INDOOR SCENE CLASSIFICATION USING MACHINE LEARNING ON OBJECT-DETECTION BASED FEATURES

SIMON YOSBOON

MASTER OF ENGINEERING
IN
COMPUTER ENGINEERING

SCHOOL OF INFORMATION TECHNOLOGY
MAE FAH LUANG UNIVERSITY

2024

INDOOR SCENE CLASSIFICATION USING MACHINE LEARNING ON OBJECT-DETECTION BASED FEATURES

SIMON YOSBOON

THIS THESIS IS A PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF ENGINEERING IN COMPUTER ENGINEERING

SCHOOL OF INFORMATION TECHNOLOGY
MAE FAH LUANG UNIVERSITY

2024

INDOOR SCENE CLASSIFICATION USING MACHINE LEARNING ON OBJECT-DETECTION BASED FEATURES

SIMON YOSBOON

THIS THESIS HAS BEEN APPROVED
TO BE A PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF ENGINEERING
IN
COMPUTER ENGINEERING
2024

EXAMINATION COMMITTEE

.....CHAIRPERSON
(Surapong Uttama, Ph. D.)

.....ADVISOR
(Khwunta Kirimasthong, Ph. D.)

.....CO-ADVISOR
(Prof. Wg. Cdr. Tossapon Boongoen, Ph. D.)

.....EXTERNAL EXAMINER
(Assoc. Prof. Wg. Cdr. Thiansiri Luangwilai, Ph. D.)

ACKNOWLEDGEMENTS

In the journey of crafting this thesis, profound gratitude is owed to the steadfast guidance and expertise of Dr. Khwunta Kirimasthong and Prof. Dr. Wg. Cdr. Tossapon Boongoen. Their support, mentorship, and scholarly insights have profoundly shaped the trajectory and quality of this research endeavor.

Acknowledgment is equally due to the esteemed members of the thesis committees, whose constructive critique and scholarly acumen have played a pivotal role in refining the depth and breadth of this work.

The indispensable support provided by Mae Fah Luang University deserves sincere appreciation. Their provision of facilities and financial backing has not only facilitated the necessary resources but has also catalyzed rigorous investigation and analysis, enriching the overall quality of this thesis. I would like to thank Mae Fah Luang University for providing a scholarship to support my studies for a master's degree. Additionally, I am deeply grateful for the grant specifically awarded for thesis research, which has enabled me to conduct thorough and focused research, significantly enhancing the depth and quality of my work.

Lastly, heartfelt appreciation is extended to the participants who generously shared their time, knowledge, and experiences, enriching the understanding within this research domain.

Simon Yosboon

Thesis Title	Indoor Scene Classification Using Machine Learning on Object-detection Based Features
Author	Simon Yosboon
Degree	Master of Engineering (Computer Engineering)
Advisor	Khwunta Kirimasthong, Ph. D.
Co-Advisor	Prof. Wg. Cdr. Tossapon Boongoen, Ph. D.

ABSTRACT

The classification of scenes from images is a fundamental task in computer vision, vital for various applications ranging from autonomous driving to surveillance systems. An ongoing challenge in this field is the identification of discriminative features for accurate classification. This study addresses this challenge by comparing the effectiveness of two approaches: object-based feature extraction and deep learning.

We propose a novel methodology that leverages YOLOv3, a state-of-the-art pre-trained model for object detection, to extract object-based features from scene images. By utilizing YOLOv3, we obtain feature vectors representing the presence and characteristics of objects within each scene. These features are then used as input for four distinct machine learning algorithms to classify scenes.

Concurrently, we develop a deep learning model using the original images, which typically requires more computational resources and time for training. We conduct comprehensive experiments to evaluate the performance of both approaches across various scene classification tasks.

Surprisingly, our results demonstrate that simple machine learning models utilizing object-level features achieve comparable performance to deep learning

methods. This finding suggests that focusing on object-based representations can effectively classify scenes while circumventing the resource-intensive nature of deep learning algorithms.

Keywords: Deep learning, Training, Machine Learning Algorithms, Computational Modeling, Object Detection, Feature Extraction, Convolutional Neural Networks

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Scene classification has become an important and widely studied problem in computer vision due to its numerous practical applications. The ability to categorize images into different scene categories has been shown to be useful in a variety of fields, including autonomous driving, surveillance, and image retrieval. In this chapter, we will provide background information and the rationale behind our research in scene classification, outlining the key challenges and opportunities in this area. We will also present our research objectives, the scope of our study, the expected results, and the location and equipment used for our experiments. By the end of this chapter, readers should have a clear understanding of our research goals and methodology, as well as a comprehensive overview of the key concepts and issues involved in scene classification.

1.1 Background and Rational

In recent years, scene classification has gained popularity as a research area among scientists due to the increasing number of studies that have emerged recently, including works by Cheng et al. (2018), Liu et al. (2019) and Zeng et al. (2018), which provide evidence. The goal of scene classification is to develop a method for automatically categorizing images of different scenes into pre- defined categories. However, unlike humans, machines have difficulty identifying the relevant features in an image that are necessary for accurate classification. As a result, scientists must determine the appropriate level of detail for the features used in the classification process, such as fine- grained textures and shapes, or recognizable objects and components. In this research work, our goal is to conduct indoor scene classification, which consists of four types of room scenes: bathroom, bedroom, living room, and

kitchen. We aim to use simple machine learning methods and compare their performance in terms of classification accuracy with the baseline Convolutional Neural Network (CNN). This will help us determine if the alternative approaches can achieve similar levels of performance in the scene classification task.

1.2 Objective

Our research aims to compare the performance of machine learning and deep learning models. To achieve this goal, we plan to train machine learning models using YOLO, a state-of-the-art object detection model. Specifically, we will develop an object-based feature extractor for machine learning model training, leveraging the capabilities of YOLO. Our assumption of this approach is to enhance the performance of the machine learning models, particularly in tasks like image classification. We selected four common types of indoor scenes, including bathroom, bedroom, living room, and kitchen, because they represent the most common indoor areas in a typical household.

Additionally, we aim to compare the performance of our machine learning models with the base-line deep leaning that is the model in terms of classification accuracy. Deep learning has recently gained a lot of attention in the field of artificial intelligence and has been shown to achieve impressive results in various tasks. By comparing the performance of simple machine learning with base-line CNN models, we hope to gain a better understanding of the advantages and limitations of different approaches to machine to perform indoor scene classification.

1.3 Scope

1.3.1 Apply object-based feature extraction for machine learning models training.

1.3.2 Train and classify four types of indoor scene data set consists of bedroom, restroom, living room and kitchen.

1.3.3 Train machine learning and base-line CNN models four types of indoor

scene data set.

