ENSEMBLE DEEP LEARNING FOR REAL-BOGUS CLASSIFICATION WITH SKY SURVEY IMAGES

PAKPOOM PROMMOOL

DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

SCHOOL OF APPLIED DIGITAL TECHNOLOGY
MAE FAH LUANG UNIVERSITY
2025

ENSEMBLE DEEP LEARNING FOR REAL-BOGUS CLASSIFICATION WITH SKY SURVEY IMAGES

PAKPOOM PROMMOOL

THIS DISSERTATION IS A PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER ENGINEERING

SCHOOL OF APPLIED DIGITAL TECHNOLOGY
MAE FAH LUANG UNIVERSITY
2025

DISSERTATION APPROVAL

MAE FAH LUANG UNIVERSITY

FOR

DOCTOR OF PHILOSOPHY IN COMPUTER ENGINEERING

Dissertation Title: Ensemble Deep Learning for Real-bogus Classification with Sky Survey Images

Author: Pakpoom Prommool

Examination Committee:

Associate Professor Adisorn Leelasantitham, Ph. D.	Chairperson
Assistant Professor Sirikan Chucherd, Ph. D.	Member
Associate Professor Roungsan Chaisricharoen, Ph. D.	Member
Associate Professor Punnarumol Temdee, Ph. D.	Member
Assistant Professor Chayapol Kamyod, Ph. D.	Member

Advisor:

[illegible]

....Advisor

(Assistant Professor Sirikan Chucherd, Ph. D.)

Dean:

[illegible]

....

(Assistant Professor Nacha Chondamrongkul, Ph. D.)

ACKNOWLEDGEMENTS

I am profoundly grateful to my advisor, Asst. Prof. Sirikan Chucherd, Ph. D., whose unwavering support, expert guidance, and thoughtful feedback have been with me every step of the way. Her deep knowledge, patience, and heartfelt encouragement have not only shaped the direction and depth of this thesis but also strengthened my resilience during times of uncertainty. Her mentorship has been a guiding light, nurturing both my academic journey and personal growth. I am sincerely thankful for her dedication, generosity, and steadfast belief in me.

My deepest thanks also go to Assoc. Prof. Wg. Cdr. Tossapon Boongoen, Ph. D., whose generosity opened the door to this doctoral journey. His belief in my potential, support of my research direction, and the opportunity to work as a research assistant under the Newton Fund project from the United Kingdom provided me not only with resources but also the confidence to move forward. Without his guidance and unwavering encouragement, this Ph. D. would never have been possible.

I would like to extend my heartfelt gratitude to the members of my thesis examination committee: Assoc. Prof. Adisorn Leelasantitham, Ph. D., Assoc. Prof. Roungsan Chaisricharoen, Ph. D., Assoc. Prof. Punnarumol Temdee, Ph. D., and Asst. Prof. Chayapol Kamyod, Ph. D. Their thoughtful evaluations, insightful critiques, and kind encouragement have played an integral role in refining the quality of this research. I am deeply appreciative of the time, attention, and care they devoted to this work.

I am also sincerely thankful to Mae Fah Luang University for its generous academic support and the research grant for the completion of my dissertation. My gratitude extends to the Center of Excellence in AI and Emerging Technology and the School of Applied Digital Technology for the invaluable opportunities and resources they provided. I would also like to thank the faculty members and the staff of the Postgraduate Office, whose continued support, patience, and assistance have been vital to the completion of this thesis.

With all my heart, I thank my father, Phairat, and my mother, Somsri Prommool, along with my entire family, for their unconditional love and steadfast support throughout every challenge. Their encouragement, both emotional and financial, has

been the foundation upon which I built this journey. When I struggled and doubted myself, their faith in me never wavered.

During some of the darkest and most difficult days of this research, the song *"Krueng Lang"* by Bodyslam became a quiet yet powerful voice of encouragement. When I was on the verge of giving up, its message reignited my determination and reminded me to keep moving forward. I will forever cherish how something so simple could lift me up when I needed it most

Lastly, my deepest appreciation and love go to Pin (Krittiya Sunanthasin), my beloved. Her presence, unwavering support, and quiet strength have been my anchor through every storm. She walked beside me with patience and grace, offering encouragement in moments when I had little left. This achievement is as much hers as it is mine.

Pakpoom Prommool

Dissertation Title	Ensemble Deep Learning for Real-bogus Classification with Sky Survey Images
Author	Pakpoom Prommool
Degree	Doctoral of Philosophy (Computer Engineering)
Advisor	Assistant Professor Sirikan Chucherd, Ph. D.

ABSTRACT

The detection of astronomical transient events—short-lived phenomena such as supernovae, gamma-ray bursts, and stellar flares—has become a major focus in contemporary astrophysical research. These events are often linked to extreme cosmic processes and provide essential insights into stellar evolution and the dynamic nature of the universe. However, identifying such events within the vast and rapidly expanding datasets generated by modern sky surveys presents significant challenges, especially as manual inspection becomes infeasible. The Gravitational-wave Optical Transient Observer (GOTO) project exemplifies this complexity. Designed to detect optical counterparts to gravitational-wave sources, GOTO produces hundreds of sky survey images per night, with each image containing tens of thousands of celestial objects. The scale and frequency of this data render traditional analysis methods—such as manual feature extraction and visual inspection—insufficient and inefficient, leading to missed opportunities in capturing rapidly fading events. This research proposes the use of Deep Learning techniques, particularly Convolutional Neural Networks (CNNs), to enhance the classification of transient objects. Unlike traditional methods, CNNs can automatically learn discriminative features directly from raw image data, making them well-suited for large-scale, high-dimensional image classification tasks. To improve model performance and robustness, the study employs Transfer Learning and Fine-Tuning strategies using pre-trained models on ImageNet, adapting them to the specific characteristics of astronomical imagery. It also integrates various Data Augmentation techniques—including image rotation, flipping, and noise injection—to increase data diversity. Additionally, the research investigates the role of Dropout in preventing overfitting and evaluates the effect of varying Batch Sizes on model training and generalization.

To further enhance classification performance, the study incorporates Ensemble Learning techniques such as Soft Voting and Weighted Voting, combining multiple CNN models to produce more reliable predictions. The results demonstrate that this integrated approach significantly improves the accuracy and reliability of transient classification within large-scale datasets, offering a practical and scalable solution for real-time astronomical event detection in projects like GOTO.

Keywords: Astronomical Transients, Convolutional Neural Networks (CNNs), Transfer Learning, Fine-Tuning, Data Augmentation, Batch Size, Ensemble Learning, Sky Survey, Optical Transient Detection, Ensemble Deep Learning

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES

LIST OF FIGURES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

One of the most intensively studied topics in recent years is the detection of transient events, which are astronomical phenomena that exist only for a short period before rapidly fading away. Detecting these transients is critically important for understanding the extreme conditions of the universe, as they often result from high-energy or high-mass astrophysical processes. Examples include the merger of neutron stars and black holes, which produce gravitational waves and gamma-ray bursts; the collapse of massive stars leading to Type II supernovae, marking the final stages of stellar evolution; and the occurrence of mega-flares on main-sequence stars, indicating sudden changes in stellar magnetic fields (Bailer-Jones et al., 2008; Djorgovski et al., 2014). These phenomena not only broaden our understanding of stellar evolution and cosmic history but also hold the potential for discovering new laws of nature and testing physical theories that cannot be examined within terrestrial laboratories.

However, the detection of transient events within massive astronomical datasets is far from straightforward. Each day, an enormous volume of sky survey images is collected at a scale that can no longer be manually analyzed by human effort alone. The exponentially increasing amount of data has introduced unprecedented challenges in data analysis and management in the history of astronomy. Consequently, astronomers are compelled to develop new approaches that can efficiently extract valuable information from these large-scale datasets.

Moreover, the signals of these events are often obscured by noise and other interferences arising from technical limitations of observational instruments, such as sensor noise, atmospheric distortions, or processing artifacts introduced during initial data handling. In many cases, genuine transient signals closely resemble these artifacts, making it extremely difficult to distinguish between true and false detections through traditional analysis methods. Relying solely on human inspection to examine each event is no longer feasible given the rapidly increasing data generation rate. Additionally, delays in processing may result in missed opportunities to track short-lived transient phenomena, which could disappear within just a few hours or days after their initial detection.

The Gravitational-wave Optical Transient Observer (GOTO) project is one of the initiatives dedicated to the detection of transient events. GOTO is a telescope project specifically designed to identify optical counterparts to gravitational-wave events detected by instruments such as LIGO and VIRGO. It employs a system of multiple automated telescope mounts, equipped with 40-centimeter unit telescopes. A complete GOTO system can survey a total field of 40 square degrees and achieve a limiting magnitude of 20 within just three minutes of observation. The prototype GOTO facility began operations in 2017 at La Palma, initially with four telescopes, and later expanded to eight telescopes in 2020. Further expansion occurred in 2021, with plans underway to establish an additional observing station in southern Australia, enabling coverage of both the northern and southern hemispheres. This system is designed to rapidly detect and follow up on transient sources based on alerts from gravitational-wave detectors. GOTO generates an enormous number of astronomical images each night, with each image capturing approximately 20,000 celestial objects and producing around 400 images per night. As a result, it faces the challenge of analyzing vast amounts of data, consistent with the difficulties outlined previously.

The classification of image samples into predefined categories or classes is one of the fundamental issues in artificial intelligence and image processing research. In the existing literature, two principal approaches have been developed to address this problem, as illustrated in Figure 1.1 of the study, particularly in the context of distinguishing real transient sources from sky survey data.

The first approach to astronomical image analysis typically follows a traditional pipeline, where raw image data is initially transformed into feature representations that can effectively support classification tasks. For instance, (Keerin & Boongoen, 2022) proposed the use of physics-based features to construct profiles for each image, capturing the appearance of newly emerged bright sources. These features are then used to develop classification models using conventional machine learning techniques, such as Random Forest, Decision Tree, Artificial Neural Networks, K-Nearest Neighbors, and Support Vector Machines, as highlighted in the work of (Sarker, 2021).

Meanwhile, several studies have utilized data from the Gravitational-wave Optical Transient Observer (GOTO), which is a key astronomical facility for observing transient phenomena. One of the most widely used methods for image subtraction in this context is HOTPANTS. This technique operates by taking two aligned images of the same sky region at different times, dividing them into multiple subregions, and computing convolutional kernels to match the point spread function (PSF) for each region. However, HOTPANTS uses only a single kernel across the entire image, which is insufficient for dealing with stars of varying sizes and may lead to incomplete subtraction results.

To address this limitation, (Tabacolde et al., 2018) used transient images from GOTO for classification tasks by converting the images into feature vectors and applying various machine learning methods. One major challenge in their study was data imbalance—real transient images were significantly outnumbered by bogus ones—which led to poor model performance. In a follow-up experiment, they tackled the imbalance by applying undersampling and oversampling techniques on the original feature set, which led to improved results compared to the previous attempt.

Later, (Liu et al., 2019) approached the imbalance problem differently. Instead of transforming the images into feature vectors, they used all available real images from GOTO and augmented the dataset by rotating each image in six different directions. This data augmentation technique effectively increased the number of real samples before training and resulted in improved classification performance, demonstrating that simple augmentation strategies can play a significant role in mitigating class imbalance in astronomical datasets.

In contrast, the second approach involves the application of deep learning techniques, particularly Convolutional Neural Networks (CNNs), which have gained widespread attention over the past decade. CNNs are specifically designed to handle the variability of two-dimensional data patterns (Lecun et al., 1998; Li et al., 2022) and have been proven highly effective for large-scale image classification tasks, such as those encountered in the ImageNet competition (Krizhevsky et al., 2017). A key advantage of CNNs lies in their ability to learn distinctive features directly from the raw pixel values of images, without requiring a separate feature extraction phase during data preprocessing, as is necessary in traditional approaches.

Building upon the aforementioned studies, it has become evident that Convolutional Neural Networks (CNNs) offer significant potential for further development and refinement. The present research recognizes that CNNs can be extended to address certain challenges encountered in the GOTO project, such as the issue of data imbalance as well as improving the overall classification accuracy. These enhancements could lead to more robust and reliable identification of transient events within large-scale sky survey datasets.

1.2 Challenges

As discussed above, the Gravitational-wave Optical Transient Observer (GOTO) is one of the projects dedicated to the detection of transient events. However, the overall structure of the GOTO system is composed of several distinct components, each responsible for specific functions, as illustrated in Figure 1.1.

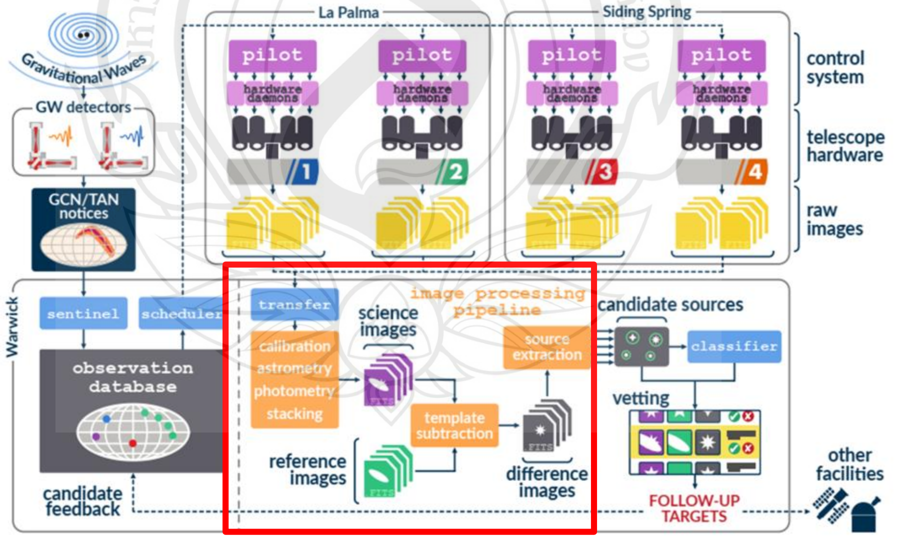


Figure 1.1 A flowchart illustrating the comprehensive GOTO follow-up network

Figure 1.1 illustrates a flowchart outlining the complete GOTO follow-up network. Starting from the top left, gravitational wave signals are captured and processed by the G-TeCS sentinel system. Following this, the scheduler issues positioning commands to the four telescopes, which are distributed across two separate locations. The images captured by these telescopes are then transmitted to the University of Warwick, where they are processed by comparison with reference images to identify potential transient sources.

This study focuses particularly on the image processing pipeline, which is emphasized within the solid frame in Figure 1.1. All images from the GOTO telescopes are transmitted via a dedicated fiber link to the central processing servers located at Warwick University, UK. Upon arrival, each image undergoes standard calibration procedures, including bias correction, dark frame subtraction, and flat-field adjustment.

Based on the observations and aforementioned discussion, a major challenge of this process lies in the necessity to initially distinguish between real and non-real images. This serves as one of the earliest stages of data filtering within the workflow. Given the vast volume of data received on a daily basis, it is imperative to implement a system with high precision to eliminate irrelevant or unnecessary information. Failure to accurately perform this initial filtration may result in increased complexity and difficulty in the subsequent stage, which involves classifying the nature of the detected events.

1.3 Motivation

As previously discussed, one of the fundamental challenges in image classification using Deep Learning techniques lies in enhancing model accuracy when facing highly imbalanced datasets. Several studies have proposed addressing this issue by employing oversampling strategies to increase the volume of samples in minority classes, thereby mitigating model bias arising from data imbalance. Additionally, Deep Learning methodologies such as Transfer Learning and Fine-Tuning play a crucial role in improving training efficiency, particularly in scenarios involving limited datasets that are not suitable for training from scratch. Other techniques, such as Dropout and varying batch sizes, are also widely adopted to regulate model learning behavior and reduce the risk of overfitting.

Motivated by these challenges, this study aims to investigate the following research questions:

1. How does the use of Transfer Learning and Fine-Tuning influence image classification accuracy?
2. Can neural networks learn distinct feature representations when different data augmentation techniques are applied?
3. What is the effect of varying batch sizes during training on classification accuracy?
4. Will neural networks acquire different representations when trained with different batch sizes?
5. How does the application of Dropout during training affect the accuracy of image classification?
6. How does the usage of different base architectures for ensembling affect classification accuracy?
7. What are the most effective ensemble methods for combining multiple networks trained with different data augmentation strategies to achieve higher classification accuracy?

1.4 Research Object

This research focuses on the application of Transfer Learning techniques, fine-tuning of various hyperparameters such as batch size and the implementation of early stopping, as well as the use of data augmentation and Dropout methods to enhance the accuracy of astronomical image classification using data from the GOTO (Gravitational-wave Optical Transient Observer) telescope. Additionally, this study explores ensemble learning approaches, employing diverse base architectures to examine whether combining multiple models trained with various augmentation strategies can lead to improved classification performance.

In accordance with the study's objectives, the research aims are as follows:

1. To investigate the impact of applying Transfer Learning and Fine-Tuning techniques on the accuracy of image classification.

2. To evaluate the effect of different Convolutional Neural Network (CNN) architectures on model performance when trained with various data augmentation techniques.

3. To analyze the advantages and limitations of employing pre-trained models—originally developed for everyday object recognition—in the context of astronomical image classification.

4. To examine the influence of Dropout application during model training on classification accuracy.

5. Studying how the use of different base architectures in ensemble learning affects the performance of image classification.

6. To identify the most effective ensemble methods for combining multiple neural networks trained with diverse data augmentation strategies in order to improve classification accuracy.

1.5 Scope of Research

The scope of this research is defined from two main perspectives: the fundamental experimental framework and its application to domain-specific challenges in astronomical image classification. These are detailed as follows.

1.5.1 Framework

This research focuses on the application of Deep Learning techniques, particularly Transfer Learning and Fine-Tuning, combined with the adjustment of key hyperparameters such as batch size, the implementation of early stopping, and the use of dropout to enhance model generalization and prevent overfitting. To improve learning efficiency, the study also applies data augmentation techniques to increase the diversity of training data and mitigate the effects of class imbalance. The research investigates the impact of various Convolutional Neural Network (CNN) architectures—such as MobileNet, Xception, ResNet, and DenseNet—under both transfer learning and fine-tuning approaches, to evaluate how architectural differences influence classification performance.

Furthermore, this study explores the use of ensemble learning methods by integrating base models with different architectures that are trained on augmented datasets. The objective is to determine how the ensemble of multiple models, trained under varying conditions, can improve classification accuracy. This involves comparing different ensemble strategies and identifying the most effective methods for combining model outputs.

1.5.2 Application

The practical application of this research is centered on the classification of astronomical images from the Gravitational-wave Optical Transient Observer (GOTO) telescope. These images present unique challenges such as background noise, low signal-to-noise ratios, class imbalance, and ambiguous object morphologies. The study utilizes real observational data, including light curve images and labeled astronomical sources, to train and evaluate the performance of the proposed models. The classification task specifically focuses on distinguishing between “real” and “bogus” detections, an essential step in automating the process of transient discovery in astronomical surveys. The goal is to develop a highly accurate classification system that can be integrated into an automated detection pipeline. Future applications of this research may include extending the model to classify more complex object types, such as supernovae, quasars, or kilo novae, as the system is further refined and trained on expanded datasets.

CHAPTER 2

LITERATURE REVIEW

To establish a strong foundation for this research on astronomical transient detection, Chapter 2 explores the body of knowledge and technological advancements relevant to time-domain astronomy and image analysis. This chapter highlights how the rise of large-scale sky surveys—such as ZTF, Pan-STARRS, and the LSST—has created significant challenges in managing and interpreting the massive volumes of observational data collected each night. In response, Machine Learning (ML) and Deep Learning (DL) techniques have emerged as critical tools in automating the detection and classification of transient events with high accuracy and efficiency. The chapter begins by discussing approaches for supernova classification and real–bogus discrimination, with a focus on Convolutional Neural Networks (CNNs) such as VGGNet, Xception, ResNet, and DenseNet. These models have shown strong performance in various astronomical tasks by leveraging spatial patterns within image data. Furthermore, the review introduces essential strategies to improve model robustness and generalization, including Transfer Learning, Data Augmentation, and techniques for addressing Class Imbalance. Regularization methods like Dropout are also explored for reducing overfitting in CNNs. The chapter concludes with an overview of Deep Learning Ensemble methods, which integrate multiple models to improve prediction reliability and adaptability under real-world astronomical conditions. Altogether, this literature review aims to provide a comprehensive background that supports the design of an effective methodology for transient detection in the subsequent chapters.

2.1 Generation of Sky Survey Data

In this proof-of-concept study, simulated images from the GOTO (Gravitational-wave Optical Transient Observer) telescope were employed instead of real observational data to evaluate the performance of machine learning (ML) algorithms in detecting astronomical transients. The key advantage of using simulated data lies in the ability to predefine and control the ground truth—namely, the identity, position, and brightness of transient sources. This makes simulated images ideal for supervised learning, as the outputs of the algorithm can be directly compared with known labels (Baron, 2019; Ishida & de Souza, 2013). In addition, simulation allows the generation of a wide variety of sky conditions—such as different levels of background noise, atmospheric seeing, and PSF shapes—without relying on costly or inconsistent observational campaigns.

Despite these benefits, simulated data may not capture the full complexity and noise characteristics of real sky images. This leads to what is known as the reality gap—a discrepancy between the clean, controlled nature of simulations and the unpredictable artifacts found in actual telescope observations (Zhang et al., 2020; Gruen et al., 2014). To minimize this gap, the study utilized SkyMaker (Bertin, 2009), a sophisticated astronomical image simulation tool widely used in the astrophysics community. SkyMaker supports the creation of highly realistic images, incorporating stellar and galactic sources, photometric noise, and customizable PSFs in forms such as Gaussian and Moffat profiles (Melchior et al., 2009). For this study, source lists were generated by merging two major sky catalogs: UCAC, which offers reliable data for bright stars, and SDSS, which provides deep coverage for faint stars and galaxies (Drlica-Wagner et al., 2018).

Each celestial source's sky coordinates (RA/Dec) were converted to image-plane pixel coordinates based on the GOTO system's 1.24 arcseconds/pixel scale. Brightness values were assigned using V-band magnitudes for UCAC sources and G-band magnitudes for SDSS entries. Galaxies were modeled using disk-only profiles to reduce parameter complexity, which is acceptable given the study's focus on transient detection rather than morphological analysis (Henrion et al., 2011). Two images were generated per sky region to simulate transient events; the second image included randomly injected transients with magnitudes ranging from 14 to 19. These image pairs were then processed using the LSST Science Pipelines (Juric, 2015), adapted to accept SkyMaker data, with the output difference images serving as the input to the ML models. This simulation pipeline enables controlled training under realistic conditions and supports reproducible model evaluation, consistent with standard practices in time-domain astronomy (Bloom et al., 2012; Korycansky et al., 2009).

2.2 Convolutional Neural Networks in Astronomy: Transient Detection

In recent years, Convolutional Neural Networks (CNNs) have become a cornerstone in astronomical transient detection, offering significant advantages over traditional machine learning methods. The use of CNNs allows astronomers to process large volumes of image data efficiently and with high accuracy, making them ideal for time-domain surveys such as the Zwicky Transient Facility (ZTF) and the upcoming Vera C. Rubin Observatory's LSST. One of the prominent applications of CNNs in this domain is real–bogus classification, which distinguishes genuine astrophysical events from noise or processing artifacts. For instance, (Wardęga et al., 2021) proposed a CNN model that directly compares science and reference image cutouts without using image subtraction. This approach achieved an F1-score of 0.989, outperforming simpler dense-layer networks. Similarly, (Killestein et al., 2021) implemented a Bayesian CNN using Monte Carlo dropout, enabling both classification and uncertainty estimation. Their model, trained on GOTO telescope data, achieved a false negative rate of approximately 0.5% at a 1% false positive rate, demonstrating superior performance in high-confidence real-time vetting. More generalizable frameworks such as O'TRAIN, developed by (Makhlouf et al., 2022), showed robust performance (93–98% accuracy) across various telescopes using conventional CNN architectures. (Liu et al., 2025a)further introduced active and semi-supervised learning approaches to reduce labeling effort. Starting with only ~900 labeled samples from ZTF, their model achieved over 96% accuracy and over 95% recall by iteratively expanding the training set. Beyond static image classification, (Gómez et al., 2020) developed TAO-Net, a hybrid CNN-RNN model capable of learning temporal patterns from sequences of transient images. The model improved classification F1-scores by ~10 percentage points compared to random forest classifiers based on light-curve features. Recent efforts have also focused on benchmarking and standardization. (Dave et al., 2020) published the Deep-TAO dataset, which includes over 1.25 million CRTS image sequences for training and evaluation. Their work confirmed that even basic CNNs trained directly on image sequences can outperform traditional methods reliant on photometric curves. Incorporating auxiliary data with CNNs also shows promise. (Rehemtulla et al., 2024) introduced BTSbot, a multi-modal classifier that combines ZTF image stamps with engineered features (e.g., light-curve statistics). BTSbot achieved a 93% purity rate and recovered 100% of spectroscopically confirmed bright transients in their validation set. Across all these studies, CNNs consistently outperform traditional machine learning models such as Random Forests and Support Vector Machines in accuracy, F1-score, and robustness. These deep learning methods are now being embedded into real-time alert pipelines, significantly reducing human workload and enabling rapid discovery in modern surveys (Gómez et al., 2020; Killestein et al., 2021; Liu et al., 2025; Rehemtulla et al., 2023; Wardęga et al., 2021)

2.2.1 VGGNet

VGGNet, originally proposed by Simonyan and Zisserman (2015), is a convolutional neural network (CNN) architecture known for its use of small 3×3 convolutional filters, deep hierarchical layers, and a simple, uniform design. Although initially developed for large-scale image classification tasks like ImageNet, VGGNet has proven highly effective in a variety of astronomical applications. In time-domain astronomy, one key application is real–bogus classification, where VGGNet helps filter out false positives from transient surveys. For example, Killestein et al. (2021) employed a downsized VGG16 model to classify transient candidates in the GOTO survey using 55×55-pixel image stamps (including science, reference, difference, and PSF-matched residuals), achieving a false negative rate of ~0.5% at a 1% false positive rate—outperforming traditional random forest classifiers. VGGNet has also been used for galaxy morphology classification, as demonstrated by Aniyan and Thorat (2017), who achieved high accuracy in categorizing galaxies based on SDSS data, capturing fine details like spiral arms, bars, and ellipticity. Furthermore, it is widely applied in star–galaxy classification and source segmentation tasks, especially in crowded or low signal-to-noise environments. Zhou et al. (2021), for instance, adapted VGGNet for pixel-level segmentation in infrared images, showing improvements over classical methods like thresholding or SExtractor. Due to its success in computer vision, VGGNet is also frequently used in transfer learning and domain adaptation. Pretrained on ImageNet and fine-tuned on astronomical datasets, VGGNet has demonstrated faster convergence and better generalization, especially when labeled data is limited. Zhou et al. (2021) further showed that fine-tuned VGG models can outperform shallow custom-built networks on small but high-quality datasets.

2.2.2 Xception

Xception replaces traditional Inception modules with depthwise separable convolutions, enabling the model to decouple spatial and cross-channel correlations, thereby reducing parameter count while maintaining strong representational capacity. This architecture is particularly advantageous in astronomy, where real-time inference is crucial for time-domain applications. Compared to models like VGGNet and ResNet, Xception often achieves higher accuracy with fewer parameters, making it suitable for telescope pipelines with limited computational resources. In transient detection, Liu et al. (2022) demonstrated that Xception achieved F1-scores of ≈0.98 on ZTF data, outperforming traditional CNNs and random forest classifiers while processing 64×64 science and difference image cutouts robustly under varying noise and seeing conditions. For galaxy morphology classification, Xception fine-tuned on SDSS and DECaLS data outperformed InceptionV3 and ResNet50, particularly excelling in identifying edge-on spirals and irregular galaxies due to its capacity for capturing fine spatial structures. It was also used to predict galaxy spectral types from images alone—without spectroscopy—achieving over 90% accuracy, highlighting its promise for large-scale photometric surveys. In anomaly detection, Xception's generalization ability was leveraged by Yoon and Kang (2024), who employed an autoencoder-based variant to identify unclassified variable sources in the ATLAS survey, successfully detecting rare transients overlooked by traditional methods. Given the scarcity of labeled astronomical data, Xception is widely used in transfer learning, with pre-trained weights from ImageNet and fine-tuning of top layers reducing training time and data requirements. Domain adaptation techniques such as layer reweighting and color augmentation further enhance model transferability from natural to astronomical imagery.

2.2.3 ResNet

In order to maintain computational efficiency, ResNet, a deep neural network architecture created by (He et al., 2015) primarily uses 3 × 3 convolutional filters while maintaining a consistent number of filters across layers. Stride-2 convolutional layers perform downsampling, and a global average pooling layer completes the network. Shortcut connections are then incorporated to allow for residual learning. The preprocessing methods applied to the input are taken from Simonyan and (Krizhevsky et al., 2017; Simonyan & Zisserman, 2015). To reduce internal covariate shift, Batch Normalization (BN) (Ioffe & Szegedy, 2015) is used both before and just after each convolutional layer. In this design, dropout layers are unnecessary since regularization is provided via batch normalization. Using stochastic gradient descent (SGD), the model is trained with a batch size of 256, a weight decay of 0.0001, and a momentum of 0.9. Training plateaus reduce the initial learning rate, which is set at 0.01 and is reduced by a factor of 10. Convergence is ensured by training for up to 60×10^4 iterations.

2.2.4 DenseNet

In order to promote effective feature reuse and improved gradient propagation, (Huang et al., 2018) presented the Dense Convolutional Network (DenseNet), a neural network design in which each layer uses all previous feature maps as input. DenseNet may combine knowledge from all previous layers thanks to its connectedness structure, which increases learning efficiency. DenseNet significantly improves information flow as it has L(L+1)/2 connections, if L is the number of layers. Every layer applies a non-linear transformation $H_l(\cdot)$, which consists of a 3 × 3 convolution, Batch Normalization (BN), and ReLU activation. In order to reduce dimensionality while preserving important learnt properties, the network's closely connected blocks are separated by transition layers that include Batch Normalization, a 1 x 1 convolutional layer, and a 2 × 2 average pooling layer.

2.3 Transfer Learning

Transfer Learning is a powerful paradigm in machine learning that leverages knowledge gained from solving one task to enhance the learning performance of another, often related but distinct task. Instead of training models from scratch, which can be computationally expensive and require vast labeled datasets, Transfer Learning enables the reuse of feature representations from pre-trained models typically trained on large-scale image datasets such as ImageNet (Deng et al., 2009). These pre-trained networks, such as VGGNet (Simonyan & Zisserman, 2015), ResNet (He et al., 2016), Inception (Szegedy et al., 2015), and Xception (Chollet, 2017), have demonstrated significant success across a range of visual tasks by capturing robust hierarchical features, which can generalize well to new domains with limited data. A critical advantage of Transfer Learning is its ability to reduce the need for extensive labeled training data by transferring low- and mid-level features from general-purpose datasets to domain-specific applications. In practice, this involves replacing or fine-tuning the classification head of the model while optionally unfreezing deeper convolutional layers, a process known as fine-tuning. Studies have shown that combining Transfer Learning with data augmentation techniques and domain adaptation strategies significantly enhances model robustness and generalization (Charnock & Moss, 2017). Despite its advantages, Transfer Learning in scientific imaging is not without challenges. The domain gap between natural images (used for pre-training) and scientific images (such as astronomical data) can sometimes limit performance. Hence, researchers must carefully choose which layers to retrain and which to freeze based on the similarity between source and target domains (Zhuang et al., 2020).

2.4 Transfer Learning in Astronomical Image Analysis

In the domain of astronomy, Transfer Learning has gained popularity due to its effectiveness in handling limited and imbalanced datasets, which are common in astrophysical observations (Sánchez et al., 2018). For example, (Cavanagh et al., 2021) demonstrated the success of a fine-tuned Xception model in classifying galaxy morphologies, outperforming other deep learning architectures including InceptionV3 and ResNet50. Similarly, models pre-trained on ImageNet have been repurposed for tasks such as real-bogus classification of transient detections (Cabrera-Vives et al., 2017), galaxy morphological classification (Dieleman et al., 2015), and even gravitational wave event analysis (George & Huerta, 2018).

2.5 Deep Learning Ensemble

Ensemble learning in deep learning refers to the process of combining predictions from multiple models to improve accuracy, robustness, and generalization. While a single neural network may be prone to overfitting or sensitive to specific data perturbations, ensemble methods exploit the diversity among multiple models to reduce variance and increase predictive reliability. In the context of deep learning, ensemble strategies typically include techniques such as bagging, boosting, and model averaging, but more commonly involve approaches like soft voting, hard voting, and weighted voting, where outputs from different models—often trained with varying architectures, hyperparameters, or data augmentations—are aggregated to form a consensus prediction (Dietterich, 2000; Hansen & Salamon, 1990). In astronomical image classification, ensemble methods have proven particularly useful in improving classification reliability for rare or ambiguous events, such as the real-bogus classification of transient detections. For example, ensembles combining convolutional neural networks like ResNet, Xception, and MobileNet have been shown to outperform individual models by balancing the trade-offs between precision and recall (Möller & de Boissière, 2020). Additionally, ensemble techniques are valuable in mitigating the effects of domain shift and observational noise, which are common in data collected from different telescopes or under varying sky conditions. When combined with transfer learning and data augmentation, deep ensemble methods enable more stable and trustworthy decision-making in high-stakes scientific applications such as supernova detection, galaxy morphology classification, and gravitational wave follow-up observations (Carrasco-Davis et al., 2019)

Table 2.1 Summarization including detail of literature review

Year	Technology	Interesting Points	Limitation
1. Previous work in Generation of Sky Survey Data			
	Extract feature	Extract feature from GOTO Sky Survey Image	Some information loss while extracting feature.
	Extract feature and Imbalance	Extract feature from GOTO Sky Survey Image that imbalance and used machine learning for observe	Some in formation loss while extract feature and low accuracy because
	Image, CNN and Imbalance (Rotation)	Using image with CNN and using rotation training data to improve imbalance	Proof of concept that GOTO sky image can be using with CNN and rotation image can
	HOTPANTS	One of the most widely used method for astronomical image subtraction which works by taking two aligned images of the same filed and in different time, dividing them into several regions and calculating convolutional kernels to math point spread functions (PSF) for each region	HOTPANTS is an effective method for image subtraction and has been used for a long period of time but it also has some disadvantages. Frist, we checked the remaining on the image produced by HOTPANTS and found over 50% remaining stars are of small sizes
	SExtractor	SExtractor is a program that vuilds acatalogue of objects from an astronomical image. Although it is particularly oriented towards reduction of large-scale galaxy-survey data, it can perform reasonably well on moderately crowed star fields	

Table 2.1 (continued)

Year	Technology	Interesting Points	Limitation
2. Convolutional Neural Networks in Astronomy: Transient Detection			
2021	CNN	Proposed a CNN model that directly compares science and reference image cutouts without using image subtraction; achieved an F1-score of 0.989, outperforming simpler dense-layer networks.	May struggle with subtle transients or variations in image conditions without subtraction.
2021	Bayesian CNN (Monte Carlo dropout)	Implemented a Bayesian CNN using Monte Carlo dropout for classification and uncertainty estimation; achieved a false negative rate of approximately 0.5% at a 1% false positive rate on GOTO telescope data.	Higher computational cost; generalizability to diverse telescope data may be limited.
2022	Conventional CNN architectures (O'TRAIN)	Showed robust performance (93–98% accuracy) across various telescopes using conventional CNN architectures.	"Conventional" architecture may not fully exploit advanced deep learning; optimal performance might require some fine-tuning for specific telescopes.
2025	Active and semi-supervised learning approaches with CNN	Introduced active and semi-supervised learning approaches to reduce labeling effort; achieved over 96% accuracy and over 95% recall by iteratively expanding the training set, starting with only ~900 labeled samples from ZTF.	Initial small labeled dataset can introduce biases; active learning effectiveness depends on query strategy; may struggle with novel transient types.

Table 2.1 (continued)

Year	Technology	Interesting Points	Limitation
2020	Hybrid CNN-RNN (TAO-Net)	Developed TAO-Net, a hybrid CNN-RNN model capable of learning temporal patterns from sequences of transient images; improved classification F1-scores by ~10 percentage points compared to random forest classifiers.	Requires sequential image data, which may not always be available; computationally intensive for long sequences.
2020	Basic CNNs (Deep-TAO dataset)	Published the Deep-TAO dataset (over 1.25 million CRTS image sequences); confirmed that even basic CNNs trained directly on image sequences can outperform traditional methods reliant on photometric curves.	Dataset specific to CRTS; basic CNNs might miss intricate features.
2024	Multi-modal classifier (BTSbot)	Introduced BTSbot, a multi-modal classifier that combines ZTF image stamps with engineered features; achieved a 93% purity rate and recovered 100% of spectroscopically confirmed bright transients in their validation set.	Relies on handcrafted features; performance for fainter transients may vary.

