65	67

Results presented in the table reflect classification performance of the model for each class (alcohol, acetone, flux, cologne, and water) during both the test and validation phases. In the test set, the "Acetone" class achieved the highest performance with 100% precision and recall, while the "Cologne" class showed the lowest performance with a precision of 57%. This indicates that the model frequently misclassified samples from other classes as "Cologne." In contrast, precision and recall rates for the remaining classes were relatively high, suggesting that the model was generally effective in distinguishing between these categories. In the validation phase, the "Acetone" class once again yielded the highest F1 score at 88%, while the "Water" class had the lowest score at 67%. Overall, the metrics in the validation phase were slightly lower than those in the test phase. This difference suggests that the model adapted better to the training data, while its generalization capability varied across different classes. These metrics were calculated based on three fundamental evaluation criteria rather than solely relying on overall accuracy. Precision measures ratio of true positive predictions for a class to all predictions made for that class. For instance, in the test set, the model correctly predicted 8 "Cologne" samples (True Positives) and incorrectly labeled 6 samples from other classes as "Cologne" (False Positives). Therefore, precision is calculated as $8 / (8 + 6) = 0.57$. Recall is the ratio of accurately predicted "Cologne" samples

to the total actual instances of the "Cologne" class, computed as $8 / (8 + 2) = 0.80$. F1 score is the harmonic mean of precision and recall. These class-specific metrics are critical for evaluating not only the overall performance but also how balanced and reliable the model is across individual categories.

The consistency between validation and test results suggests that the model generalizes reasonably well. In machine learning, not only validation and test performances but also training performance plays a critical role in evaluating a model. While test and validation results indicate how well the model generalizes to unseen data, training results provide evidence of how effectively the model has learned from the data it was exposed to. A balanced assessment across all three components ensures a more reliable and comprehensive evaluation of the model's behavior. MS validation and training confusion matrix results are given Figure 6.10.

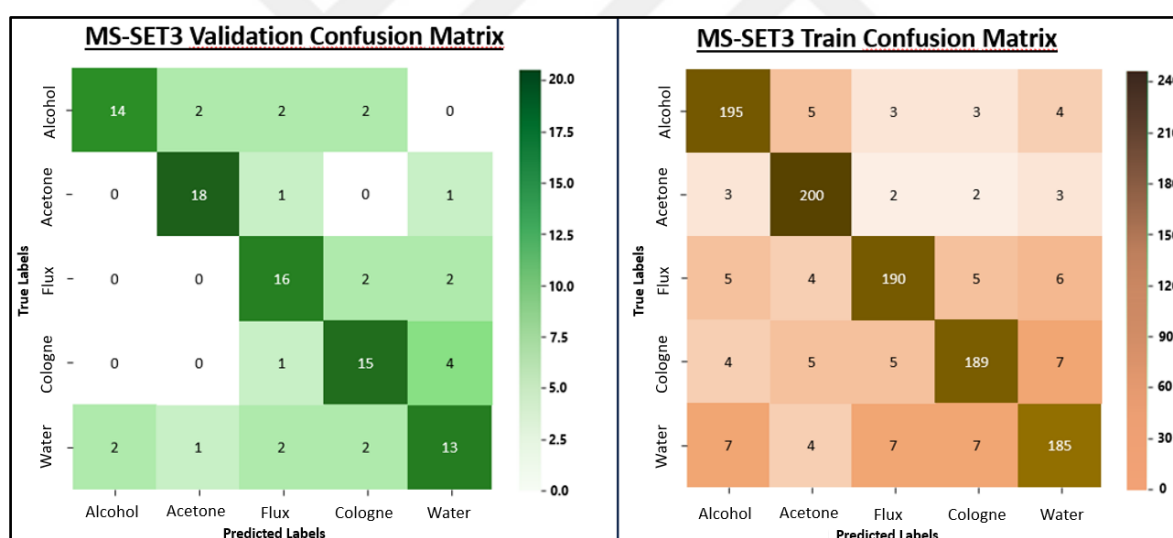


Figure 6.10. Confusion matrix for MS-SET3 validation and training datasets

This figure presents confusion matrices for the training and validation sets of the MS-SET3 dataset. The training confusion matrix (right) shows that the model performs very well on the training data, with a high number of correct predictions and minimal misclassifications across all classes.

The outcomes indicate that the model has effectively learned the training patterns. Overall, this comparison highlights the importance of evaluating not just test and validation sets, but also the training performance, to gain a full understanding of the model's behavior.

Figure 6.11. presents the confusion matrices for the SWIR-SET3 dataset, showing the model's performance on the test set (left) and validation set (right). In both matrices, there are noticeable misclassifications across multiple classes, indicating that the model struggles to distinguish clearly between certain categories. The validation matrix shows slightly better performance compared to the test set, with a higher number of correct classifications in some classes. These results suggest that the model may be slightly overfitting to the training data or that the class features in the SWIR domain are not distinct enough. Further optimization or additional data augmentation may be required to improve generalization performance.