1.3.4 Perform scene classification with machine learning and base-line CNN model.

1.3.5 Compare the accuracy performance of machine learning and baseline CNN model.

1.4 Expected Result

1.4.1 Able to extract features using YOLOv3 from images for machine learning models.

1.4.2 Train and validate machine learning and base-line CNN models.

1.4.3 Get classification accuracy results for machine learning and base line CNN model.

1.4.4 Able to compare the performance of machine learning and base-line CNN model.

1.5 Equipment

1.5.1 Software

1.5.1.1 Python version 3.8

1.5.1.2 Ubuntu 18.04

1.5.1.3 YOLOv3

1.5.1.4 Jupyter Notebook

1.5.2 Hardware

1.5.2.1 CPU 16 cores

1.5.2.2 Memory 16 GB

1.5.2.3 Storage 500 GB

CHAPTER 2

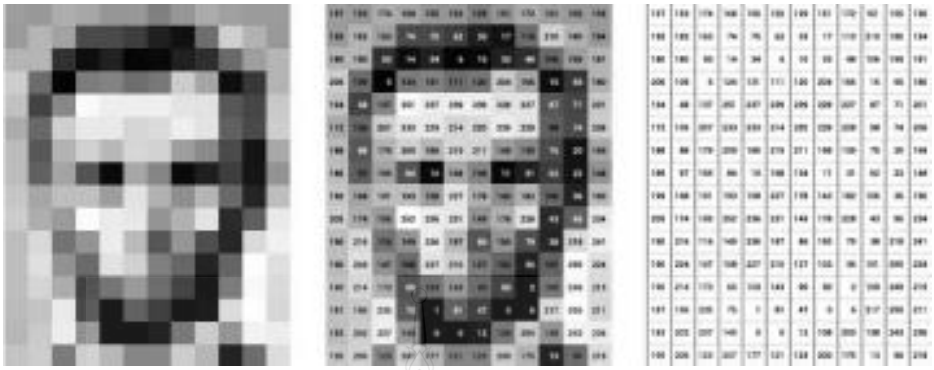
LITERATURE REVIEW

In this chapter, we will provide a comprehensive review of the literature on scene classification. Specifically, we will examine recent advancements in scene classification research, including the related technology and existing work that have focused on improving scene classification accuracy.

2.1 Related Technology

2.1.1 Computer Vision and Image Processing

Computer vision and image processing are important fields of study in the area of artificial intelligence, which have gained significant attention in recent years due to their many real-world applications. Computer vision involves developing algorithms that can automatically extract, analyze, and understand information from visual data, while image processing involves manipulating digital images to improve their quality, extract features, or perform other operations. These fields have numerous applications, such as object recognition (Xie et al., 2021), autonomous driving (Chen, Li et al., 2024), and medical imaging (Shen et al., 2017). Recent advancements in deep learning have led to significant improvements in computer vision tasks such as object detection, segmentation, and classification. For a comprehensive overview of the latest advancements in computer vision and image processing, see the book *Computer Vision: Models, Learning, and Inference* by Jeremy (2014). The diagram in figure 2.1 displays pixel data of an image of Lincoln with values ranging from 0 to 255. It highlights digital image representation, essential for image processing and computer vision.

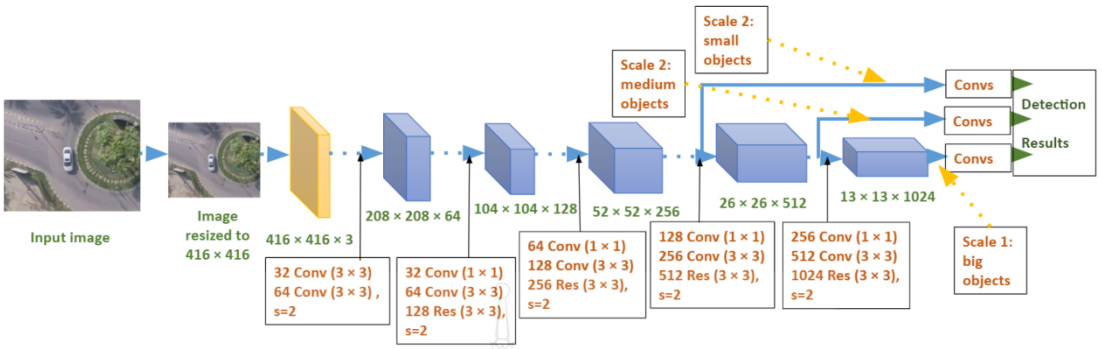


Source Nguyen et al. (2018)

Figure 2.1 Example of image processing

2.1.2 YOLOv3

According to Redmon and Farhadi (2018), YOLOv3 is a popular object detection algorithm that utilizes a grid-based approach to predict bounding boxes and class probabilities for multiple object categories simultaneously, as depicted in Figure 2.2, illustrating the architecture of YOLOv3. The detection employs a multi scale feature extraction technique and includes features like feature pyramid networks and residual blocks, which make it more accurate than its predecessor, YOLOv2. The algorithm has achieved state-of-the-art accuracy on a variety of object detection benchmarks and is widely used in computer vision applications.

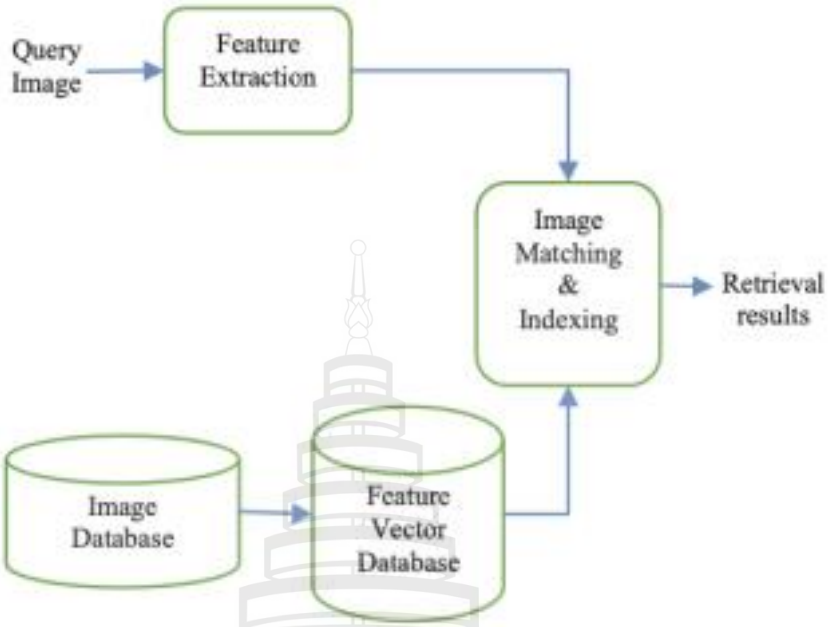


Source Ammar et al. (2021)

Figure 2.2 YOLOv3 architecture

2.1.3 Feature Extraction

Content Based Image Retrieval (CBIR) is a process for retrieving relevant images from a large image database. The goal is to identify images that are visually similar based on specific features such as color, shape, texture, etc. Feature extraction is a crucial step in CBIR as it determines the quality of the results. There are several techniques used for feature extraction, such as color histogram, color correlogram, color co-occurrence matrix, Tamura texture feature, steerable pyramid, wavelet transform, and Gabor wavelet transform. Each technique has its own strengths and weaknesses, and it is important to understand the differences between them to select the most appropriate method for a given CBIR task (Patel & Gamit, 2016). Refer to Figure 2.3 for the block diagram illustrating the fundamental structure of a CBIR system.