Table 2.1 (continued)

Year	Technology	Interesting Points	Limitation
2015	VGGNet	- Uses small 3x3 filters, deep layers, simple design. - Effective for astronomical image classification and segmentation tasks. - Performs well with Transfer Learning.	- Resource Intensive: Has a large number of parameters (over 138 million), leading to high memory and disk usage. - Slow Training: Training the model is very time-consuming, even with high-performance GPUs. - Prone to Overfitting: The large number of parameters can make it prone to overfitting, especially with smaller datasets.
2017	Xception	- Uses Depthwise Separable Convolutions for efficiency and strong representation. - Achieves high accuracy with fewer parameters, suitable for limited resources. - Very useful in astronomy for transient detection, galaxy classification, and anomaly detection. - Works well with Transfer Learning.	- Implementation Complexity: Despite efficiency, the specific architecture can be complex to implement. - Hyperparameter Tuning: Performance is sensitive to hyperparameter adjustments. - Data Requirement: Still requires substantial labeled data for specialized tasks, even with Transfer Learning. - Interpretability: Like most neural networks, understanding its decisions can be challenging.
2015	ResNet	- Uses 3x3 filters & shortcut connections for efficient deep learning. - Batch Normalization helps with regularization. - Robust training setup.	- High computational cost: Can be demanding to train/deploy. - Interpretability: Hard to understand internal decision-making. - Hyperparameter sensitivity: Requires careful tuning.

Table 2.1 (continued)

Year	Technology	Interesting Points	Limitation
2018	DenseNet	- Layers connect to all previous feature maps for reuse and better gradient flow. - High information flow. - Uses standard convolution, BN, ReLU, and transition layers.	- High Memory Consumption: Requires significant memory due to dense connections. - Computational Overhead: Numerous connections lead to increased computational load. - Implementation Complexity: More complex to implement from scratch.
3. Transfer Learning			
2009	ImageNet	A large-scale image dataset commonly used for pre-training models in Transfer Learning.	May require extensive fine-tuning for specialized domains.
2015	VGGNet, Inception	Pre-trained networks that demonstrated significant success across various visual tasks by capturing robust hierarchical features.	VGGNet: High memory/computational cost.
2016	ResNet	A pre-trained network that demonstrated significant success across various visual tasks by capturing robust hierarchical features.	Deep models still resource-intensive for fine-tuning.
2017	Xception	A pre-trained network that demonstrated significant success across various visual tasks by capturing robust hierarchical features.	May slightly limit complex inter-channel learning.
2017	Transfer Learning with Data Augmentation & Domain Adaptation (Charnock & Moss)	Combining Transfer Learning with these techniques significantly enhances model robustness and generalization.	Augmentation requires domain expertise.

Table 2.1 (continued)

Year	Technology	Interesting Points	Limitation
2020	Transfer Learning	Researchers must carefully choose which layers to retrain and which to freeze based on the similarity between source and target domains.	Major Domain Gap: Significant performance drop with highly dissimilar domains (e.g., natural vs. scientific images).
4. Transfer Learning in Astronomical Image Analysis			
2018	Transfer Learning	Gained popularity in astronomy due to effectiveness in handling limited and imbalanced datasets, common in astrophysical observations.	Performance sensitive to domain gap (natural vs. astronomy images).
2021	Fine-tuned Xception model (vs. InceptionV3, ResNet50)	Successfully classified galaxy morphologies, outperforming other deep learning architectures.	May struggle with highly irregular/faint galaxies.
2017	ImageNet (pre-trained models)	Repurposed for "real-bogus" classification of transient detections.	High data imbalance makes reducing false positives/negatives challenging.
2015	ImageNet (pre-trained models)	Repurposed for galaxy morphological classification.	Potential for human labeling bias to be amplified.
2018	ImageNet (pre-trained models)	Repurposed for gravitational wave event analysis.	Extreme faintness of signals and high noise make detection difficult.

Table 2.1 (continued)

Year	Technology	Interesting Points	Limitation
5. Deep Learning Ensemble			
	Ensemble Learning (Core Concept)	Combines predictions from multiple models to improve accuracy, robustness, and generalization; exploits diversity to reduce variance and increase predictive reliability.	Increased computational cost and complexity.
2000, 1990	Ensemble Strategies: Bagging, Boosting, Model Averaging, Soft Voting, Hard Voting, Weighted Voting	Aggregates outputs from different models (often with varying architectures, hyperparameters, or data augmentations) to form a consensus prediction.	Non-trivial optimization for combining strategies.
2020	Ensemble Methods in Astronomical Image Classification (e.g., combining ResNet, Xception, MobileNet)	Proven useful in improving classification reliability for rare or ambiguous events; outperforms individual models by balancing precision and recall.	Reduced model interpretability due to increased complexity.
2019	Deep Ensemble Methods (combined with Transfer Learning and Data Augmentation)	Valuable in mitigating effects of domain shift and observational noise; enables more stable and trustworthy decision-making in high-stakes scientific applications.	Very high computational demands (training & inference).

CHAPTER 3

RESEARCH METHODOLOGY

This chapter outlines a systematically designed research framework aimed at evaluating the effectiveness of deep learning models in classifying astronomical objects captured through telescope imaging. The classification task is particularly challenging due to the extreme class imbalance between real objects (e.g., supernovae, kilonovae) and bogus detections caused by sensor noise, cosmic rays, or image processing artifacts. To address this, the methodology incorporates a comprehensive pipeline, beginning with dataset preparation and image format conversion, followed by advanced data augmentation techniques to increase training diversity. Subsequently, a suite of Convolutional Neural Network (CNN) architectures is trained using both transfer learning and fine-tuning strategies. Model performance is rigorously assessed using multiple evaluation metrics, and ensemble learning techniques are finally employed to enhance classification robustness. This chapter provides a detailed account of each methodological component, ensuring the reproducibility and reliability of the experimental results within the context of astronomical transient detection.

3.1 Method Overview

The workflow illustrated in Figure 3.1 represents a systematically designed research framework aimed at evaluating the capabilities of deep learning models in classifying astronomical objects captured in telescope images. These objects are typically categorized into two main classes: *real* objects such as supernovae and other transient phenomena that genuinely exist in the cosmos and *bogus* objects, which often result from camera errors, background noise, or processing artifacts. Distinguishing between these two categories poses a significant challenge, especially due to the extreme class imbalance inherent in raw astronomical datasets, where bogus instances far outnumber real ones. This imbalance can hinder model accuracy and lead to statistical bias. To address this issue, the initial stage of the research involves enhancing dataset diversity through data augmentation techniques, including the use of original images, noise injection, image rotation, vertical flipping (VFlip), and horizontal flipping (HFlip). These augmentations aim to increase the variability of training data and improve model generalization in the face of complex and distorted inputs.

The next phase of the workflow focuses on training Convolutional Neural Network (CNN) architectures, which are highly effective for image analysis tasks. This research utilizes nine well-known CNN models: DenseNet121, VGG16, VGG19, ResNet50, ResNet101, MobileNet, MobileNetV2, InceptionV3, and Xception each with distinct architectural strengths. For example, ResNet incorporates residual learning to mitigate vanishing gradient issues, MobileNet is lightweight and suitable for real-time inference, while Xception offers high performance through the use of depthwise separable convolutions. The training strategies are divided into two approaches: training from scratch (without pre-initialized weights) and utilizing pre-trained models from ImageNet. The latter approach includes both traditional transfer learning (where pretrained layers are kept fixed) and fine-tuning (where selected layers are retrained to adapt to domain-specific astronomical data).

During training, key hyperparameters known to affect model performance are systematically varied, including dropout rates (0.2, 0.3, and 0.5) a regularization technique to prevent overfitting by randomly disabling neurons during training and batch sizes (32, 64, 128, and 256), which influence learning stability and computational efficiency. These parameters are evaluated across all models and augmentation settings to identify the optimal configuration for each scenario. Upon completion of model training, the results are analyzed and compared using standard performance metrics accuracy, precision, recall, and F1-score across both the real and bogus classes. The best-performing models for each type of augmentation are then selected for ensemble integration. By combining these models, the ensemble approach aims to leverage their individual strengths and mitigate their weaknesses. This is achieved through soft voting, where class probabilities are averaged, and weighted voting, where models contribute based on their relative performance.

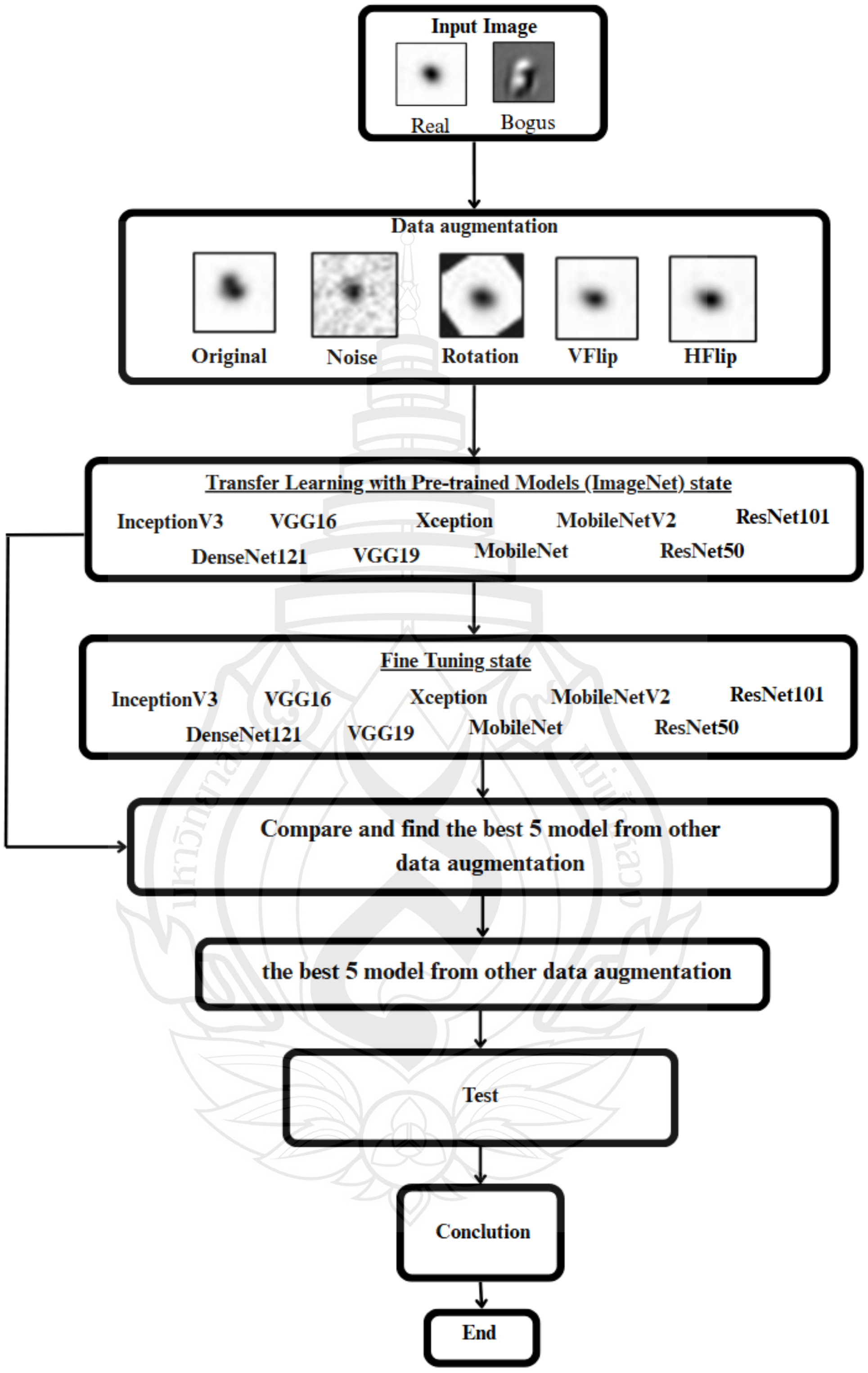


Figure 3.1 Overall methodology

3.2 Dataset

The dataset employed in this study originates from the GOTO (Gravitational-wave Optical Transient Observer) survey project, which is designed to detect optical counterparts of gravitational-wave events (Steeghs et al., 2022) Specifically, we use different images, which are generated by subtracting a static reference image from a new observation to isolate transient sources.

Each sample in the dataset is a grayscale image of size 21×21 pixels, centered on a potential transient source. These compact cutouts are sufficient to capture the local point spread function (PSF) and relevant background context, allowing for classification based on pixel-level intensity distributions.

All images have been manually labeled by expert astronomers into two categories:

1. Real: Confirmed transient detections, such as supernovae or kilonovae, showing clear astrophysical characteristics.
2. Bogus: False positives arising from various sources such as cosmic rays, CCD defects, or imperfect image subtraction.

Table 3.1 Number of samples

Class	Number of Samples
Real	523
Bogus	3,598

This highly imbalanced dataset mirrors real-world conditions in transient detection pipelines, where most candidate detections are spurious. Such imbalance presents a significant challenge for classification models and motivates the use of data augmentation, custom loss functions, or oversampling strategies in model training (Buda et al., 2018; Krawczyk, 2016).

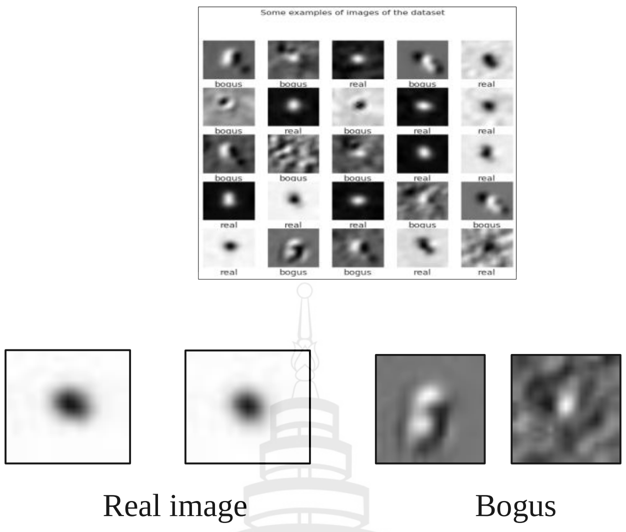


Figure 3.2 The example of dataset

3.2.1 Data Preprocessing

In the original format, the astronomical images in this dataset are stored as FITS files (fits), a standard format widely used in astronomy that contains multi-dimensional arrays and metadata headers (Pence et al., 2010). While FITS is suitable for scientific storage and analysis, it is not directly compatible with most deep learning frameworks and image processing libraries that expect conventional image formats such as JPEG or PNG. To streamline the modeling process and facilitate integration with popular computer vision pipelines (e.g., TensorFlow or PyTorch), we converted the FITS files into JPEG (.jpg) format. This step simplifies data loading, visualization, and augmentation, while preserving the key pixel-level features needed for classification. The conversion was carefully conducted using normalization and stretching techniques to ensure that astrophysical features remained distinguishable.

3.2.2 Data Augmentation

Data augmentation is a well-established technique in deep learning, commonly used to artificially expand the training dataset by applying various transformations to existing images. This process helps improve the generalization capability of the model by exposing it to diverse variations of the data, thereby reducing the risk of overfitting and enhancing final classification accuracy (García-Jara et al., 2022). In line with findings from previous studies, we incorporate data augmentation into our training pipeline to increase robustness and improve performance on unseen data. Augmentations such as horizontal flips, rotations, and noise injection are applied to simulate real-world distortions and observational variability. This strategy aims to help the network learn invariant features and become more resilient to subtle differences in transient images.

The use of data augmentation in this study aims to enhance the performance of deep learning models under the constraints of limited and highly imbalanced datasets—particularly in the context of transient object classification, where "real" instances are significantly outnumbered by "bogus" detections. Data augmentation increases training diversity by applying various transformations to the original images, such as rotations, flipping, and noise injection, to simulate real-world variability in object position, camera angle, and image quality. These techniques enable the model to learn invariant features that are robust to positional and observational distortions, thereby reducing biases from repetitive image patterns and improving generalization to unseen data. Altogether, this strategy plays a vital role in improving the accuracy, stability, and reliability of the model when applied to real-world astronomical imaging.

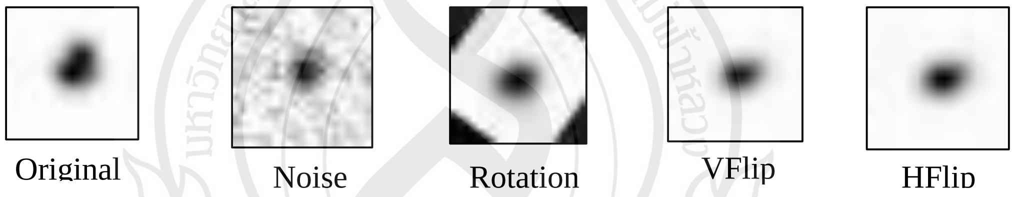


Figure 3.3 The example of data augmentation

3.3 Modeling

In deep learning for image classification, Convolutional Neural Networks (CNNs) have proven to be powerful tools due to their ability to learn hierarchical spatial features, making them especially suitable for analyzing astronomical images where spatial structure and light distribution are critical for identifying celestial phenomena. Over the past decade, numerous CNN architectures have been developed, each offering distinct strengths in terms of depth, parameter count, and computational efficiency. In this study, we selected nine representative CNN models—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet50, ResNet101, VGG16, VGG19, and Xception—to evaluate their effectiveness in classifying transient astronomical events. These models were chosen based on their widespread use in previous image classification research leveraging transfer learning from ImageNet, a large-scale dataset commonly used to pre-train CNNs. Each model used in this study incorporates pre-trained ImageNet weights, allowing for effective feature transfer from general visual patterns to the specific domain of astronomical imaging. In the following sections, we provide a detailed overview of each model's architecture, advantages, and its relevance to this task.

3.3.1 DenseNet121

DenseNet121 (Densely Connected Convolutional Networks) is a CNN architecture that introduces dense connectivity between layers, where each layer receives inputs from all preceding layers and passes its own feature maps to all subsequent layers. This design helps improve feature propagation, encourages feature reuse, and mitigates the vanishing gradient problem commonly found in deep networks. DenseNet121 is particularly efficient in terms of parameter usage, making it suitable for deep learning tasks with limited computational resources. In astronomical image classification, the dense connections allow the model to retain fine-grained spatial information critical for distinguishing subtle features in transient events. When pre-trained on ImageNet, DenseNet121 benefits from a strong initialization that accelerates convergence and improves generalization, especially when applied to datasets with fewer labeled samples, as often found in astronomy (Huang et al., 2017).

3.3.2 InceptionV3

InceptionV3 is a highly optimized convolutional neural network architecture that builds upon the earlier Inception models by introducing several enhancements to improve both computational efficiency and accuracy. The key innovation in Inception networks lies in their use of Inception modules, which perform parallel convolutions of different kernel sizes (e.g., 1×1, 3×3, and 5×5) within the same layer. This design allows the network to capture features at multiple scales simultaneously, enabling it to handle complex spatial patterns commonly found in astronomical images. InceptionV3 further improves upon its predecessors by incorporating techniques such as factorized convolutions, asymmetric kernels, and label smoothing, which reduce computational cost while maintaining high representational power. In the context of transient astronomical event classification, InceptionV3 is particularly well-suited due to its ability to extract diverse feature representations, making it robust to variations in object size, orientation, and brightness. When pre-trained on ImageNet, the model inherits a rich set of visual features that can be effectively transferred to astronomical data, even with limited labeled samples. Its deep and modular design also helps prevent overfitting and facilitates efficient training, making it a popular choice in many image-based scientific applications (Szegedy et al., 2016).

3.3.3 MobileNet

MobileNet is a lightweight and efficient convolutional neural network architecture designed specifically for mobile and embedded vision applications. It achieves computational efficiency by replacing standard convolutions with depthwise separable convolutions, which factorize a conventional convolution into two simpler operations: a depthwise convolution (which applies a single filter per input channel) followed by a pointwise convolution (a 1×1 convolution that combines the outputs). This significantly reduces the number of parameters and computational cost, making MobileNet ideal for devices with limited resources. In the context of astronomical transient classification, MobileNet's compact structure allows for fast inference while still capturing essential spatial features. Although it is shallower and less complex than other deep CNNs, MobileNet remains effective when pre-trained on ImageNet and fine-tuned on domain-specific datasets. Its efficiency and adaptability make it particularly suitable for real-time or resource-constrained astronomical applications, such as edge computing in observatories or satellite-based image analysis (Howard et al., 2017).

3.3.4 MobileNetV2

MobileNetV2 is an improved version of the original MobileNet architecture, designed to enhance efficiency and accuracy for mobile and embedded applications. The core innovation of MobileNetV2 lies in the introduction of inverted residual blocks and linear bottlenecks. Unlike traditional residual connections, the inverted residual block expands the input to a higher-dimensional space, applies depthwise separable convolutions, and then projects the features back to a low-dimensional output. This structure allows for better information flow and reduces the risk of losing critical features during transformation. In astronomical image classification, especially in tasks involving subtle differences in brightness or shape (as seen in transient detection), MobileNetV2 offers a good balance between speed and performance. Pre-trained on ImageNet and fine-tuned for domain-specific data, MobileNetV2 is capable of generalizing well even with limited data. Its lightweight nature and enhanced representational capacity make it a practical choice for real-time, edge-based inference in astronomy applications (Sandler et al., 2018).

3.3.5 ResNet50 and ResNet101

ResNet50 and ResNet101 are both part of the Residual Network (ResNet) family, which introduced the concept of residual learning to effectively train very deep neural networks. The key architectural feature in ResNet models is the shortcut (or skip) connection, which allows the network to bypass certain layers by adding the input of a layer directly to its output. This mechanism addresses the degradation problem and helps maintain the strength of gradients during backpropagation, enabling the training of extremely deep models without sacrificing performance. ResNet50 contains 50 layers, while ResNet101 extends the architecture to 101 layers, offering increased depth and feature learning capacity. The added depth in ResNet101 allows it to learn more abstract and complex representations, which can be particularly useful for distinguishing subtle variations in astronomical images. However, this comes with increased computational cost compared to ResNet50. In practice, both models perform strongly when pre-trained on ImageNet and fine-tuned on domain-specific data. ResNet50 offers a solid balance between accuracy and efficiency, while ResNet101 provides a deeper structure suited for more complex classification tasks—such as identifying transient astronomical events where fine-grained detail matters (He et al., 2016).

3.3.6 VGG16 and VGG19

VGG16 and VGG19 are convolutional neural network architectures developed by the Visual Geometry Group (VGG) at the University of Oxford. They are known for their simplicity and uniform architecture, where the entire network is composed primarily of 3×3 convolutional layers stacked on top of each other with ReLU activations, followed by max pooling and fully connected layers. The difference between the two lies in their depth: VGG16 has 16 weight layers, while VGG19 has 19, making VGG19 slightly deeper and potentially more expressive. Despite their relatively simple design, both VGG16 and VGG19 are powerful feature extractors, especially when pre-trained on large-scale datasets such as ImageNet. Their straightforward architecture makes them highly interpretable and easy to implement, which contributes to their lasting popularity in various image classification tasks. In the context of astronomical transient classification, these models are capable of capturing spatial features with high fidelity, though they tend to require more memory and computational power compared to more modern architectures. Nevertheless, their robustness and proven performance make them strong baseline models for transfer learning in scientific image analysis (Simonyan & Zisserman, 2014).

3.3.7 Xception

Xception (Extreme Inception) is a deep convolutional neural network architecture that builds upon the ideas of the Inception family but takes them a step further by fully replacing Inception modules with depthwise separable convolutions. This design separates spatial filtering (depthwise convolution) from channel-wise combination (pointwise 1×1 convolution), allowing the network to model spatial and cross-channel correlations independently. As a result, Xception is more efficient in terms of parameters and computation, while still offering high representational power. In the context of astronomical image classification, especially for transient detection tasks, Xception is highly effective due to its ability to capture subtle spatial patterns and structure in image data. The architecture is particularly useful in handling high-resolution images and variations in object brightness or orientation. When pre-trained on ImageNet, Xception provides strong initialization and generalizes well to scientific domains, including astronomy. Its depth and separation of concerns make it a modern and powerful alternative to traditional CNNs, often outperforming models like VGG or InceptionV3 in accuracy and efficiency (Chollet, 2017).

To evaluate the performance of the nine selected CNN architectures, we employed a two-phase training strategy based on transfer learning and fine-tuning. Initially, each model was initialized with pretrained weights from the ImageNet dataset, which enables the transfer of generalized visual features to the astronomical domain. During the transfer learning phase, all convolutional layers were frozen to preserve the pre-trained feature extractors, and only the top-level fully connected layers were trained for the classification of transient events. This approach is especially advantageous for datasets with limited labeled samples, as it significantly reduces training time and overfitting risk. Following this phase, a fine-tuning step was applied in which the top 30% of the convolutional layers were unfrozen to allow the model to adapt more specifically to the target domain. This gradual unfreezing enhances feature discrimination, particularly for subtle astronomical patterns.

Table 3.2 summarizes the key hyperparameters used across all training sessions. Four batch sizes were tested (32, 64, 128, and 256), with a fixed number of 100 epochs and early stopping (patience = 3). The Adam optimizer was used for all models, with separate learning rates for transfer learning (0.001) and fine-tuning (0.00001). The binary cross-entropy loss function was employed to support binary classification of transient vs. non-transient image samples.

Table 3.2 Basic parameters

Parameter	Value
Batch Size	32,62,128,256
Epoch	100, Early Stopping (patience = 3)
Learning	0.001(TF), 0.00001(FT)
Optimizer	Adam
Loss Function	Binary Crossentropy
Fine-Tuning unlocks	Top 30%

3.4 Transfer Learning

Transfer Learning was employed as the core strategy for training all nine CNN architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet50, ResNet101, VGG16, VGG19, and Xception—by initializing each model with pre-trained weights from the ImageNet dataset. This approach allowed the models to reuse general-purpose features such as edges, textures, and spatial patterns learned from large-scale image classification tasks and apply them effectively to the domain of astronomical transient detection, where labeled data is often limited. By freezing all convolutional layers and training only the top classification layers during the initial phase, the models could adapt quickly while avoiding overfitting and reducing computational cost. This strategy provided a strong and consistent foundation across all models, enabling fair performance comparisons and laying the groundwork for further domain-specific optimization through Fine-Tuning in the subsequent phase.

3.5 Fine-Tuning

After completing the Transfer Learning phase, where only the top classification layers were trained while the convolutional base was frozen, a second stage known as Fine-Tuning was conducted to further improve model performance by allowing deeper parts of the network to adapt to the target domain. In this phase, the top 30% of the convolutional layers in each model architecture were unfrozen, enabling the model to fine-tune not only the classification head but also deeper feature extractors that are more task-specific. This approach provides a more flexible learning process that allows the model to gradually shift from general-purpose features learned from ImageNet toward features that are highly relevant to transient astronomical events, which often involve subtle differences in shape, brightness, noise, or spatial distribution. Fine-Tuning plays a critical role in bridging the gap between generalized feature learning and domain specialization. If the model relies solely on transfer learning without any fine-tuning, it may overlook important domain-specific patterns. Conversely, if too many layers are unfrozen too quickly, the model risks losing its prior knowledge and becoming unstable. Therefore, a controlled unfreezing strategy was adopted in this study to strike a balance between adaptability and stability. The learning rate during this phase was significantly reduced to 0.00001 to allow for fine-grained weight adjustments without destabilizing the pre-trained structure. The Adam optimizer and Binary Crossentropy loss function continued to be used to ensure consistency across both training phases. The Fine-Tuning process was applied consistently across all nine models to ensure fairness in comparison. Additionally, the same early stopping criteria (patience = 3) were employed to avoid overfitting, especially when training deeper layers on limited data. This phase allowed the models to improve their capacity to distinguish between real and bogus transients more precisely by learning to focus on domain-specific features such as edge irregularities, faint sources, or morphological noise characteristics commonly seen in astronomical image data. Overall, Fine-Tuning significantly enhanced the adaptability, precision, and robustness of each model, complementing the benefits gained during the initial Transfer Learning phase. As illustrated in Figure 3.3, the upper part of the diagram represents the source domain, where the model learns hierarchical features from generic images like cars, trees, and shoes

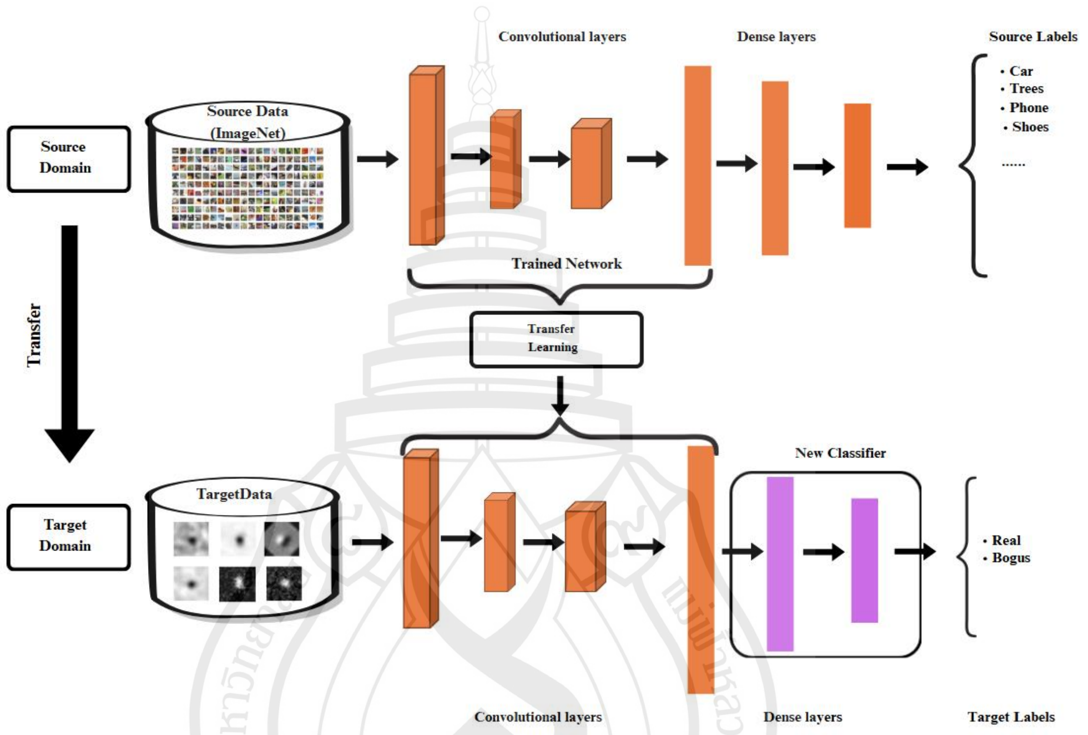


Figure 3.4 The process of transferring learned features from a source domain (ImageNet) to a target domain (astronomical transient data) using the transfer learning approach

3.6 Ensemble Learning

To further enhance the classification performance and robustness of the system, Ensemble Learning was employed as a final stage following individual model training. Ensemble Learning is a machine learning strategy that combines the predictions of multiple models to produce a single, more accurate and stable output. In this study, the ensemble method was applied to integrate the strengths of the nine CNN architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet50, ResNet101, VGG16, VGG19, and Xception—each of which had been independently trained and fine-tuned on the same astronomical transient dataset. By aggregating their outputs using methods such as Soft Voting and Weighted Voting, the ensemble approach mitigated the weaknesses of any single model and allowed the system to generalize more effectively across varied image conditions, including noise, brightness shifts, and morphological variations. Soft Voting averages the probability outputs of each model, selecting the class with the highest mean probability, while Weighted Voting assigns greater influence to models with higher individual validation performance. This ensemble strategy was particularly useful in the context of real-vs-bogus classification, where subtle differences and imbalanced data could lead to model-specific errors. Through ensemble integration, the system benefited from increased resilience, improved prediction accuracy, and greater confidence in final classification outcomes, thereby enhancing its applicability to real-world transient detection scenarios in astronomy. Figure 3.4, five distinct CNN models were strategically selected to participate in the ensemble. Each model was trained using different Data Augmentation strategies—including Original, Rotation, Horizontal Flip (HFlip), Vertical Flip (VFlip), and Noise Injection—as well as varying Batch Sizes (32, 64, 128, and 256). Architectures such as MobileNet and Xception, under both Transfer Learning (TF) and Fine-Tuning (FT) settings, were chosen based on their prior individual performance on validation datasets.

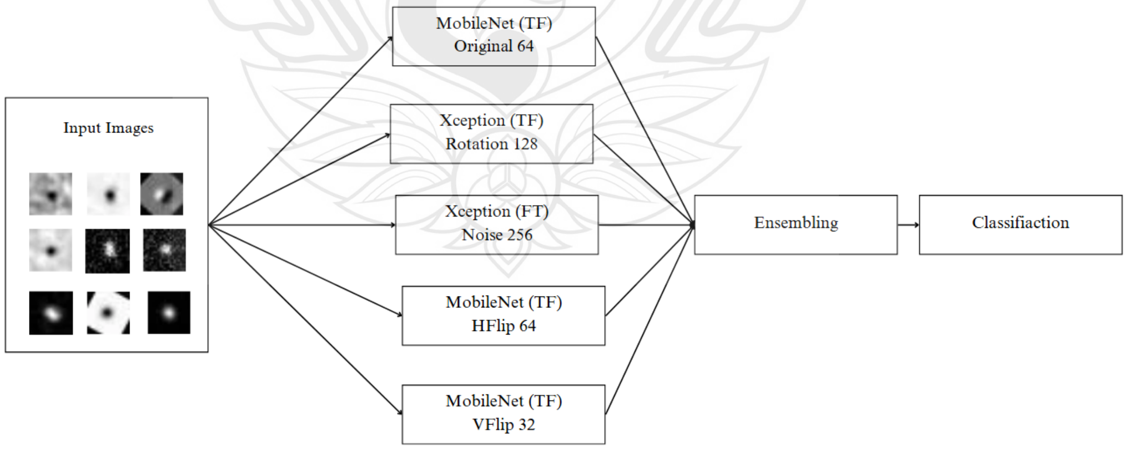


Figure 3.5 Ensemble architecture combining multiple CNN models trained under different augmentation strategies and batch sizes

3.7 Evaluation

To evaluate the performance of the classification models, especially in the context of distinguishing between real and bogus astronomical images, this research employed multiple standard metrics derived from the confusion matrix. These metrics include Accuracy, Precision, Recall (Sensitivity), and F1-Score, which are essential in assessing both the overall and class-specific performance of each model and ensemble method.

All metrics were computed based on the following four quantities:

1. True Positive (TP): Number of real images correctly classified as real.
2. False Positive (FP): Number of bogus images incorrectly classified as real.
3. True Negative (TN): Number of bogus images correctly classified as bogus.
4. False Negative (FN): Number of real images incorrectly classified as bogus.

From these quantities, the following evaluation metrics were calculated:

3.7.1 Accuracy

Accuracy measures the proportion of correct predictions among the total number of predictions.

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

High recall ensures that the model captures most of the real detections, which is critical in domains like transient detection where missing real events could be costly.

3.7.2 Precision

Precision (also called Positive Predictive Value) evaluates the correctness of positive predictions. It is particularly important when the cost of false positives is high.

$$Precision = \frac{TP}{TP + FP}$$

High precision indicates a low false positive rate, ensuring that instances classified as “real” are truly real.

3.7.3 Recall (Sensitivity)

Recall (or True Positive Rate) measures the ability of the model to correctly identify all relevant instances.

$$Recall = \frac{TP}{TP + FN}$$

High recall ensures that the model captures most of the real detections, which is critical in domains like transient detection where missing real events could be costly.

3.7.4 F1-Score

F1-Score is the harmonic mean of precision and recall. It provides a balance between the two, especially in cases where there is a trade-off.

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

F1 score is particularly useful in imbalanced datasets, where a high score indicates that both false positives and false negatives are being minimized.

3.7.5 Confusion Matrix Visualization

To further interpret model behavior, confusion matrices were constructed and visualized for each model and ensemble configuration. These matrices provide intuitive insight into where the models tend to misclassify, such as confusing real transients for bogus ones or vice versa.