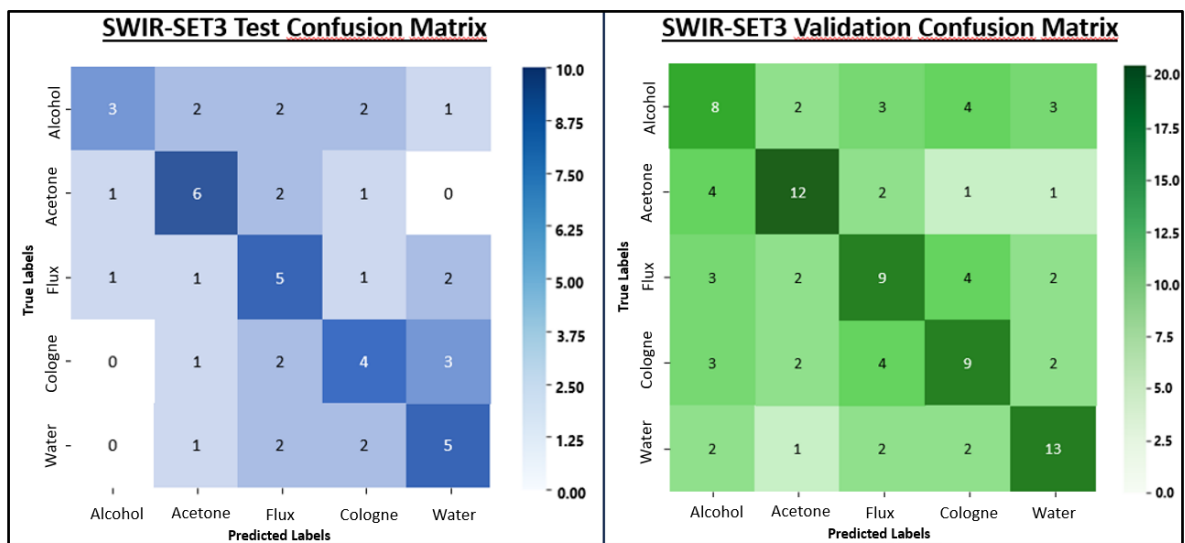


Figure 6.11. Confusion matrix for SWIR-SET3 test and validation datasets

The information content of the classes for SWIR-SET3 is detailed in the table below. Accordingly, "Test Precision", "Validation Precision", "Test Recall", "Validation Recall", "Test F1 Score", and "Validation F1 Score" are calculated separately for each class.

Table 6.10. Classification performance metrics for SWIR-SET3 confusion matrix

Classes	Test Precision (%)	Test Recall (%)	Test F1 Score (%)	Validation Precision (%)	Validation Recall (%)	Validation F1 Score (%)
Alcohol	50	30	37	40	44	42
Acetone	54	60	57	63	60	61
Flux	38	50	43	45	45	45
Cologne	40	40	40	50	45	47
Water	45	45	45	61	65	63

The overall distribution suggests that the model's ability to generalize using SWIR data is limited and could benefit from further tuning, additional data, or enhanced preprocessing to improve class separability. The confusion matrix results for the RGB data set are shared in Figure 6.12. According to these results, it is seen that the model experienced confusion between some classes in both test and validation datasets.

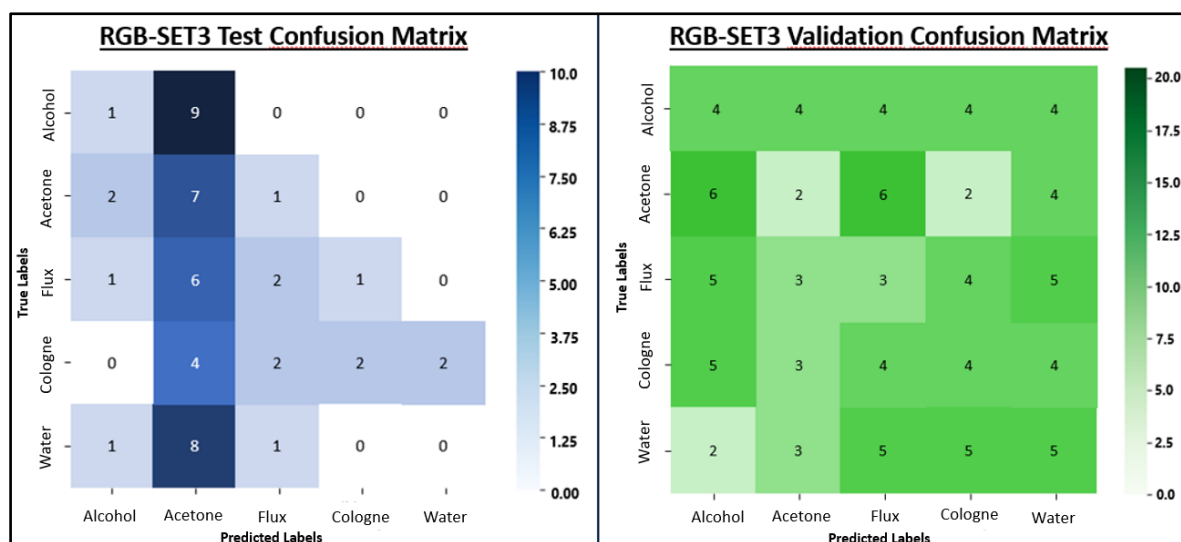


Figure 6.12. Confusion matrix for RGB-SET3 test and validation datasets

The test and validation confusion matrices created for the RGB data set show that the model experiences high confusion between certain classes. In particular, it is seen that some classes are frequently confused with each other and the model has low accuracy rates in certain classes. Model struggles to differentiate specific features between these classes, highlighting the need for further optimization. For the SWIR data set, it can be said that this situation is better than the RGB data set. The separation of the data is made more easily. However, it is not at the desired level. The confusion matrix distribution for the MS data set is generally consistent when the obtained test and validation accuracy values are examined.

Table 6.11. Classification performance metrics for RGB-SET3 confusion matrix

Classes	Test Precision (%)	Test Recall (%)	Test F1 Score (%)	Validation Precision (%)	Validation Recall (%)	Validation F1 Score (%)
Alcohol	25	10	14	18	20	19
Acetone	19	70	30	10	11	10
Flux	33	20	25	26	20	22
Cologne	50	20	28	20	25	22
Water	0	0	0	27	25	26

The information content of the classes for RGB-SET3 is detailed in the table below. Accordingly, "Test Precision", "Validation Precision", "Test Recall", "Validation Recall", "Test F1 Score", and "Validation F1 Score" are calculated separately for each class.