Source Patel and Gamit (2016)

Figure 2.3 Block diagram of basic CBIR system

2.1.4 Machine Learning (ML)

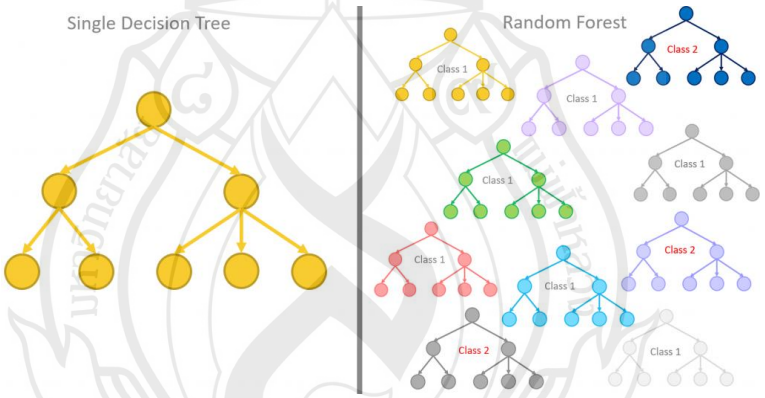
Machine learning involves the creation of computer algorithms that are capable of learning and making predictions or decisions without explicit programming. This is achieved through training the computer system on large amounts of input data, which allows it to develop its own understanding and improve its accuracy over time. Machine learning is now widely used across various industries and applications, such as email filtering, computer vision, speech recognition, and finance, among others. Technology is becoming increasingly valuable due to the vast amounts of data being generated and collected, and the potential for machine learning to revolutionize the way we live, and work is immense. The field is expected to continue growing and evolving, making it a worthwhile area of study for those interested in computer science and artificial intelligence.

The algorithms discussed in this work are Decision Tree, K-Nearest Neighbors, SVM, and Naive Bayes. These algorithms are considered simple but play an important

role in the field of machine learning.

2.1.4.1 Decision Tree

Decision tree algorithms have been widely used for classification and regression tasks in the field of machine learning. They are simple to understand and interpret and can handle both categorical and numerical data. The tree structure can also be visualized and analyzed to gain insights into the decision-making process of the algorithm. Random Forests (Breiman, 2001), XGBoost (Chen et al., 2016), and Stochastic Gradient Boosting (Friedman, 2002) are some of the popular variants of the decision tree algorithm that have been shown to improve accuracy and reduce overfitting. These algorithms have been successfully applied in various domains, including image classification, text classification, and financial forecasting, among others as displayed in Figure 2.3.



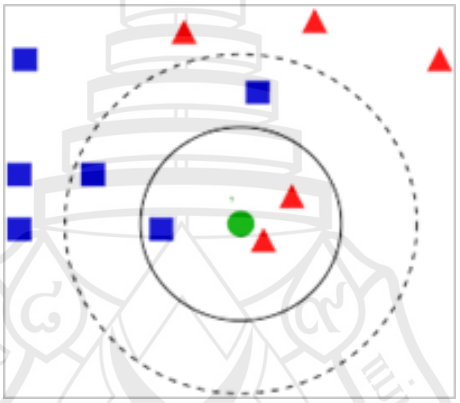
Source Sharma (2021)

Figure 2.4 Decision tree and random forest

2.1.4.2 K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a popular supervised machine learning algorithm used for both classification and regression tasks (Alpaydin, 2010; Hastie et al., 2009). The algorithm functions by identifying the K nearest data points in the training set to a new data point, subsequently assigning either the most frequent class among these neighbors for classification or the average value of the neighbors for

regression, as demonstrated in Figure 2.5 (Alpaydin, 2010). KNN is a simple algorithm and can be used as a baseline for more complex machine learning algorithms (Hastie et al., 2009). However, the choice of parameter K is crucial for the algorithm's effectiveness, and it can be computationally expensive when dealing with large datasets. Feature scaling is also necessary to ensure all features are given equal weight (Alpaydin, 2010). KNN is commonly used in applications such as image recognition, natural language processing, and recommendation systems (Alpaydin, 2010; Hastie et al., 2009).



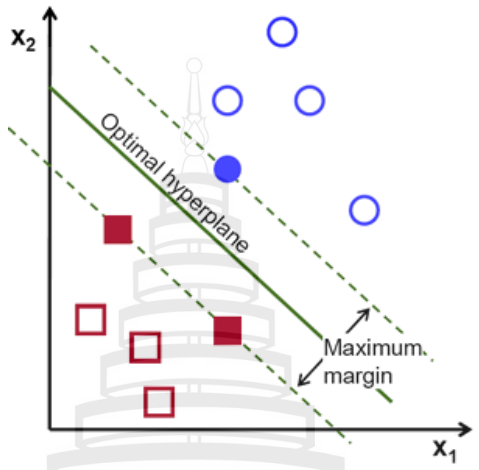
Source Srivastava (2018)

Figure 2.5 K-Nearest neighbors

2.1.4.3 Support Vector Machine

Support Vector Machines (SVM) is a popular supervised machine learning algorithm used for classification, regression, and outlier detection tasks (Cortes & Vapnik, 1995; Hastie et al., 2009). The algorithm works by finding the hyperplane that best separates the data into different classes, with the maximum margin between the classes as demonstrated in Figure 2.6 (Cortes & Vapnik, 1995). SVM can handle both linear and nonlinear data by using different kernel functions to transform the data into a higher-dimensional space (Hastie et al., 2009). SVM has been shown to be effective in various applications, including image and text classification, bioinformatics, and finance (Cortes & Vapnik, 1995; Hastie et al., 2009). However, the algorithm's

performance can be affected by the choice of kernel function and the regularization parameter (Hastie et al., 2009). In addition, SVM can be computationally expensive when dealing with large datasets (Hastie et al., 2009).