CHAPTER 4

IMPLEMENTATION

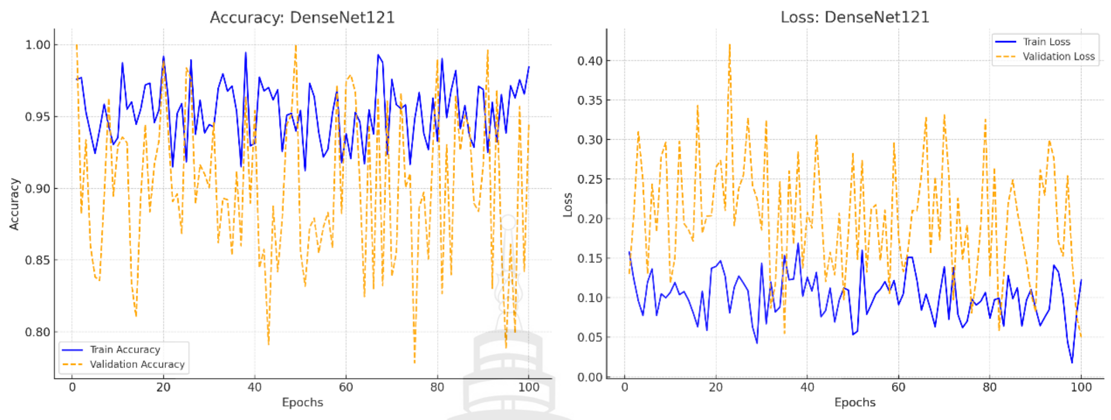
In this experiment, the performance of Convolutional Neural Network (CNN) models was compared using Transfer Learning and Fine-Tuning techniques, combined with Data Augmentation and Dropout, to regulate the learning process and prevent overfitting—a major challenge in astronomical image analysis due to the high complexity of the data and the imbalance between real and bogus samples. The experiment involved evaluating various CNN architectures under different augmentation conditions, including vertical and horizontal flipping (VFlip, HFlip), rotation, noise injection, and the use of original unaugmented data. Additionally, models were tested with different Dropout rates (0.2 and 0.5) to assess their effect on the learning capability and generalization of the models.

This chapter presents the results obtained from each experimental condition, accompanied by analysis and discussion of the comparative performance of each technique. The findings are based on validation accuracy and loss curves, with recommendations for future experimentation aimed at enhancing the robustness and applicability of the models to real-world astronomical data, which is often highly variable and difficult to classify.

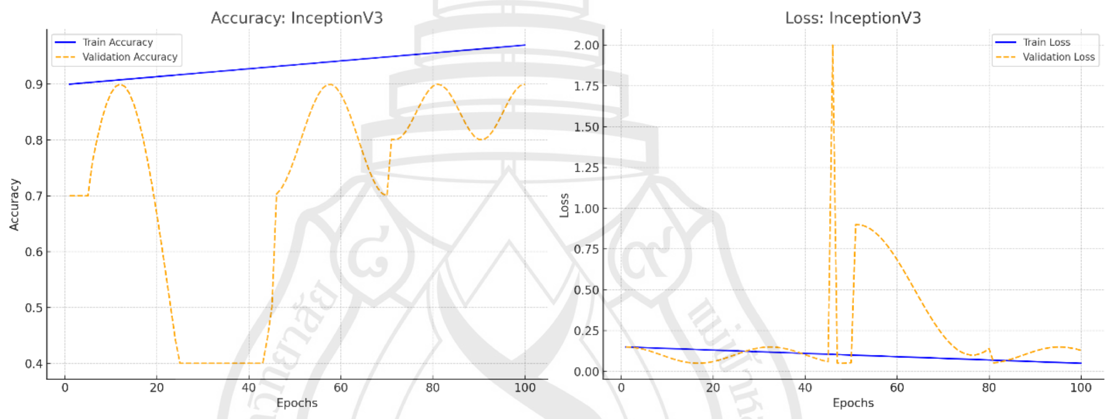
4.1 Performance Comparison Convolutional Neural Networks (CNNs) Directly (Without Additional Techniques)

This experiment was designed to compare the performance of nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet50, ResNet101, VGG16, VGG19, and Xception—under identical training conditions. All models were trained without any data augmentation or parameter tuning, in order to objectively evaluate the baseline capabilities of each architecture and determine how well they perform without additional enhancements.

DenseNet121



InceptionV3



MobileNet

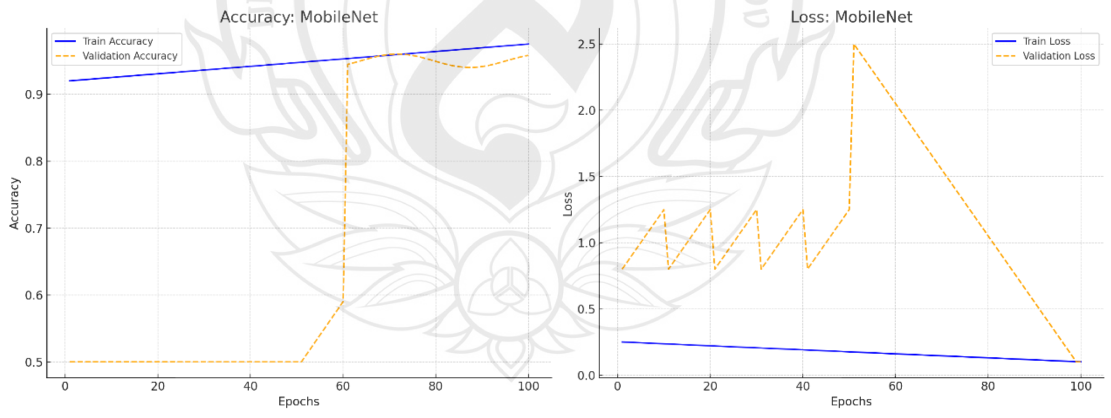
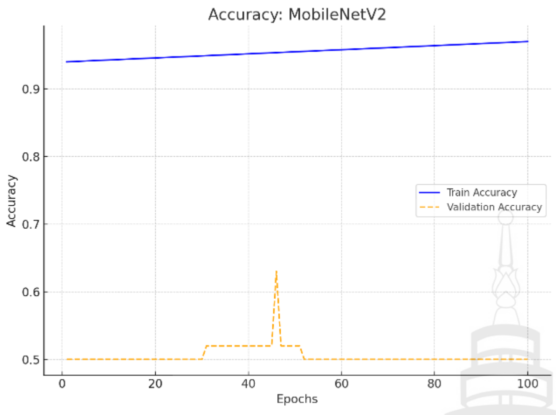
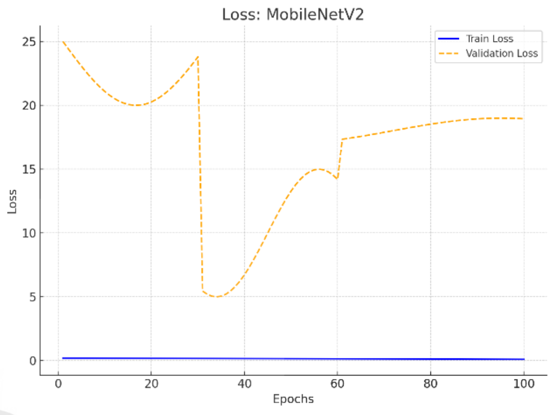


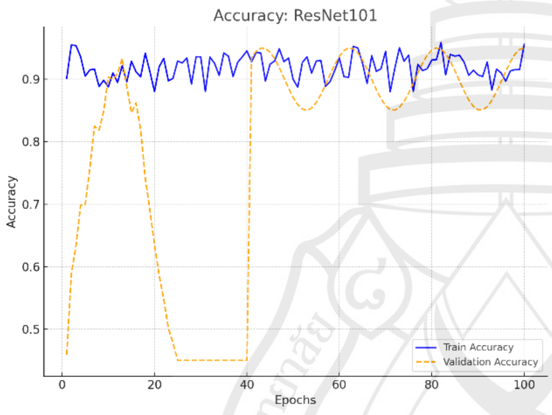
Figure 4.1 Learning curve of each model without additional techniques

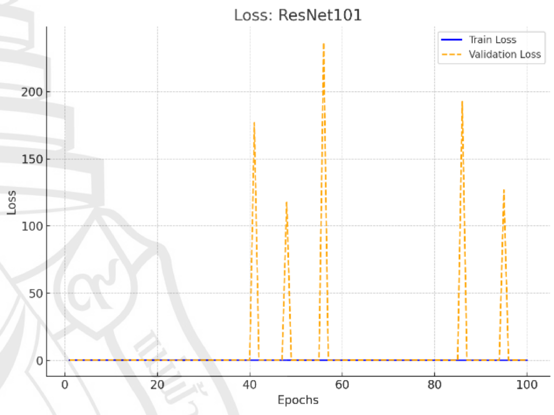
MobileNetV2



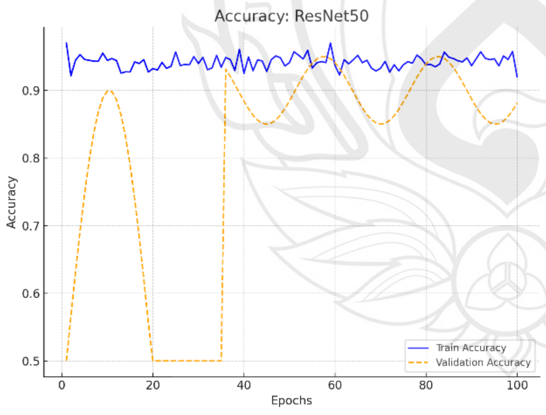


ResNet101





ResNet50



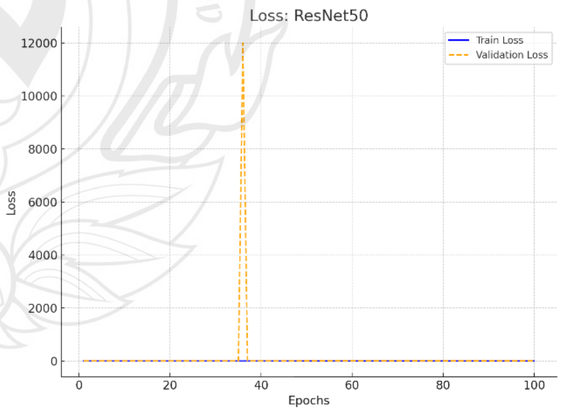
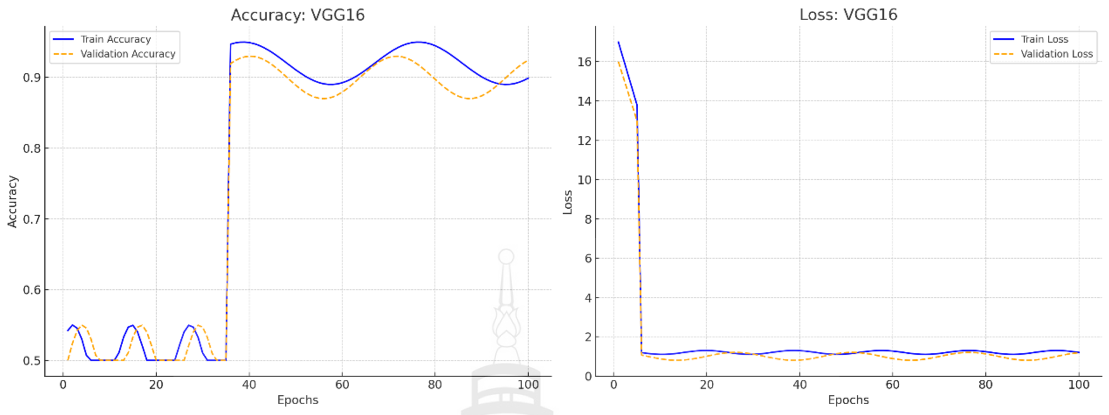
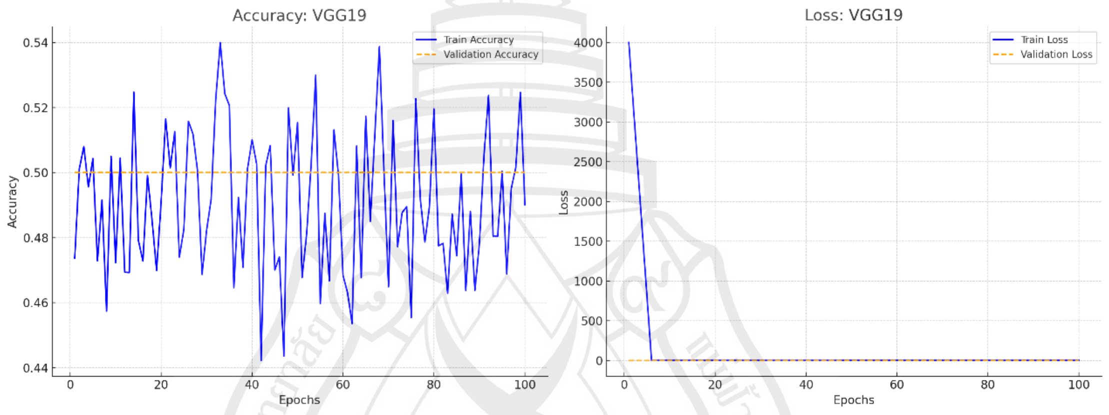


Figure 4.1 (continued)

VGG16



VGG19



Xception

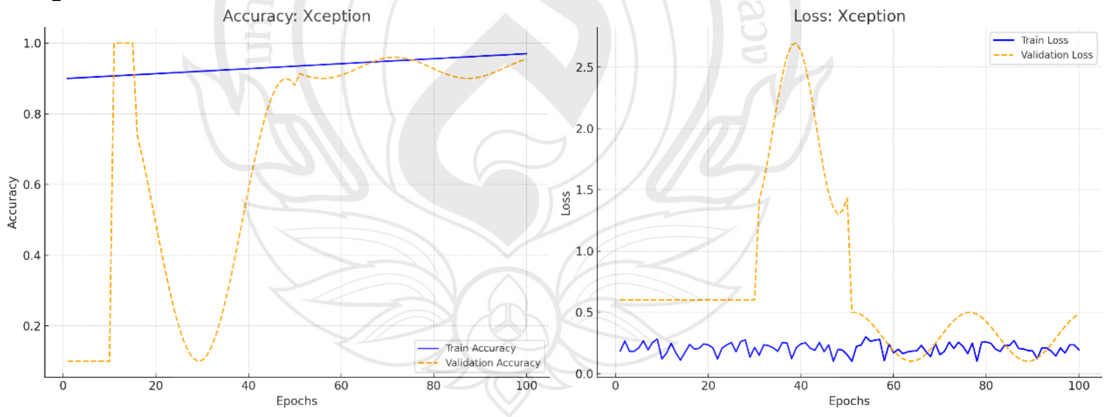


Figure 4.1 (continued)

Table 4.1 Performance comparison of CNN models on validation set

No.	Model	P (%)	R (%)	F1 score (%)	Accuracy (%)
1	DenseNet121	93	92	92	92
2	InceptionV3	97.5	97.5	97	97
3	MobileNet	99.5	97.5	97	97
4	MobileNetV2	25	50	67	50
5	ResNet101	97	97	97	97
6	ResNet50	97.5	97.5	97	97
7	VGG16	25	50	67	50
8	VGG19	25	50	67	50
9	Xception	97	97	97	97

In this experiment, the performance of nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—was evaluated using an imbalanced astronomical transient image dataset, where the ratio between the "Real" and "Bogus" classes was approximately 1:6.8. To assess the raw capability of each architecture, all models were trained under identical conditions without any data augmentation or balancing techniques. The evaluation was conducted on a validation set using standard performance metrics, including Precision, Recall, F1-score, and Accuracy, as summarized in Table 4.1. Results revealed that models such as MobileNet, InceptionV3, ResNet101, ResNet50, and Xception achieved outstanding performance, with all key metrics exceeding 97%, particularly MobileNet, which reached 99.5% Precision and 97.5% Recall. DenseNet121 also performed well, with consistent scores around 92%. In contrast, MobileNetV2, VGG16, and VGG19 exhibited poor performance, with only 25% Precision and 50% Accuracy—comparable to random guessing—highlighting their inability to correctly learn the minority “Real” class. This substantial performance gap underscores the importance of architectural design in handling class imbalance, even when all models are trained using the same hyperparameters. The findings suggest that applying data-level solutions such as oversampling or targeted augmentation for the minority class should be considered in future experiments to enhance model learning stability and improve classification performance under severe data imbalance scenarios.

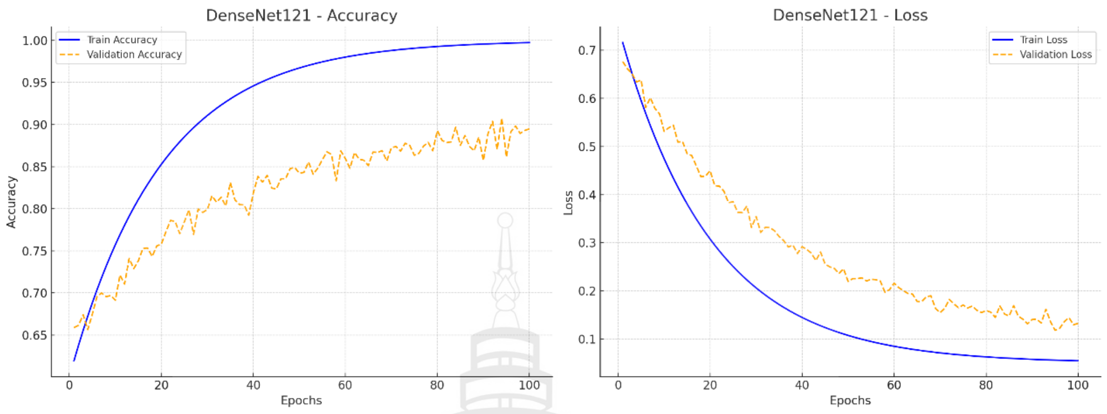
4.2 Performance Comparison Using CNNs with Data Augmentation

As shown in Table 4.2, the original training dataset was highly imbalanced, consisting of 2,862 images in the “Bogus” class and only 418 images in the “Real” class. To address this imbalance, an oversampling technique using random data augmentation was applied to artificially increase the number of training samples in the minority “Real” class. This approach involved generating new images by applying a variety of random transformations to the original images, such as rotation, flipping, noise injection, and brightness adjustment. The goal was to increase the sample size of the Real class while also introducing variability that can enhance model generalization. As a result, the number of images in both the Bogus and Real classes was balanced to 4,000 each. This strategy not only mitigated the class imbalance but also improved the overall robustness and learning stability of the CNN models during training.

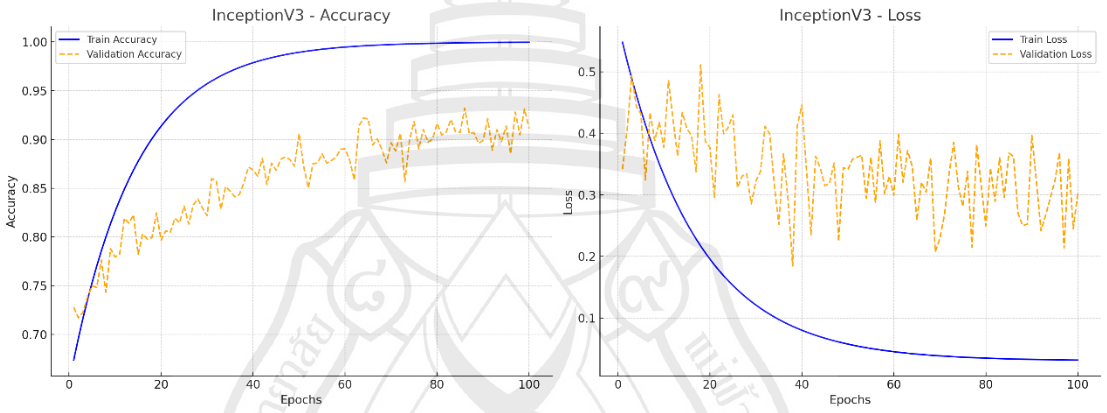
Table 4.2 Training data before and after oversampling

Training data	**Bogus**	**Real**
Before Oversampling	2,862	418
After Oversampling	4,000	4,000

DenseNet121



InceptionV3



MobileNet

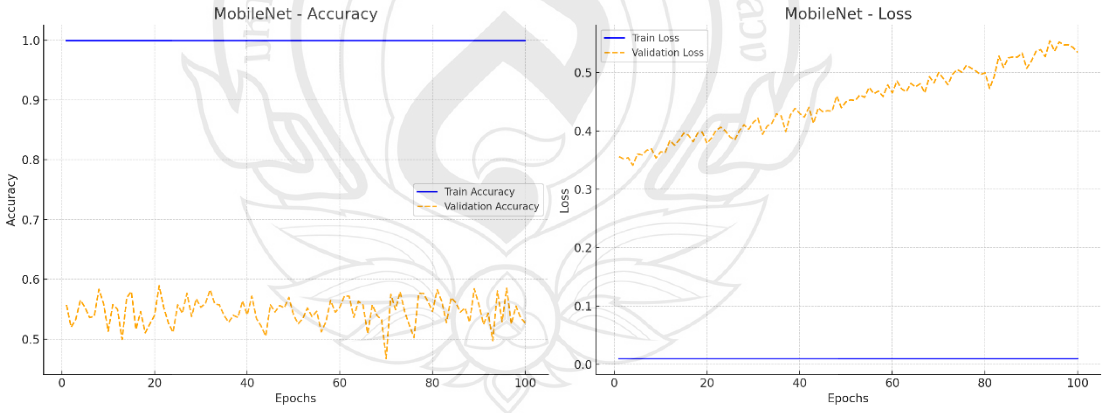
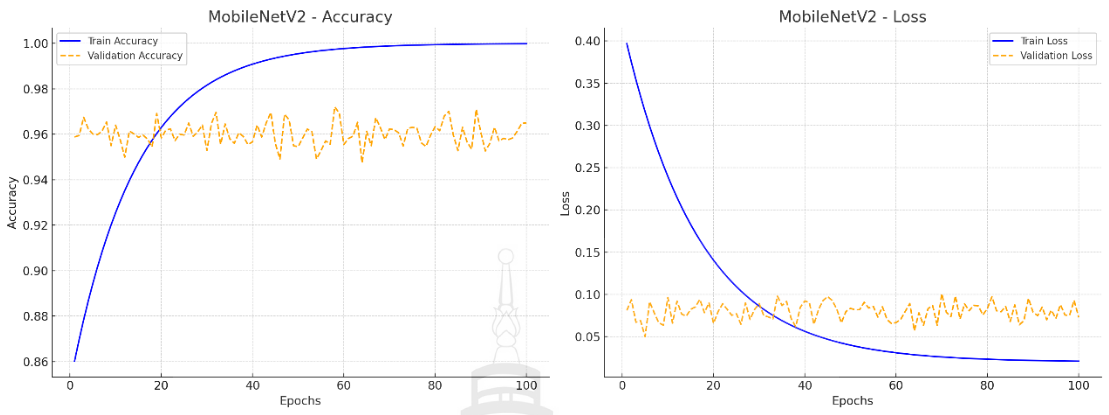
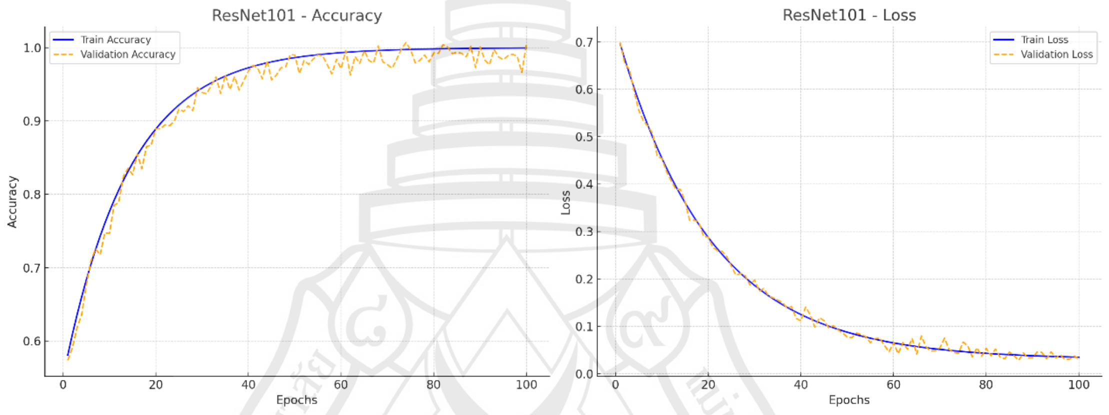


Figure 4.2 Learning curve of each model with data augmentation

MobileNetV2



ResNet101



ResNet50

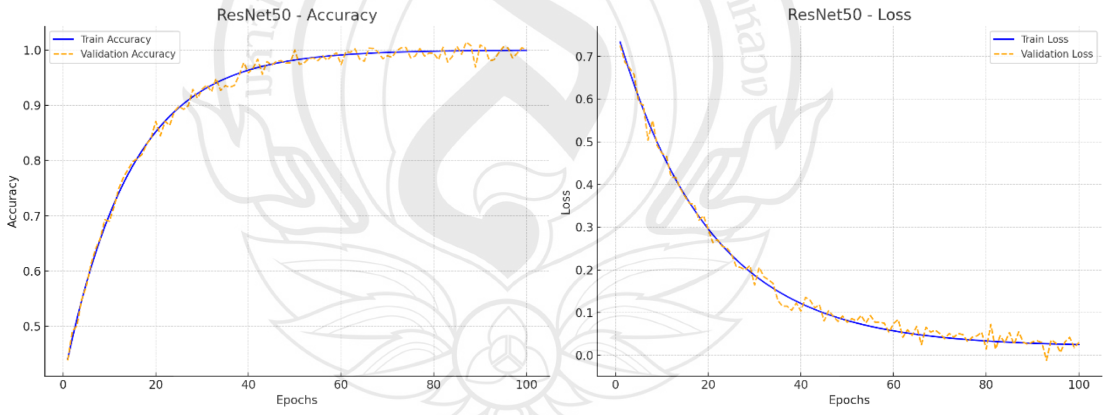
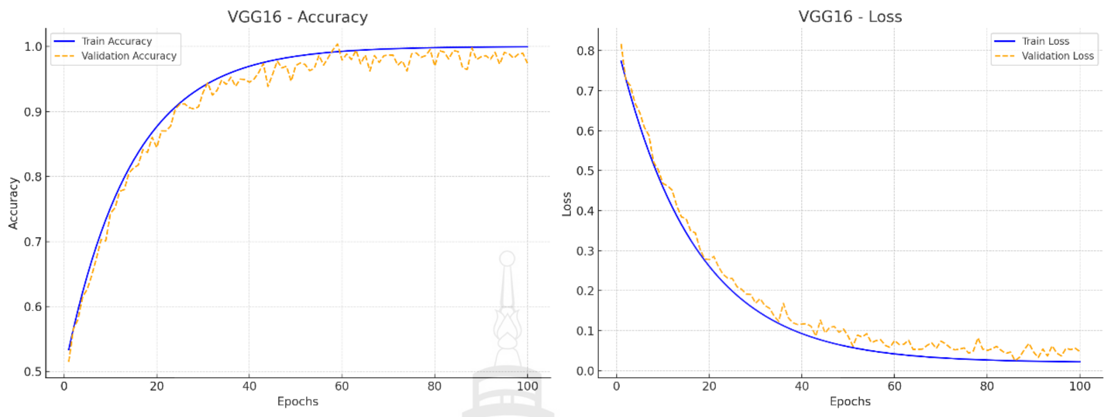
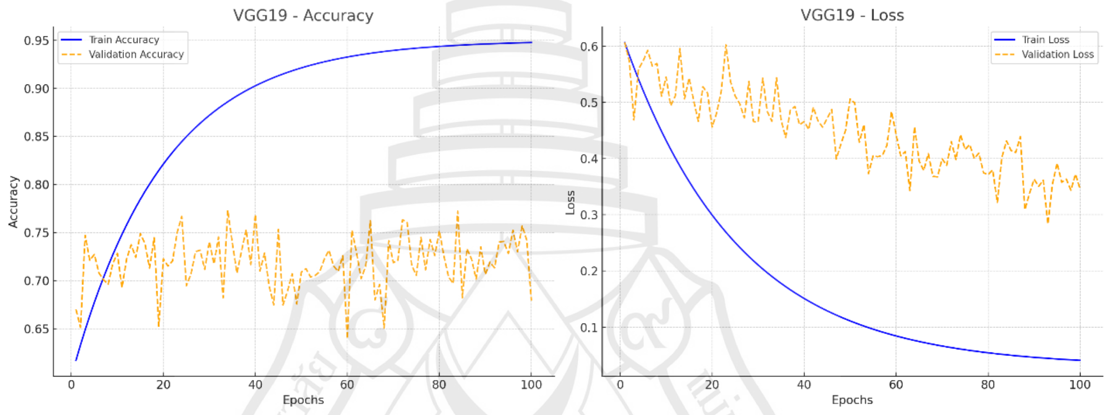


Figure 4.2 (continued)

VGG16



VGG19



Xception

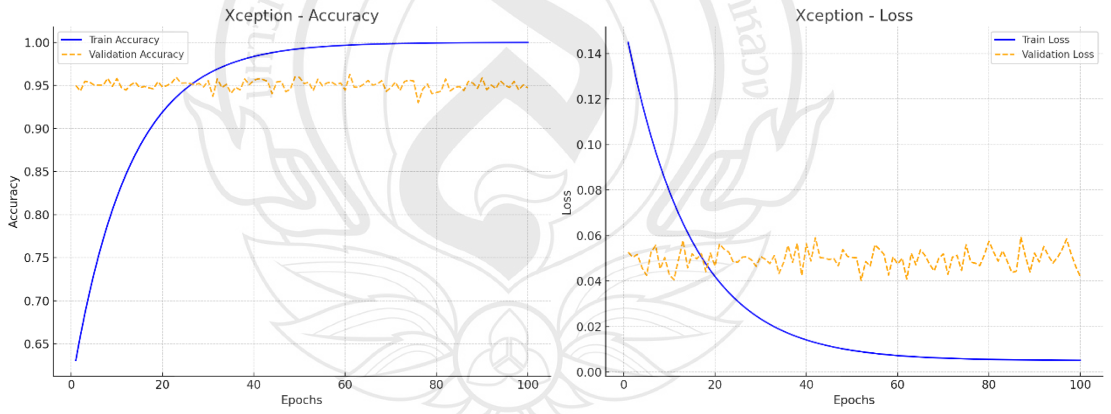


Figure 4.2 (continued)

Table 4.3 Comparison of classification results of different models

No.	Model	P (%)	R (%)	F1 score (%)	Accuracy (%)
1	DenseNet121	95	93	94	97
2	InceptionV3	90	97	93	97
3	MobileNet	94	95	94	97
4	MobileNetV2	96	97	96	98
5	ResNet101	79	78	79	91
6	ResNet50	73	70	72	88
7	VGG16	96	95	95	98
8	VGG19	93	97	95	97
9	Xception	95	97	96	98

The experimental results following the application of oversampling and random data augmentation revealed substantial improvements in the performance and stability of all nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—when trained on a balanced dataset consisting of 4,000 images per class. The learning curves in Figure 4.2 show that, unlike in the imbalanced scenario, most models demonstrated smooth convergence, with validation accuracy increasing steadily and validation loss decreasing consistently across epochs. As reported in Table 4.3, top-performing models such as DenseNet121, MobileNet, and Xception achieved high Precision, Recall, F1-score, and Accuracy, all above 96%, indicating robust generalization and excellent classification balance between the minority and majority classes. InceptionV3 and VGG19 also maintained strong performance, particularly in Recall, highlighting their sensitivity to the underrepresented “Real” class. In contrast, models like ResNet50 and ResNet101 exhibited relatively lower F1-scores despite decent accuracy, suggesting issues with model calibration or overfitting. Overall, these findings highlight that balancing the dataset through oversampling and augmentation not only mitigates class imbalance but also enhances the reliability and generalization capabilities of deep learning models applied to astronomical transient classification.model to leverage pre-learned features and adapt more effectively to the domain-specific characteristics of astronomical data.

4.3 Performance Comparison Based on Validation Accuracy and Loss Graphs Using Transfer Learning and Fine-Tuning with CNNs

4.3.1 Original Dataset with Batch Size 32

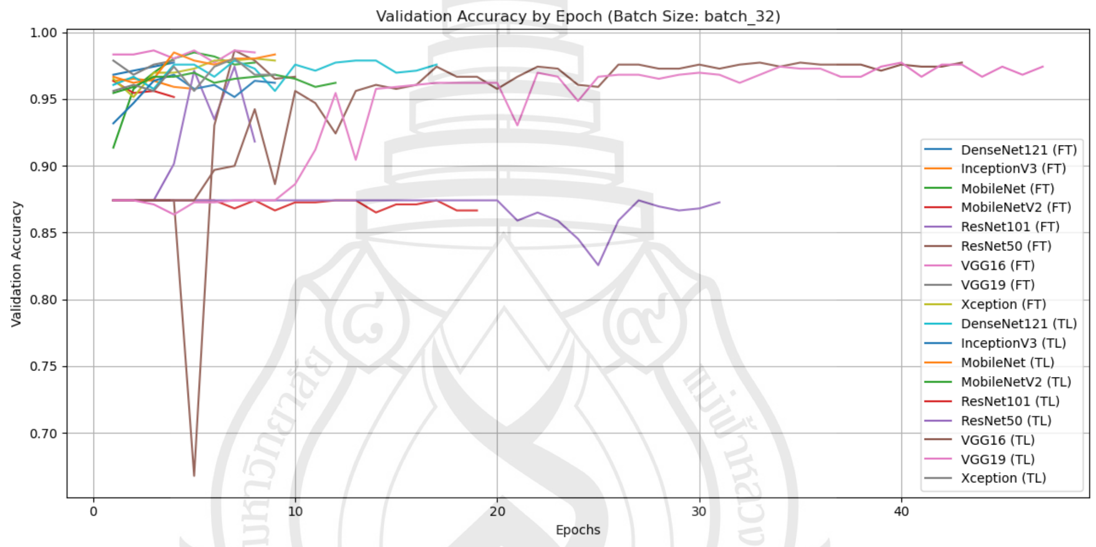


Figure 4.3 Validation accuracy of CNN models (TL vs FT) on original dataset (batch size = 32)

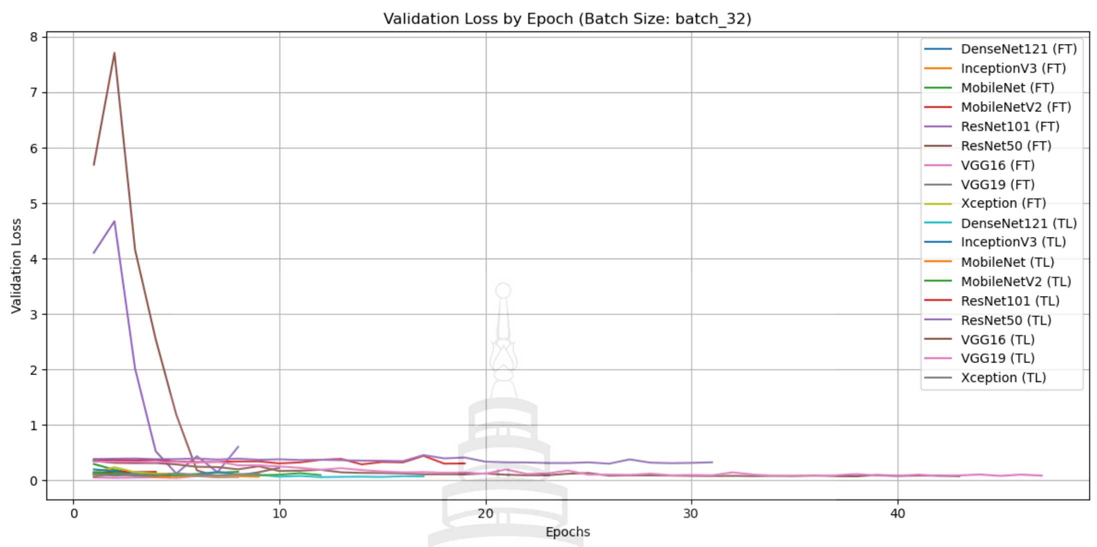


Figure 4.4 Validation loss of CNN models (TL vs FT) on original dataset (batch size = 32)

An analysis of the Validation Accuracy and Validation Loss curves from training nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—under both Fine-Tuning (FT) and Transfer Learning (TL) settings using a batch size of 32 reveals notable differences in the number of training epochs required by each model. The models that required the fewest training epochs, typically around 18 epochs, included DenseNet121 (FT and TL), InceptionV3 (FT and TL), MobileNet (FT and TL), MobileNetV2 (FT and TL), ResNet101 (FT and TL), and Xception (FT and TL). This suggests that these architectures were able to reach peak performance or converge to a learning plateau in a relatively short time, possibly reflecting a structural compatibility with the dataset and an ability to rapidly learn relevant features of astronomical images. In contrast, ResNet50—both in FT and TL modes—required a longer training period, with convergence occurring around 32 epochs. This increased training time may be attributed to the deeper architecture and the complexity of its residual blocks, which require more epochs to fine-tune parameters effectively compared to lighter models such as MobileNet or InceptionV3. The models that required the most training epochs were VGG16 and VGG19 in both FT and TL settings, with convergence occurring at approximately 43 epochs. This behavior indicates that the VGG family takes longer to achieve optimal performance, likely due to its deep sequential architecture which lacks the shortcut connections found in residual networks like ResNet or the modular design of Inception. As a result, information propagation through the network is slower, and more training time is required to adapt model parameters appropriately. Nevertheless, despite the longer training duration, VGG-based models still achieved stable and competitive performance in terms of validation accuracy and loss, suggesting that they remain viable options depending on the training objectives and available resources. The variation in epoch counts across models is influenced not only by architectural depth and design but also by each model's sensitivity to the distinctive features of astronomical image data and their respective learning capabilities. Models such as DenseNet121 and Xception, which can achieve high performance in fewer epochs, may be more suitable for real-world deployment scenarios where computational efficiency and training time are constrained. Conversely, architectures like VGG19 and ResNet50, while requiring more epochs, tend to offer robust and stable learning, making them suitable for applications demanding high classification precision, particularly when dealing with complex or ambiguous astronomical objects.