6.2.3. Performance visualization

This part focuses on the graphical representation of the model performance, including metrics such as test and validation accuracies also test and validation losses. Visualizations such as accuracy-loss curves, confusion matrices, and feature maps are provided to illustrate the results and analyze the model's effectiveness.

Graphs of losses and accuracies for test and validation output, architecture created using DenseNet and Adam selected as the initial approach in this work and yielding the best results are presented in Figure 6.13.

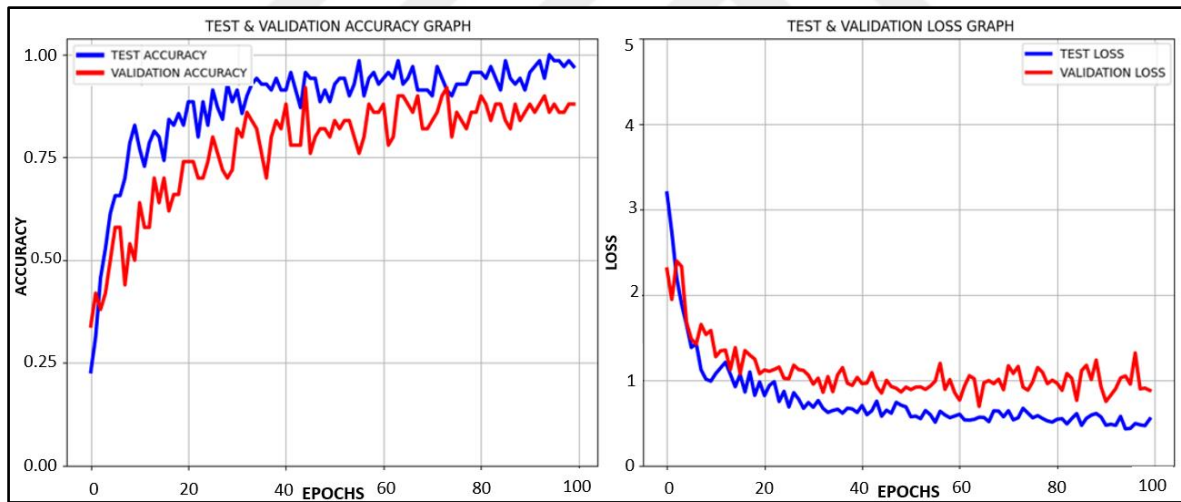


Figure 6.13. Multispectral imaging (SET3) accuracy (left) and loss (right) outputs for 5-class DenseNet – Nadam classification at 100 epochs

Performance of the model is good on the test data, with validation accuracy stabilizing around 80%. The test loss decreases consistently, demonstrating that the model is minimizing errors on the test set.

Validation loss initially decreases, and since there is an overall decreasing trend in losses. Results obtained from the DenseNet-Adam experiment conducted with different epoch numbers are presented in Figure 6.14.

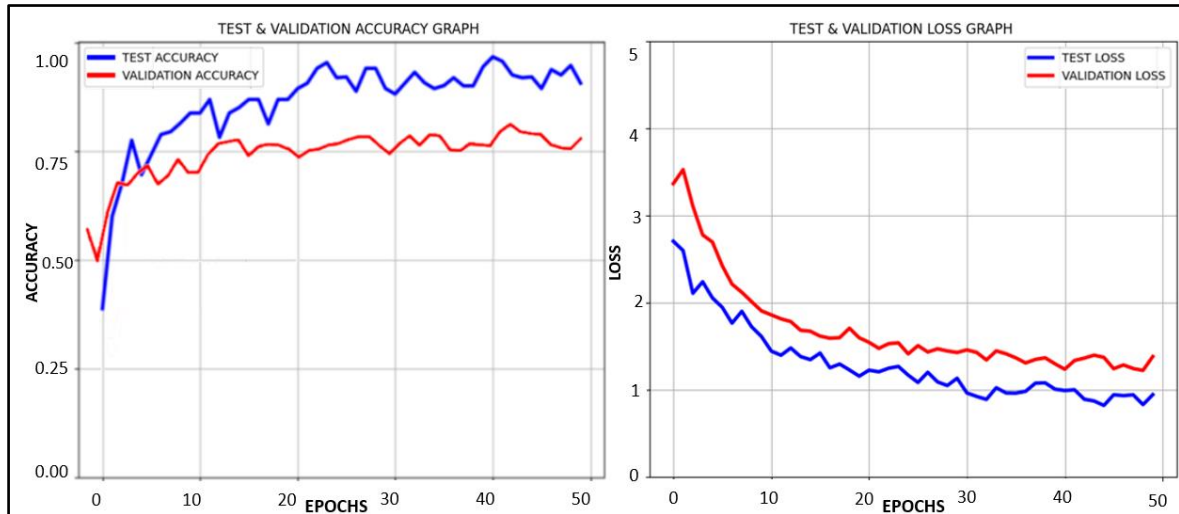


Figure 6.14. Multispectral imaging (SET3) accuracy (left) and loss (right) outputs for 5-class DenseNet – Nadam classification at 50 epochs

In addition, results of test performed using SWIR camera with the DenseNet - Adam configuration are shown in Figure 6.15. A test accuracy of 59.96% and a validation accuracy of 47.46% were obtained. Some oscillation in accuracy can be observed. Although there was a decreasing trend in the loss values, the validation loss still showed oscillations.