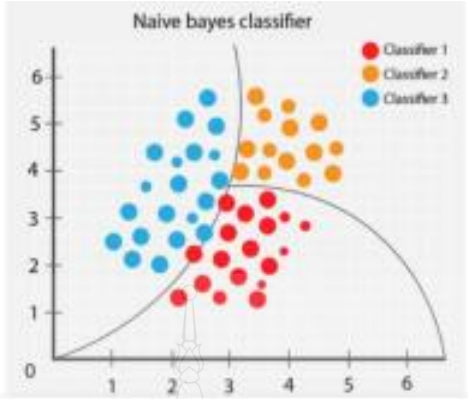


Source Hastie et al. (2009)

Figure 2.6 Support vector machines

2.1.4.4 Naive Bayes

Naive Bayes is a popular supervised machine learning algorithm used for classification tasks, particularly in natural language processing and document classification (Rish, 2001; Manning, Raghavan & Schütze, 2008). The algorithm is based on Bayes theorem and assumes that the features are conditionally independent given the class variable, hence the term naive (Rish, 2001). Naive Bayes is simple, fast, and performs well on small and high-dimensional datasets (Manning et al., 2008). However, it may not perform well when there are correlated features or when the training data is imbalanced (Manning et al., 2008). Different variants of Naive Bayes exist, such as Gaussian Naive Bayes for continuous data and Multinomial Naive Bayes for discrete data (Rish, 2001). Naive Bayes has been applied in various fields, including spam filtering, sentiment analysis, and medical diagnosis (Rish, 2001; Manning et al., 2008).

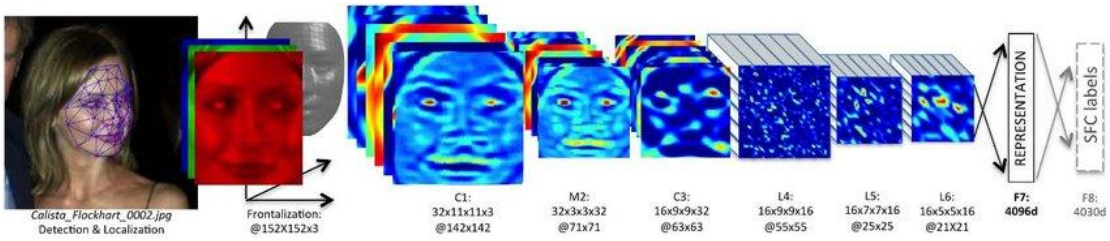


Source Chaudhuri (2022)

Figure 2.7 Naive bayes

2.1.5 Convolutional Neural Network (CNN)

Deep Learning, also known as Deep Neural Networks (ANN), involves the use of multiple layers in artificial neural networks to handle large amounts of data. This technology has become very popular in recent decades and is now widely used for pattern recognition in various fields. One of the most widely used deep neural networks is the Convolutional Neural Network (CNN), named after the mathematical linear operation called convolution. The structure of a CNN includes multiple layers, such as the convolutional layer, non-linearity layer, pooling layer, and fully connected layer, as shown in Figure 2.8. The CNN is particularly effective in machine learning applications that involve image data, such as the ImageNet dataset, computer vision, and natural language processing. In this paper, we will focus on explaining the important elements of CNN and how they work, as well as the parameters that affect its efficiency. It is assumed that readers have a basic understanding of machine learning and artificial neural networks (Albawi et al., 2017).



Source Albawi et al. (2017)

Figure 2.8 Visualizing convolutional deep neural network layers

2.2 Related Work

The study of scene classification techniques has seen growth in recent years, with Convolutional Neural Network (CNN) emerging as a popular choice. According to research (King, 2017), a baseline CNN model was trained on 5,000 images and 365 classes, however, it did not perform well with unseen samples. To improve its performance, a new model was proposed (Zhang et al., 2019). This model uses CNN as a feature extractor and applies CapsNet for classification in the field of remote sensing. The remote sensing data is first processed by CNN to create feature maps, which are then passed on to CapsNet to produce the final classification result.

The research work discussed in (Li et al., 2020) proposes a framework that leverages feature maps obtained from CNN to train a classifier using Graph Neural Network. According to the findings reported in (Özyurt, 2019), these feature maps extracted from CNN can be utilized to build effective classification models through various algorithms such as VGG16, VGG19, ResNet, SqueenzeNet, and AlexNet. This method has shown to improve the accuracy and efficiency of scene classification. Many studies have explored the use of state-of-the-art models that incorporate CNN as a feature extractor, with the aim of addressing the limitations of the baseline CNN (Cheng et al., 2018; Liu et al., 2019; Zeng et al., 2018).

While Convolutional Neural Network (CNN) is widely considered as the state-of the-art approach in scene classification, there are still alternative methods that offer

competitive accuracy. One such method is proposed by (Walther et al., 2011), which involves using fMRI to analyze data by drawing lines and determining the scene. Another feature-based machine learning approach is presented in (Mandhala et al., 2014), which explores the use of Support Vector Machine (SVM) with three different types of kernels (linear, polynomial, and RBF). Both methods offer alternative solutions for scene classification and have the potential to achieve similar levels of accuracy compared to CNN.

In the related work section, it can be inferred that the previous research has primarily focused on enhancing the accuracy of scene classification using machine learning and deep learning algorithms. However, in contrast, the present study aims to evaluate and compare the performance in accuracy of machine learning and base-line CNN techniques for scene classification, rather than improving the existing model. The outcomes of this study provide valuable insights into the relative effectiveness of these methods, which can aid in selecting appropriate techniques for scene classification for the four-room type scene dataset that were selected for this experiment.

CHAPTER 3

METHODOLOGY

The purpose of this research is to compare the results of machine learning and deep learning algorithms, considering that their training processes are different. While Convolutional Neural Networks (CNNs) can be trained by feeding images directly, machine learning algorithms require feature extraction. The system flow Figure 3.1 and Figure 3.3 show that this experiment will encompass machine learning, CNNs, and feature extraction, which will be discussed in further detail in subsequent sections.