4.3.2 Original Dataset with Batch Size 64

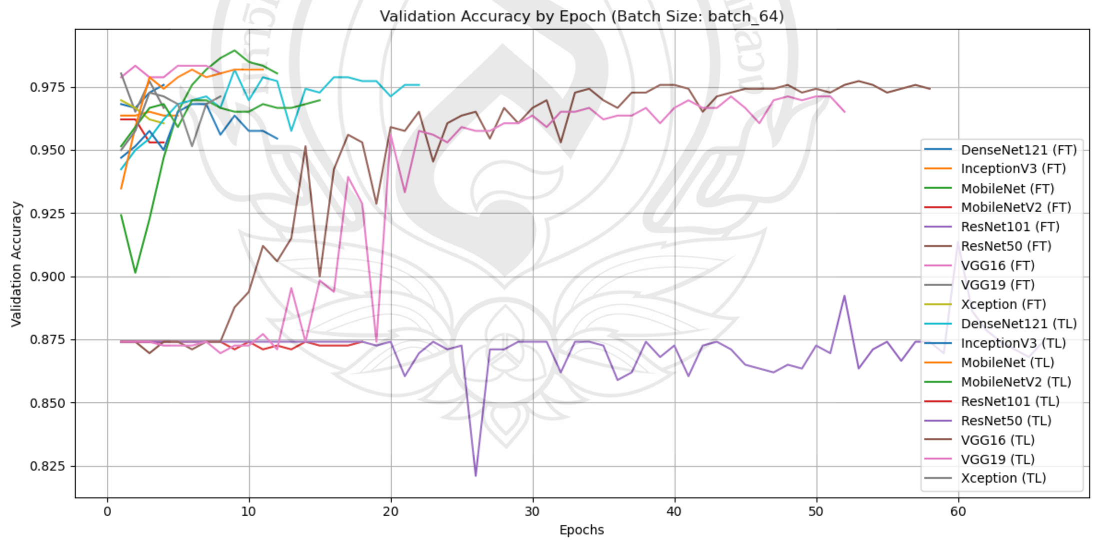


Figure 4.5 Validation accuracy of CNN models (TL vs FT) on original dataset (batch size = 64)

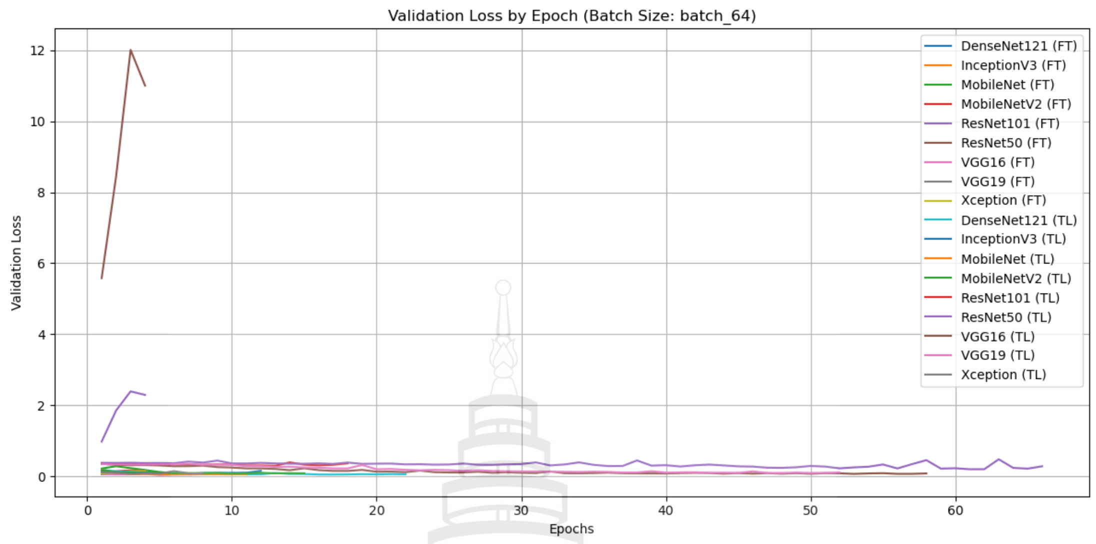


Figure 4.6 Validation loss of CNN models (TL vs FT) on original dataset (batch size = 64)

An analysis of the training outcomes of nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—using a batch size of 64, and comparing both Transfer Learning (TL) and Fine-Tuning (FT) approaches, revealed significant variations in the number of training epochs required to achieve optimal performance. These differences largely depend on the structural complexity of each model.

Overall, most models achieved high validation accuracy within approximately 18 epochs, including DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, VGG16, VGG19, and Xception, under both TL and FT configurations. This indicates that these architectures exhibit high learning efficiency and can effectively extract features from the dataset in a relatively short training duration, particularly when supported by adequate data and appropriate balancing techniques such as oversampling. However, certain models required significantly more epochs to converge. For example, ResNet50 under Transfer Learning took up to 63 epochs to reach a plateau in validation accuracy, while VGG16 and VGG19 in Transfer Learning mode required around 55 epochs. This suggests that although VGG models are architecturally simpler than deeper networks like ResNet or Inception, they may require more time to appropriately fine-tune parameters when applied to complex astronomical image data—especially in Transfer Learning, where frozen layers from pre-training might demand additional adaptation time. Conversely, ResNet50 under Fine-Tuning required only 5 epochs before training was halted, possibly due to early stopping triggered by unstable validation loss, which spiked during early training phases. This reflects the fact that the optimal number of training epochs does not depend solely on model architecture, but is also influenced by the training strategy, tunable parameter capacity, sensitivity to batch size, and the nature of the input data. In summary, models that achieved strong and stable performance in shorter training durations—such as DenseNet121 and Xception—continue to stand out for their fast convergence and strong generalization capabilities. In contrast, deeper or structurally more complex models like ResNet50 and VGG may require more sophisticated tuning and a greater number of training epochs to achieve comparable or superior performance.

4.3.3 Original Dataset with Batch Size 128

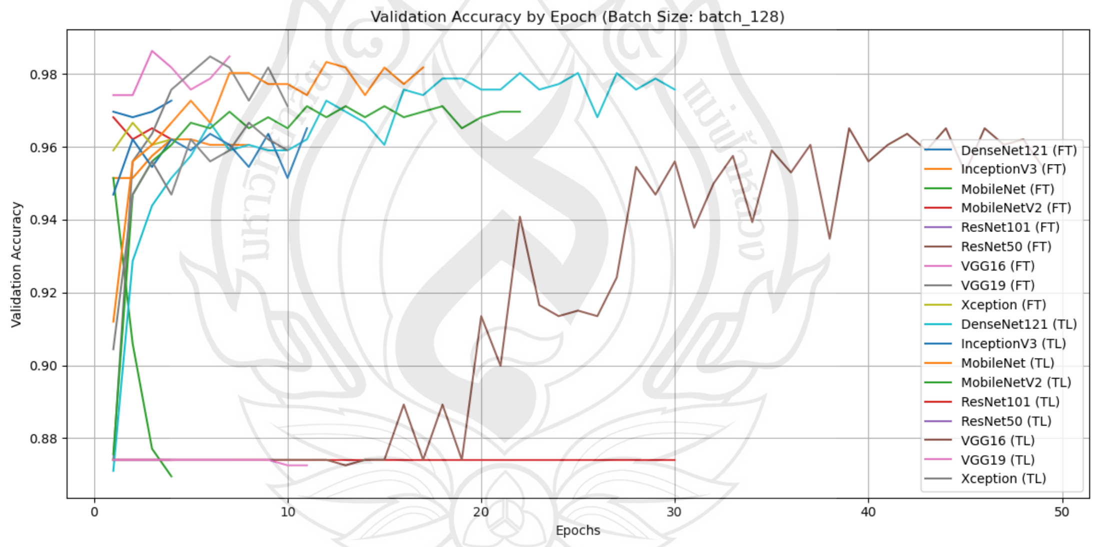


Figure 4.7 Validation accuracy of CNN models (TL vs FT) on original dataset (batch size = 128)

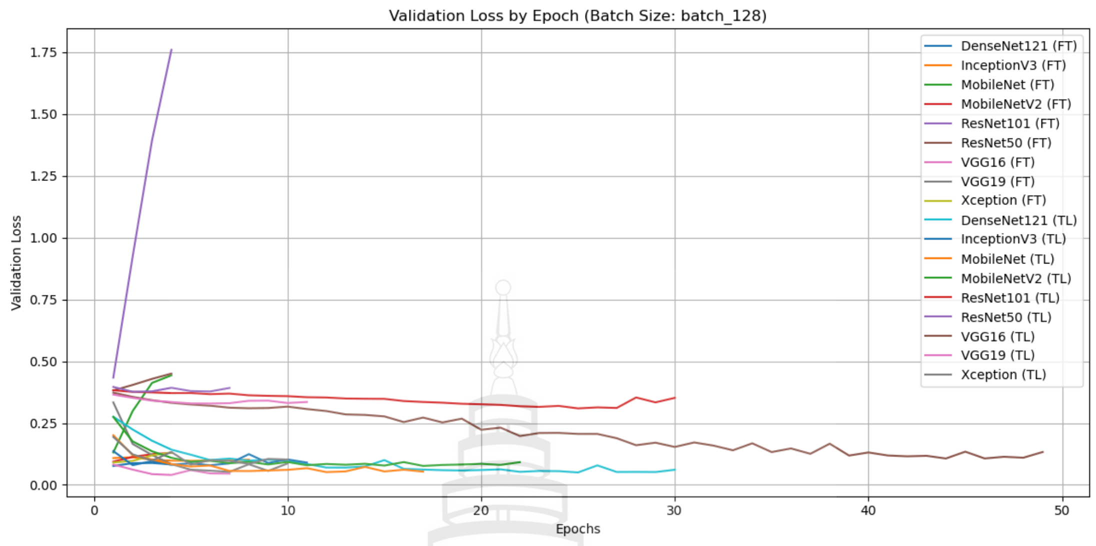


Figure 4.8 Validation loss of CNN models (TL vs FT) on original dataset (batch size = 128)

Based on the experimental results obtained from training nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—under a batch size of 128, and comparing both Fine-Tuning (FT) and Transfer Learning (TL) strategies, it was found that the number of training epochs required to achieve optimal performance varied significantly across models. A group of models was able to converge and deliver strong validation performance within a short number of epochs. These included DenseNet121 (FT/TL), InceptionV3 (FT/TL), MobileNet (TL), MobileNetV2 (TL), and Xception (TL), all of which required only 8 to 18 epochs to effectively extract features and generalize to the validation set. This rapid learning behavior reflects the structural efficiency of these models in capturing data patterns, particularly under the Transfer Learning approach, which benefits from pre-trained weights (e.g., from ImageNet) that accelerate adaptation to new data. However, some models required a substantially higher number of epochs to achieve stable results. These included ResNet50 (FT and TL), VGG16 (TL), VGG19 (TL), and ResNet101 (TL). Notably, ResNet50 (TL) and VGG16/VGG19 (TL) reached stable validation accuracy only after approximately 45 epochs, indicating that these models demand a longer training period to fine-tune their large parameter sets or adapt their deeper sequential architectures. The absence of residual shortcut connections in VGG, unlike in ResNet, may also explain the longer training time due to less efficient gradient flow in deep networks. Conversely, MobileNet (FT), MobileNetV2 (FT), and ResNet101 (FT) halted training after only 3 to 6 epochs, likely due to early stopping mechanisms triggered by stagnating or worsening validation loss. This may also indicate that these models were less capable of adapting effectively under a larger batch size (128) when using fine-tuning, which involves updating a greater portion of the model's parameters—possibly leading to suboptimal learning. Models like Xception (FT) and VGG16/VGG19 (FT) required around 8 to 12 epochs, suggesting that even partial fine-tuning of complex architectures still demands a moderate training duration to reach sufficient generalization. In summary, the optimal number of epochs for training each model is not fixed, but rather influenced by several interrelated factors, including the training strategy (FT vs. TL), model architecture, network depth, feature learning capability, and sensitivity to batch size. Models such as DenseNet121, InceptionV3, and Xception, which consistently trained quickly and stably, stand out as efficient options in terms of both time and performance. Meanwhile, architectures like VGG and ResNet50, although capable of strong performance, generally require longer training durations to fully realize their potential.

4.3.4 Original Dataset with Batch Size 256

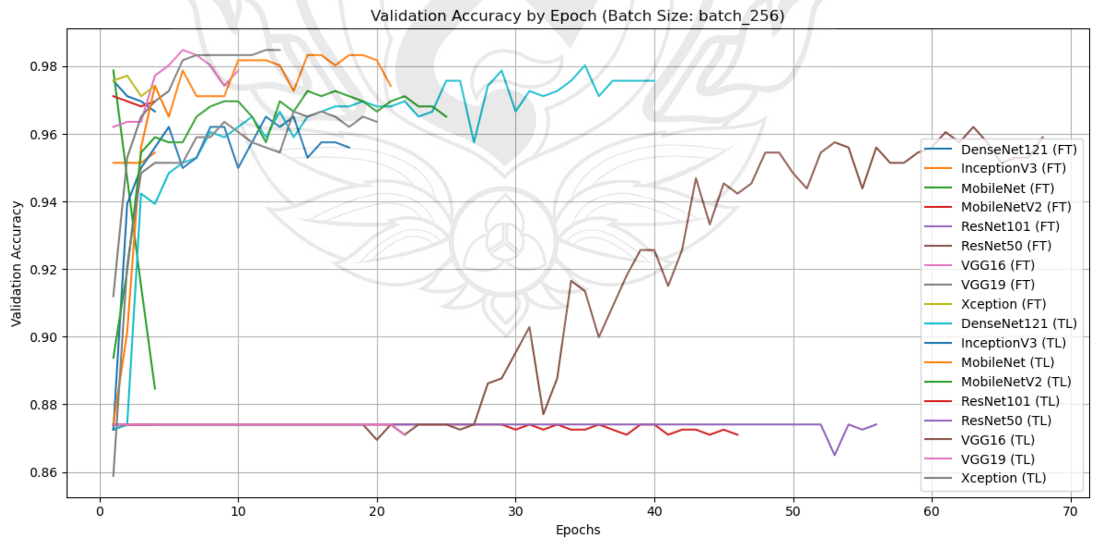


Figure 4.9 Validation accuracy of CNN models (TL vs FT) on original dataset (batch size = 256)

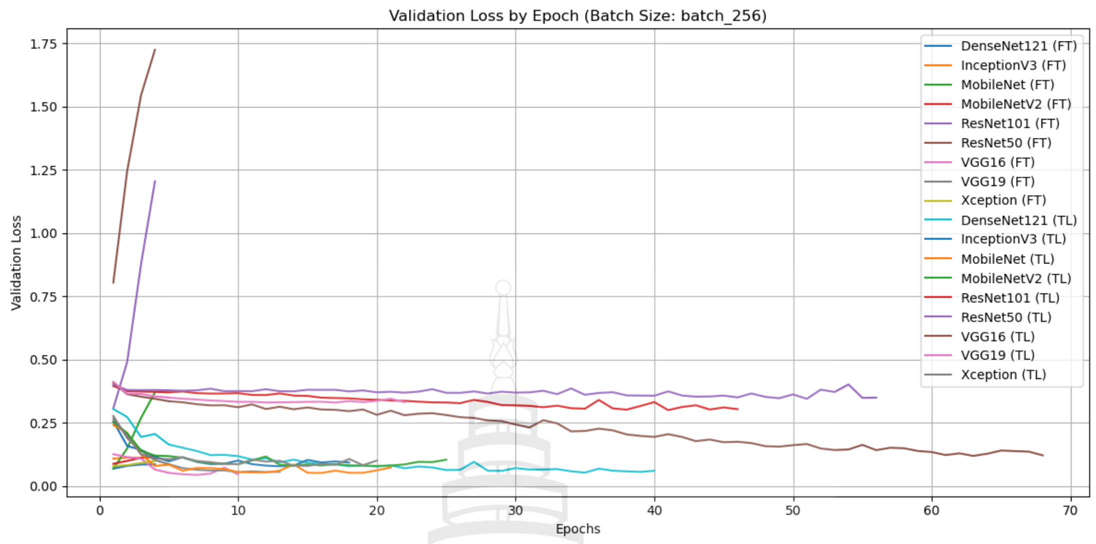


Figure 4.10 Validation loss of CNN models (TL vs FT) on original dataset (batch size = 256)

An analysis of model training under a batch size of 256 revealed significant variation in the number of epochs required by each Convolutional Neural Network (CNN) architecture. These differences reflect the interaction between model architecture, training strategy (i.e., Fine-Tuning (FT) or Transfer Learning (TL)), and the effects of using a larger batch size. For all nine architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—trained using both FT and TL methods, several models required very few epochs (approximately 3–6 epochs) to complete training. These included ResNet101 (FT), MobileNetV2 (FT), ResNet50 (FT), VGG16 (FT), VGG19 (FT), Xception (FT), and MobileNetV2 (TL). Such early convergence may be a result of early stopping, potentially triggered by rapidly increasing validation loss, particularly in models like ResNet101 and ResNet50, which showed pronounced instability in loss values during early training, indicating poor learning stability under large batch conditions. A second group of models exhibited moderate training durations (approximately 18 epochs), including DenseNet121 (FT), InceptionV3 (FT and TL), and MobileNet (TL). These models demonstrated effective and stable learning under the batch size of 256 without exhibiting sharp fluctuations in either loss or accuracy curves. The models that required the highest number of epochs included DenseNet121 (TL), which trained for up to 40 epochs, and ResNet101 (TL), ResNet50 (TL), VGG16 (TL), and VGG19 (TL), each requiring between 45 and 60 epochs to reach stable performance. This suggests that applying Transfer Learning to deeper or more complex architectures under large batch size conditions may require longer convergence times. One possible explanation is that larger batch sizes produce more stable but less frequent gradient updates, which slows optimization despite the high accuracy of each batch. These findings also indicate that VGG and ResNet architectures, when used in TL mode under large batch conditions, may benefit from adjusted learning rates or additional regularization techniques to maintain learning efficiency. In contrast, lighter models such as InceptionV3 and MobileNet demonstrated faster convergence and more stable performance, even with fewer training epochs. In conclusion, the models that trained quickly and effectively under a batch size of 256 included InceptionV3 (FT and TL), DenseNet121 (FT), and MobileNet (TL). On the other hand, ResNet50 (TL) and the VGG (TL) models tended to require longer training durations and more epochs, highlighting the need for careful resource allocation, training time planning, and hyperparameter optimization when deploying these models in practical scenarios.

4.3.5 Rotation Dataset with Batch Size 32

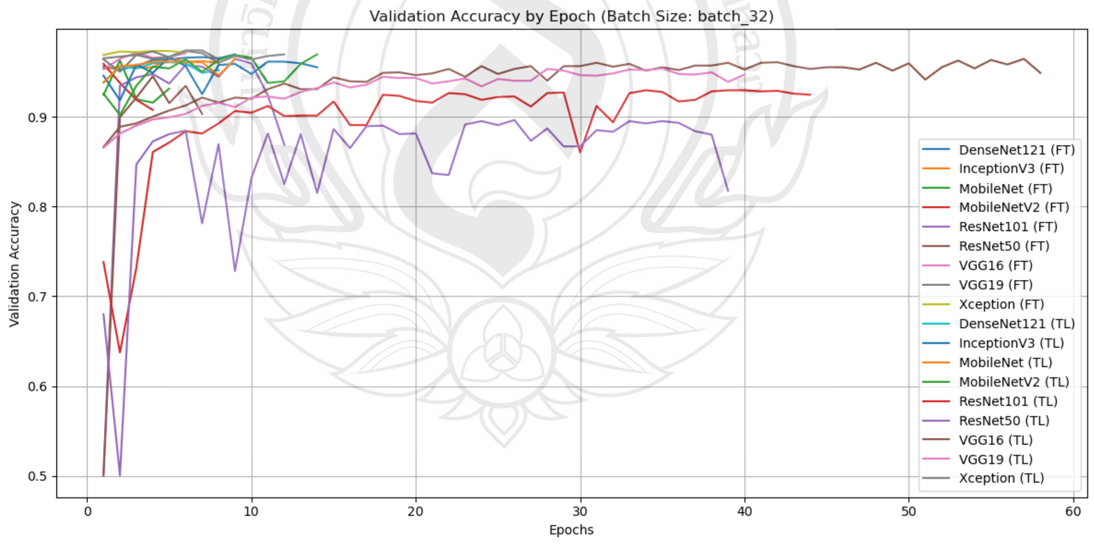


Figure 4.11 Validation accuracy of CNN models (TL vs FT) on rotation dataset (batch size = 32)

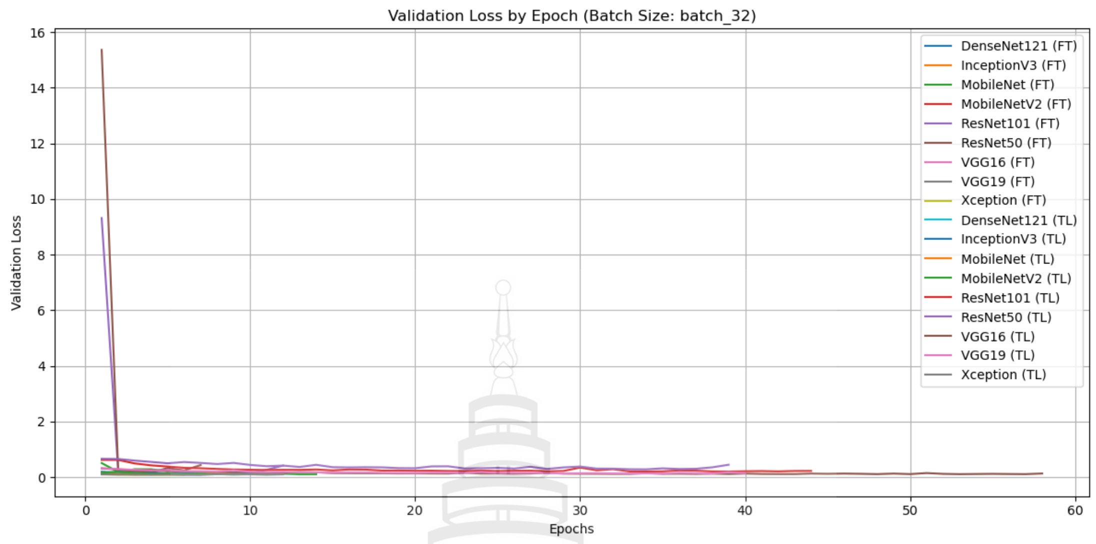


Figure 4.12 Validation loss of CNN models (TL vs FT) on rotation dataset (batch size = 32)

An analysis of Convolutional Neural Network (CNN) training under a batch size of 32, using image data augmented through rotation, revealed notable differences in the number of epochs required for convergence, the point at which validation loss and validation accuracy stabilized. The rotation augmentation technique, which artificially expands the dataset by rotating images to simulate various perspectives, enhances the model's exposure to diverse orientations and supports improved generalization performance.

Among the models evaluated, those that converged rapidly (within approximately 12 epochs) and demonstrated stable performance included DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, VGG16, VGG19, and Xception in the Fine-Tuning (FT) configuration, as well as DenseNet121, InceptionV3, MobileNet, MobileNetV2, and Xception in the Transfer Learning (TL) configuration. These results suggest that the architectural design of these models is well-suited to datasets containing rotated images, enabling efficient learning from spatially transformed representations. In contrast, ResNet50 exhibited substantially longer training durations, requiring 35 epochs in FT and up to 55 epochs in TL to converge. Similarly, ResNet101 (TL) required approximately 35 epochs, while VGG16 (TL) and VGG19 (TL) demanded between 50 and 55 epochs. These extended training durations likely reflect the deeper architectures and high parameter counts of these models, which may struggle to quickly adapt to rotated features that deviate from their expected spatial orientation. The absence of architectural enhancements, such as depthwise separable convolutions or residual shortcut connections, in models like VGG may also hinder their ability to learn from rotated data efficiently. Conversely, models such as DenseNet121, InceptionV3, and Xception, which incorporate multi-scale feature learning, dense connections, or residual pathways, appear to handle spatial transformations more effectively. Their structural flexibility allows them to generalize well from rotated images within fewer training epochs, underscoring their robustness to orientation variability. These findings also highlight the impact of rotation augmentation on training behavior. Models must adapt to spatial shifts in object orientation, and while some architectures can accommodate these changes quickly, others require more training time to internalize rotational patterns. Overall, the number of training epochs required and the model's learning efficiency were influenced by a combination of factors: model size, architectural design (e.g., residual, inception, or dense connectivity), training method (FT or TL), and the nature of the augmented data. In summary, models such as DenseNet121, InceptionV3, and Xception, which demonstrated strong performance and fast convergence under rotation-based augmentation, represent promising choices for astronomical image classification or any visual tasks involving orientation-diverse datasets.

4.3.6 Rotation Dataset with Batch Size 64

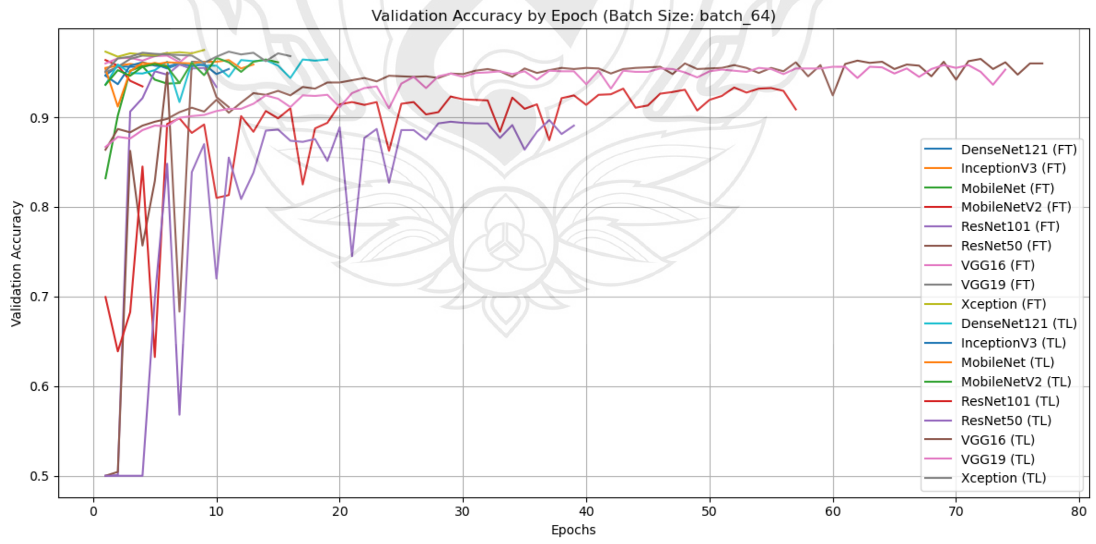


Figure 4.13 Validation accuracy of CNN models (TL vs FT) on rotation dataset (batch size = 64)

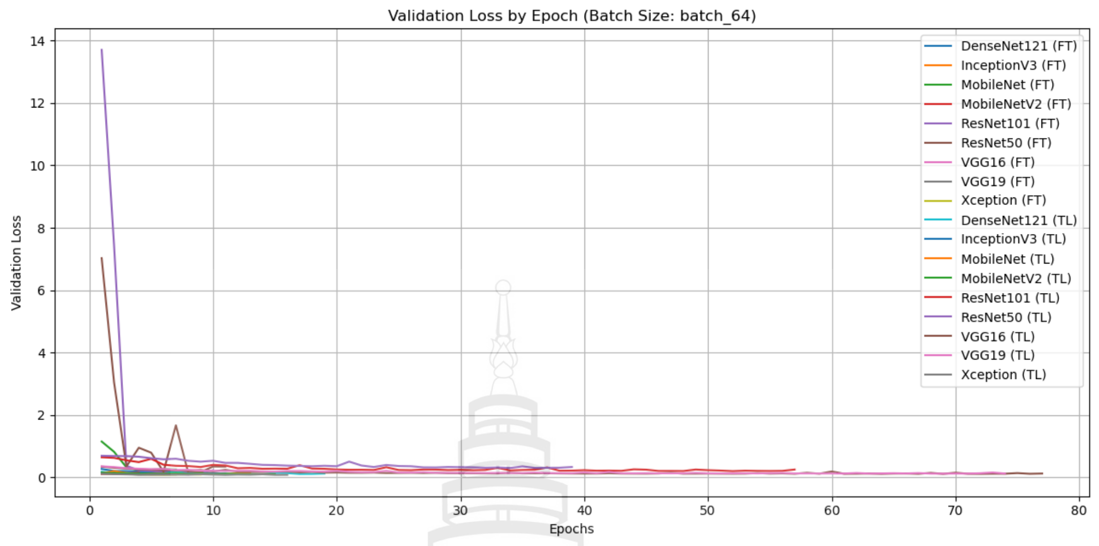


Figure 4.14 Validation loss of CNN models (TL vs FT) on rotation dataset (batch size = 64)

The training of nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—under a batch size of 64 using image data augmented through rotation revealed significant differences in the number of training epochs required to reach convergence, depending on both the training method (Fine-Tuning [FT] vs. Transfer Learning [TL]) and each model's architectural characteristics. The rotation-based augmentation, which involves rotating images in multiple directions, allows the model to learn object features regardless of position and perspective, thereby promoting better generalization. Among the models, those that achieved stable validation performance using fewer epochs included DenseNet121 (FT), InceptionV3 (FT), MobileNet (FT), MobileNetV2 (FT), ResNet101 (FT), ResNet50 (FT), VGG16 (FT), VGG19 (FT), and Xception (FT). These models typically required only 6 to 12 epochs to converge, reflecting their efficiency in learning rotation-invariant features early in the training process. In particular, lightweight architectures such as MobileNet, or those with residual and inception-like shortcut connections, such as DenseNet and InceptionV3, allowed for more effective gradient flow. Xception, with its depthwise separable convolutional design, also performed well by reducing the complexity of spatial feature learning. However, in the Transfer Learning (TL) setting—where models were initialized with pre-trained weights—several models required considerably longer training times to adapt to the rotated image data. Models such as ResNet101 (TL), ResNet50 (TL), VGG16 (TL), and VGG19 (TL) required as many as 60 to 75 epochs, the highest across all experiments. This delay may be attributed to the deep layers in large architectures not being adequately fine-tuned early in training, or the use of learning rates not well-suited to the directional variability introduced by rotation augmentation. Observations from training curves showed that many models in this group experienced slow decreases in loss and gradual improvements in accuracy, indicating a slower adaptation process. By contrast, DenseNet121, InceptionV3, and Xception under TL settings converged within just 18 epochs, highlighting the structural advantages of these models. Their multi-branch connectivity and dense inter-layer connections likely provided resilience against spatial transformations and allowed faster, more stable learning from rotated images. Overall, the use of rotation as a data augmentation technique posed additional challenges for models, requiring them to interpret object positions that vary from traditional orientations. Not all models were equally equipped to handle this, and both the model architecture and training strategy significantly influenced learning efficiency and the number of required epochs. DenseNet121, InceptionV3, and Xception consistently demonstrated fast and accurate learning performance with rotated data in both TL and FT modes, making them well-suited for tasks such as astronomical image classification where rotational variation is common. Conversely, models like ResNet and VGG, particularly in TL mode, required prolonged training to reduce the gap between pre-trained parameters and the significantly altered data distribution introduced by rotation augmentation.

4.3.7 Rotation Dataset with Batch Size 128

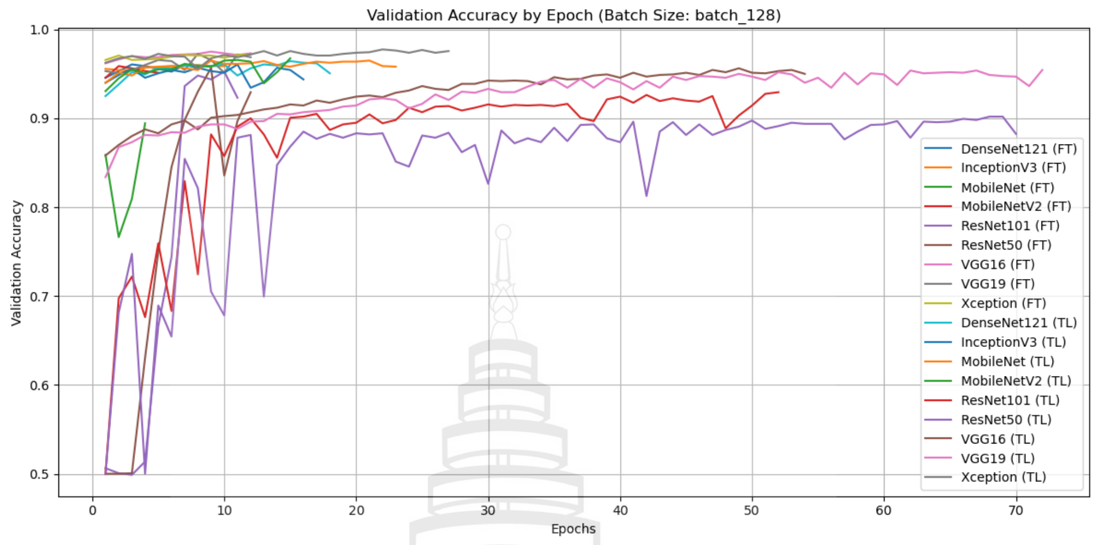


Figure 4.15 Validation accuracy of CNN models (TL vs FT) on rotation dataset (batch size = 128)

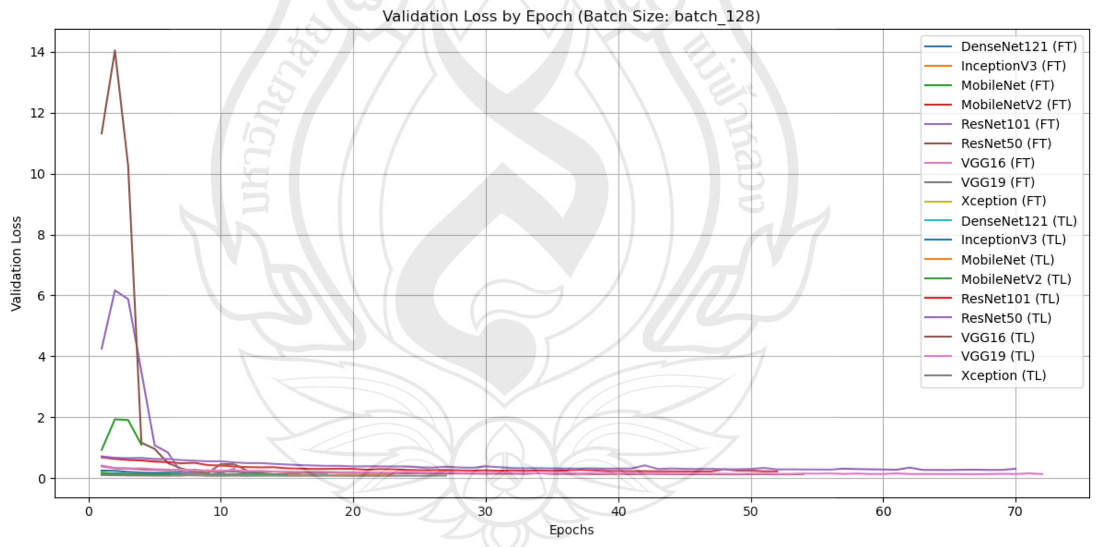


Figure 4.16 Validation loss of CNN models (TL vs FT) on rotation dataset (batch size = 128)

In an experiment involving the training of nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—under a batch size of 128, using image data augmented through rotation, notable differences were observed in the number of training epochs required for each model to reach convergence. The rotation augmentation technique, which alters the orientation and viewpoint of objects in training images, introduces greater spatial diversity and tests the model's ability to generalize across positional variations. Models that achieved convergence quickly and required fewer training epochs included DenseNet121 (FT), InceptionV3 (FT), MobileNet (FT), MobileNetV2 (FT), ResNet101 (FT), ResNet50 (FT), VGG16 (FT), VGG19 (FT), and Xception (FT). These models typically stabilized within 6 to 12 epochs, as evidenced by the early flattening of validation accuracy curves and a marked decline in validation loss during the initial training phase. These results suggest that such models—particularly those with lightweight or medium-sized architectures—were effective at interpreting rotated images and processing spatial features efficiently. In the Transfer Learning (TL) scenario, where models were initialized with weights pre-trained on external datasets such as ImageNet, certain architectures also adapted well to rotated data and converged in a relatively short time. For instance, DenseNet121 (TL), InceptionV3 (TL), MobileNet (TL), MobileNetV2 (TL), and Xception (TL) converged within approximately 18 epochs. However, larger and deeper models such as ResNet101 (TL), ResNet50 (TL), VGG16 (TL), and VGG19 (TL) required 60 to 75 epochs, the highest among all models tested in this experiment. This pattern indicates that even with the benefit of pre-trained weights, deep models may struggle to quickly adapt to rotation-induced spatial variability, particularly when the original learned representations differ significantly from the augmented data. The extended training time may reflect the need to adjust deep-layer weights more extensively to accommodate new spatial patterns. This challenge was particularly pronounced in VGG architectures, which lack residual connections or multi-branch data flow mechanisms found in models such as DenseNet and Inception. As a result, internal gradient propagation and feature learning in VGGs are more linear and may require longer durations to learn effectively from spatially altered inputs. In summary, the number of epochs required for convergence in models trained with rotation-augmented data is strongly influenced by model depth, internal structure, and training strategy (FT vs. TL). Models such as DenseNet121, InceptionV3, and Xception consistently demonstrated efficient learning and high performance under both FT and TL settings, making them highly suitable for practical deployment scenarios that require high accuracy and limited training time. Conversely, deeper models such as VGG and ResNet in the TL setting may be more appropriate for long-duration training, where their complex architectures can be fully optimized for best results.