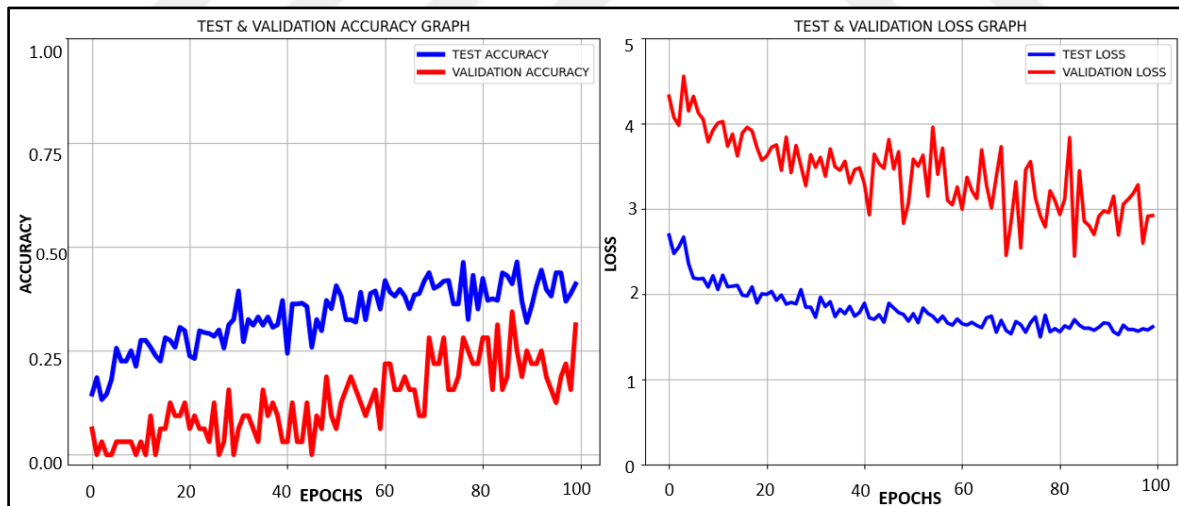


Figure 6.15. SWIR imaging (SET3) accuracy (left) and loss (right) outputs for 5-class DenseNet – Adam classification at 100 epochs

When the data obtained using the RGB imaging method for 5 different classes was tested with the DenseNet – Adam architecture. While the test loss remains stable, the validation loss value increases as the number of epochs increases and shown in Figure 6.16.

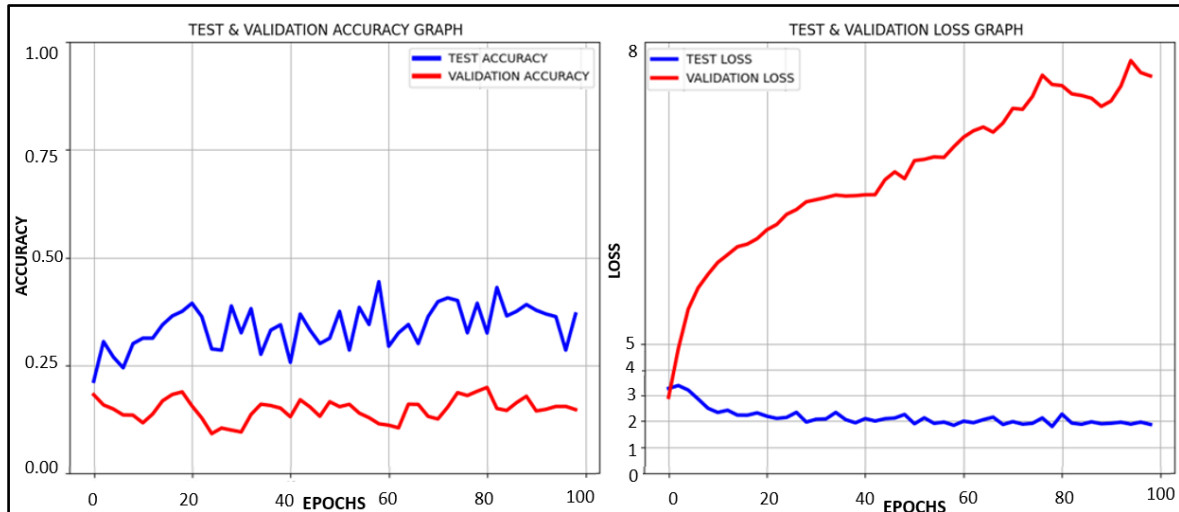


Figure 6.16. RGB imaging (SET3) accuracy (left) and loss (right) outputs for 5-class DenseNet – Nadam classification at 100 epochs

6.3 Additional Applied Techniques

In addition to the primary experimental setup, supplementary techniques were applied to further analyze and improve the model's output and robustness. The methods under this section helped to evaluate the consistency of results, reduce overfitting risk, and examine model performance in more generalized contexts.

6.3.1. K-Fold cross validation

K-Fold Cross Validation is a generally used evaluation method in machine learning that supports assess a model's ability to generalize to unseen data. Instead of relying on a single train-test split, which may lead to biased or unstable performance measurements, dataset is divided into K equal-sized folds. The model is then trained and validated K times, each time using a different fold as the validation set and remaining folds for training. After all iterations are done, the results are averaged to provide a more reliable estimate of the model's performance.

This method helps reduce variance and ensures that sample in the dataset is used exactly once for validation and K-1 times for training, thus offering a balanced and robust evaluation. Method is useful when the dataset is limited in size, as it maximizes the usage of available data.

In this study, 5-Fold Cross Validation was applied to validate the model trained with Nadam optimizer and LeakyReLU activation function. Original validation accuracy of 89.09% (obtained through a fixed validation set) for MS-SET3 slightly decreased to 85.86% after applying the K-Fold strategy.