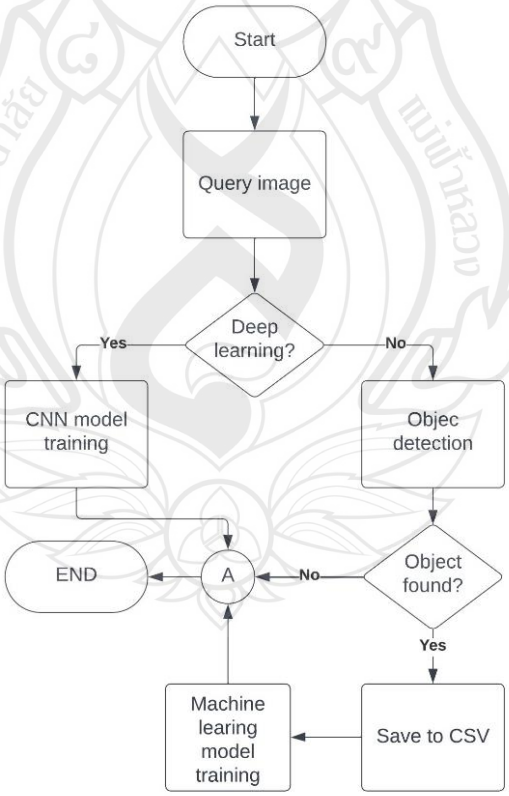


Figure 3.1 System flow of model training

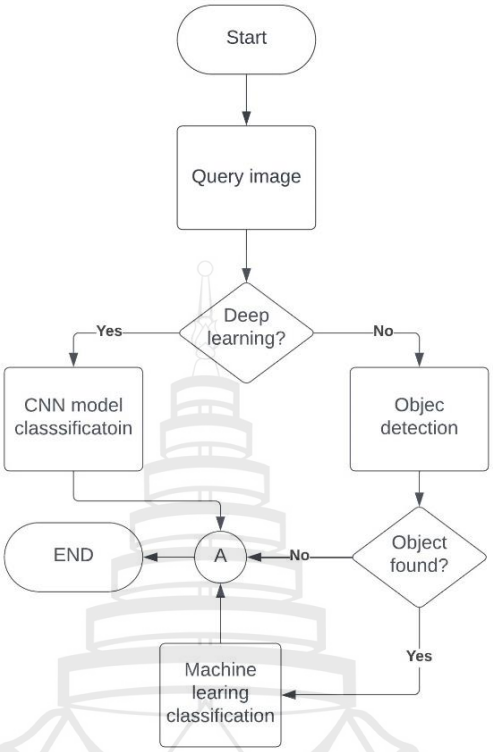


Figure 3.2 System flow of classification process

3.1 Indoor Scene Dataset

In this research, we use a dataset of 400 images of house scenes that were randomly collected from the internet as the training dataset. These images represent four different types of rooms: living room, bathroom, bedroom, and kitchen. These four classes serve as the targets for the development and evaluation of the model. The distribution of images is 100 images each for the four classes, and examples of class-specific images from the dataset can be seen in Figure 3.3. The testing dataset comprises a total of 500 new images of house scenes that were randomly selected from the internet. The dataset includes 120 images for each class.

Figure 3.3 Examples of image data set, living room, bathroom, bedroom, and kitchen

3.2 Feature Extraction

In the first part of this experiment, feature extraction is performed using an object detection model, YOLOv3 (Zhao & Li, 2020). YOLOv3 is a pre-trained object detection model that is optimized for real-time object detection. Its simplicity and lightweight nature make it suitable for fast classification. The list of features extracted by YOLOv3 is displayed in Figure 3.4. The dataset used in this experiment consists of 400 images, and YOLOv3 extracts 41 features from each image. The four target classes are bedroom, bathroom, living room, and kitchen, each containing roughly 100 images.

bottle	orange	bed	oven
wine glass	broccoli	dining table	toaster
cup	carrot	toilet	sink
fork	hot dog	tv	refrigerator
knife	pizza	laptop	book
spoon	donut	mouse	clock
bowl	cake	remote	vase
banana	chair	keyboard	scissors
apple	couch	cell phone	teddy bear
sandwitch	pottedplant	microwave	hair drier
toothbrush			

Figure 3.4 List of objects that can be extracted from YOLOv3

3.3 Machine Learning

In the model training phase, four basic models were used to train the model from the extracted dataset. The models are Decision Tree, K-Nearest Neighbors (KNN), Naive Bayes, and Support Vector Machine (SVM).

3.3.1 Decision Tree

The Decision Tree is an algorithm based on a tree structure. The depth of the tree can have a significant impact on the accuracy of the resulting model. Therefore, it is necessary to specify the maximum depth of the tree. To determine the optimal maximum depth, we need to train the model with different depths starting from 1 and evaluate the accuracy to identify the maximum depth of 17 that provides the best performance as shown in Figure 3.5.

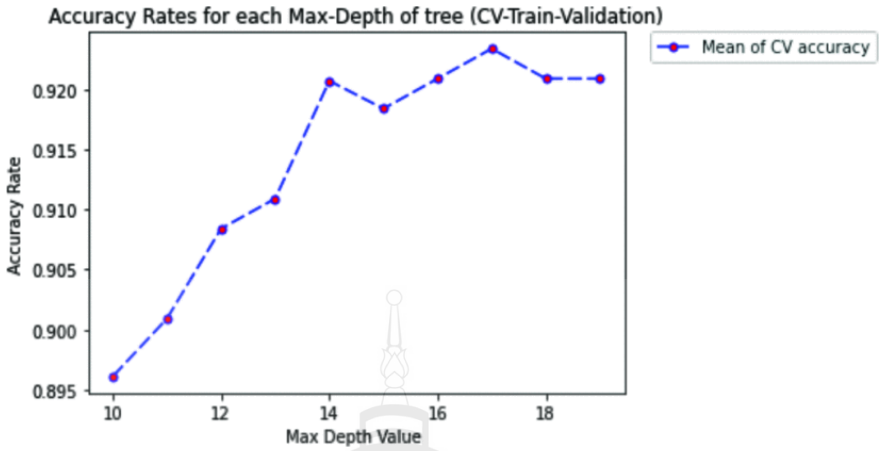


Figure 3.5 Accuracy of each max depth

3.3.2 K-Nearest Neighbors

In the K- Nearest Neighbors algorithm, a new data point is classified by identifying the K closest data points in the training set, and then assigning the class that appears most frequently among these neighbors. For regression, the algorithm assigns the average value of the target variable among the K nearest neighbors to the new data point. This required us to train with different K values to get the best result. For the distance metrics we chose Euclidean distance for this experiment because it is the one of the most popular as it is the default metric of SKlearn library.

Source Fiori (2020)

Figure 3.6 Euclidean distance

In Figure 3.6, it is an illustration of Euclidean distance, showing how it measures the straight-line distance between two points in a two-dimensional space. This visualization helps in understanding how the algorithm calculates the distances between data points.

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Source Fiori (2020)

Figure 3.7 Euclidean distance equation

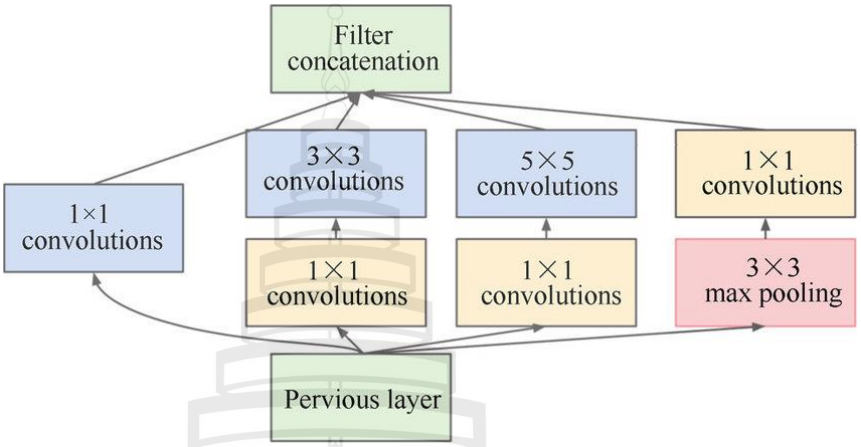
Furthermore, Figure 3.7, represents the Euclidean distance equation, providing a mathematical representation of how the distance between two points is computed. This equation serves as the foundation for calculating distances in the K-Nearest Neighbors algorithm, aiding in the understanding of its implementation and optimization.