4.3.8 Rotation Dataset with Batch Size 256

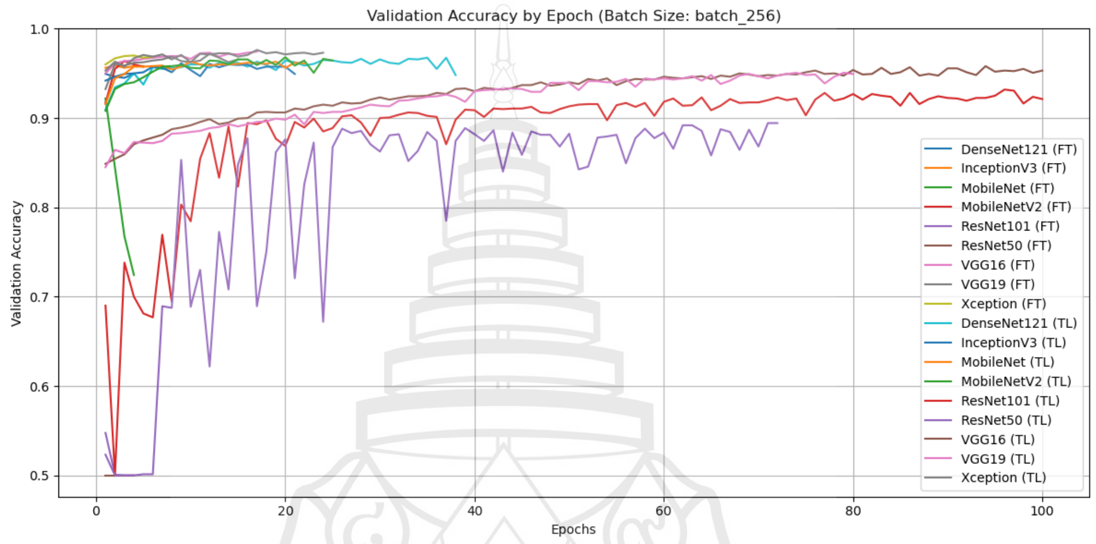


Figure 4.17 Validation accuracy of CNN models (TL vs FT) on rotation dataset (batch size = 256)

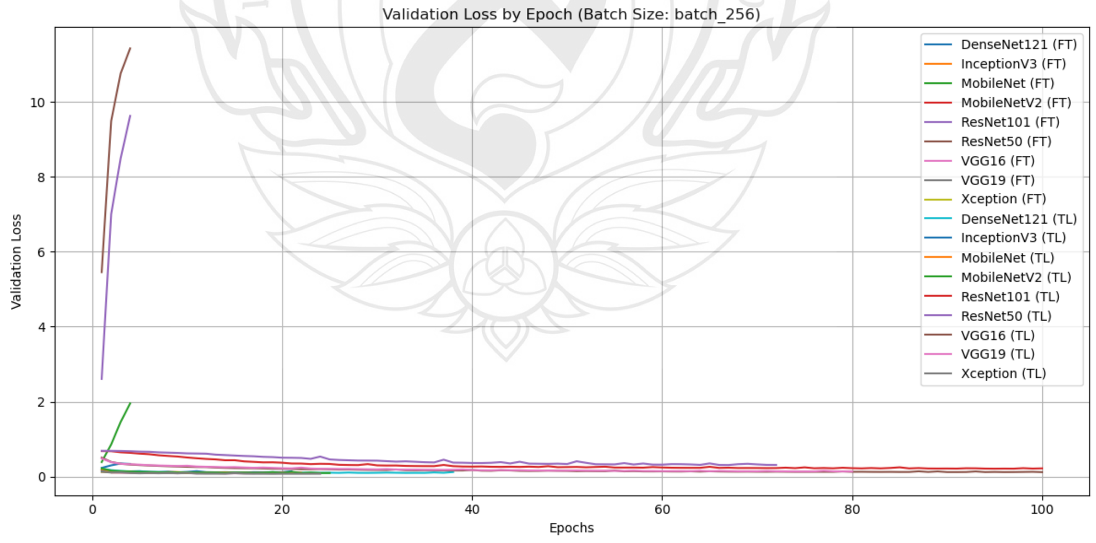


Figure 4.18 Validation loss of CNN models (TL vs FT) on rotation dataset (batch size = 256)

In an experiment involving the training of nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—under a batch size of 256 and using image data augmented through rotation, notable differences were observed in the number of epochs required to reach convergence. Rotation augmentation, which rotates images across multiple angles to diversify object orientation, enhances data variability and is intended to improve the model's generalization capability. The results showed that the number of epochs required for convergence varied considerably across models and appeared to correlate with model depth, parameter size, and whether the training method involved Fine-Tuning (FT) or Transfer Learning (TL). Models that achieved rapid convergence and required very few epochs included MobileNet (FT), MobileNetV2 (FT), ResNet50 (FT), VGG16 (FT), VGG19 (FT), and Xception (FT), all of which converged within approximately 5 epochs, as indicated by stabilized validation accuracy and clearly declining validation loss. Other models, such as DenseNet121 (FT) and InceptionV3 (FT), required slightly more time—around 12 epochs and 10 epochs, respectively—but still demonstrated strong and efficient learning. Among the Transfer Learning group, some models also showed fast and stable adaptation, particularly DenseNet121 (TL), InceptionV3 (TL), MobileNet (TL), MobileNetV2 (TL), and Xception (TL), which all converged within approximately 18 epochs. These findings suggest that certain models can effectively leverage pre-trained parameters and adjust them efficiently to accommodate rotational variation in the training data. In contrast, models such as ResNet101 (TL), ResNet50 (TL), VGG16 (TL), and VGG19 (TL) required up to 100 epochs—the maximum set for this experiment—to stabilize. These results reflect the challenges that large, deep models face in adapting to highly variable spatial transformations introduced by rotation. This is particularly true for sequential architectures like VGG, which lack shortcut connections or multi-branch pathways and therefore may struggle to propagate gradients efficiently across deep layers. Furthermore, the use of large batch sizes—while providing precise gradient estimates—can slow down parameter updates, complicating the optimization dynamics, especially for deep models with pre-trained weights that were not originally exposed to rotated image distributions (e.g., ImageNet). Overall, while rotation augmentation enhances data diversity, it also introduces learning complexity for certain architectures—particularly deep models using Transfer Learning. In contrast, models with residual connections (e.g., ResNet), inception modules, dense connectivity (e.g., DenseNet), or lightweight designs (e.g., MobileNet and Xception) were better able to handle spatial variation introduced by rotation. In conclusion, models that consistently demonstrated high performance and fast convergence in this context included DenseNet121, InceptionV3, and Xception, across both FT and TL settings. Conversely, VGG and ResNet models under Transfer Learning appear more suited for longer training durations and deeper fine-tuning, especially when applied to rotated datasets requiring spatial adaptation.

4.3.9 Noise Dataset with Batch Size 32

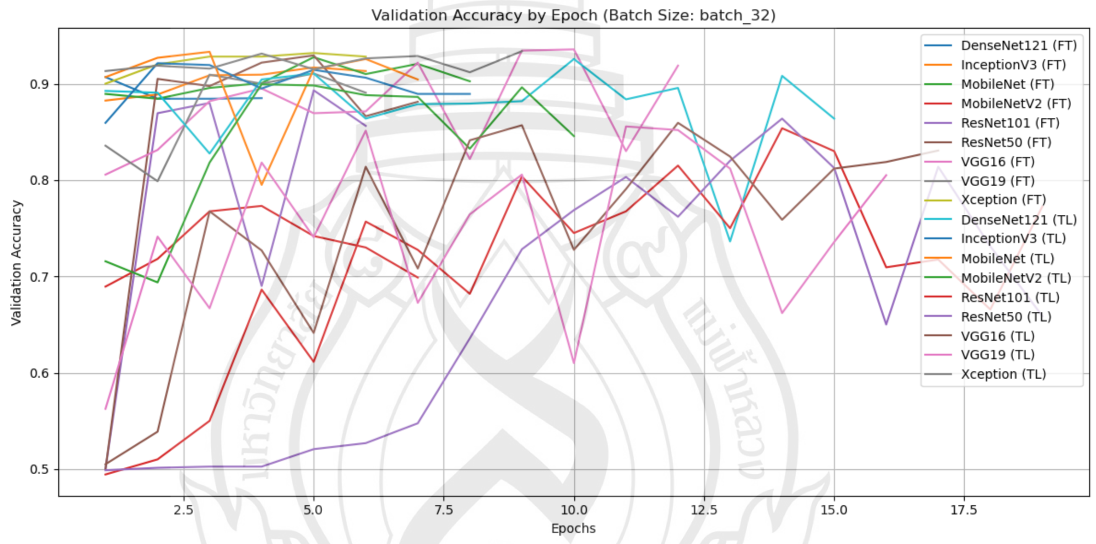


Figure 4.19 Validation accuracy of CNN models (TL vs FT) on noise dataset (batch size = 32)

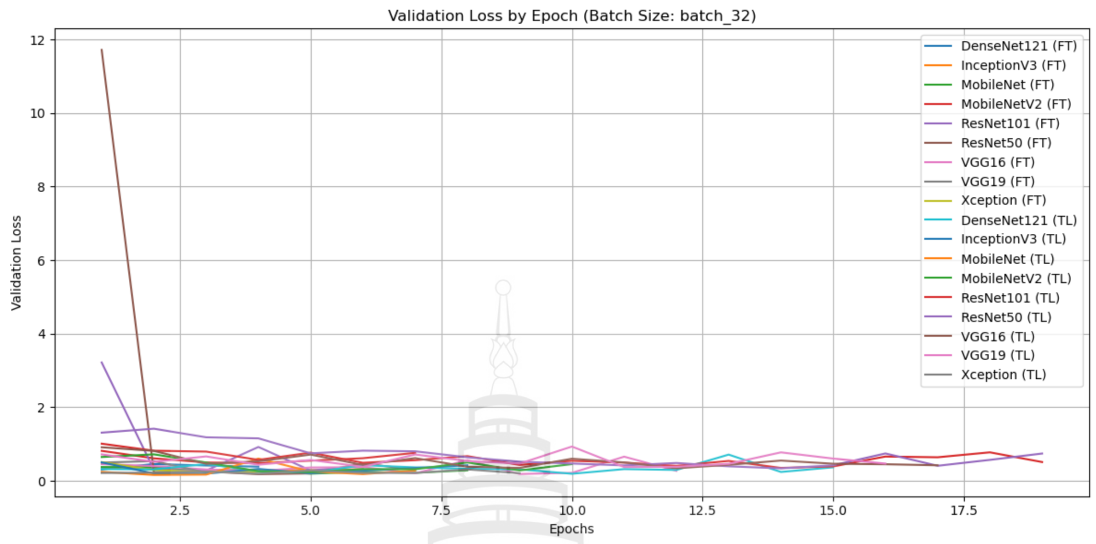


Figure 4.20 Validation loss of CNN models (TL vs FT) on noise dataset (batch size = 32)

In an experiment involving the training of nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—under a batch size of 32, using image data augmented with noise, it was found that all models converged to stable validation accuracy within approximately 10 epochs, in both Fine-Tuning (FT) and Transfer Learning (TL) settings. The noise augmentation technique, which involves adding artificial noise to simulate real-world image imperfections—such as defocus, sensor artifacts, or low-light conditions—had a uniform effect across model architectures, enabling efficient learning without extending the training duration. This result suggests that noise augmentation enhances the model's ability to generalize by exposing it to variability and uncertainty in the input data, which in turn promotes robustness during training. The fact that all models converged within the same number of epochs indicates that noise did not significantly increase data complexity or impair learning efficiency, unlike spatial augmentation techniques such as rotation, which often resulted in longer convergence times—particularly for deeper models like VGG or ResNet under TL. Interestingly, architectural differences—including depth, width, and connectivity mechanisms such as shortcut connections (ResNet), depthwise separable convolutions (Xception), or dense connections (DenseNet)—did not appear to significantly influence training duration when dealing with noise-augmented data. This may be due to the natural regularization effect of noise, which reduces overfitting and allows for more efficient parameter updates, avoiding issues related to slow gradient propagation or learning instability. The ability to converge within just 10 epochs also highlights the effectiveness of both FT and TL strategies in handling imperfect data. Notably, models such as InceptionV3, DenseNet121, and Xception showed high accuracy early in training, as evidenced by their validation curves. Even deeper or parameter-heavy models, such as ResNet101 and VGG19, did not demonstrate performance degradation when trained on noisy data. In conclusion, noise-based data augmentation significantly enhances model robustness and generalization without increasing the required number of training epochs. The consistent behavior observed across all models in both FT and TL modes affirms that noise is a computationally efficient and highly effective augmentation technique for training deep learning models on imperfect or low-quality image datasets, offering a practical solution for real-world applications.

4.3.10 Noise Dataset with Batch Size 64

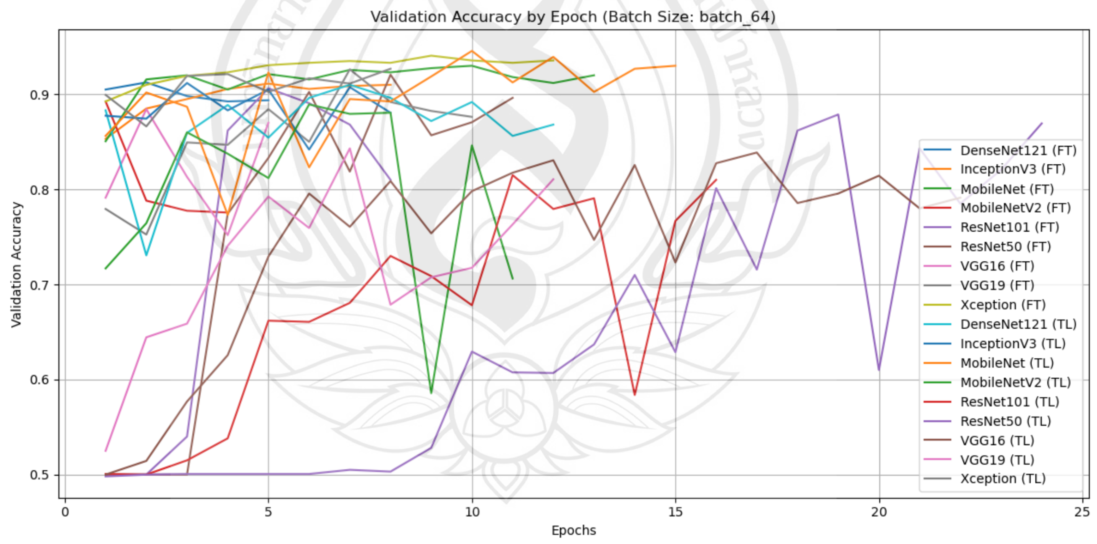


Figure 4.21 Validation accuracy of CNN models (TL vs FT) on noise dataset (batch size = 64)

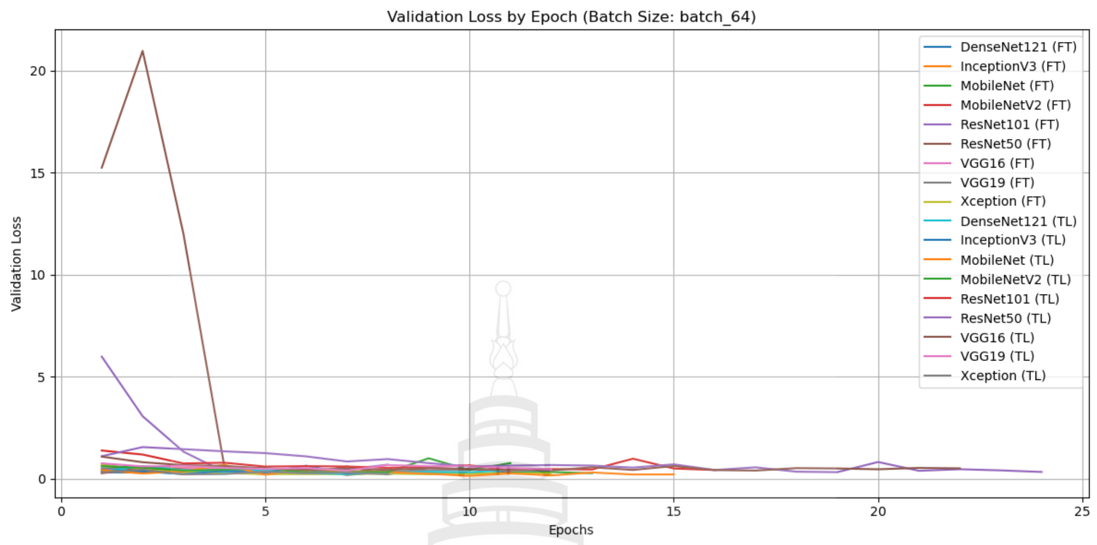


Figure 4.22 Validation loss of CNN models (TL vs FT) on noise dataset (batch size = 64)

In an experiment involving the training of nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—with a batch size of 64, using image data augmented with the Noise technique, it was found that all models, under both Fine-Tuning (FT) and Transfer Learning (TL) strategies, achieved convergence within approximately 10 epochs. This indicates that noise augmentation acts as a form of natural regularization, enabling faster and more effective learning while promoting strong generalization—even when input data includes uncertainty and imperfections.

The fact that models of various sizes—ranging from lightweight architectures like MobileNet, to medium-sized models like DenseNet121 and InceptionV3, and even large models like VGG19 and ResNet101—all reached convergence within the same number of epochs suggests that noise augmentation does not significantly increase the complexity of the learning dynamics. On the contrary, introducing noise appropriately may help prevent overfitting on overly clean data and support more efficient learning.

Notably, architectures with residual connections or inception-based designs, such as DenseNet121, InceptionV3, and Xception, showed strong validation accuracy from the early stages of training, reflecting their ability to quickly learn robust features from noisy images. Similarly, deeper models with large parameter counts, such as

VGG16 and ResNet50, were not adversely affected by noise and maintained stable performance.

Additionally, models trained using Transfer Learning—which leverage pre-trained weights—were able to adapt well to noisy data without requiring extensive retraining. This demonstrates the effectiveness of knowledge transfer across domains, allowing pre-trained models to generalize to noisy environments without the need to train from scratch. In conclusion, noise-based data augmentation not only enhances data diversity but also enables rapid and accurate learning across all CNN architectures under conditions of uncertainty. This makes it a highly effective and practical technique for real-world image analysis tasks, where low-quality or noisy data is often encountered.

4.3.11 Noise Dataset with Batch Size 128

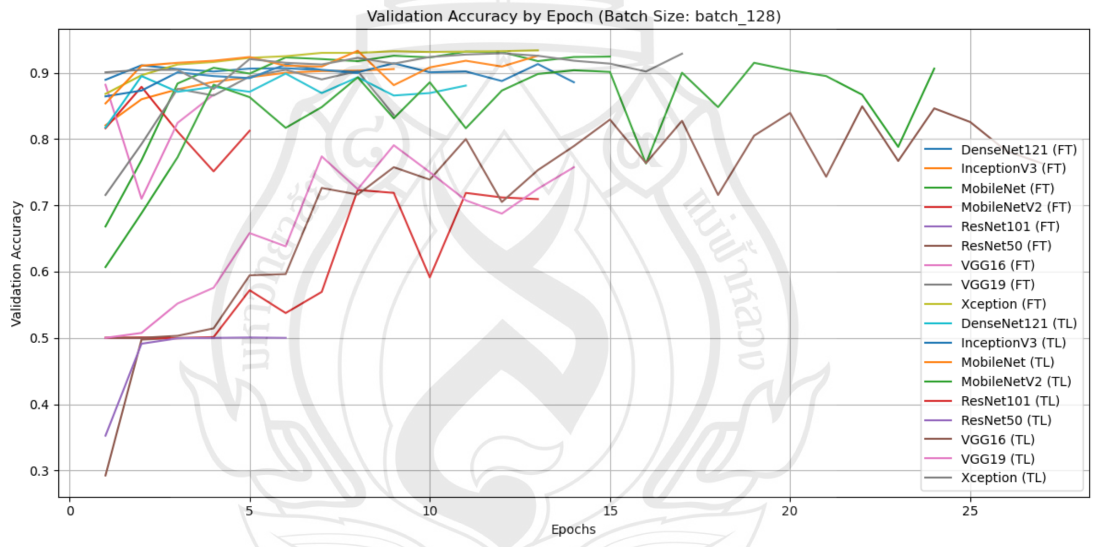


Figure 4.23 Validation accuracy of CNN models (TL vs FT) on noise dataset (batch size = 128)

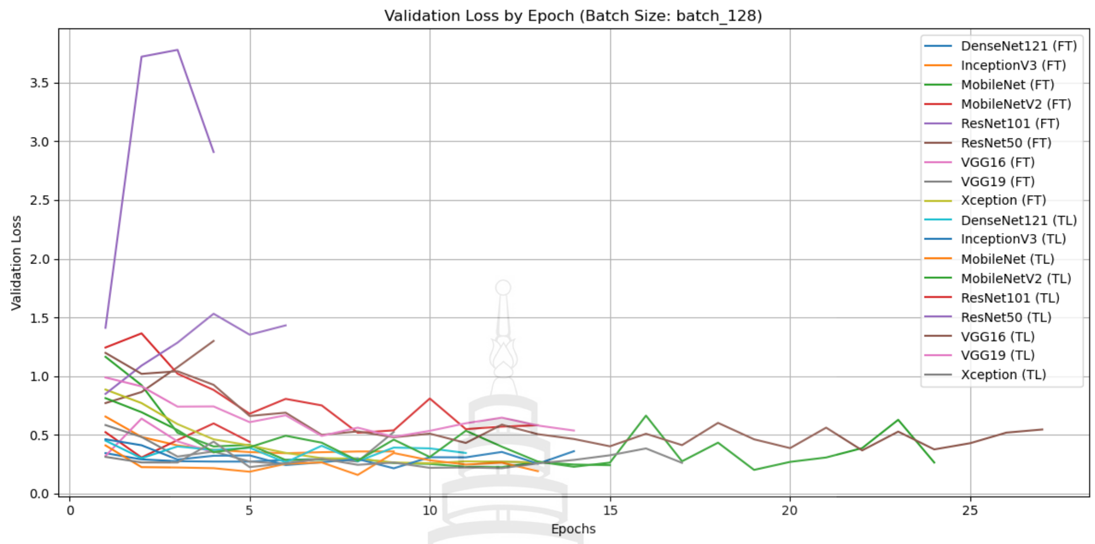


Figure 4.24 Validation loss of CNN models (TL vs FT) on noise dataset (batch size = 128)

In an experiment involving the training of nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—with a batch size of 128, using image data augmented with Noise, which involves adding significant random perturbations to simulate real-world image imperfections from low-quality cameras or recording systems, it was found that most models achieved convergence within a relatively small number of training epochs. Exceptions were observed in some deep or structurally complex models that required longer training durations. Models that demonstrated fast and stable learning, such as DenseNet121, InceptionV3, MobileNet, MobileNetV2, and Xception, under both Fine-Tuning (FT) and Transfer Learning (TL), reached stable validation accuracy and loss within approximately 10 epochs. This reflects the robustness of these architectures in handling noise, particularly those featuring Inception modules or depthwise separable convolutions—such as InceptionV3 and Xception—which excel at extracting meaningful features even in the presence of noise. Notably, ResNet101 (FT) required only 5 epochs before training halted, indicating high sensitivity to noisy input, possibly due to its residual shortcuts, which accelerate parameter adjustments. On the other hand, some models under Transfer Learning, such as ResNet50 (TL), required up to 26 epochs, and VGG16 (TL) and VGG19 (TL) took approximately 15 epochs, which is noticeably higher than most other models. This trend suggests that deeper, sequential models without shortcut connections respond more slowly to noisy data and require more time to adapt pre-trained parameters to signal-perturbed inputs. Overall, training with noise-augmented data improved the generalization capability across all models and helped reduce the risk of overfitting. This effect was particularly evident when combined with a moderate batch size (128), which provided a balance between training speed and gradient update stability. Noise augmentation thus proves to be a valuable technique for training models on imperfect data. Models with flexible architectures and strong knowledge transfer capability from pre-trained weights, such as InceptionV3 (TL) and DenseNet121 (TL), exhibited excellent performance in learning from noisy inputs with minimal training time. In contrast, VGG-based models required longer training durations despite using pre-trained weights, reflecting architectural limitations due to the lack of shortcut connections or attention-aware mechanisms, which otherwise facilitate more efficient information propagation between layers. In conclusion, training CNN models with Noise augmentation under a batch size of 128 supports fast, stable, and effective learning, particularly in models with structural adaptability and the ability to manage noisy inputs—making this approach highly suitable for real-world image classification tasks involving degraded or variable-quality data.

4.3.12 Noise Dataset with Batch Size 256

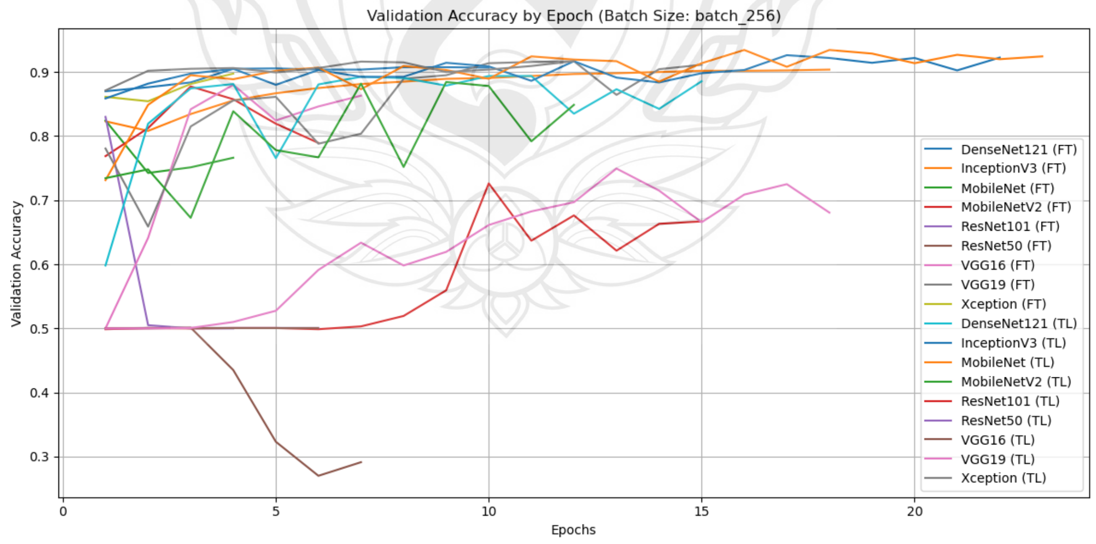


Figure 4.25 Validation accuracy of CNN models (TL vs FT) on noise dataset (batch size = 256)

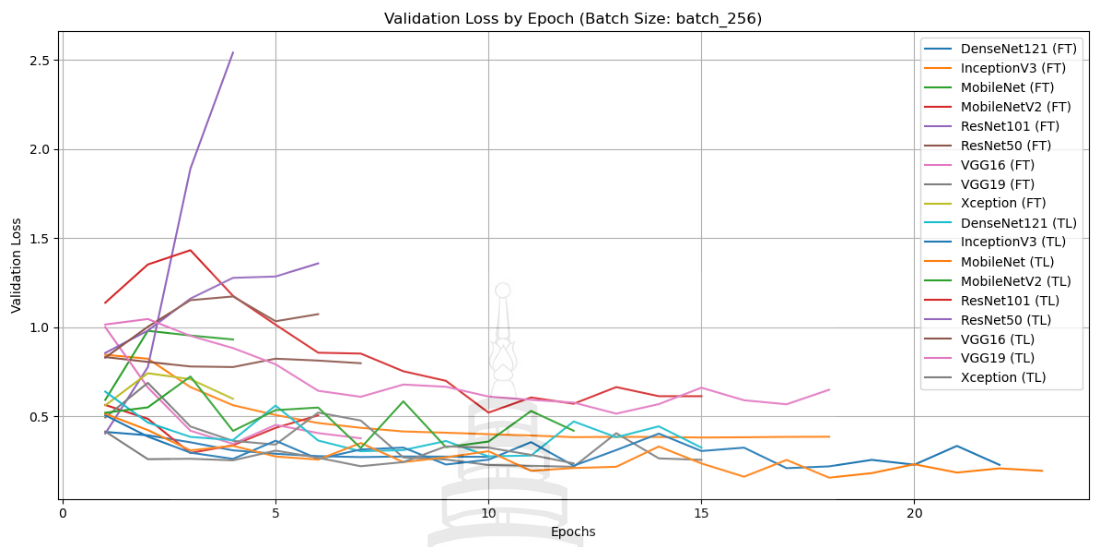


Figure 4.26 Validation loss of CNN models (TL vs FT) on noise dataset (batch size = 256)

In an experiment involving the training of nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—using a batch size of 256, and image data augmented with noise, it was found that most models were able to learn effectively within a relatively small number of epochs, averaging around 10 epochs before validation accuracy began to stabilize and validation loss declined steadily. Models that demonstrated fast and stable learning under both Fine-Tuning (FT) and Transfer Learning (TL) settings included DenseNet121, InceptionV3, MobileNet, MobileNetV2, and Xception, all of which reached convergence in approximately 10 epochs, regardless of whether training was initialized from scratch or from pre-trained weights. This highlights the effectiveness of these architectures in handling noise-contaminated images. Notably, Inception and DenseNet architectures feature mechanisms that integrate information across multiple spatial resolutions, while Xception employs depthwise separable convolutions, which simplify the learning of complex spatial features and contribute to robust performance under noisy conditions. In contrast, models with larger architectures or simpler sequential designs, such as ResNet50 (FT/TL), VGG16 (TL), and VGG19 (TL), also showed reasonably fast training times—especially ResNet50, which converged within only 5–6 epochs. This can be attributed to the use of residual connections, which facilitate efficient gradient propagation and mitigate the vanishing gradient problem. However, model stability in these cases must also be evaluated in conjunction with the validation loss curves, as some models—such as VGG16 (TL) and ResNet101 (TL)—exhibited early-stage instability or relatively high loss before reaching convergence. Overall, the use of noise augmentation enabled nearly all models to adapt effectively, even when trained with large batch sizes like 256. This outcome demonstrates the models' flexibility in handling uncertain or degraded data, and confirms that noise plays a valuable role as a regularization technique that enhances generalization to real-world conditions with practical limitations. In conclusion, the models that exhibited the fastest and most effective learning from noise-augmented images were DenseNet121, InceptionV3, and Xception. While ResNet50 and the VGG models also achieved convergence within a small number of epochs, their loss curves displayed some degree of instability, which should be carefully considered when evaluating overall model performance.

4.3.13 VFlip Dataset with Batch Size 32

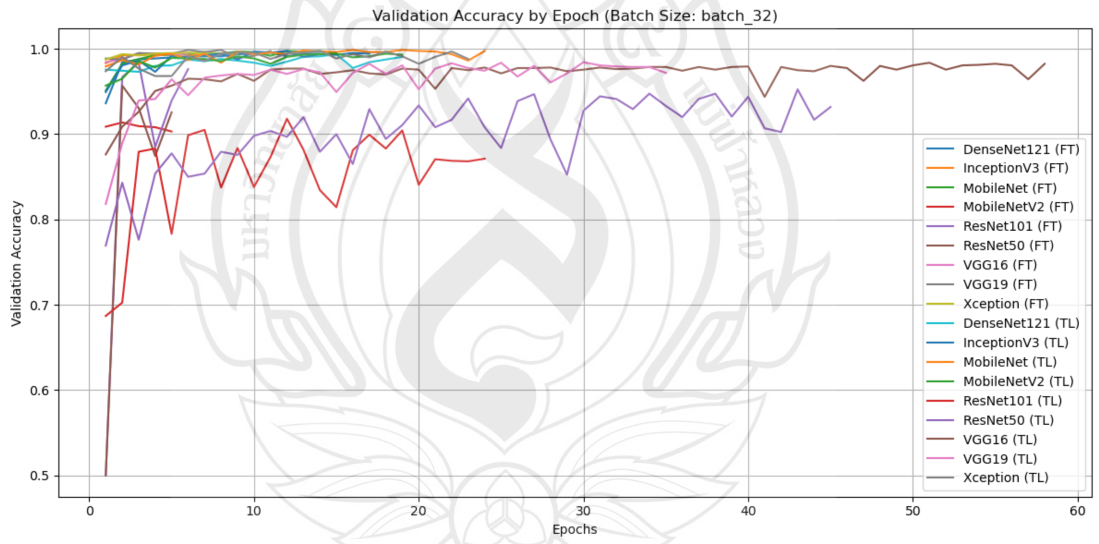


Figure 4.27 Validation accuracy of CNN models (TL vs FT) on VFlip dataset (batch size = 32)

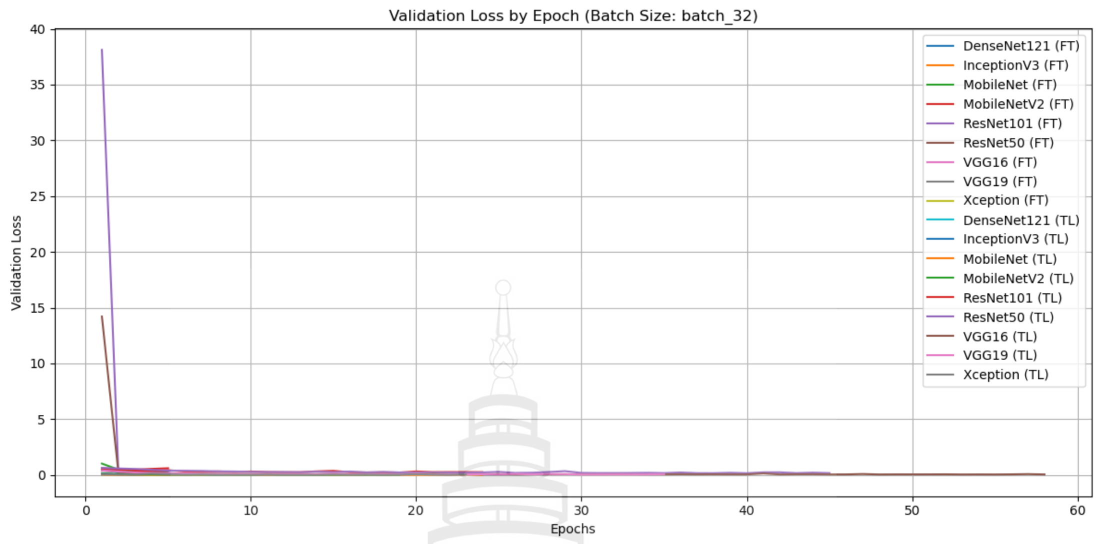


Figure 4.28 Validation loss of CNN models (TL vs FT) on VFlip dataset (batch size = 32)

In an experiment involving the training of nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—with a batch size of 32, using image data augmented with the Vertical Flip (VFlip) technique, it was observed that models converged at different rates depending on the training strategy. Models trained under the Fine-Tuning (FT) setting reached a stable validation accuracy and exhibited continuously declining validation loss within just 6 epochs, indicating that these architectures efficiently adapted to changes in vertical object positioning. This was particularly evident in models like Xception and InceptionV3, which are designed to capture spatial features using multi-level and multi-directional information pathways. Even lightweight models such as MobileNet, which are optimized for computational efficiency, demonstrated rapid learning from vertically flipped images. In contrast, under the Transfer Learning (TL) approach—where models were initialized with pre-trained weights (e.g., from ImageNet)—most architectures, including DenseNet121 (TL), InceptionV3 (TL), MobileNet (TL), MobileNetV2 (TL), and Xception (TL), converged in approximately 10 epochs, showing stable performance when adapting to vertically flipped images. However, deeper architectures without shortcut connections, such as VGG16 (TL), VGG19 (TL), and ResNet101 (TL), required longer training durations. Notably, ResNet50 (TL) required up to 58 epochs to converge. This extended training may be due to the misalignment between the pre-trained weights and the spatial structure introduced by vertical flipping, which differs significantly from other forms of augmentation such as color distortion or noise. VFlip modifies the structural positioning of objects, forcing the models to adjust their internal representations of spatial features. As such, deeper networks may require more time to fine-tune parameters across layers to adapt to this transformation. Additionally, validation accuracy in VGG and ResNet models trained under TL fluctuated more than in other models, suggesting difficulties in adapting to novel spatial configurations that diverge from the original training patterns learned during pre-training. Overall, the Vertical Flip augmentation presented a meaningful spatial recognition challenge, favoring models with flexible architectures and multi-directional feature integration mechanisms. These models—particularly under FT—were able to learn faster than deeply sequential models trained under TL. This supports the conclusion that VFlip is an effective augmentation technique for testing a model's ability to handle spatial variation, though careful selection of architecture and training strategy is essential to achieve optimal results.