Table 6.12. presents the results of 5-Fold Cross Validation performed on the model trained with the MS-SET3 dataset using the Nadam optimizer and LeakyReLU activation function. The average accuracy obtained across the five folds was 85.86%, with an average loss of 0.39.

Table 6.12. 5-fold cross validation results for MS-SET3

FOLD	Validation Accuracy (%)	Loss	Precision (%)	Recall	Validation F1 Score (%)
1	84.20	0.42	86.75	0.83	84.85
2	87.10	0.39	88.02	0.86	87.00
3	85.90	0.37	89.30	0.84	86.60
4	86.40	0.38	87.85	0.86	86.90
5	85.70	0.40	86.10	0.82	84.95
Average	85.86	0.39	87.60	0.84	86.06

Compared to fixed validation set results, a moderate decline in performance is observed. This decrease aligns with the expectations outlined in model evaluation literature and reflects a more realistic estimation of the model's generalization capability across varying data splits. These results confirm that although the initial fixed-split validation yielded higher scores, the model still maintains strong and consistent performance when evaluated using a robust K-Fold validation strategy. Method also shows that initial fixed validation split may have provided a somewhat optimistic estimate.

Moreover, K-Fold Cross Validation allowed for a deeper analysis of class-level performance consistency across folds. It was observed that certain classes were consistently predicted more accurately, while others showed higher variability in predictions. This insight can guide future work in data balancing, feature engineering, or class-specific augmentation to improve weaker areas of the model. Structure of 5-Fold cross validation and performance averaging is shown in Figure 6.17.

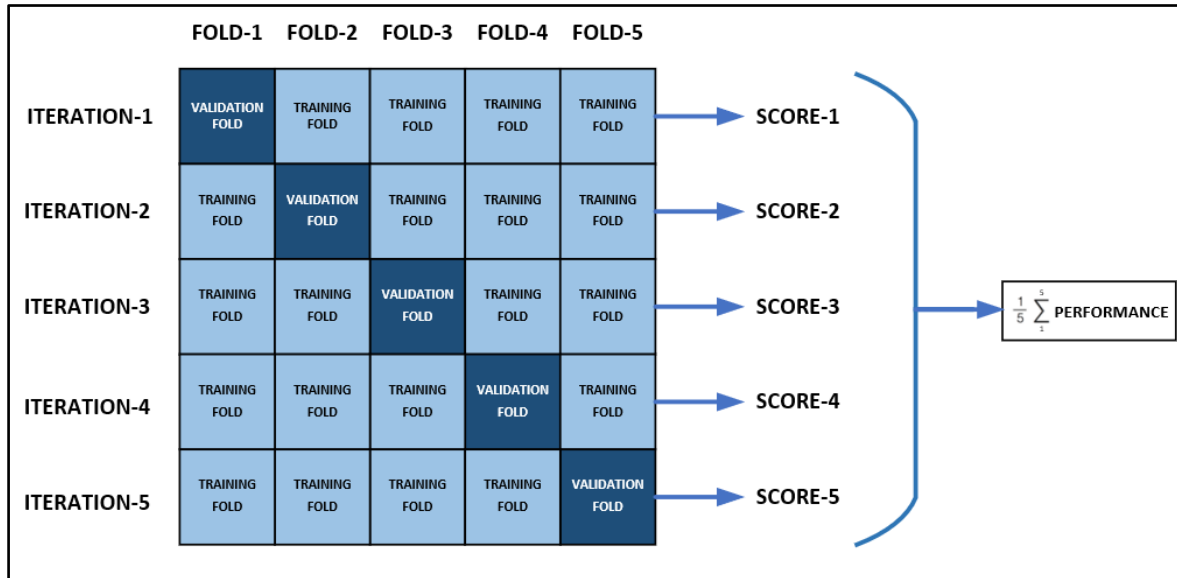


Figure 6.17. Structure of 5-Fold cross validation and performance averaging

The figure demonstrates the process of 5-Fold Cross Validation, where the dataset is divided into five equal parts. In each iteration, one part is used for validation while the remaining four are used for training. This process is repeated five times, ensuring that every subset is used once as the validation set. At the end of all iterations, the performance scores from each round are averaged to obtain a reliable estimate of the model's overall performance. This approach provides a more balanced and generalizable evaluation of the model.

6.3.2. Data merge application

The experiments on MS, SWIR and RGB data sets were carried out separately. In this part, all data were combined and the algorithm was tested. The results obtained from the experiments conducted at 100 epochs are shown in Table 6.13.

Table 6.13. Merged image dataset results

Results	Merged Image Dataset (MS, SWIR, RGB)
Test Accuracy (%)	71.21
Validation Accuracy (%)	58.97
Test Loss	0.94
Validation Loss	1.75
F1 Score	59.43

Table shows the test results of the dataset created by combining MS, SWIR and RGB data. The test accuracy was calculated as 71.21%, which indicates that the model has limited performance on the combined dataset. The validation accuracy was determined as 58.97%, which is slightly lower than the test accuracy.

Test loss was calculated as 0.94 and the validation loss was calculated as 1.75. A higher validation loss than test loss may indicate that model made more errors on validation data. The F1 score was determined as 59.43, which may indicate that the model does not exhibit a balanced classification performance.



7. CONCLUSION AND FUTURE WORK

This study investigates the use of multispectral, short-wave infrared, and visible imaging techniques in combination with deep learning architectures to classify five visually similar liquid substances that cannot be distinguished by the human eye or standard RGB cameras. These materials share similar visual characteristics, making them particularly challenging to classify using conventional imaging approaches. Emphasizing the effectiveness of multispectral imaging, which captures spectral responses at discrete wavelengths, the study demonstrates that MS imaging is far more successful in differentiating such substances based on their spectral signatures. Experimental findings revealed that MS imaging achieved a 5-class classification accuracy of 96.46%, significantly outperforming SWIR (59.96%) and RGB (44.16%) methods.