To maintain simplicity in the models, we used the baseline version without any modifications for Naive Bayes and SVM.

3.4 Deep Learning

Convolutional Neural Networks (CNNs) are a type of Artificial Neural Network (ANN). They are specifically designed to solve image-based pattern recognition problems. In recent times, CNNs have become a state-of-the-art method for scene classification. In this experiment, we use a simple set of CNNs with Inception module. The inception module is a neural network architecture that incorporates a combination of parallel convolutions of different sizes, including 1x1, 3x3, and 5x5. The purpose of the 1x1 convolutions is to perform dimensionality reduction before the more

computationally expensive 3x3 and 5x5 convolutions are applied (Noor et al., 2012). A single inception module can be visualized in Figure 3.8. Due to its sparsely connected architecture, the inception module can reduce the number of computational resources required for training and inference.



Source Zhao et al. (2017)

Figure 3.8 Inception module

3.5 Evaluation

The evaluation framework of 10-fold cross-validation (Smith & Johnson, 2019) is utilized to generate and assess the quality metric of accuracy. This metric is estimated from the corresponding confusion matrix, where accuracy is calculated as equation (3.1), with TP representing true positives, TN indicating true negatives, FP referring to false positives, and FN denoting false negatives, respectively.

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Figure 3.9 Accuracy result calculation equation

CHAPTER 4

EXPERIMENT AND RESULTS

This chapter is all about putting our research into action**.** It is divided into two parts**:** the experiment and the results**.** In the experiment section, we will talk about the process of experiment, while in the results section, we will share what we found out**.** It is where theory meets practice.

4.1 Experiment

This section details the experimental procedures and methodologies conducted to investigate the research question proposed at the beginning of this thesis. As the experiment's design was presented in Chapter 3, we now delve into the practical implementation of the experiment. This chapter encompasses a comprehensive exploration of the research design, data collection, and the distinct training processes of machine learning and deep learning algorithms, with a focus on Convolutional Neural Networks (CNNs).

4.1.1 Data Set Preparation

Before we can start our experiment, we need to get our data ready. As we mentioned in Chapter 3, the data we're using in this research consists of images. First, we randomly picked images of four different types of rooms: bedrooms, bathrooms, kitchens, and living rooms from the internet. We didn't worry about how big or small the images were. We decided not to make any changes to these images because we want to compare machine learning and deep learning without any extra help. To train our machine learning model, we took the objects out of these images using a special tool called YOLOv3, a pre-trained object detection model. The information about these objects was then saved in a CSV file, as you can see in

Table 4.1. For deep learning, we're going to use the images themselves to teach our model directly. No extra steps or changes will be made to the images.

The machine learning model will use the features extracted from the object detection process, while CNN will use the raw images to train the model. The reason behind this approach is to highlight the differences between traditional machine learning methods and deep learning techniques. Machine learning models often rely on pre-processed and feature-extracted data to perform well, whereas deep learning models like CNNs are designed to automatically learn and extract features from raw data, which can lead to more effective and scalable solutions for image classification tasks.

Table 4.1 Extracted data set example

bottle	cup	fork	knife	pizza	book	bowl	cake	…	class
1	1	1	1	1	0	1	1	…	0
0	0	0	0	0	1	1	1	…	1
0	0	1	0	1	1	0	0	…	2
1	1	1	0	1	0	0	0	…	3

Note 0 indicates the absence of an object, while 1 signifies the presence of an object, and classes 1-4 represent the specific types of room.

4.1.2 Machine Learning Model Training

This section will go through the process of training machine learning models. As mentioned in Chapter 3, we will be experimenting on 4 basic machine learning models and they are Decision Tree, KNN, Naïve Bayes and SVM.

Firstly, we will talk about Decision Tree. In the context of decision tree modeling, we find the best maximum depth setting by identifying the depth value that gives us the highest training accuracy. Figure 4.1 shows a clear trend where training accuracy increases as the maximum depth ranges from 10 to 18. Based on this evidence, we determine that the best parameter value is 17, and we'll use this value during the upcoming testing phase.

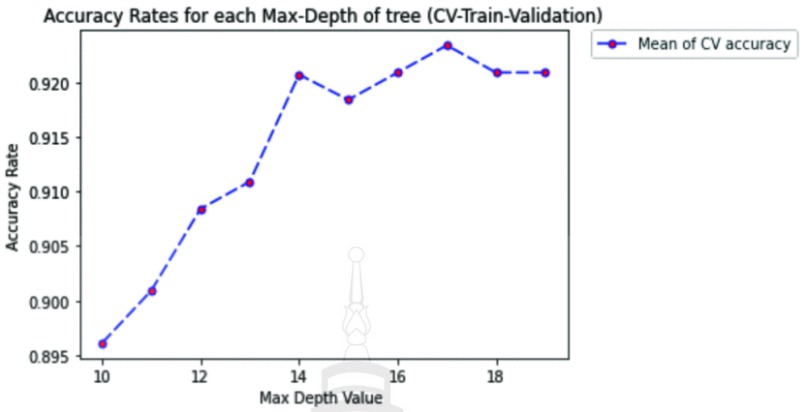


Figure 4.1 Best maximum depth for decision tree

Secondly, we delved into a comparable exploration aimed at pinpointing the most effective number of neighbors (k) for the KNN technique. Within Figure 4.2, a range of training accuracy values is presented for KNN, spanning from 1 to 9 for different k values. Our scrutiny of this data demonstrates a clear preference for the value k = 4 as the most advantageous. As a result, our choice for the forthcoming testing phase aligns with k = 4.