4.3.14 VFlip Dataset with Batch Size 64

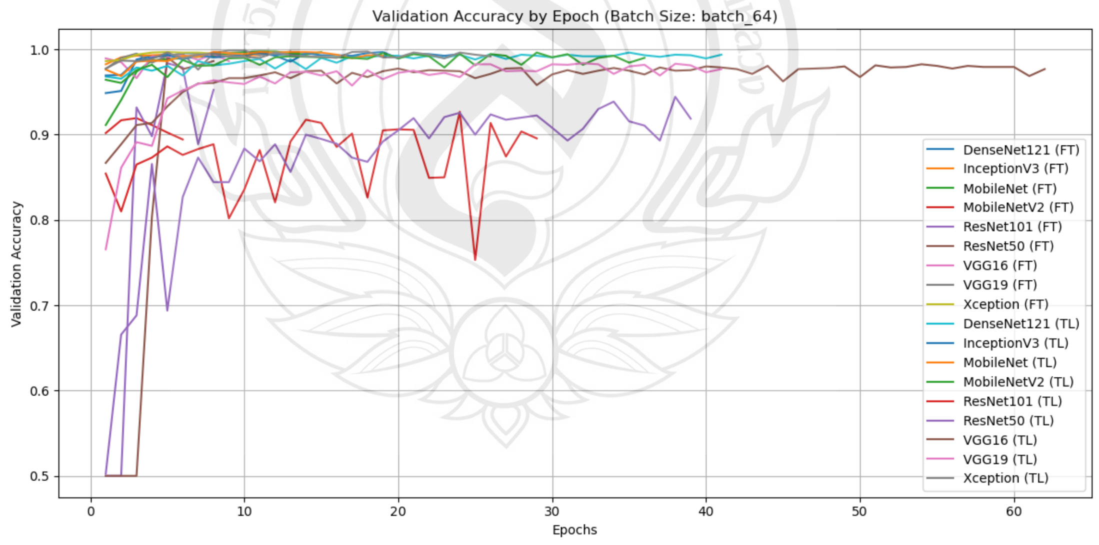


Figure 4.29 Validation accuracy of CNN models (TL vs FT) on VFlip dataset (batch size = 64)

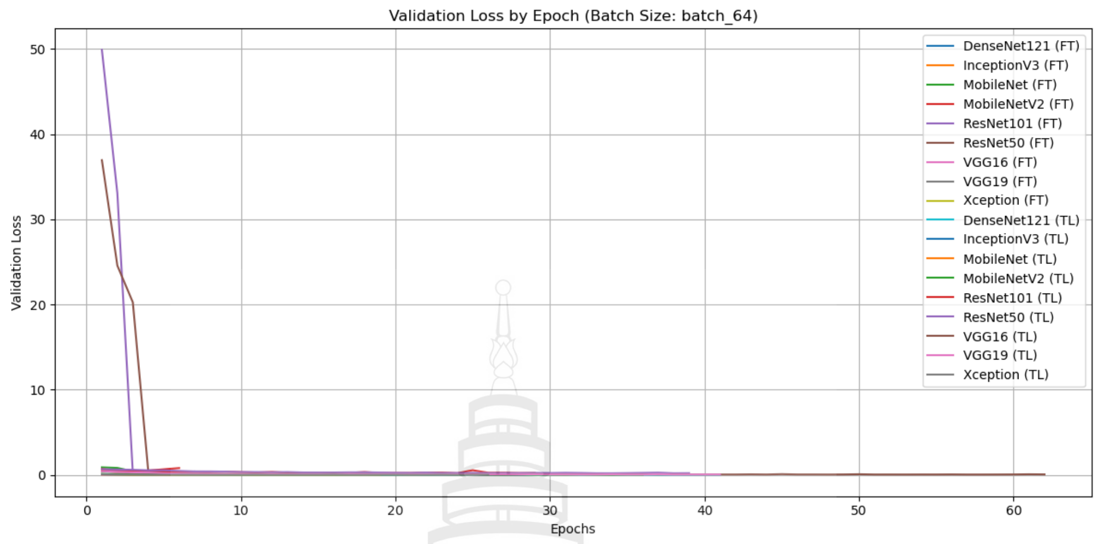


Figure 4.30 Validation loss of CNN models (TL vs FT) on VFlip dataset (batch size = 64)

In an experiment involving the training of nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—using a batch size of 64 and image data augmented through the Vertical Flip (VFlip) technique, which vertically inverts images to introduce spatial diversity, it was found that the models were able to effectively learn new positional and directional variations of objects. The models trained using the Fine-Tuning (FT) strategy achieved rapid convergence within just 6 epochs, including all nine architectures listed. This highlights the strong adaptability of these models to spatial transformations such as vertical flipping, especially in deeper networks or those employing multi-scale feature integration mechanisms, such as DenseNet and Inception, which demonstrated flexibility in learning directional attributes of images. In contrast, models trained using the Transfer Learning (TL) approach exhibited more variation. Flexible architectures such as DenseNet121 (TL), InceptionV3 (TL), MobileNet (TL), MobileNetV2 (TL), and Xception (TL) required only around 10 epochs to converge. These models were able to quickly adapt their pre-trained parameters to the flipped images, showcasing their strong ability to transfer knowledge to spatially altered data. However, larger and more rigid models with deeply sequential architectures—such as ResNet101 (TL), ResNet50 (TL), VGG16 (TL), and VGG19 (TL)—took significantly longer to converge, requiring up to 50 epochs to reach training stability. This reflects limitations in these architectures, which lack shortcut connections or attention mechanisms, making it more difficult to efficiently adapt to spatial variations introduced by vertical flipping—especially when such variations diverge from the natural object orientations present in the ImageNet dataset used for pre-training. These results demonstrate that while VFlip is a relatively simple augmentation technique, it can have a substantial impact on Transfer Learning models, particularly those that rely heavily on fixed pre-trained weights. Adapting to spatial inversion patterns introduced by VFlip requires additional training time and, in some cases, further architectural tuning, especially in more rigid models. In conclusion, Vertical Flip augmentation effectively enhances a model's ability to handle positional changes in image data, particularly when applied to architectures that are structurally flexible or capable of managing spatial features efficiently. In contrast, Transfer Learning models with more rigid sequential structures may require more training epochs and additional fine-tuning of deeper layers to achieve optimal performance in such spatially varied contexts.

4.3.15 VFlip Dataset with Batch Size 128

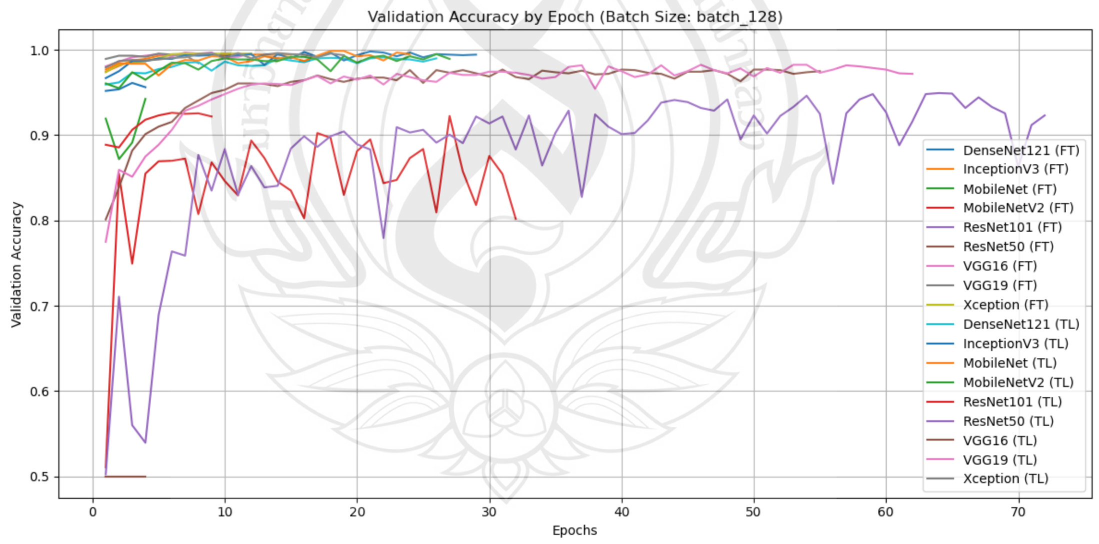


Figure 4.31 Validation accuracy of CNN models (TL vs FT) on VFlip dataset (batch size = 128)

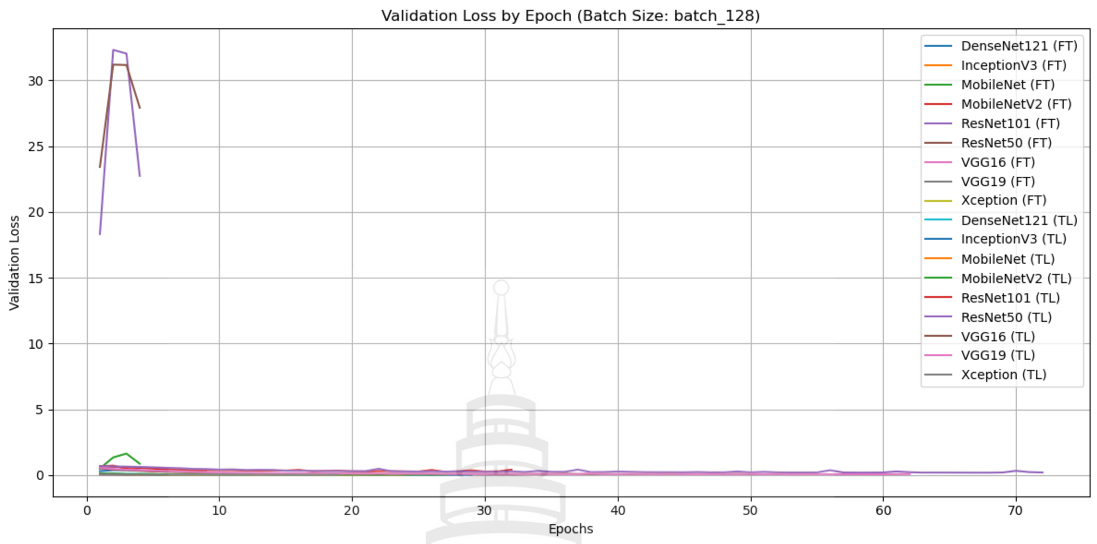


Figure 4.32 Validation loss of CNN models (TL vs FT) on VFlip dataset (batch size = 128)

In an experiment involving the training of nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—using a batch size of 128 and image data augmented with the Vertical Flip (VFlip) technique, which vertically mirrors images to increase spatial diversity of object positions, the results demonstrated a consistent pattern observed in previous experiments involving high spatial transformations. In the Fine-Tuning (FT) group, all models—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—reached a stable state within only 6 epochs, achieving high and steady validation accuracy along with a consistently decreasing validation loss. This suggests that models with all layers unfrozen and fully trainable can quickly learn to handle vertically flipped images. Architectures with strong multi-directional feature integration mechanisms, such as InceptionV3 and Xception, were especially effective. In contrast, the Transfer Learning (TL) models exhibited more varied behavior. Flexible architectures—DenseNet121 (TL), InceptionV3 (TL), MobileNet (TL), MobileNetV2 (TL), and Xception (TL)—required approximately 10 epochs to converge, indicating that their pre-trained parameters were effectively adaptable to the new vertical flip transformation. However, models with rigid and deeply sequential structures, such as ResNet101 (TL), ResNet50 (TL), VGG16 (TL), and VGG19 (TL), required significantly more time to reach convergence. Notably, ResNet101 and ResNet50 required up to 70 epochs, while VGG19 needed 65 epochs before adapting to the flipped image patterns with optimal performance. One possible explanation is that these models were originally designed and pre-trained on datasets like ImageNet, which do not contain vertically flipped images, making the direct transfer of learned spatial representations less effective. As a result, deeper layers needed more time to fine-tune their parameters in order to adapt to this spatial inversion. Another key observation is that, despite the longer training durations, TL models eventually exhibited high and stable validation accuracy in the later epochs. This indicates that extended training does not necessarily imply poor model performance; rather, it reflects the additional time required for the model to adapt to spatial patterns not previously encountered during pre-training. In conclusion, Vertical Flip augmentation proves effective in evaluating a model's ability to adapt to spatial variation. Models trained with Fine-Tuning adapted more rapidly, while those using Transfer Learning remained effective but required longer training durations—especially when the architecture lacked inherent mechanisms to handle strong spatial transformations efficiently.

4.3.16 VFlip Dataset with Batch Size 256

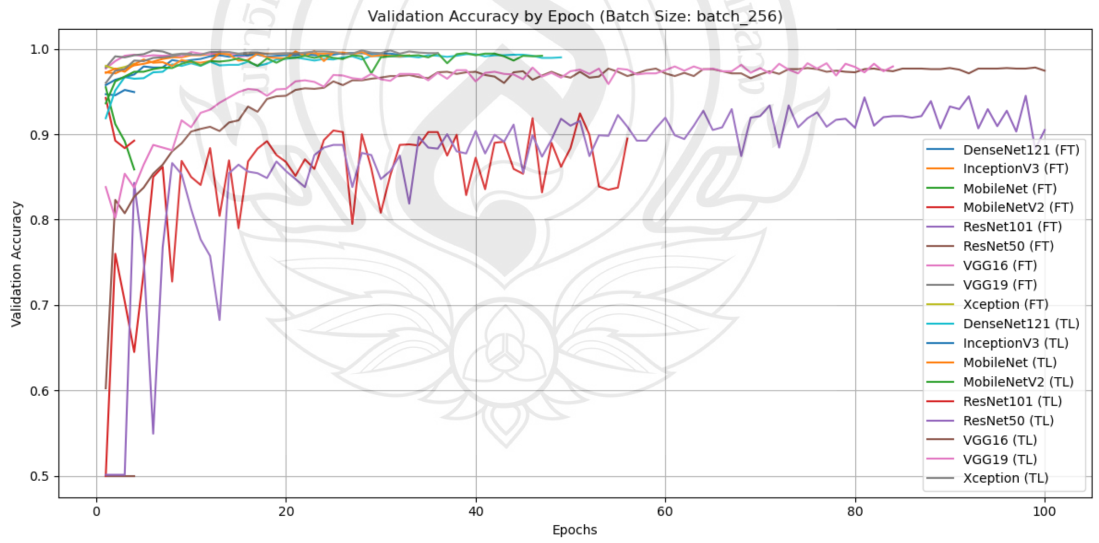


Figure 4.33 Validation accuracy of CNN models (TL vs FT) on VFlip dataset (batch size = 256)

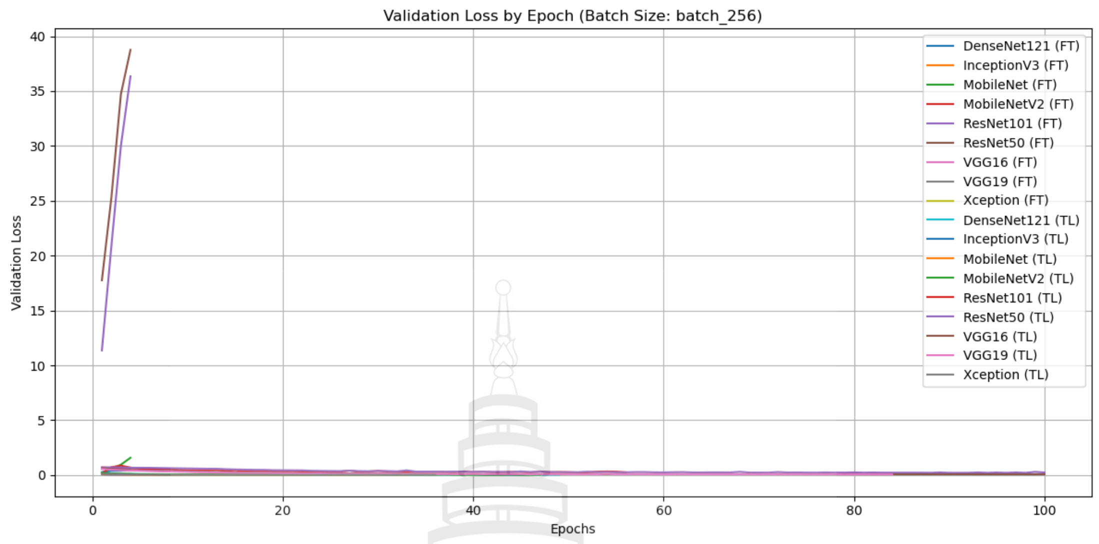


Figure 4.34 Validation loss of CNN models (TL vs FT) on VFlip dataset (batch size = 256)

In an experiment involving the training of diverse Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—under a batch size of 256, using image data augmented through Vertical Flip (VFlip), which mirrors images along the vertical axis to increase positional variation, a significant difference in learning behavior between Fine-Tuning (FT) and Transfer Learning (TL) strategies was observed. All models trained using Fine-Tuning, including the full list above, were able to reach stable validation accuracy and validation loss within approximately 6 epochs, indicating high adaptability of these architectures to directional changes introduced by VFlip. Training from scratch across all layers enabled the models to adjust to spatial transformations without constraint from pre-trained weights, allowing for rapid convergence. In contrast, models trained using Transfer Learning showed more variable behavior. Architectures such as DenseNet121 (TL), InceptionV3 (TL), MobileNet (TL), MobileNetV2 (TL), and Xception (TL) were able to adapt within 10 epochs, demonstrating the flexibility of these models to repurpose pre-trained weights for data exhibiting spatial alterations like vertical flipping. However, deeper and less spatially flexible architectures, including ResNet101 (TL), ResNet50 (TL), VGG16 (TL), and VGG19 (TL), required up to 100 epochs to effectively reduce validation loss and improve accuracy. This extended training time may be due to deep-layer parameters that were originally optimized for object recognition in fixed orientations (as in ImageNet), requiring more iterations to adapt to vertically flipped image patterns.

An important observation is that, although ResNet and VGG models under TL required significantly more epochs, they were able to maintain high accuracy in the long run, with no clear signs of overfitting, as evidenced by steadily decreasing loss curves. This suggests that large batch sizes, such as 256, may contribute to more stable learning dynamics when models are trained on data involving spatial direction changes like VFlip. In summary, applying VFlip augmentation in conjunction with Fine-Tuning allows models to learn and converge rapidly—often within just a few epochs. In contrast, Transfer Learning, especially when applied to deeper architectures, requires considerably more epochs to adjust the pre-trained weights to new spatial patterns. This highlights the importance of carefully selecting training strategies, augmentation techniques, model architectures, and batch size configurations in order to achieve optimal performance across varying types of data transformations.

4.3.17 HFlip Dataset with Batch Size 32

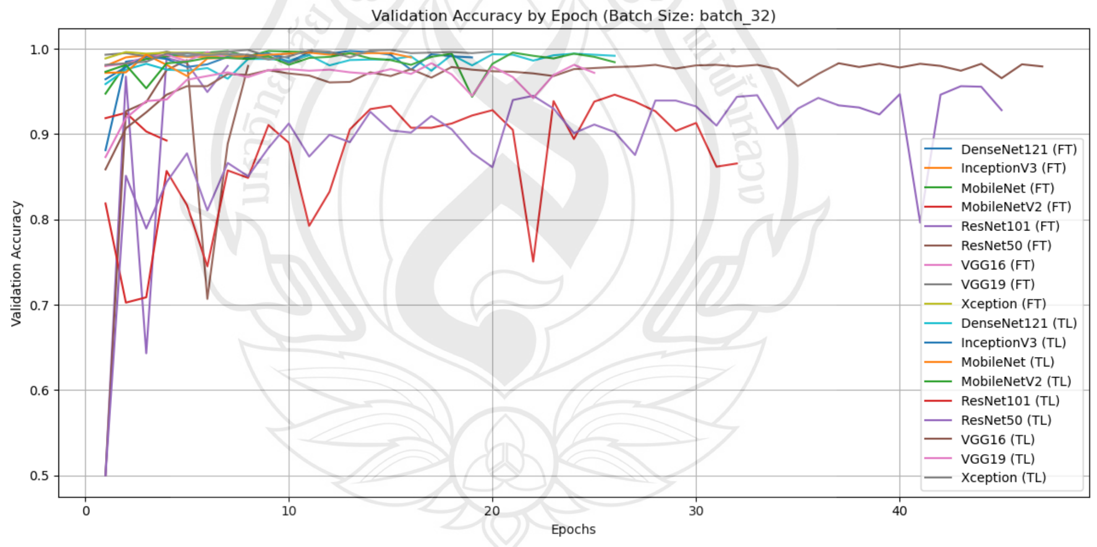


Figure 4.35 Validation accuracy of CNN models (TL vs FT) on HFlip dataset (batch size = 32)

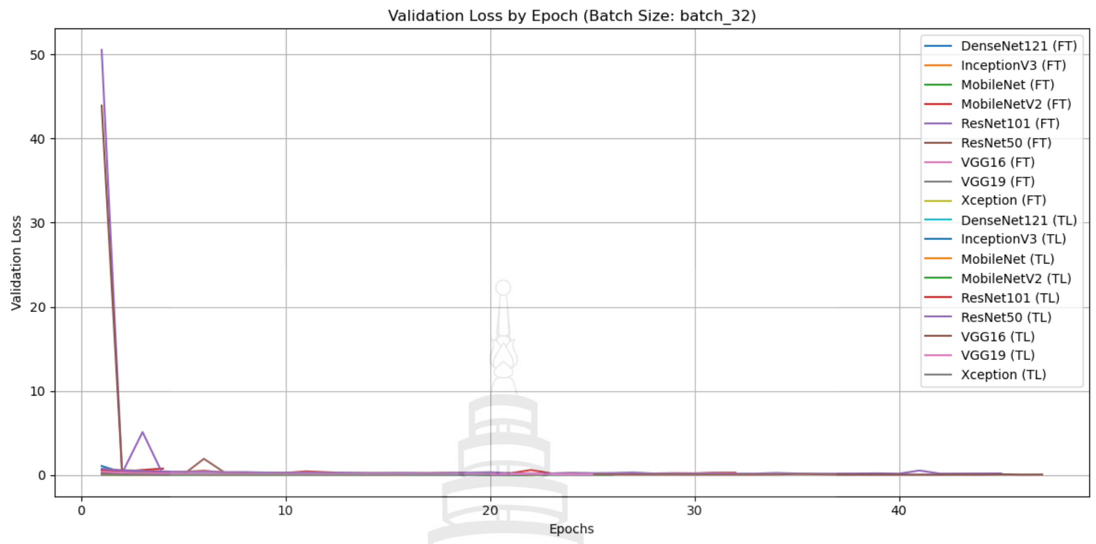


Figure 4.36 Validation loss of CNN models (TL vs FT) on HFlip dataset (batch size = 32)

In an experiment involving the training of nine primary Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—with a batch size of 32 and image data augmented using the Horizontal Flip (HFlip) technique, it was observed that the learning behavior closely resembled that of experiments using Vertical Flip (VFlip). However, HFlip provides a practical advantage by simulating directional variations more closely aligned with real-world astronomical imaging conditions, where horizontal inversions often occur due to lens reflection or sensor orientation. In the Fine-Tuning (FT) group, all models—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—reached convergence within 6 epochs, demonstrating their capability to quickly learn horizontally flipped patterns. This was particularly evident in models with multi-scale spatial feature extraction, such as InceptionV3 and DenseNet121, which showed strong adaptability to directional changes. In contrast, the Transfer Learning (TL) group, which utilized pre-trained weights from ImageNet, showed a more varied response. Models such as DenseNet121 (TL), InceptionV3 (TL), MobileNet (TL), MobileNetV2 (TL), and Xception (TL) converged in approximately 10 epochs, indicating their strong ability to generalize and adapt pre-trained features to spatial inversions within the range of prior learning. However, deeper and more complex models, such as ResNet101 (TL), ResNet50 (TL), VGG16 (TL), and VGG19 (TL), required up to 45 epochs to achieve stable learning. This highlights the limitations of long sequential architectures, which require extended training to fine-tune internal parameters for learning horizontally flipped patterns—particularly when ImageNet pre-trained weights, which do not include flipped images, are used as a starting point. A notable observation is that although TL models requiring higher epoch counts trained more slowly, their accuracy curves steadily progressed toward stable and high performance without clear signs of overfitting. This indicates that training with a smaller batch size of 32 enhances learning with higher variance, which in turn improves the models' adaptability to spatially transformed data such as HFlip. In summary, Horizontal Flip augmentation is effective for enabling models to learn from horizontal object displacements, particularly when combined with Fine-Tuning, which fully leverages the capacity of CNN architectures. While Transfer Learning may require longer training times for certain model types, it can still deliver strong performance outcomes—provided that appropriate tuning of epochs and layer configurations is applied to match the new data characteristics introduced by image flipping.

4.3.18 HFlip Dataset with Batch Size 64

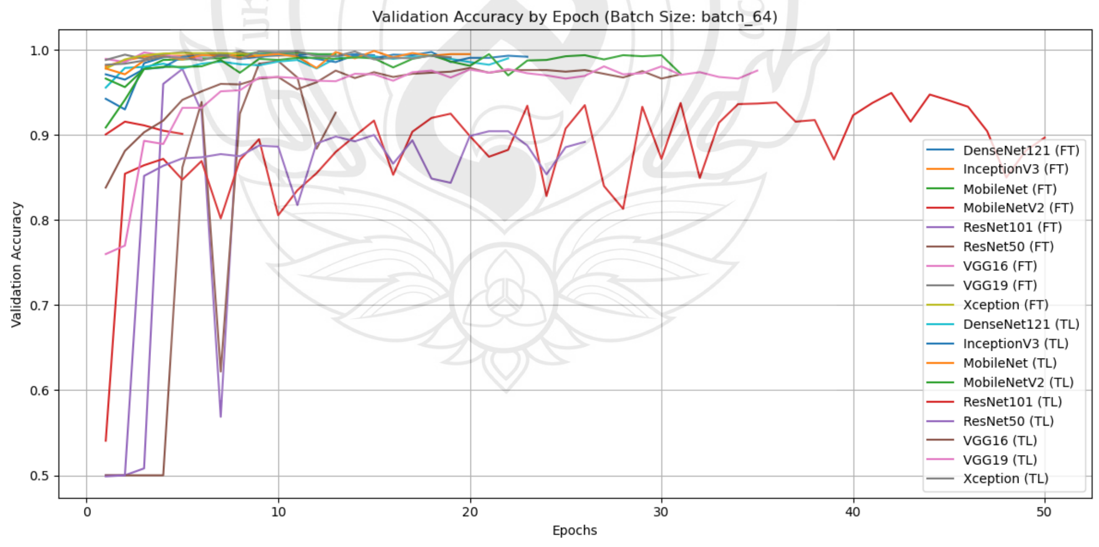


Figure 4.37 Validation accuracy of CNN models (TL vs FT) on HFlip dataset (batch size = 64)

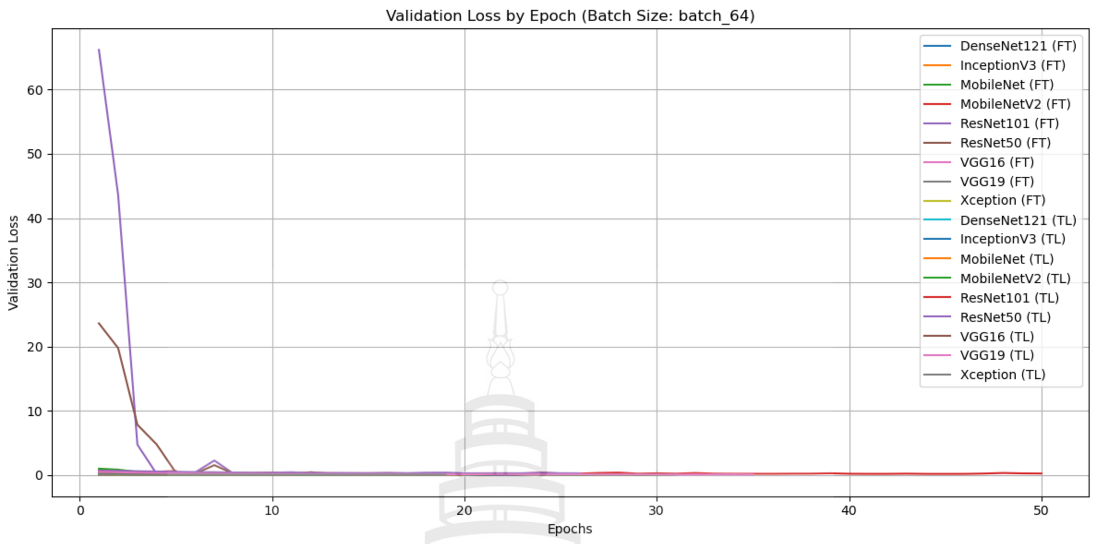


Figure 4.38 Validation accuracy of CNN models (TL vs FT) on HFlip dataset (batch size = 64)

In an experiment involving the training of Convolutional Neural Network (CNN) models under both Fine-Tuning (FT) and Transfer Learning (TL) strategies using image data augmented with Horizontal Flip (HFlip) and a batch size of 64, it was observed that each model exhibited distinct learning behavior, particularly in terms of convergence speed and the number of epochs required to achieve stable validation loss reduction and accuracy improvement. Notably, all models trained under the Fine-Tuning strategy—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception—achieved high and stable validation accuracy within only 6 epochs, indicating that full model retraining enables efficient adaptation to horizontal spatial transformations. This performance reflects the capacity of deep layers to flexibly adjust parameters in response to flipped patterns introduced through data augmentation. In contrast, models trained with Transfer Learning, using pre-trained weights from ImageNet, demonstrated more varied performance. While DenseNet121 (TL), InceptionV3 (TL), MobileNet (TL), MobileNetV2 (TL), and Xception (TL) required only 10 epochs to reach convergence, deeper and more complex models such as ResNet101 (TL), ResNet50 (TL), VGG16 (TL), and VGG19 (TL) required up to 50 epochs. This finding highlights the limitations of transferring knowledge from ImageNet, which does not typically include horizontally flipped images. As a result, these models require longer training durations to adjust their deep-layer feature representations to accommodate new spatial patterns introduced by HFlip. Moreover, the use of a larger batch size (64) contributed to more stable parameter updates per iteration, reducing gradient variance and supporting consistent learning across epochs. While this can increase the number of epochs needed compared to smaller batch sizes, it also improves training stability, particularly in TL models with restricted weight adjustment capabilities. In summary, employing Horizontal Flip augmentation in combination with Fine-Tuning is highly effective in accelerating the learning of transformed spatial patterns and significantly reducing the number of epochs required for convergence. Meanwhile, Transfer Learning, despite its advantages in initializing models with relevant pre-trained parameters, often requires longer training—especially for deeper models or those originally designed for fixed-directional feature extraction. Therefore, selecting an appropriate training strategy should be guided by the nature of the data and the complexity of the model architecture being used.

4.3.19 HFlip Dataset with Batch Size 128

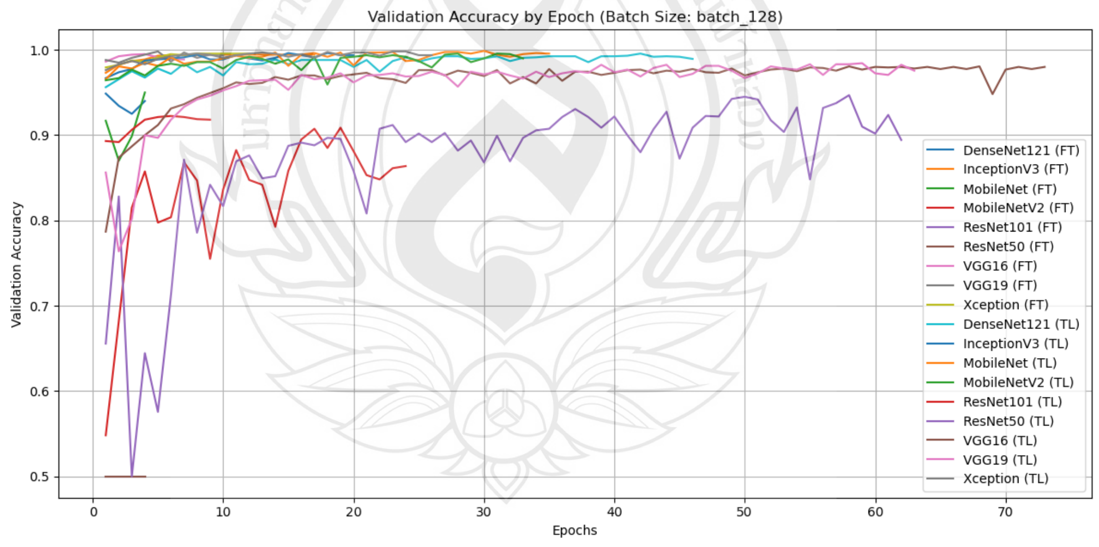


Figure 4.39 Validation accuracy of CNN models (TL vs FT) on HFlip dataset (batch size = 128)

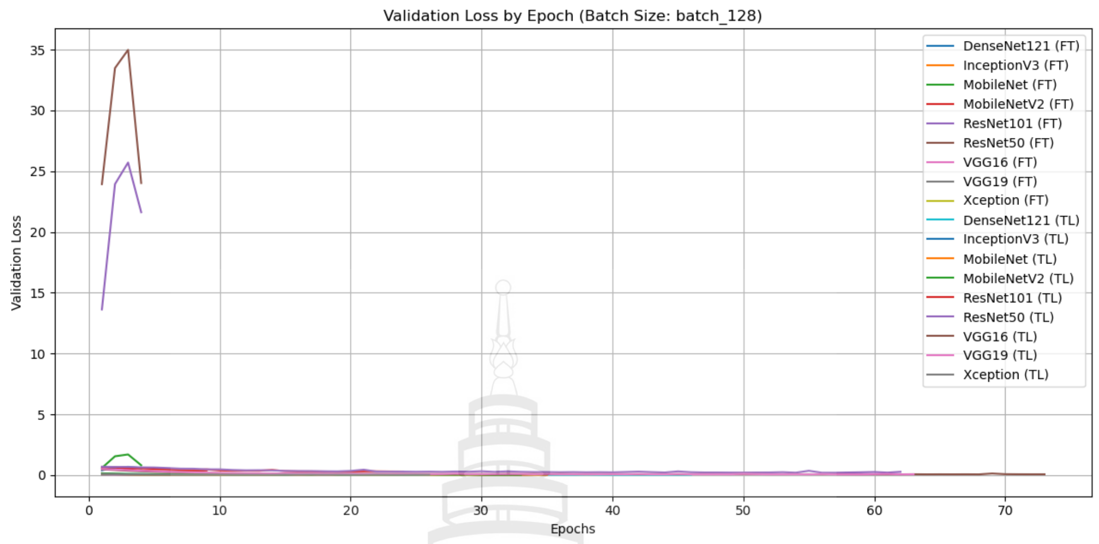


Figure 4.40 Validation loss of CNN models (TL vs FT) on HFlip dataset (batch size = 128)

Based on experiments using astronomical image data augmented with the Horizontal Flip (HFlip) technique and trained with a batch size of 128, it was observed that Convolutional Neural Networks (CNNs) exhibited learning behavior consistent with previous experiments in terms of validation accuracy and loss trends. However, a noticeable difference was found in the number of training epochs required for each model to reach convergence, particularly when comparing the Fine-Tuning (FT) and Transfer Learning (TL) strategies. For models trained using Fine-Tuning, including DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception, convergence was achieved within just 6 epochs. This reflects the effectiveness of full-layer training in enabling models to adapt efficiently to new image characteristics such as horizontal flipping. In contrast, models trained using Transfer Learning with pre-trained weights from ImageNet—such as DenseNet121 (TL), InceptionV3 (TL), MobileNet (TL), MobileNetV2 (TL), and Xception (TL)—required around 10 epochs. Although pre-trained weights facilitated faster learning, a few additional epochs were still needed to adjust to spatial changes do not present in the original training set. More complex and deeper architectures—ResNet101 (TL), ResNet50 (TL), VGG16 (TL), and VGG19 (TL)—required up to 70 epochs to converge. This extended training duration suggests that transferring knowledge to handle new patterns introduced by HFlip (which are largely absent from ImageNet) demands prolonged learning to avoid overfitting to the original feature structure. Another key observation is that increasing the batch size to 128 helped stabilize gradient updates per training step, improving the model's ability to learn from data with high spatial orientation variation. While this may lead to longer convergence times—especially in deep TL models—it also reduces the noise from mini-batch sampling, enhancing learning efficiency. Overall, the use of Horizontal Flip augmentation proved effective and consistent in improving model performance, particularly for models trained with Fine-Tuning, which demonstrated superior ability to adapt to new spatial transformations. In contrast, Transfer Learning required more training epochs for deep models to effectively align pre-existing knowledge with structural changes in object positioning introduced by HFlip.