In terms of model selection, Convolutional Neural Networks were found to be the most suitable architecture for this classification task due to their powerful feature extraction capabilities. Various configurations were systematically tested, including different activation functions, optimization algorithms, epoch values, and data augmentation parameters. Among these, the combination of LeakyReLU activation function and Nadam optimizer produced the most successful results, with the highest accuracy and lowest loss values observed in both test and validation phases. This finding underscores the importance of selecting appropriate training configurations when working with complex image-based classification tasks.

Given the relatively small size of the original dataset, the study placed particular emphasis on the use of data augmentation techniques to artificially increase number of training samples and improve model generalization. Techniques such as rotation, scaling, width and height shifting, shearing, and flipping were applied at empirically determined values. Notably, a rotation range of 40 degrees and a transformation factor of 0.3 for other augmentation parameters were found to be the most effective. These enhancements led to measurable improvements in performance, validating the use of augmentation in data-limited environments.

To further support the model evaluation, confusion matrices were generated and analyzed, providing insight into class-level accuracy and misclassification patterns. These matrices confirmed the consistency between overall test accuracy and per-class performance, illustrating that the results were both statistically sound and practically interpretable. Additionally, when a mixed-data training approach was applied by combining all MS, SWIR, and RGB datasets the model achieved a test accuracy of 55.21% and a validation accuracy of 51.06%, suggesting that multi-modal integration, while slightly better than RGB alone, may introduce complexity without yielding proportional benefits.

In conclusion, this study demonstrates that MS imaging combined with deep learning constitutes a highly effective approach for the classification of visually similar liquid substances. Findings highlight the potential of MS data in applications where subtle spectral differences must be detected. For future research, the integration of multiple architectures, larger and more diverse datasets, and advanced ensemble learning strategies could further enhance performance. Additionally, exploring novel imaging technologies and refining model configurations based on specific application requirements may lead to more robust and scalable solutions in the domains of food authentication, material inspection, and quality control.

REFERENCES

1. Salimi, M., Roshanfar, M., Tabatabaei, N. and Mosadegh, B. (2023). Machine learning-assisted short-wave infrared (SWIR) techniques for biomedical applications: Towards personalized medicine. *Journal of Personalized Medicine*, 14(1), 33.
2. Gerken, M. and Schlemmer, H. (2022). *Multispectral imaging in the SWIR: theory, practical approach, and expected performance*. Infrared Sensors, Devices, and Applications XII. SPIE, San Diego, CA. 12234, 122340D.
3. Bertozzi, M., Fedriga, R. I., Miron, A. and Reverchon, J. L. (2013). *Pedestrian detection in poor visibility conditions: Would SWIR help?* In A. Petrosino (Ed.), *Image Analysis and Processing – ICIAP 2013*. Springer, Berlin, Heidelberg. 629–638
4. Rosenberger, M., Dittrich, P. G., Illmann, R., Horn, R., Golomoz, A., Notni, G., Eiser, S., Hirst, O. and Jahn, N. (2022). Multispectral imaging system for short wave infrared applications. *Sensors*, 22(9), 3472.
5. Elmasry, G., Mandour, N., Al-Rejaie, S., Belin, E., and Rousseau, D. (2019). Recent Applications of Multispectral Imaging in Seed Phenotyping and Quality Monitoring—An Overview. *Sensors*, 19(5), 1090.
6. Jenal, A., Bareth, G., Bolten, A., Kneer, C., Weber, I., and Bongartz, J. (2019). Development of a VNIR/SWIR multispectral imaging system for vegetation monitoring with unmanned aerial vehicles. *Remote Sensing*, 11(17), 1994.
7. Adige, S., Kurban, R., Durmuş, A., and Karaköse, E. (2022). Classification of apple varieties based on their types using image processing techniques. *European Journal of Science and Technology*, (37), 131–138.
8. Hou, J., Che, Y., Fang, Y., Bai, H., and Sun, L. (2024). Early bruise detection in apple based on an improved Faster R-CNN model. *Computers and Electronics in Agriculture*, 212, 108233.
9. Pavlović, M. S., Milanović, P. D., Stanković, M. S., Perić, D. B., Popadić, I. V., and Perić, M. V. (2022). Deep learning-based SWIR object detection in long-range surveillance systems: An automated cross-spectral approach. *Sensors*, 22(14), 5284.
10. Puerto, A., Pedraza, C., Jamaica-Tenjo, D. A., and Osorio Delgado, A. (2020). A deep learning approach for weed detection in lettuce crops using multispectral images. *Sensors*, 20(18), 5137.
11. Yuan, K., Zhuang, X., Schaefer, G., Feng, J., Guan, L., and Fang, H. (2021). Deep-learning-based multispectral satellite image segmentation for water body detection. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 1306–1317.
12. Wang, S., Xia, X., Ye, L., and Yang, B. (2021). Automatic detection and classification of steel surface defects using deep convolutional neural networks. *Applied Sciences*, 11(3), 1000.