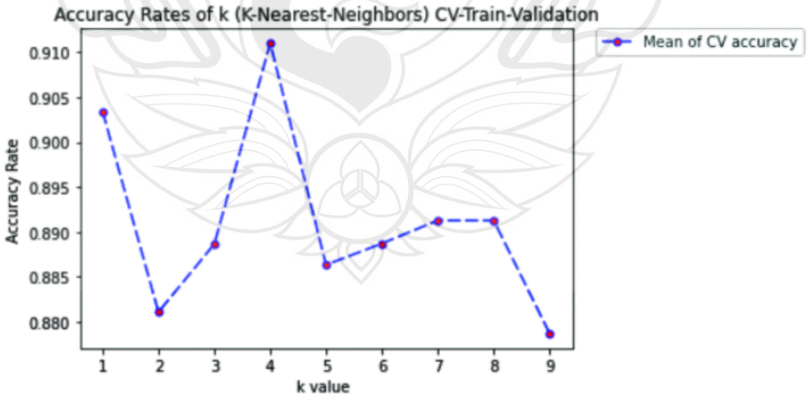


Figure 4.2 Training accuracy to find the best K value for KNN training parameters

Finally, the last Two models of machine learning Naïve Bayes and SVM. For Naïve Bayes, we will be using the Gaussian distribution and for SVM, we will be using the linear kernel.

4.1.3 CNN Model Training

Regarding the CNN approach, which showcases the utilization of low-level feature descriptors, we've opted for Inception-v3 (Nguyen et al., 2018) to showcase its applicability in addressing this problem. It's important to note that we've configured specific settings for this demonstration, including a batch size of 1, an epoch count of 100, and the utilization of the SoftMax activation function.

4.2 Result

In this section, we present the findings obtained from our research, highlighting the outcomes of our model configurations, training, and evaluation efforts. The results provide valuable insights into the performance of different models in addressing the objectives of our study. As we delve into the outcomes, it becomes evident that certain models have demonstrated remarkable accuracy, while others have exhibited distinctive characteristics. This section offers a detailed analysis of these findings, laying the foundation for a deeper understanding of our research outcomes and their implications.

After configuring our models, we proceeded with their training. The results of this training phase revealed a distinct performance ranking. As illustrated in table 4.2, the cross-validation accuracy result shows that the Decision Tree model emerged as the top performer, boasting an impressive accuracy rate of 91.6%. It was closely followed by the KNN, SVM, and Naïve Bayes models, arranged in descending order of accuracy.

Table 4.2 Cross validation accuracy

Models	Cross Validation Result
Decision Tree	91.6%
KNN	91.1%
SVM	90.1%
Naïve Bayes	78.0%

To evaluate the models, we calculated accuracy using Equation (1). The results, as presented in Table 4.3, clearly indicate the model best suited for indoor scene classification. The Decision Tree model outperforms the others with the highest at 84.7%. It achieved an impressive accuracy rate, followed by SVM at 84.6%, KNN at 83.9%, CNN at 81%, and Naïve Bayes at 46%.

Table 4.3 Accuracy results of the models

Models	Accuracy Result
Decision Tree	84.7%
SVM	84.6%
KNN	83.9%
CNN	81.0%
Naïve Bayes	46.0%

CHAPTER 5

CONCLUSION AND DISCUSSION

5.1 Conclusion

In this thesis, we presented a comparative study of machine learning and deep learning models for indoor scene classification. We trained four basic machine learning models (Decision Tree, KNN, Naïve Bayes, and SVM) and a simple CNN model on a dataset of indoor scene images. We evaluated the performance of the models using 10-fold cross-validation.

Our results showed that the Decision Tree model outperformed all the other models, including the CNN model. The Decision Tree achieved an accuracy of 84.7%, while the CNN achieved an accuracy of 81%. This suggests that simple machine learning models can still be effective for indoor scene classification, even when compared to more complex deep learning models. Inception-v3 is a convolutional neural network architecture introduced by Google, known for its ability to capture features at multiple scales and its computational efficiency. The use of the Inception-v3 model, which I did not develop myself, ensures that these findings can be trusted.

5.2 Discussion

One possible explanation for the good performance of the decision tree is that it can learn complex relationships between the features of the data. For example, the Decision Tree may be able to learn that a bathroom is more likely to contain a toilet than a kitchen, or that a bedroom is more likely to contain a bed than a living room.

Another possible explanation is that the Decision Tree is less susceptible to overfitting than the other models. Overfitting occurs when a model learns the training data too well and is unable to generalize to new data. The decision tree regularization

mechanism, such as pruning, may help to prevent overfitting.

The CNN model performed well overall, but it was slightly outperformed by the Decision Tree model in terms of accuracy. This may be because the CNN model is more complex and requires more training data to achieve optimal performance. The dataset used in this experiment is relatively small, so it is possible that the CNN model was not able to learn the underlying patterns in the data as effectively as the Decision Tree model.

However, in terms of execution time, CNNs often outperform traditional machine learning models. This advantage stems from their streamlined approach to object detection and feature extraction, which eliminates the additional steps that traditional ML methods require, thereby reducing processing time.

The KNN model performed well overall, but it was slightly outperformed by the Decision Tree and CNN models. This may be because the KNN model is sensitive to the choice of distance metric and the number of neighbors. In this experiment, we used the Euclidean distance metric and the K=4 nearest neighbors. It is possible that different values for these parameters would have resulted in better performance.

The Naïve Bayes model performed the worst of all the models in this experiment. This is likely due to several factors, including:

1. The Naïve Bayes model assumes that the features are independent of each other. This is not necessarily true in the case of indoor scene classification. For example, the presence of a bed in a room is likely to be correlated with the presence of other bedroom furniture, such as a dresser or nightstand.
2. The Naïve Bayes model is sensitive to noise in the data. The dataset used in this experiment is relatively small and may contain some noise. The Naïve Bayes model may be more susceptible to the effects of noise than the other models.
3. The Naïve Bayes model is a simple model that does not have a lot of parameters to tune. This can make it difficult to improve the performance of the model on complex tasks.

In addition to these factors, it is also possible that the Naïve Bayes model is simply not well-suited for the task of indoor scene classification. Indoor scene classification is a complex task that requires the model to be able to learn complex relationships between the different features of the data. The Naïve Bayes model may

not be able to learn these relationships as effectively as the other models.

Overall, the Naïve Bayes model is a simple but effective model for classification tasks. However, it may not be the best choice for complex tasks, such as indoor scene classification.

5.3 Limitations and Future Work

One limitation of this experiment is that the dataset is relatively small. With more training data, it is possible that the CNN model would have outperformed the Decision Tree model. Additionally, a larger data set would allow for more rigorous evaluation of the models, such as using a holdout test set.

Another limitation of this experiment is that we only used a simple CNN model. There are more complex CNN models that could be used, such as ResNet or DenseNet. Additionally, we could experiment with different feature extraction methods.

In future work, we plan to address these limitations by using a larger data set and experimenting with different CNN models and feature extraction methods. We also plan to explore the use of machine learning and deep learning models for indoor scene classification in real-w0orld applications.

For example, we could develop a mobile app that uses indoor scene classification to help blind or visually impaired people navigate their surroundings. We could also develop a system that uses indoor scene classification to monitor the safety and security of a building.