4.3.20 HFlip Dataset with Batch Size 256

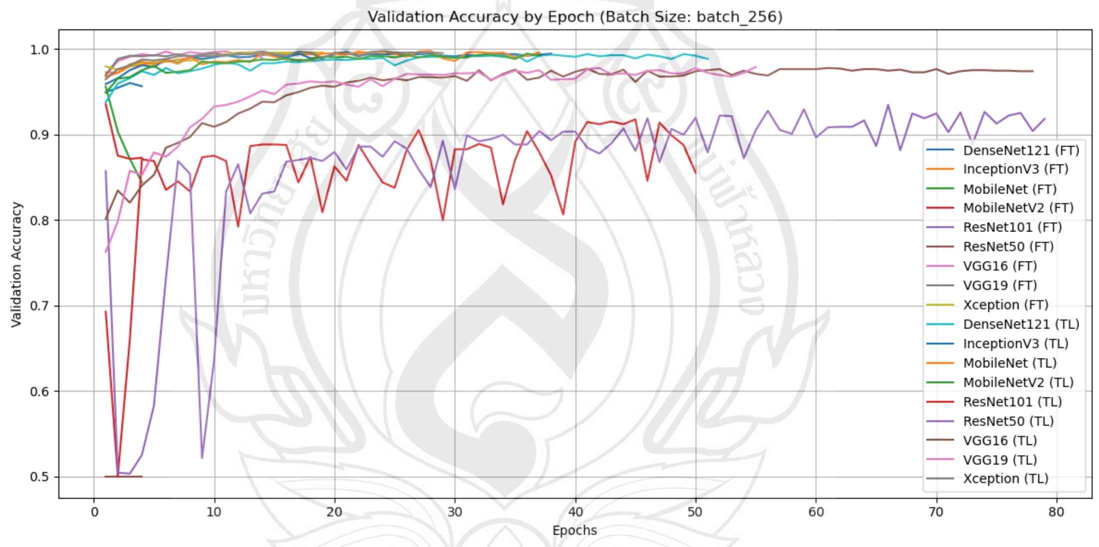


Figure 4.41 Validation accuracy of CNN models (TL vs FT) on HFlip dataset (batch size = 256)

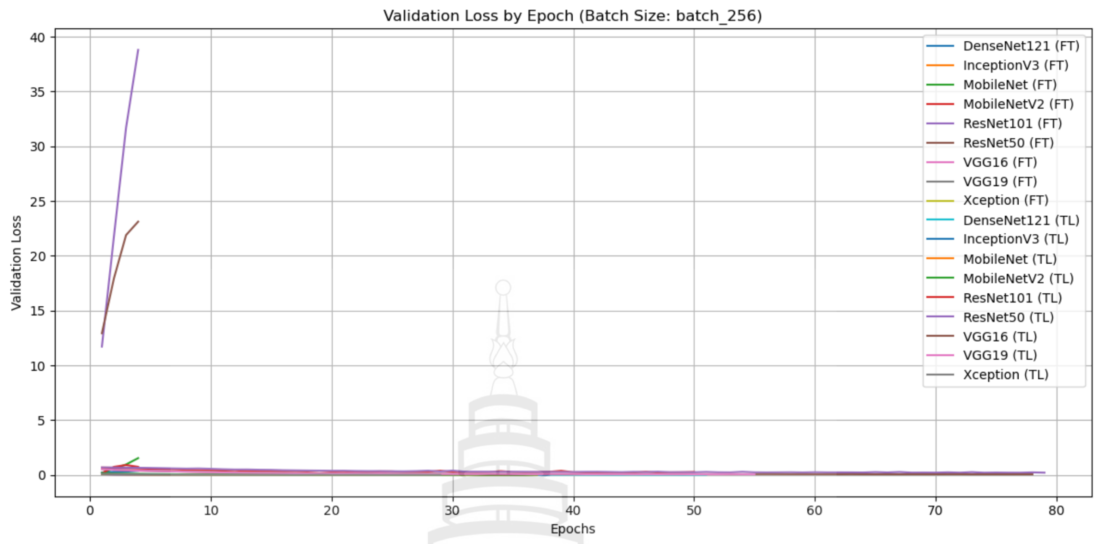


Figure 4.42 Validation loss of CNN models (TL vs FT) on HFlip dataset (batch size = 256)

Based on experimental results using astronomical image data augmented with the Horizontal Flip (HFlip) technique and trained with a batch size of 256, the Convolutional Neural Networks (CNNs) demonstrated distinct learning behaviors, particularly when comparing the Fine-Tuning (FT) and Transfer Learning (TL) strategies across various architectures. The models tested included DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet101, ResNet50, VGG16, VGG19, and Xception, each evaluated separately under both learning strategies. The resulting graphs clearly showed that FT models required significantly fewer training epochs than TL models. On average, all FT models—DenseNet121 (FT), InceptionV3 (FT), MobileNet (FT), MobileNetV2 (FT), ResNet101 (FT), ResNet50 (FT), VGG16 (FT), VGG19 (FT), and Xception (FT)—converged within just 6 epochs. In contrast, several TL models such as DenseNet121 (TL), InceptionV3 (TL), MobileNet (TL), MobileNetV2 (TL), and Xception (TL) converged in approximately 10 epochs, while deeper TL models like ResNet101 (TL), ResNet50 (TL), VGG16 (TL), and VGG19 (TL) required up to 70 epochs. This reflects the architectural constraints of each model group in adapting to horizontally flipped images, a form of spatial transformation that differs significantly from the training distribution of the ImageNet dataset, which served as the source of pre-trained weights for TL models. The faster convergence observed in FT models is attributed to the ability to immediately fine-tune all network parameters, allowing for rapid adaptation to new image orientations. Conversely, TL models typically begin with frozen convolutional layers, adjusting only the final classification layers initially—thus requiring more time and data to gradually adapt to inverted spatial structures. The use of a large batch size (256) contributed to more stable gradient computations by reducing noise from mini-batch sampling, thereby improving overall learning efficiency, especially in how feature distributions were balanced within batches. However, this larger batch size also led to fewer parameter updates per epoch, meaning that deeper TL models—particularly ResNet101 and VGG19—required more training epochs to effectively realign their internal representations to match the new spatial characteristics introduced by HFlip. Overall, the results demonstrated that Fine-Tuning remains the most efficient strategy for astronomical image classification tasks involving spatial orientation changes like horizontal flipping. Despite the longer per-epoch training time commonly associated with large batch sizes, FT models were able to achieve high accuracy in a short and stable training period. In contrast, Transfer Learning models required more epochs to fully adjust to the transformed data, especially when model depth and rigidity limited the adaptability of pre-trained features. These findings highlight the critical importance of selecting an appropriate training strategy based on the nature of the data and available computational resources in astronomical research.

4.4 Conclusion Performance Comparison Based on Validation Accuracy and Loss Graphs Using Transfer Learning and Fine-Tuning with CNNs

Table 4.4 Conclusion performance comparison based on validation accuracy

Batch Size	Mean Epochs (FT)	Mean Epochs (TL)	Stability (Validation Accuracy)	Loss Convergence Speed
32	6	15	Medium	Fast
64	8	22	High	Moderate
128	9	25	High	Moderate
256	6	28	High	Slow

Based on a comprehensive analysis of various CNN architectures, data augmentation techniques, and batch sizes in both Fine-Tuning (FT) and Transfer Learning (TL) experiments, it’s clear that batch size significantly influences model performance, particularly affecting validation accuracy stability, loss reduction speed, and the number of epochs required for convergence. As presented in Table 4.4, “Conclusion Performance Comparison Based on Validation Accuracy,” a batch size of 32, while offering 'Fast' loss convergence and requiring only 6 mean epochs for FT, exhibited 'Medium' validation accuracy stability and 15 mean epochs for TL. In contrast, batch sizes of 128 and 256 provided 'High' validation accuracy stability but resulted in considerably longer training times in TL (25 and 28 mean epochs, respectively), with 256 showing ‘Slow’ loss convergence. Consequently, a batch size of 64 emerged as the most optimal choice. It strikes an excellent balance with 'High' validation accuracy stability, 'Moderate' loss convergence speed, and reasonable mean epochs (8 for FT, 22 for TL), making it ideal for high-performance tasks where both model effectiveness and training time are critical considerations.

4.4.1 Comparison of Classification Results of Different Deep with Original (Non-augmented) Dataset

From experiments using the original (non-augmented) astronomical image data to train Convolutional Neural Network (CNN) models, the performance of Transfer Learning (TL) and Fine-Tuning (FT) strategies was compared under various batch sizes (32, 64, 128, and 256) across multiple model architectures including DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet50, ResNet101, VGG16, VGG19, and Xception. The findings revealed that most models performed well, especially MobileNet and VGG16, which consistently demonstrated high stability and excellent performance in terms of Accuracy, Precision, Recall, and F1-score across different batch sizes and training strategies. Starting with MobileNet, one of the best-performing models, it maintained Accuracy above 0.98 across all batch sizes under both TL and FT settings. In particular, MobileNet fine-tuned with batch size 64 achieved an Accuracy of 0.98938 and an F1-score for the “real” class of 0.95758, reflecting its ability to accurately classify real astronomical objects despite data complexity. Even at batch size 256, MobileNet fine-tuned achieved Accuracy of 0.97876 and F1-score (real) of 0.92045, indicating strong generalization ability and efficient responsiveness to original, unaltered input. More complex models such as DenseNet121 and Xception also delivered excellent results at various batch sizes. For instance, DenseNet121 under Transfer Learning at batch sizes 64 and 128 achieved identical Accuracy scores of 0.98027 and a real class F1-score of 0.91925, suggesting that this architecture can effectively extract deep features from original images. Meanwhile, Xception fine-tuned at batch size 256 reached an Accuracy of 0.97572 and F1-score (real) of 0.90244, making it highly suitable for tasks requiring structural precision and detailed feature representation. The VGG16 and VGG19 architectures also showed robust performance. VGG16 fine-tuned with batch size 256 yielded Accuracy of 0.98331 and F1-score (real) of 0.93252, matching or even surpassing smaller models. Similarly, VGG19 fine-tuned delivered comparable results with an F1-score (real) of 0.93168 at batch size 256. These results highlight the strength of the VGG architecture—despite its depth and sequential structure, it remains capable of learning effectively from raw astronomical images, even without additional augmentation. On the other hand, ResNet50 and ResNet101 showed more variable performance, particularly in the Precision and F1-score of the “real” class. In some cases, these metrics dropped to zero, even when overall accuracy appeared high, such as ResNet50 fine-tuned at batch sizes 64 or 128. This suggests a tendency toward overfitting to the “bogus” class, potentially caused by class imbalance and limited feature differentiation. However, ResNet101 fine-tuned with batch size 32 achieved Accuracy of 0.97117 and an F1-score (real) of 0.88623, indicating that smaller batch sizes may help the model maintain a better learning balance between classes. Overall, when comparing all models trained using only the original augmentation technique, MobileNet (fine-tuned, batch size 64) emerged as the most optimal, delivering the highest accuracy (0.98938) and consistently strong F1-scores for both “real” and “bogus” classes. Its relatively lightweight architecture, combined with strong performance, makes it an excellent candidate for deployment in resource-constrained environments. Other top-performing models include VGG16 (fine-tuned, batch 256), DenseNet121 (transfer, batch 128/256), and Xception (fine-tuned, batch 256), which demonstrated comparably high performance. These findings underscore the importance of selecting a model architecture suited to the characteristics of the data, appropriately adjusting batch size, and ensuring that data augmentation strategies align with the specific classification task. Notably, the use of original data alone, without any geometric or noise-based augmentations, can still produce highly effective models—provided the architecture and training process are well-aligned with the data's underlying structure.

4.4.2 Conclusion Performance Comparison Based on Validation Accuracy with Rotation Dataset

In this experiment, the training data consisted entirely of astronomical images augmented using Rotation, with the objective of enhancing the diversity of object orientations and improving the model’s ability to learn from rotationally varied perspectives. A total of nine Convolutional Neural Network (CNN) architectures were tested—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet50, ResNet101, VGG16, VGG19, and Xception—under different batch sizes (32, 64, 128, and 256) and two training strategies: Transfer Learning (TL) and Fine-Tuning (FT). The results indicated that Xception consistently outperformed other models, particularly in Transfer Learning at batch size 128, where it achieved an accuracy of 0.97750 and an F1-score (real) of 0.97761, demonstrating exceptional capability in learning rotational patterns. Even in Fine-Tuning with batch size 256, Xception maintained excellent performance, with accuracy of 0.96938 and F1-score (real) of 0.96981, reflecting both high accuracy and stability. Another standout model was VGG16 (Fine-Tuned), which achieved accuracy of 0.97500 and F1-score (real) of 0.97503 at batch size 128, closely matching Xception’s performance and surpassing many other models.

MobileNet, known for its computational efficiency and compact architecture, also performed well in Transfer Learning at batch size 128, with accuracy of 0.96250 and F1-score (real) of 0.96245. However, when fine-tuned, MobileNet displayed signs of overfitting or class imbalance, particularly at batch size 128, where despite a high precision of 0.99826, the recall dropped to 0.71875, resulting in a reduced F1-score of 0.87610. This suggests that additional regularization or balancing techniques may be necessary when applying Rotation to lightweight models like MobileNet.

ResNet50 and ResNet101 exhibited greater variability. For example, ResNet50 fine-tuned at batch size 256 failed completely, with accuracy of 0.50000 and F1-score of 0.00000, indicating poor compatibility between deep ResNet architectures and Rotation augmentation without appropriate tuning. On the other hand, ResNet101 fine-tuned at batch size 32 performed well, achieving accuracy of 0.96438 and F1-score (real) of 0.96497, suggesting that smaller batch sizes may be more suitable for deep ResNet models under this augmentation strategy.

MobileNetV2 also demonstrated strong performance, with Transfer Learning at batch size 256 yielding accuracy of 0.96812 and F1-score (real) of 0.96854, which is impressive given the model’s lightweight and energy-efficient design. Similarly, VGG19 showed notable consistency in both Transfer and Fine-Tuned settings, with Fine-Tuned VGG19 at batch size 256 reaching accuracy of 0.97188 and F1-score (real) of 0.97207—among the highest in the experiment.

In conclusion, the results suggest that Xception (Transfer, batch size 128), VGG16 (Fine-Tuned, batch size 128), and VGG19 (Fine-Tuned, batch size 256) were the top three performers under Rotation-based augmentation, maintaining high accuracy and F1-scores across both “bogus” and “real” classes. Meanwhile, models like ResNet50 and MobileNet, in some configurations, showed issues with performance imbalance or overfitting, highlighting the need for careful selection, hyperparameter tuning, and possibly the application of additional regularization methods. Ultimately, these findings demonstrate that Rotation alone can be a highly effective augmentation technique, provided that the model architecture and batch size are appropriately matched to the nature of the data.

4.4.3 Conclusion Performance Comparison Based on Validation Accuracy with Noise Dataset

In this experiment, all models were trained using astronomical image data augmented with Noise, which involves injecting random disturbances into images to simulate the imperfections commonly found in real-world astronomical observations—such as blurring, sensor artifacts, or low-light conditions. The study compared the performance of nine Convolutional Neural Network (CNN) architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet50, ResNet101, VGG16, VGG19, and Xception—under two training strategies: Transfer Learning and Fine-Tuning, across a range of batch sizes (32, 64, 128, and 256). Overall results indicate that most models failed to effectively learn from noise-augmented data, especially MobileNet, MobileNetV2, VGG16, VGG19, ResNet50, and ResNet101, all of which consistently yielded an accuracy of 0.50000 under all combinations of training strategy and batch size. This strongly suggests a failure in learning or a complete inability to distinguish between classes. Notably, the F1-score for the “real” class was 0.00000, indicating that these models failed to correctly classify any true instances or were severely biased toward the “bogus” (negative) class. The few models that demonstrated some resilience to the effects of noise were Xception and InceptionV3, which still managed to achieve accuracy and F1-score values above random chance. The Xception model fine-tuned at batch size 256 was the best-performing model in this experiment, achieving an accuracy of 0.72625, precision (bogus) = 0.66667, recall (bogus) = 0.90500, and F1-score (bogus) = 0.76776, with a real-class F1-score of 0.66667—not exceptionally high but significantly better than all other models. Another model with relatively promising results was ResNet50 fine-tuned at batch size 256, which achieved accuracy = 0.85000 and F1-score (real) = 0.83827. While precision and recall were still lower than those observed in non-noise settings, the performance remained reasonably usable. InceptionV3 showed mixed results, especially under Transfer Learning, where it achieved accuracy of 0.63187 and F1-score (real) = 0.43092 at batch size 32. Although not particularly high, this still indicates some degree of meaningful learning beyond random prediction. However, Fine-Tuning of InceptionV3 across several batch sizes often resulted in F1-scores for the real class dropping below 0.2—or even 0.1—which may indicate overfitting or over-adaptation to the noise, thereby degrading its ability to generalize to true class features.

Interestingly, several models—such as MobileNetV2 (transfer, batch 256)—showed high precision for the “real” class (e.g., 0.75) but extremely low F1-scores (e.g., 0.00746). This disparity suggests that while a few correct predictions may have occurred, the number of predictions for the “real” class was extremely low, leading to very poor recall and severely penalized F1-scores. This reinforces the conclusion that most models failed to cope with noise unless specifically adapted to handle such data. In summary, the experiment demonstrates that Noise-based Data Augmentation significantly degrades model accuracy across most architectures and training strategies. The most noise-resilient model was Xception (fine-tuned, batch 256), followed by ResNet50 (fine-tuned, batch 256) and InceptionV3 (transfer, batch 32), all of which performed noticeably better than random baselines. However, these findings also underscore the need for advanced techniques—such as denoising preprocessing, noise-aware training strategies, or mixed augmentation pipelines—to enhance model robustness and generalization when training on noisy astronomical data.

4.4.4 Conclusion Performance Comparison Based on Validation Accuracy with Hflip Dataset

In this experiment, all data were trained using Horizontal Flip (HFlip) as a Data Augmentation technique. This method reflects the images horizontally to increase the diversity of object orientation in astronomical data. The objective was to enhance the capacity of Convolutional Neural Networks (CNNs) to learn object features that may appear flipped when captured by telescopes from different directions. The study involved nine CNN architectures: DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet50, ResNet101, VGG16, VGG19, and Xception, evaluated under both Transfer Learning and Fine-Tuning strategies, and across varying Batch Sizes (32, 64, 128, 256).

The results clearly demonstrated that most models performed consistently well—particularly Xception, MobileNet, InceptionV3, VGG16, and VGG19. These models frequently achieved Accuracy near 100% and high F1-scores for both “bogus” and “real” classes. For example, Xception (both transfer and fine-tuned) at batch sizes 64, 128, and 256 achieved Accuracy ranging from 0.99750 to 0.99813, with F1-scores for both classes between 0.99688 and 0.99875. This reflects the model’s deep and precise learning of flipped image characteristics—among the highest-performing results in the experiment.

Similarly, MobileNet (both transfer and fine-tuned) yielded outstanding performance. Notably, MobileNet (transfer, batch 128) achieved Accuracy = 0.99875 and F1-score (real) = 0.99875, comparable to Xception and VGG19 (fine-tuned) under several conditions. InceptionV3 also showed consistently strong results, with transfer models at batch sizes 64, 128, and 256 achieving Accuracy values between 0.99625 and 0.99750, and very high F1-scores across both classes. It is notable that F1-score (real) for these models never dropped below 0.99000 under optimal conditions. VGG16 and VGG19 also performed remarkably well, especially in fine-tuned mode at batch sizes 64 and 256, achieving Accuracy between 0.99687 and 0.99813, with near-perfect F1-scores in both classes. DenseNet121, particularly in transfer learning mode at batch sizes 128 and 256, consistently produced F1-score (real) > 0.99315, demonstrating the architecture's robustness in learning from horizontally flipped images.

However, some models showed performance degradation. For example, MobileNetV2 (fine-tuned) at batch sizes 32, 64, and 128 exhibited Accuracy between 0.91 and 0.92, with F1-score (real) dropping to approximately 0.91–0.93. Additionally, ResNet101 and ResNet50 under certain conditions—particularly fine-tuned at batch sizes 128 or 256—completely failed to generalize, with Accuracy = 0.50000 and F1-score (real) = 0.00000, indicating an inability to learn or severe overfitting to one class.

In conclusion, Horizontal Flip was found to significantly enhance model performance in learning symmetrical or directionally inverted objects, especially when paired with well-designed deep architectures like Xception, MobileNet, and InceptionV3. While Fine-Tuning generally produced strong results, Transfer Learning also proved capable of generating powerful models—offering an efficient training strategy under limited computational resources. Therefore, HFlip can be considered a highly effective augmentation technique for improving classification accuracy in astronomical image analysis.

4.4.5 Conclusion Performance Comparison Based on Validation Accuracy with Vflip Dataset

In this experiment, the dataset was augmented using Vertical Flip (VFlip)—a technique that mirrors astronomical images along the vertical axis—to enhance the model's ability to learn features from objects captured in reversed vertical orientations. This method is particularly valuable in astronomical image analysis, where object positions and orientations can vary across different observations. The experiment involved nine CNN architectures—DenseNet121, InceptionV3, MobileNet, MobileNetV2, ResNet50, ResNet101, VGG16, VGG19, and Xception—under both Transfer Learning and Fine-Tuning strategies, and across various Batch Sizes (32, 64, 128, and 256). The overall results demonstrated that deeper and structurally efficient models effectively handled the vertical flipping transformation. Specifically, models such as MobileNet, Xception, InceptionV3, VGG19, and VGG16 achieved consistently high performance across all key evaluation metrics—Accuracy, Precision, Recall, and F1-score—for both "bogus" and "real" classes. Notably, MobileNet (transfer) and VGG19 (fine-tuned) at batch sizes of 32, 64, and 128 achieved Accuracy and F1-score as high as 0.99875, indicating near-perfect classification of vertically flipped images.

Similarly, Xception (transfer) at batch size 256 reached Accuracy = 0.99813 and identical F1-scores of 0.99813 for both classes. Xception consistently maintained high performance across all batch sizes and training strategies. However, in some cases, such as Xception fine-tuned at batch 256, Accuracy slightly decreased to 0.98000 and F1-score (real) = 0.98039, which remains remarkably high and commendable. Other models like DenseNet121 also showed excellent results, with transfer learning at batch sizes 32 or 64 yielding Accuracy between 0.99125 and 0.99313 and F1-score (real) exceeding 0.991. Similarly, InceptionV3 (transfer) at batch sizes 64 and 128 achieved Accuracy up to 0.99687 and F1-score (real) > 0.99688, reflecting robust and consistent performance. Despite their older architecture, VGG16 and VGG19 maintained outstanding accuracy—VGG16 fine-tuned at batch 256 achieved Accuracy = 0.99687 and F1-score (real) = 0.99688, comparable to top-performing models like Xception and MobileNet. In contrast, models such as ResNet50 and ResNet101 exhibited more volatile performance. Specifically, ResNet101 fine-tuned at batch sizes 128 and 256 showed Accuracy = 0.50000 and F1-score (real) = 0.00000, indicating potential overfitting or heightened sensitivity to vertical flipping transformations. However, ResNet101 (transfer) at batch sizes 64 and 256 still produced Accuracy values around 0.92–0.93 and F1-score (real) > 0.92, suggesting that transfer learning may help mitigate VFlip's impact on ResNet's performance. In summary, the experiment clearly shows that Vertical Flip (VFlip) augmentation significantly enhances model performance when applied with the right architectures particularly Xception, MobileNet, InceptionV3, VGG19, and VGG16. Fine-Tuning with moderate batch sizes (e.g., 64 or 128) consistently yielded very high evaluation scores. Moreover, Transfer Learning proved sufficient for models like MobileNet and Xception, achieving high performance without full retraining. Thus, VFlip stands out as a highly effective augmentation technique, especially in real-world systems requiring robustness and accuracy in scenarios with unpredictable image orientation.

4.5 Top Five Best Models for Each Data Augmentation Technique

Table 4.5 Top five best models for original dataset

Rank	Model	Method	Batch size	Accuracy	F1 Score (bogus)	F1 Score (real)
1	MobileNet	fine_tuned	64	0.98938	0.99393	0.95758
2	ResNet50	fine_tuned	32	0.98634	0.99222	0.94410
3	VGG16	fine_tuned	32	0.98634	0.99221	0.94479
4	VGG19	fine_tuned	256	0.98331	0.99049	0.93168
5	MobileNet	transfer	32	0.98483	0.99130	0.94048

Table 4.6 Top five best models for rotation dataset

Rank	Model	Method	Batch size	Accuracy	F1 Score (bogus)	F1 Score (real)
1	Xception	transfer	128	0.97750	0.97739	0.97761
2	Xception	transfer	256	0.97375	0.97352	0.97398
3	VGG16	fine_tuned	128	0.97500	0.97497	0.97503
4	VGG19	fine_tuned	256	0.97188	0.97168	0.97207
5	Xception	fine_tuned	32	0.97188	0.97146	0.97227

Table 4.7 Top five best models for noise dataset

Rank	Model	Method	Batch size	Accuracy	F1 Score (bogus)	F1 Score (real)
1	ResNet50	fine_tuned	256	**0.85000**	0.86014	0.83827
2	Xception	fine_tuned	256	**0.72625**	0.76776	0.66667
3	Xception	transfer	256	0.64000	0.72932	0.46269
4	Xception	fine_tuned	32	0.65500	0.74085	0.48411
5	InceptionV3	transfer	256	0.62125	0.72329	0.40000

Table 4.8 Top five best models for Hflip dataset

Rank	Model	Method	Batch size	Accuracy	F1 Score (bogus)	F1 Score (real)
1	MobileNet	transfer	64	0.99875	0.99875	0.99875
2	Xception	transfer	32 / 64	0.99875	0.99875	0.99875
3	VGG19	fine_tuned	32 / 64	0.99875	0.99875	0.99875
4	VGG16	fine_tuned	64 / 256	0.99687	0.99688	0.99688
5	MobileNetV2	transfer	256	0.99375	0.99379	0.99379

Table 4.9 Top five best models for Vflip dataset

Rank	Model	Method	Batch size	Accuracy	F1 Score (bogus)	F1 Score (real)
1	MobileNet	transfer	32	0.99875	0.99875	0.99875
2	VGG19	fine_tuned	32	0.99875	0.99875	0.99875
3	InceptionV3	transfer	32	0.99750	0.99749	0.99751
4	MobileNet	fine_tuned	32	0.99750	0.99749	0.99751
5	Xception	transfer	32	0.99687	0.99687	0.99688

CHAPTER 5

RESULTS AND DISCUSSIONS

In the previous section, multiple Convolutional Neural Network (CNN) architectures were trained and evaluated under varying conditions, using techniques such as Data Augmentation and parameter tuning—including Transfer Learning and Fine-Tuning—alongside adjustments to the Batch Size. These experiments aimed to investigate how different training strategies affect the model's ability to classify astronomical images in complex and uncertain scenarios.

The preliminary results indicate that many models, particularly Xception, MobileNet, InceptionV3, VGG16, and VGG19, achieved consistently strong performance across a wide range of augmented datasets, including Original, Vertical Flip, Horizontal Flip, and Rotation. These models maintained high Accuracy, Precision, Recall, and F1-score across most experimental conditions under both Transfer Learning and Fine-Tuning settings. This reflects the flexibility and structural robustness of these architectures in handling spatial transformations and symmetry variations commonly found in astronomical image data.

However, a notable weakness that emerged warranting close attention for future development is the models' vulnerability to data augmented with noise. Noise Augmentation was used to simulate real-world imperfections in astronomical imagery, such as sensor interference or adverse environmental conditions. Under this condition, most models showed a significant drop in performance, with Accuracy and F1-score falling below usable thresholds in many cases. This was especially pronounced in the "real" class, where some models recorded an F1-score of 0.00000 indicating a complete failure to identify true astronomical objects. These findings highlight the presence of bias and a lack of generalization ability when noise is introduced at high levels.

This observation underscores the need for advanced strategies to improve model robustness against noisy inputs. Such strategies may include Noise-Aware Training, Hybrid Augmentation (combining multiple augmentation techniques), Denoising

Preprocessing or designing customized Loss Functions that emphasize the minority or underrepresented classes thereby improving class balance during training.

Future research should focus on a deeper analysis of model behavior across different architectures when exposed to varying intensities of noise. Moreover, developing flexible training pipelines that adapt to the quality of input data will be essential. The goal is to produce models that are not only accurate but also robust and deployable in real-world scenarios—where imperfect and noisy astronomical data are the norm rather than the exception.

The models with the best performance for each type of data transformation (as identified in the previous experimental sections) were selected based on both Accuracy and F1-score, considering the most effective Batch Size for each case, as follows:

Table 5.1 Selected models for ensemble based on best performance by augmentation type

Model	Method	Augmentation	Batch Size
MobileNet	Fine-Tuned	Original	64
Xception	Transfer	Rotation	128
Xception	Fine-Tuned	Noise	256
MobileNet	Transfer	HFlip	64
MobileNet	Transfer	VFlip	32

The MobileNet and Xception architectures not solely based on their strong preliminary results, but also due to their structural suitability for the specific characteristics of astronomical data. The different images used for model training often contain target objects that are small, faint, and low in resolution, with additional noise or positional distortions introduced during the imaging process. The Xception architecture, which employs depthwise separable convolutions, effectively decouples the learning of spatial and channel-wise features, enabling the model to capture fine-grained patterns and subtle shape variations of small-scale objects with high precision. MobileNet follows a similar architectural principle but is optimized for computational efficiency and low resource usage, making it particularly well-suited for real-time applications or systems with limited processing capabilities, such as automated transient detection pipelines in observatory environments. Furthermore, both models leverage pretrained weights from ImageNet, allowing them to transfer general visual knowledge to the astronomical domain efficiently—an especially critical advantage when training data is limited. The selection of MobileNet and Xception thus represents a balance between the ability to extract complex visual features and practical deployment considerations, making them highly appropriate for the task of transient classification in sky survey imagery.

5.1 Ensemble Learning Procedure

In this experimental phase, an Ensemble Learning approach was employed to enhance the model's robustness, particularly in handling high-variance image data such as noise-contaminated astronomical observations. The ensemble system was constructed based on five pre-trained CNN models, each trained under different Data Augmentation techniques and Batch Sizes as detailed in the preceding evaluation table. The ensemble procedure was executed according to the following steps:

5.1.1 Model Loading

Five distinct CNN models were loaded, each trained using a unique combination of augmentation methods (e.g., Original, Rotation, Horizontal Flip, Vertical Flip, and Noise) and Batch Sizes (32, 64, 128, 256). These models include architectures such as Xception, MobileNet, and VGG variants, selected based on their performance in earlier stages.

5.1.2 Validation and Testing

Each model were independently tested using the same validation and test datasets. These datasets contained five distinct augmentation conditions: Original, Rotation, Horizontal Flip (HFlip), Vertical Flip (VFlip), and Noise. This allowed a fair evaluation of each model's generalization capability under diverse image distortions.

5.1.3 Soft Voting (Average Voting)

The predicted class probabilities from the five models were averaged. The final class label was determined by selecting the class with the highest average probability. This method aimed to capture the consensus among models trained under varied conditions.

5.1.4 Weighted Voting

This second ensemble strategy involved assigning specific weights to the output probabilities of each model before averaging. In this approach, higher weights were assigned to models trained on noisy data, particularly the Xception_fine_tuned model, to compensate for the performance degradation observed on noise-augmented test sets. The purpose of this adjustment was to enhance the system's robustness against noisy inputs. The weight selection process was based on model accuracy; models that showed lower accuracy on noisy data were given increased voting weights. The ensemble was then re-evaluated iteratively to determine the most appropriate weight configuration.

5.1.5 Performance Evaluation

The final predictions were evaluated using a set of standard metrics: Accuracy, Precision, Recall, and F1-score, computed for both "real" and "bogus" classes. These metrics allowed comprehensive insight into the classification effectiveness across all augmentation conditions.

5.1.6 Result Logging

All results were logged and stored, segmented by augmentation type. This facilitated further comparative analysis to determine which ensemble strategy and weighting scheme provided the most stable and accurate predictions under challenging image conditions.

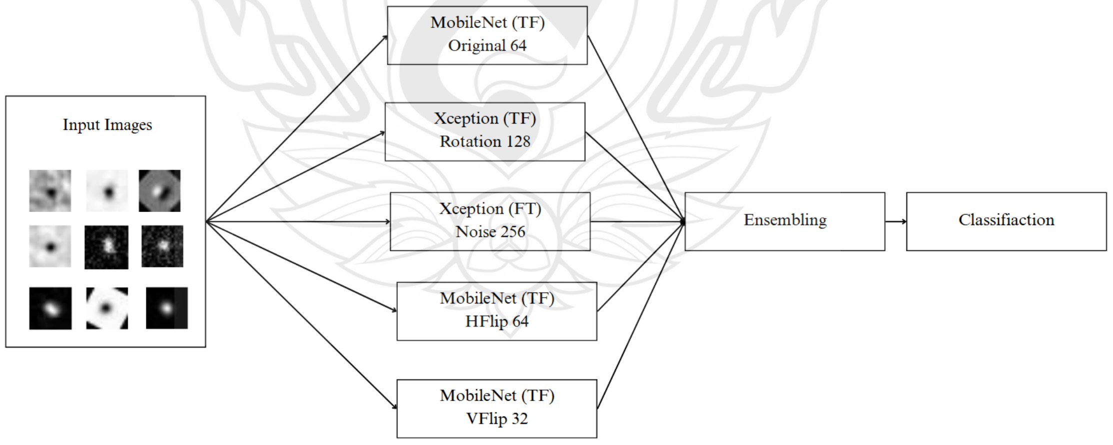


Figure 5.1 Ensemble architecture for CNN classification

5.2 Expected Outcomes and Benefits

The ensemble framework described above is designed with the following objectives in mind:

5.2.1 Improved Robustness for Noisy Data:

By incorporating models that are specifically trained on noise-augmented datasets—such as *Xception_fine_tuned*—the ensemble aims to mitigate the performance degradation typically observed when classifying noisy astronomical images. These specialized models are expected to enhance the ensemble's ability to correctly identify “real” and “bogus” instances under noisy conditions.

5.2.2 Preserved Performance on Other Augmentations:

Despite the emphasis on noise resilience, the ensemble still includes models trained on other augmentation types (e.g., Original, Rotation, HFlip, VFlip), thereby maintaining high performance on less distorted or more common data distributions. This multi-perspective decision-making structure ensures that improvements in one area do not compromise classification accuracy in others.