13. Li, Z., Liu, F., Yang, W., Peng, S. and Zhou, J. (2020). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12), 6999–7019.
14. Zhu, C., Wang, L., Zhao, W. and Lian, H. (2024). Image classification based on tensor network DenseNet model. *Applied Intelligence*, 54, 6120–6136.
15. Sharma, R., Singh, A., Kavita, Jhanjhi, N., Masud, M. and Verma, P. (2021). Plant disease diagnosis and image classification using deep learning. *Computers, Materials and Continua*, 66(2), 1581–1596.
16. Yang, L., Xu, S., Yu, X. Y., Long, H. B. and Zhang, H. H. (2023). A new model based on improved VGG16 for corn weed identification. *Frontiers in Plant Science*, 14, 1123456.
17. Reddi, S. J., Kale, S. and Kumar, S. (2018). *On the convergence of Adam and beyond*. Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, Canada.
18. Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning*. Cambridge, MA: MIT Press.
19. Ogundokun, R., Maskeliūnas, R., Misra, S., Damaševičius, R. (2022). *Improved CNN based on batch normalization and Adam optimizer*. *Sensors*, 22(17), 6549.
20. Rogalski, A. (2020). Infrared detector characterization. *Infrared Physics and Technology*, 104, 103098.
21. Goss, T., Viljoen, J. and Fourie, H. (2019). *SWIR sensor design considerations*. Proceedings of the 2019 International Conference on Infrared, Millimeter, and Terahertz Waves. Paris, France. 1–3.
22. Kamalov, F., Nazir, A., Safaraliev, M., Cherukuri, A. K. and Zgheib, R. (2021). *Comparative analysis of activation functions in neural networks*. Proceedings of the 2021 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS). Dubai, United Arab Emirates. 1–4.
23. Raschka, S. and Mirjalili, V. (2019). *Python Machine Learning* (2nd ed.). Birmingham: Packt Publishing.
24. Nalepa, J. (2021). Recent advances in multi- and hyperspectral image analysis. *Sensors*, 21(18), 6002.
25. Elmasry, G., Mandour, N., Al-Rejaie, S., Belin, E. and Rousseau, D. (2019). Recent applications of multispectral imaging in seed phenotyping and quality monitoring—An overview. *Sensors*, 19(5), 1090.
26. Thomas, J.-B. (2020). Multispectral imaging for computer vision. *Signal and Image Processing: An International Journal*, 11(1), 1–12.

27. Kazemi, A. and Andersson, J. (2013). *Infrared detectors: Advances, challenges and new technologies*. In IOP Conference Series: Materials Science and Engineering. Yokohama, Japan. 51, 012005. IOP Publishing.
28. Hou, F., Zhang, Y., Zhou, Y., Zhang, M., Lyu, B. and Wu, J. (2022). Review on infrared imaging technology. *Infrared Physics & Technology*, 123, 104139.
29. Gani, M., Kuiry, S., Das, A., Nasipuri, M. and Das, N. (2021). *Multispectral object detection with deep learning*. In 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, pp. 121–125.
30. Parelus, E. J. (2023). A review of deep-learning methods for change detection in multispectral remote sensing images. *Remote Sensing*, 15(8), 2092.
31. Rawat, W. and Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9), 2352–2449.
32. Bishop, C. M. (2006). *Pattern recognition and machine learning* (1st ed.). New York: Springer.
33. Zwinkels, J. (2015). Light, electromagnetic spectrum. In *Encyclopedia of Color Science and Technology* (pp. 1–5). New York: Springer.
34. Zhang, Y., Yilmaz, A., Popa, M. and Brewster, C. (2023). Analysis of the impact of data augmentation on the performance of deep learning models in multispectral food authenticity identification. *Sensors*, 23(5), 2150.
35. Rostami, B., Anisuzzaman, D. M., Wang, C., Gopalakrishnan, S., Niezgoda, J. and Yu, Z. (2021). Multiclass wound image classification using an ensemble deep CNN-based classifier. *Computers in Biology and Medicine*, 134, 104469.
36. Cheng, X., Sun, Y., Zhang, W., Wang, Y., Cao, X. and Wang, Y. (2023). Application of deep learning in multitemporal remote sensing image classification. *Remote Sensing*, 15(4), 945.
37. Yavuz, B. (2020). *Recognition and classification of wheat kernel images obtained using RGB, VNIR and SWIR cameras through different fusion strategies*. Master's Thesis, Eskişehir Osmangazi University, Eskişehir, Department of Computer Engineering.
38. Jorden, P. R., Jerram, P., Fryer, M. and Stefanov, K. (2017). e2v CMOS and CCD sensors and systems for astronomy. *Journal of Instrumentation*, 12, C03088.
39. Fengou, L., Lianou, A., Tsakanikas, P., Mohareb, F. and Nychas, G.-J. (2021). Detection of meat adulteration using spectroscopy-based sensors. *Foods*, 10(6), 1300.
40. Alan, A. ve Karabatak, M. (2020). Veri seti - sınıflandırma ilişkisinde performansa etki eden faktörlerin değerlendirilmesi. *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 32(1), 35–45.
41. Polatoğlu, A. and Özkesen, İ. C. (2022). Working principles of CCD and CMOS sensors and their place in astronomy. *Journal of Anatolian Physics and Astronomy*, 2(1), 51–59.