We believe that indoor scene classification is a promising area of research with a wide range of potential applications. We are excited to continue our work in this area and to contribute to the development of new and innovative indoor scene classification systems.

REFERENCES

REFERENCES

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*, 1–6. https://doi.org/10.1109/icengtechnol.2017.8308186

Alpaydin, E. (2010). Toward a minor architecture. In *The MIT Press eBooks* (2nd ed.). The MIT Press. https://doi.org/10.7551/mitpress/9304.001.0001

Ammar, A. (2021). Vehicle detection from aerial images using deep learning. *A Comparative Study. Electronics, 7*, 820. https://doi.org/10.3390/electronics7100820

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/a:1010933404324

Chaudhuri, K. D. (2022, March 21). *Building naive bayes classifier from scratch to perform sentiment analysis*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/03/building-naive-bayes-classifier-from-scratch-to-perform-sentiment-analysis

Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., & Li, H. (2024). *End-to-end autonomous driving: Challenges and frontiers*. IEEE. https://doi.org/10.48550/arXiv.2306.16927

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16* (pp. 785–799). https://doi.org/10.1145/2939672.2939785

Cheng, X., Lu, J., Feng, J., Yuan, B., & Zhou, J. (2018). Scene recognition with objectness. *Pattern Recognition*, *74*, 474–487. https://doi.org/10.1016/j.patcog.2017.09.025

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Fiori, L. (2020, May 22). *Distance metrics and k-nearest neighbor (KNN)*. Medium. https://medium.com/@luigifiori/distance-metrics-and-k-nearest-neighbor-knn-3c7e3e2f93d0

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367–378. https://doi.org/10.1016/s0167-9473(01)00065-2

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H (2009). *The elements of statistical learning: Data mining inference, and prediction* (Vol. 2, pp. 1–758) Springer. https://doi.org/10.1007/978-0-387-84858-7

Jeremy, S. (2014). *Computer vision: models, learning, and inference* (p. 83). Cambridge University Press.

King, J., Kishore, V., & Ranalli, F. (n.d.). *Scene classification with convolutional neural networks*. Retrieved August 2, 2024, from https://cs231n.stanford.edu/reports/2017/pdfs/102.pdf

Li, Y., Chen, R., Zhang, Y., Zhang, M., & Chen, L. (2020). Multi-Label remote sensing image scene classification by combining a convolutional neural network and a graph neural network. *Remote Sensing*, *12*(23), 4003. https://doi.org/10.3390/rs12234003

Li, Y., Dixit, M., & Vasconcelos, N. (2017). Deep scene image classification with the MFAFVNet. In *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 5746–5754). IEEE. https://doi.org/10.1109/iccv.2017.613

Liu, S., Tian, G., & Xu, Y. (2019). A novel scene classification model combining ResNet based transfer learning and data augmentation with a filter. *Neurocomputing*, *338*, 191–206. https://doi.org/10.1016/j.neucom.2019.01.090

Mandhala, V. N., Sujatha, V., & Devi, B. R. (2014). Scene classification using support vector machines. In *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies* (pp. 1807–1810). IEEE. https://doi.org/10.1109/icaccct.2014.7019421

Manning, C. D., Raghavan, P., & Schutze, H. (2008). *Introduction to information retrieval.* Cambridge University Press. https://doi.org/10.1017/cbo9780511809071

Nguyen, L. D., Lin, D., Lin, Z., & Cao, J. (2018). Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation. *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. https://doi.org/10.1109/iscas.2018.8351550

Noor, M. A., Al-Amin, A. Q., & Kabir, M. E. (2012). Developing an effective e-learning system for tertiary level education in Bangladesh. *International Journal of Emerging Technology in Learning*, *7*(4). https://doi.org/10.3991/ijet.v7i4.2408

Özyurt, F. (2019). Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures. *The Journal of Supercomputing*, *75*, 8413–8431. https://doi.org/10.1007/s11227-019-03106-y

Patel, J. M., & Gamit, N. C. (2016). A review on feature extraction techniques in Content Based Image Retrieval. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 2259–2263). IEEE. https://doi.org/10.1109/wispnet.2016.7566544

Rish, I. (2001). An empirical study of the naive Bayes classifier. *IJCAI-01 Workshop on Empirical Methods in Artificial Intelligence*, *3*, 41–46.

Sharma, P. (2021, April 29). *Decision tree classification*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/04/decision-tree-classification-guide/

Shen, D., Wu, G., & Suk, H.-I. (2017). Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, *19*(1), 221–248. https://doi.org/10.1146/annurev-bioeng-071516-044442

Smith, J., & Johnson, A. (2019). Comparing supervised learning algorithms for predicting house prices: A case study. *Journal of Machine Learning Research*, *20*(3), 451–467.

Srivastava, T. (2018, March 25). *KNN algorithm*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2018/03/knn-algorithm-latest-guide/

Walther, D. B., Chai, B., Caddigan, E., Beck, D. M., & Fei-Fei, L. (2011). Simple line drawings suffice for functional MRI decoding of natural scene categories. *Proceedings of the National Academy of Sciences of the United States of America*, *108*(23), 9661–9666. https://doi.org/10.1073/pnas.1015666108

Xie, X., Cheng, G., Wang, J., Yao, X., & Han, J. (2021). Oriented R-CNN for object detection. In *Proceeding of the IEEE/CVF International Conference on Computer Vision* (pp. 3520–3529). https://doi.org/10.48550/arxiv.2108.05699

Zeng, D., Chen, S., Chen, B., & Li, S. (2018). Improving remote sensing scene classification by integrating global-context and local-object features. *Remote Sensing*, *10*(5), 734. https://doi.org/10.3390/rs10050734

Zhang, W., Tang, P., & Zhao, L. (2019). Remote sensing image scene classification using CNN-CapsNet. *Remote Sensing*, *11*(5), 494. https://doi.org/10.3390/rs11050494

Zhao, B., Feng, J., Wu, X., & Yan, S. (2017). A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, *14*(2), 119–135. https://doi.org/10.1007/s11633-017-1053-3

Zhao, L., & Li, S. (2020). Object detection algorithm based on improved YOLOv3. *Electronics*, *9*(3), 537. https://doi.org/10.3390/electronics9030537

CURRICULUM VITAE

CURRICULUM VITAE

NAME Simon Yosboon

EDUCATIONAL BACKGROUND

2021 Bachelor of Engineering
Computer Engineering
Mae Fah Luang University

SCHOLARSHIP

2021 Post-Graduate Tuition Scholarship

PUBLICATION

Yosboon, S. (2022). Scene Classification with Simple Machine Learning and Convolutional Neural Network. In *2022 International Conference on Decision Aid Sciences and Applications (DASA)* (pp. 616–619). IEEE. https://doi.org/10.1109/dasa54658.2022.9764995