42. Malinen, J., Kemppainen, A., Kansakoski, M. and Hietala, E. (2003). Application of NIR sensor modules for process measurements. *Sensors and Actuators B: Chemical*, 91(1–3), 223–228.
43. Ofer, O., Elishkov, R., Friedman, R., Reches, G., Hirsch, I., Langoff, L. and Fraenkel, R. (2013). *Multi-function InGaAs detector for SWIR imaging*. Proceedings of SPIE – The International Society for Optical Engineering, 8704, 87040T. SPIE, Baltimore, MD.
44. Razeghi, M., Dehzangi, A. and Li, J. (2021). Multi-band SWIR-MWIR-LWIR Type-II superlattice based infrared photodetector. *Results in Optics*, 2, 100058.
45. Wilson, A., Gupta, K., Koduru, B. H., Kumar, A., Jha, A. and Cenkeramaddi, L. R. (2023). Recent advances in thermal imaging and its applications using machine learning: A review. *IEEE Sensors Journal*, 23(6), 4561–4575.
46. Nalepa, J. (2021). Recent advances in multi- and hyperspectral image analysis. *Sensors*, 21(17), 6002.
47. Özen, Ö., Akkoyun, F., Görgüç, A. and Yılmaz, F. (2024). Multispektral ve hiperspektral görüntüleme tekniklerinin meyve-sebze işleme tesislerinde kullanım olanakları. *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi*, 27(1), 643–656.
48. Lu, B., He, Y. and Dao, P. (2019). Comparing the performance of multispectral and hyperspectral images for estimating vegetation properties. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(5), 1784–1797.
49. Shin, J. I., Seo, W. W., Kim, T., Park, J. and Woo, C. S. (2019). Using UAV multispectral images for classification of forest burn severity: A case study of the 2019 Gangneung Forest fire. *Forests*, 10(12), 1105.
50. LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
51. Ma, Y. and Tang, J. (2021). Deep learning on graphs: An introduction. In *Deep Learning on Graphs* (pp. 1–16). Cambridge: Cambridge University Press.
52. Ang, J. C., Mirzal, A., Haron, H. and Hamed, H. N. A. (2015). Supervised, unsupervised, and semi-supervised feature selection: A review on gene selection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 13(5), 971–989.
53. Thomas, P. (2009). Semi-Supervised Learning (Review). *IEEE Transactions on Neural Networks*, 20(3), 542.
54. Jing, L. and Tian, Y. (2021). Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7), 3252–3271.
55. Naeem, M., Rizvi, S. and Coronato, A. (2020). A gentle introduction to reinforcement learning and its application in different fields. *IEEE Access*, 8, 209320–209344.

56. Kaddoura, S., Popescu, D. and D, J. (2022). A systematic review on machine learning models for online learning and examination systems. *PeerJ Computer Science*, 8, e123.
57. Ren, P., Xiao, Y., Chang, X., Huang, P. Y., Li, Z., Gupta, B. and Wang, X. (2021). A survey of deep active learning. *ACM Computing Surveys (CSUR)*, 54(9), 1–40.
58. Wang, F., Cao, Z., Tan, L. and Zong, H. (2020). Survey on learning-based formal methods: Taxonomy, applications and possible future directions. *IEEE Access*, 8, 181482–181498.
59. Hammad, M. (2024). Artificial neural network and deep learning: Fundamentals and theory. *Artificial Intelligence Review*, 57, 1101–1120.
60. Yuan, J., Pan, F., Zhou, C., Qin, T. and Liu, T. Y. (2021). Learning structures for deep neural networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 12(2), 1–21.
61. Akgül, İ. (2023). Activation functions used in artificial neural networks. *International Journal of Engineering Research and Development*, 15(2), 45–52.
62. Sundaram, N. M. and Sivanandam, S. N. (2018). Softmax activation function for neural network multi-class classifiers. *Karpagam Journal of Computer Science*, 12(4), 23–27.
63. Hamdan, M. and Roach, D. (2022). The sigmoid neural network activation function and its connections to Airy's and the Nield-Kuznetsov functions. *PROOF*, 2(1), 50–61.
64. Dubey, S. R., Singh, S. K. and Chaudhuri, B. B. (2021). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503, 1–38.
65. Li, M. (2024). Comprehensive review of backpropagation neural networks. *Academic Journal of Science and Technology*, 8(1), 25–38.
66. Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6, 60.
67. Rosset, S. (2008). Sparse, flexible and efficient modeling using L1 regularization. *Computational Statistics and Data Analysis*, 52(1), 481–490.
68. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929–1958.
69. Ioffe, S. and Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. In Proceedings of the 32nd International Conference on Machine Learning. Lille, France. 448–456
70. Terven, J., Cordova-Esparza, D. M., Ramirez-Pedraza, A. and Chávez Urbiola, E. (2023). Loss functions and metrics in deep learning: A review. *Artificial Intelligence Review*, 56, 1349–1383.
71. Sazlı, M. (2006). A brief review of feed-forward neural networks. *Communications Faculty of Sciences University of Ankara Series A2-A3*, 50, 11–17.

72. Ghosh, A., Sufian, A., Sultana, F., Chakrabarti, A. and De, D. (2020). Fundamental concepts of convolutional neural network. *International Journal of Advanced Computer Science and Applications*, 11(12), 1–13.
73. Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404, 132306.
74. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). *Generative adversarial nets*. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, Canada, pp. 2672–2680.
75. He, K., Zhang, X., Ren, S. and Sun, J. (2016). *Deep residual learning for image recognition*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, pp. 770–778.
76. Simonyan, K. and Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, USA.
77. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2015). *Going deeper with convolutions*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, USA, 1–9.
78. Huang, G., Liu, Z., Pleiss, G., Van Der Maaten, L., Weinberger, K. Q. (2019). Convolutional networks with dense connectivity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8), 1865–1880.
79. Bianco, S., Cadène, R., Celona, L., Napoletano, P. (2018). Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6, 64270–64277.
80. Zhang, C., Benz, P., Argaw, D. M., Lee, S., Kim, J., Rameau, F., Bazin, J. C., Kweon, I. (2021). *ResNet or DenseNet? Introducing dense shortcuts to ResNet*. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, Hawaii, USA, 3549–3558.
81. Kim, D., Heo, B., Han, D. (2021). *DenseNets Reloaded: Paradigm Shift Beyond ResNets and ViTs*. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, Hawaii, USA, 3549–3558.



Gazili olmak ayrıcalıktır