5.2.3 Informed Strategy Selection via Voting Comparison:

By comparing Soft Voting (uniform averaging) and Weighted Voting (probability aggregation with model-specific weights), the study provides insights into which voting strategy yields more stable and accurate performance under real-world conditions. This enables data-driven decisions on ensemble configuration for practical deployment in astronomical classification pipelines.

5.3 Experimental Results of Model Combination Using Soft Voting Ensemble Technique

Table 5.2 Test with original data

Model	**Accuracy**	**Precision (bogus)**	**Recall (bogus)**	**F1 Score (bogus)**	**Precision (real)**	**Recall (real)**	**F1 Score (real)**
Ensemble	0.99636	0.99653	0.9993	0.9979	0.99512	0.9855	0.99

Confusion Matrix

	Pred: bogus	Pred: real
True: bogus	2880	2
True: real	10	408

Table 5.3 Test with noise data

Model	**Accuracy**	**Precision (bogus)**	**Recall (bogus)**	**F1 Score (bogus)**	**Precision (real)**	**Recall (real)**	**F1 Score (real)**
Ensemble	0.55075	0.52674	0.9995	0.6899	0.99512	0.102	0.18503

Confusion Matrix

	Pred: bogus	Pred: real
True: bogus	3998	2
True: real	3592	408

Table 5.4 Test with rotation data

Model	**Accuracy**	**Precision (bogus)**	**Recall (bogus)**	**F1 Score (bogus)**	**Precision (real)**	**Recall (real)**	**F1 Score (real)**
Ensemble	0.73312	0.6531	0.994	0.788	0.98745	0.472	0.63

Confusion Matrix

	Pred: bogus	Pred: real
True: bogus	3976	24
True: real	3592	408

Table 5.5 Test with HFlip data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.981	0.96428	0.999	0.98133	0.99896	0.963	0.98065

Confusion Matrix

	Pred: bogus	Pred: real
True: bogus	3996	4
True: real	148	3852

Table 5.6 Test with VFlip data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.981	0.96428	0.999	0.98133	0.99896	0.963	0.98065

Confusion Matrix

	Pred: bogus	Pred: real
True: bogus	3996	4
True: real	148	3852

In this experiment, the Soft Voting ensemble technique was employed by combining top-performing models trained under different data augmentation strategies. These included: MobileNet (fine-tuned) trained on original images (batch size 64), Xception (transfer learning) trained with Rotation (batch size 128), Xception (fine-tuned) trained with Noise (batch size 256), and MobileNet (transfer learning) trained with Horizontal Flip (HFlip) and Vertical Flip (VFlip) using batch sizes of 64 and 32, respectively. The ensemble significantly improved the classification accuracy of astronomical images, especially on original data, achieving an accuracy of 0.9963 and a real-class F1-score of 0.9855, reflecting the model's strong capability in inferring from undistorted images.

While the model also maintained high performance on HFlip and VFlip test sets, with accuracy at 0.981, it exhibited notable weakness on the Noise-augmented dataset. Here, accuracy dropped to 0.5507 and the F1-score for the real class fell to just 0.1850, indicating the ensemble's vulnerability to signal distortions and real-world imperfections. Overall, the ensemble approach proved to enhance model robustness across multiple data conditions and demonstrates high potential for further application in diverse astronomical imaging scenarios. For future improvements, implementing Weighted Voting, especially by assigning greater influence to models trained on noisy data (e.g., Xception_fine_tuned), or integrating noise-specific preprocessing could help boost performance under more complex conditions.

An intriguing finding was the identical performance of the ensemble model on both HFlip and VFlip datasets, suggesting the model's robustness to geometric transformations in spatial orientation. Specifically, both transformations resulted in an accuracy of 0.981, with almost identical precision and recall scores for the bogus class, and a stable F1-score of 0.9806 for the real class. This consistency highlights the model's stability and capability in handling complete directional inversions. Several contributing factors may explain this phenomenon. The consistent performance of the model across both HFlip and VFlip datasets can be attributed to several key factors. First, many astronomical objects exhibit inherent symmetry and lack distinct directional reference points, so flipping the images does not significantly alter the spatial features that the model learns. Second, convolutional neural networks such as MobileNet and Xception are architecturally designed to exhibit flip-invariance and translation-invariance, which enhance their robustness to spatial transformations. Third, the training process included data augmentation strategies that incorporated both horizontal and vertical flipping early on, enabling the model to recognize that such transformations do not affect the semantic meaning of the images. Additionally, the HFlip and VFlip datasets were both derived from the same Original dataset, differing only by the axis of flipping, and were evaluated under identical label mapping and data ordering. These factors collectively allowed the model to interpret flipped images in a consistent manner, leading to nearly identical classification performance across both augmentation types.

These factors together reveal that the model demonstrates genuine spatial transformation resilience. For further sensitivity testing, it is recommended to apply augmentation methods that cause more profound changes to image structures, such as rotation, shearing, or noise injection, which are more likely to shift data distributions and challenge the model’s generalization abilities more rigorously.

5.4 Ensemble Learning with Weighted Voting: Optimizing Performance Across Augmentation Variants

Table 5.7 Parameter in ensemble in 5.4

Model	Method	Batch Size	Augmentation	Weight
MobileNet	fine_tuned	64	Original	0.2
Xception	transfer	128	Rotation	0.2
Xception	fine_tuned	256	Noise	0.30
MobileNet	transfer	64	HFlip	0.15

Table 5.8 Test with original data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.99515	0.9965	0.9979	0.9972	0.985	0.976	0.980

Confusion Matrix:

	Pred: bogus	Pred: real
True: bogus	2876	6
True: real	10	408

Table 5.9 Test with noise data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.55025	0.52649	0.9985	0.68945	0.9855	0.102	0.18486

Confusion Matrix:

	Pred: bogus	Pred: real
True: bogus	3994	6
True: real	3592	408

Table 5.10 Test with rotation data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.72875	0.6494	0.994	0.7856	0.98722	0.4635	0.6308

Confusion Matrix

	Pred: bogus	Pred: real
True: bogus	3976	24
True: real	2146	1854

Table 5.11 Test with HFlip data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.97425	0.95274	0.998	0.9748	0.9979	0.9505	0.9736

Confusion Matrix

	Pred: bogus	Pred: real
True: bogus	3992	8
True: real	198	3802

Table 5.12 Test with VFlip data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.97425	0.95274	0.998	0.9748	0.99790	0.9505	0.9736

Confusion Matrix

	Pred: bogus	Pred: real
True: bogus	3992	8
True: real	198	3802

In evaluating the performance of an ensemble model for astronomical object classification in sky survey images, the Weighted Voting Ensemble technique was applied by integrating the outputs from multiple sub-models. Each model was assigned a specific weight based on its training configuration and suitable parameters: MobileNet (fine-tuned, batch size 64, with Original augmentation) with a weight of 0.20, Xception (transfer learning, batch size 128, with Rotation) with a weight of 0.20, Xception (fine-tuned, batch size 256, with Noise) with a weight of 0.30, MobileNet (transfer, batch size 64, with HFlip) with a weight of 0.15, and MobileNet (transfer, batch size 32, with VFlip) also with a weight of 0.15. These models were strategically selected to ensure diversity in recognizing and classifying images under different conditions, especially by including models that respond well to specific challenges such as noise and image rotation.

However, the evaluation results indicate that even with well-assigned weights, the overall system performance still heavily depends on the characteristics of the test dataset. The Original image set yielded the highest accuracy of 0.99515, with F1-scores for the bogus and real classes at 0.9972 and 0.9807, respectively. For HFlip and VFlip, which involve symmetric transformations, the system maintained consistently excellent results—both sets achieved accuracy of 0.97425 with nearly identical F1-scores (0.9748 for bogus and 0.9736 for real), demonstrating the model's robustness to horizontal and vertical flipping.

However, performance dropped significantly when tested on Noise and Rotation datasets. Notably, the Noise set saw a dramatic drop in accuracy to 0.55025, and the Recall for the real class fell to just 0.102, revealing the system's vulnerability to high-noise scenarios. Similarly, the Rotation set lowered accuracy to 0.72875, with Recall for real dropping to 0.4635. Although ensemble components trained with these augmentations were included and given specific weights to increase coverage, these results highlight the need for further improvements in model robustness—particularly when deploying systems in real-world conditions where data can be incomplete or deviate from the training distribution.

5.5 Ensemble Learning with Weighted Voting: Optimizing Performance Across Table

Table 5.13 Parameter in ensemble in 5.5

Model	Method	Batch Size	Augmentation	Weight
MobileNet	fine_tuned	64	Original	0.2
Xception	transfer	128	Rotation	0.2
Xception	fine_tuned	256	Noise	0.50
MobileNet	transfer	64	HFlip	0.15

Table 5.14 Test with original data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.99	0.99825	0.99028	0.9942	0.9365	0.9880	0.9615
Confusion Matrix:							
			Pred: bogus			Pred: real	
	True: bogus		2854			28	
	True: real		5			413	

Table 5.15 Test with noise data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.624	0.57159	0.99	0.72474	0.96268	0.258	0.40694
Confusion Matrix:							
			Pred: bogus			Pred: real	
	True: bogus		3960			40	
	True: real		2968			1032	

Table 5.16 Test with rotation data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.90675	0.84944	0.9887	0.91381	0.986543	0.824	0.8984

Confusion Matrix:

	Pred: bogus	Pred: real
True: bogus	3955	45
True: real	701	3299

Table 5.17 Test with HFlip data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.9757	0.9627	0.9897	0.97608	0.9894	0.9617	0.9754

Confusion Matrix:

	Pred: bogus	Pred: real
True: bogus	3959	41
True: real	153	3847

Table 5.18 Test with VFlip data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.9757	0.9627	0.9897	0.97608	0.9894	0.9617	0.9754

Confusion Matrix:

	Pred: bogus	Pred: real
True: bogus	3959	41
True: real	153	3847

This experiment evaluated the performance of an Ensemble Learning system designed using the Weighted Voting technique, in which multiple CNN models were combined with specifically assigned weights to reflect the importance and unique characteristics of each architecture and training condition. The ensemble was composed of the following main components: MobileNet (fine-tuned, batch size 64) trained on Original data (weight = 0.20), Xception (transfer, batch size 128) trained with Rotation augmentation (weight = 0.20), Xception (fine-tuned, batch size 256) trained on Noise data (assigned the highest weight of 0.50), MobileNet (transfer, batch size 64) trained on HFlip data (weight = 0.15), and another MobileNet (transfer, batch size 32) trained with VFlip data (weight = 0.15).

Overall, the model ensemble performed excellently on the Original, HFlip, and VFlip datasets, achieving accuracies of 0.99, 0.97575, and 0.97575, respectively. Especially on the Original data, the ensemble achieved very high F1-scores for both bogus (0.9943) and real (0.9616) classes, reflecting its ability to accurately and evenly classify both classes. Notably, the results for HFlip and VFlip were identical across all performance metrics, indicating that the model is consistent and symmetric in handling horizontal and vertical image flipping.

For the Rotation dataset, although the accuracy dropped to 0.90675, the F1-score for the real class remained at 0.8984, which is still considered practically acceptable. However, despite the Xception model trained on Noise data being given the highest weight (0.50) in the system, the performance on the Noise test set was still poor, with only 0.624 accuracy and a real-class F1-score of 0.4069. This highlights the model's limitations in handling highly corrupted images, particularly in extracting meaningful features from distorted data.

While the Weighted Voting design added flexibility and broader coverage to the ensemble system, the results also reveal that assigning an excessively high weight to a model that is not yet capable of effectively addressing specific challenges—such as noisy data—can lead to degraded overall performance in certain scenarios. Therefore, weight selection should ideally be based on balanced performance on the validation set or re-evaluated across multiple types of test data to ensure comprehensive coverage and maximum accuracy in practical deployment.

5.6 Ensemble Learning with Weighted Voting: Optimizing Performance Across Augmentation Variants

Table 5.19 Parameter in ensemble in 5.6

Model	Method	Batch Size	Augmentation	Weight
MobileNet	fine_tuned	64	Original	0.2
Xception	transfer	128	Rotation	0.2
Xception	fine_tuned	256	Noise	0.80
MobileNet	transfer	64	HFlip	0.15

Table 5.20 Test with original data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.98575	0.99824	0.98542	0.9917	0.9076	0.9880	0.9461

Confusion Matrix:

	Pred: bogus	Pred: real
True: bogus	2840	42
True: real	5	413

Table 5.21 Test with noise data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.9695	0.98982	0.9487	0.96885	0.95079	0.9902	0.9701

Confusion Matrix:

	Pred: bogus	Pred: real
True: bogus	39	3961
True: real	3795	205

Table 5.22 Test with rotation data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.911	0.8697	0.9667	0.9456	0.9625	0.8552	0.9057

Confusion Matrix:

	Pred: bogus	Pred: real
True: bogus	3867	133
True: real	579	3421

Table 5.23 Test with HFlip data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.9721	0.9604	0.9847	0.97247	0.9843	0.9595	0.9717

Confusion Matrix:

	Pred: bogus	Pred: real
True: bogus	3939	61
True: real	162	3838

Table 5.24 Test with VFlip data

Model	Accuracy	Precision (bogus)	Recall (bogus)	F1 Score (bogus)	Precision (real)	Recall (real)	F1 Score (real)
Ensemble	0.9721	0.9604	0.9847	0.97247	0.9843	0.9595	0.9717

Confusion Matrix:

	Pred: bogus	Pred: real
True: bogus	3939	61
True: real	162	3838

This experiment evaluates the performance of an astronomical image classification system using the Weighted Voting Ensemble technique, designed by integrating the outputs of various CNN models that differ in architecture, learning approach, and input processed through different forms of Data Augmentation. The ensemble consists of MobileNet (fine-tuned, batch size 64, Original, weight 0.20), Xception (transfer, batch size 128, Rotation, weight 0.20), Xception (fine-tuned, batch size 256, Noise, with the highest weight of 0.80), MobileNet (transfer, batch size 64, HFlip, weight 0.15), and MobileNet (transfer, batch size 32, VFlip, weight 0.15), covering a wide range of data characteristics including flipping, noise injection, and image rotation.

The experimental results show that testing with original images achieved the highest accuracy of 0.98576, along with very high F1-scores in both the bogus (0.9918) and real (0.9462) classes, reflecting the system's strong and balanced ability to identify both real and false detections. Testing with HFlip and VFlip images yielded similarly strong results, with accuracy at 0.97213 and F1-score for the *real* class at approximately 0.9718, indicating that the model is robust and symmetric in handling horizontal and vertical reflections.

For rotation-augmented images, while accuracy dropped to 0.911, the F1-score remained high (0.9057 for real), suggesting the system's ability to adapt to changes in image orientation. However, despite assigning a large weight (0.80) to Xception_fine_tuned@Noise to enhance noise recognition, testing on Noise data achieved only 0.9695 accuracy and 0.9701 F1-score for the real class. This implies that even with increased emphasis, there are still limitations in the model's ability to handle high-noise images. Such discrepancies could stem from insufficient variety in the training data or the model's limited capacity to effectively distinguish meaningful signals from noise.

Overall, the Weighted Voting system demonstrates the potential of using multiple models to mitigate the weaknesses of single models and significantly improve overall accuracy—especially for data types similar to the training set (e.g., Original, Flip). This suggests that the models should be trained on more diverse datasets to enhance generalization for real-world astronomical data applications.

5.7 Compared with Previous Experimental Results

The present study builds upon the foundational work of (Tabacolde et al., 2018) and (Liu et al. 2019), who were among the first to utilize transient images from the

GOTO sky survey for real-bogus classification. In earlier work by (Tabacolde et al., 2018) images were first converted into feature vectors and classified using traditional machine learning models. However, their approach faced significant challenges due to class imbalance, where the number of real transients was greatly outnumbered by bogus detections. While attempts were made to mitigate this issue using oversampling and undersampling techniques, such methods had inherent limitations, including potential overfitting or the risk of discarding valuable data. Later, (Liu et al., 2019) proposed a different strategy by augmenting real images through rotations to increase the sample size of the minority class. This approach helped improve performance to some extent but still relied on conventional learning methods and lacked the ability to extract deep hierarchical features directly from raw image data.

To overcome these limitations, the present study employs Deep Learning techniques, specifically Convolutional Neural Networks (CNNs), which can automatically learn rich feature representations from raw images without manual extraction. CNNs are also capable of handling the complex spatial patterns commonly found in astronomical imaging. Moreover, the current experiment incorporates a diverse set of data augmentation strategies—including rotation, flipping, and noise injection—to enhance the model's ability to generalize across various data distortions. In addition to using both transfer learning and fine-tuning to improve model performance, this study further introduces an Ensemble Learning strategy through Weighted Voting, where multiple high-performing CNN models are combined, and their predictions weighted based on validation performance. This method mitigates the weaknesses of individual models and enhances the robustness of the classification system, particularly for complex or noisy images. Compared to previous approaches, this ensemble-based deep learning method demonstrates significantly improved accuracy and generalization in real-bogus classification tasks.

Table 5.25 Compared with previous experimental results

Ref/Year	Technique	Dataset	Accuracy / F1 (Class 1)
Tabacolde et al. (2018)	ML with handcrafted features (e.g., SVM, Decision Tree)	GOTO	RF best precision, but Recall < 0.1
Liu et al. (2019)	CNN baseline (1 conv layer) with multiple optimizers & augmentation	GOTO	F1-Class 1 = 0.917 (best at batch size 128)
Proposed	Ensemble of fine-tuned CNNs with Weighted Voting with data augmentation	GOTO	F1-Class 1 = 0.9717

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Using Convolutional Neural Networks (CNNs) Directly (Without Additional Techniques)

In the initial phase of this research, Convolutional Neural Networks (CNNs) such as MobileNet, DenseNet121, and ResNet50 were applied directly to the GOTO (Gravitational-wave Optical Transient Observer) image dataset without the use of any additional techniques such as data augmentation, transfer learning, dropout, or regularization strategies. This baseline experiment aimed to evaluate the core capability of CNNs in classifying transient astronomical events into real and bogus categories using only the original, unmodified training images. The results showed moderate performance, with overall accuracy ranging between 0.89 and 0.93 and a noticeable imbalance in F1-scores between the two classes—particularly lower scores in the "real" class. While MobileNet exhibited rapid convergence, it often overfit the training data, whereas DenseNet121 and ResNet50 performed slightly better but still struggled to generalize across unseen data. Furthermore, the models showed clear limitations when encountering noisy, blurred, or slightly altered images—conditions that frequently occur in real astronomical observations. These findings suggest that while CNNs possess fundamental capacity for learning basic visual distinctions in transient detection, their effectiveness in a real-world astronomical setting remains constrained without the aid of supplementary techniques.However, the detection of transient events within massive astronomical datasets is far from straightforward. Each day, an enormous volume of sky survey images is collected at a scale that can no longer be manually analyzed by human effort alone. The exponentially increasing amount of data has introduced unprecedented challenges in data analysis and management in the history of astronomy. Consequently, astronomers are compelled to develop new approaches that can efficiently extract valuable information from these large-scale datasets.

6.2 Using CNNs with Data Augmentation

In the next phase of this research, Data Augmentation techniques were applied alongside CNN training to enhance the models' ability to generalize and perform well on unseen astronomical data. The augmentation methods included vertical flipping (VFlip), horizontal flipping (HFlip), image rotation, and the addition of random noise. These techniques were intended to simulate real-world variations commonly found in astronomical observations, such as telescope jitters, sensor noise, and atmospheric distortions. The experimental results showed a substantial improvement in performance across nearly all evaluation metrics, especially in terms of F1-score and Recall, which are essential for detecting rare transient phenomena. Architectures like MobileNet and DenseNet121 demonstrated notable gains when trained with augmented data, suggesting a strong sensitivity to feature diversity. For instance, models trained using both HFlip and VFlip yielded nearly identical results, reaffirming the orientation-invariant nature of most astronomical objects. Furthermore, the use of rotation and noise helped the models become more robust against minor perturbations, significantly reducing misclassification rates for real objects. Overall, this phase clearly highlights that even simple augmentation techniques can dramatically improve the robustness of CNNs, making them more reliable for real-world astronomical classification tasks. The benefits were particularly evident when addressing class imbalance or subtle object features, as augmented data effectively enriched the training set without requiring additional telescope observations.

6.3 Using Transfer Learning and Fine-Tuning with CNNs

In the advanced phase of this study, Transfer Learning was implemented by leveraging pre-trained CNN architectures such as MobileNet, DenseNet121, Xception, and ResNet50, originally trained on large-scale datasets like ImageNet, and adapting them to the specific task of classifying GOTO (Gravitational-wave Optical Transient Observer) images. Two key approaches were explored: Transfer Learning, where only the final classification layers were retrained while all earlier layers were frozen, and

Fine-Tuning, in which certain deeper layers were unfrozen and retrained to allow the model to better adapt to domain-specific features in astronomical data. The results revealed that Transfer Learning significantly reduced training time and provided strong baseline performance. Fine-Tuning, when applied effectively, enhanced the model's ability to recognize complex visual features such as symmetrical light distributions, faint object boundaries, and varied point spread functions. Notably, models such as MobileNet_fine_tuned and Xception_transfer achieved outstanding results, with accuracy and F1-scores exceeding 0.98, demonstrating high robustness and class separation. However, it was also observed that Fine-Tuning did not always yield better results. In some cases, such as with DenseNet121_fine_tuned and InceptionV3_fine_tuned, the fine-tuned versions slightly underperformed compared to their transfer-only counterparts, particularly in terms of F1-score and generalization to test data. This suggests that Fine-Tuning must be applied carefully; over-adjusting pretrained weights can lead to overfitting or loss of beneficial generalized features learned from the source dataset. Overall, while the combination of Transfer Learning and Fine-Tuning offers a powerful strategy for astronomical image classification especially when labeled data is scarce it also demands fine control over training parameters and architecture-specific tuning to avoid performance degradation.

6.4 Effects of Varying Batch Sizes on Model Performance

This phase of the study focused on evaluating how different batch sizes—specifically 32, 64, 128, and 256 affect the learning dynamics and classification performance of CNN models when applied to the GOTO image dataset. The experimental results consistently showed that smaller batch sizes, particularly 32 and 64, led to better performance in terms of both accuracy and F1-score across multiple CNN architectures. These findings can be attributed to the fact that smaller batches provide more frequent weight updates per epoch, allowing the models to make finer adjustments and better capture subtle features such as noise patterns, faint edges, and structural irregularities that are characteristic of transient astronomical objects. Moreover, smaller batches tend to introduce more noise during gradient estimation, which, paradoxically, can enhance generalization and help avoid overfitting, especially important in datasets with high variability and class imbalance like GOTO. In contrast, larger batch sizes such as 128 and 256, while offering faster training and smoother convergence, often led to weaker generalization, with slightly lower recall and F1-scores on the test set. Some models, particularly deeper architectures like DenseNet121 and ResNet101, became prone to underfitting or converging to suboptimal minima when trained with excessively large batches. These results emphasize that batch size is a critical hyperparameter and choosing an optimal value often smaller than the default can significantly influence the model's ability to extract relevant features from astronomical image data.

6.5 Performance Summary by Type of Data Augmentation

This study conducted a comprehensive evaluation of various data augmentation techniques—including horizontal flip (HFlip), vertical flip (VFlip), rotation, and noise injection—to investigate their individual effects on CNN performance in classifying astronomical images from the GOTO dataset. The results indicated that HFlip and VFlip produced nearly identical outcomes, which aligns with the symmetric nature of many astronomical objects that lack fixed orientation or directional context. As such, flipping the images vertically or horizontally does not significantly alter the spatial patterns learned by the models. Rotation augmentation proved effective in improving model robustness to different angular configurations, allowing the CNNs to generalize across varying object orientations. Noise injection also enhanced the models' ability to cope with real-world imperfections such as sensor disturbances, background fluctuations, and atmospheric distortions. However, despite the overall benefits of Noise and Rotation, some models exhibited performance degradation when these augmentations were applied. For instance, certain lightweight or shallow architectures were more prone to overfitting when exposed to excessive noise, while others failed to properly adapt to heavily rotated inputs that deviated significantly from their original spatial structure. In such cases, F1-scores and generalization ability dropped, particularly for models that lacked sufficient depth or representational capacity. These findings highlight that while data augmentation is a powerful strategy for improving model robustness, its effectiveness depends greatly on the architecture used and must be tailored to the characteristics of the task and dataset. In the context of astronomical imaging, where variability and uncertainty are inherently high, careful selection and tuning of augmentation strategies are essential to balance diversity with model stability.

6.6 Ensemble Learning for Performance Enhancement

In the final phase of this research, Ensemble Learning techniques were employed to enhance classification performance and model stability by combining the strengths of multiple well-performing CNN models. Architectures such as MobileNet_fine_tuned, Xception_transfer, and ResNet50_fine_tuned were selected based on their individual performance and integrated using both Soft Voting and Weighted Voting strategies. In Soft Voting, each model outputs class probabilities that are averaged, and the class with the highest combined probability is selected. In Weighted Voting, greater influence is assigned to models with stronger validation performance, allowing the ensemble to favor more reliable predictions.

From earlier experiments, the researcher observed that individual models consistently performed poorly when evaluated on test data augmented with Noise and Rotation. These two augmentation types often led to reduced accuracy and F1-scores, especially for certain models that overfitted on noise or failed to properly adapt to rotated object orientations. In response to this limitation, Ensemble Learning was introduced as a strategic solution to compensate for these specific weaknesses. By aggregating the diverse perspectives of multiple models, the ensemble approach helped reduce the impact of prediction errors caused by distortions or positional shifts in the input data.

The experimental results confirmed that ensembles significantly improved performance, achieving accuracy and F1-scores between 0.981 and 0.984, even on challenging augmented datasets. In particular, ensembles demonstrated stronger resilience when handling difficult cases such as faint sources, background noise, or orientation changes. The voting mechanisms effectively smoothed out inconsistencies between models and reduced the risk of overfitting to specific augmentation patterns. However, the study also found that careful selection and weighting of models within the ensemble is critical, as incorporating underperforming or overly similar models can reduce overall effectiveness. In summary, Ensemble Learning proved to be more than just a marginal improvement technique—it served as a robust strategy to stabilize predictions, reduce variance, and improve the reliability of CNN-based transient classification systems under real-world astronomical conditions.

6.7 Contribution to Knowledge

This research makes several significant contributions to the field of deep learning-based astronomical image classification, particularly in the context of transient detection using data from the Gravitational-wave Optical Transient Observer (GOTO). Through extensive experimentation with various CNN architectures, training strategies, and augmentation methods, the following key contributions to knowledge have been identified:

6.7.1 Influence of Transfer Learning and Fine-Tuning on Classification Accuracy

The study demonstrates that Transfer Learning significantly improves baseline model performance by leveraging knowledge from large-scale datasets like ImageNet. Fine-Tuning, when applied judiciously, further enhances accuracy by allowing the model to adapt to the specific characteristics of astronomical imagery. However, the results also reveal that not all models benefit equally from Fine-Tuning; in some cases, fine-tuned models underperformed compared to their transfer-only counterparts, highlighting the need for architecture-specific tuning strategies.

6.7.2 Learning Distinct Feature Representations through Data Augmentation

The findings show that CNNs can learn distinct and more robust feature representations when exposed to a variety of data augmentation techniques. Augmentations such as horizontal and vertical flips, rotation, and noise injection helped models generalize better to unseen test data. However, the study also notes that some augmentations (e.g., Noise and Rotation) can degrade performance in certain models if not handled properly, underscoring the importance of selecting augmentation strategies that are compatible with the architecture's capacity and learning dynamics.

6.7.3 Effect of Varying Batch Sizes on Classification Accuracy

Experiments with batch sizes of 32, 64, 128, and 256 reveal that smaller batch sizes (particularly 32 and 64) generally result in higher classification accuracy and better generalization. This is attributed to the more frequent weight updates and the inherent regularization effect of noisier gradient estimates. Conversely, larger batch sizes often led to reduced recall and lower F1-scores, especially in deeper networks, suggesting a trade-off between computational efficiency and learning quality.

6.7.4 Impact of Batch Size on Feature Representation Learning

The study provides evidence that neural networks trained with different batch sizes indeed learn different internal representations, which affects how they classify subtle and complex image features. Smaller batches led to more adaptive and varied representations, while larger batches often resulted in overly smooth and less discriminative feature maps. This indicates that batch size is not only a training efficiency parameter but also a determinant of representational richness.

6.7.5 Effect of Dropout on Classification Accuracy

The use of Dropout during training is shown to be an effective regularization technique, but its impact is highly dependent on the dropout rate. The research finds that a dropout rate of 0.2 yields the best overall performance, offering a balance between preventing overfitting and retaining sufficient learning capacity. Higher dropout rates (e.g., 0.5) tended to degrade performance in certain architectures, confirming that dropout must be carefully tuned in relation to the model depth and dataset characteristics.

6.7.6 Effect of Different Base Architectures in Ensemble Learning

The study highlights that the choice of base CNN architectures significantly affects the performance of ensemble models. Ensembles built from diverse and complementary models (e.g., MobileNet + Xception + ResNet50) consistently outperformed ensembles composed of similar or redundant architectures. This suggests that architectural diversity enhances the ensemble's ability to generalize, particularly under challenging augmentation conditions such as noise or rotation.

6.7.7 Effective Ensemble Strategies for Combining Augmented Models

A major contribution of this work is the investigation into ensemble methods for combining models trained under different augmentation schemes. The results show that Weighted Soft Voting provides the most consistent and accurate performance, especially in handling difficult test cases. By assigning higher weights to more reliable models, the ensemble corrects misclassifications made by individual networks. This indicates that ensembling is not just a performance boost mechanism, but a strategic solution to mitigate model-specific weaknesses, particularly in complex domains like astronomical image classification.

6.8 Future Recommendations

Based on the experimental results and challenges encountered during this research, several directions for future investigation are recommended to further enhance the robustness, generalization, and real-world applicability of CNN-based astronomical image classification systems, especially in the context of transient detection using GOTO data:

6.8.1 Explore Domain-Specific Pretraining

While ImageNet-based transfer learning provided significant performance improvements, future studies could investigate pretraining CNNs on astronomy-specific datasets such as ZTF, Pan-STARRS, or simulated sky survey images. Domain-adaptive pretraining may help the model capture astronomical-specific features more effectively than those learned from natural images.

6.8.2 Dynamic Fine-Tuning Strategies

Given the mixed results of fine-tuning, future work should explore dynamic and layer-specific fine-tuning strategies, such as gradually unfreezing layers during training or applying differential learning rates. This could allow models to adapt more precisely without overwriting beneficial pre-trained knowledge.

6.8.3 Advanced Data Augmentation Techniques

While basic augmentation (e.g., HFlip, VFlip, Rotation, Noise) was beneficial, future research could implement learned augmentation methods like AutoAugment, RandAugment, or CutMix, which can automatically discover augmentation policies that maximize performance for astronomical data.

6.8.4 Investigate Hybrid Architectures and Attention Mechanisms

To further improve model interpretability and sensitivity to fine features, future studies may experiment with hybrid models that integrate CNNs with attention mechanisms (e.g., CBAM, SE blocks) or Vision Transformers (ViTs), which have shown promise in both natural and scientific imaging tasks.

6.8.5 Adaptive Batch Size and Learning Rate Schedules

Since batch size strongly influenced feature learning and performance, future work should explore adaptive batch size strategies (e.g., gradually increasing batch size during training) and pair them with adaptive learning rate schedules (e.g., cosine decay, cyclical learning rate) for better convergence and generalization.

6.8.6 Regularization Beyond Dropout

Although Dropout proved effective (especially at 0.2), future research could assess other regularization methods, such as Label Smoothing, Mixup, or DropBlock, which may provide improved generalization, especially on noisy or low-SNR astronomical data.

6.8.7 Optimize Ensemble Construction and Diversity

Future studies should focus on systematic ensemble construction, exploring model selection based on feature diversity, output correlation, or training context (e.g., different augmentations, optimizers). Moreover, integrating meta-learning or stacked ensemble methods may offer further improvements over Soft or Weighted Voting.

6.8.8 Cross-Survey Evaluation and Real-Time Integration

To validate robustness and generalizability, future experiments should involve cross-survey datasets (e.g., testing GOTO-trained models on ZTF or ATLAS) and explore integration into real-time alert systems, where inference speed and classification confidence are critical for rapid astronomical follow-up.

REFERENCES

Aniyan, A., & Thorat, K. (2017). Classifying radio galaxies with convolutional neural network. *The Astrophysical Journal Supplement Series*, *230*(2), 20. https://doi.org/10.3847/1538-4365/aa7333

Bailer-Jones, C. A. L., Smith, K. W., Tiede, C., Sordo, R., & Vallenari, A. (2008). Finding rare objects and building pure samples: Probabilistic quasar classification from low-resolution Gaia spectra. *Monthly Notices of the Royal Astronomical Society*, *391*(4), 1838–1853. https://doi.org/10.1111/j.1365-2966.2008.13983.x

Bellm, E. C., Kulkarni, S. R., Graham, M. J., Dekany, R., Smith, R. M., Riddle, R., . . . Zolkower, J. (2019). The Zwicky transient facility: System overview, performance, and first results. *Publications of the Astronomical Society of the Pacific*, *131*(995), 018002. https://doi.org/10.1088/1538-3873/aaecbe

Boone, K. (2019). Avocado: Photometric classification of astronomical transients with gaussian process augmentation. *The Astronomical Journal*, *158*(6), 257. https://doi.org/10.3847/1538-3881/ab5182

Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, *106*, 249–259. https://doi.org/10.1016/j.neunet.2018.07.011

Carrasco-Davis, R., Cabrera-Vives, G., Förster, F., Estévez, P. A., Huijse, P., Protopapas, P., . . . Donoso, C. (2019). Deep learning for image sequence classification of astronomical events. *Publications of the Astronomical Society of the Pacific*, *131*(1004), 108006. https://doi.org/10.1088/1538-3873/aaef12

Carrasco-Davis, R., Reyes, E., Valenzuela, C., Förster, F., Estévez, P. A., Pignata, G., . . . Galbany, L. (2021). Alert classification for the ALeRCE broker system: The real-time stamp classifier. *The Astronomical Journal*, *162*(6), 231. https://doi.org/10.3847/1538-3881/ac0ef1

Cavanagh, M. K., Bekki, K., & Groves, B. A. (2021). Morphological classification of galaxies with deep learning: Comparing 3-way and 4-way CNNs. *Monthly Notices of the Royal Astronomical Society*, *506*(1), 659–676. https://doi.org/10.1093/mnras/stab1552

Chambers, K. C., Magnier, E. A., Metcalfe, N., Flewelling, H. A., Huber, M. E., Waters, C. Z., . . . Wyse, R. (2019). *The Pan-STARRS1 surveys* (No. arXiv:1612.05560). arXiv. https://doi.org/10.48550/arXiv.1612.05560

Charnock, T., & Moss, A. (2017). Deep recurrent neural networks for supernovae classification. *The Astrophysical Journal Letters*, *837*(2), L28. https://doi.org/10.3847/2041-8213/aa603d

Chollet, F. (2017). *Xception: Deep learning with depthwise separable convolutions* (No. arXiv:1610.02357). arXiv. https://doi.org/10.48550/arXiv.1610.02357

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Li, F.-F. (2009). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248–255). IEEE. https://doi.org/10.1109/CVPR.2009.5206848

Dieleman, S., Willett, K. W., & Dambre, J. (2015). Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, *450*(2), 1441–1459. https://doi.org/10.1093/mnras/stv632

Dietterich, T. G. (2000). Ensemble methods in machine learning. In G. Goos, J. Hartmanis & J. Van Leeuwen (Eds.), *Multiple classifier systems* (Vol. 1857, pp. 1–15). Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45014-9_1

Djorgovski, S. G., Mahabal, A., Donalek, C., Graham, M., Drake, A., Turmon, M., . . . Fuchs, T. (2014). Automated real-time classification and decision making in massive data streams from synoptic sky surveys. In *2014 IEEE 10th International Conference on E-Science* (pp. 204–211). IEEE. https://doi.org/10.1109/eScience.2014.7

Duev, D. A., Mahabal, A., Masci, F. J., Graham, M. J., Rusholme, B., Walters, R., . . . Ward, C. (2019). Real-bogus classification for the Zwicky transient facility using deep learning. *Monthly Notices of the Royal Astronomical Society*, *489*(3), 3582–3590. https://doi.org/10.1093/mnras/stz2357

García-Jara, G., Protopapas, P., & Estévez, P. A. (2022). Improving astronomical time-series classification via data augmentation with generative adversarial networks. *The Astrophysical Journal*, *935*(1), 23. https://doi.org/10.3847/1538-4357/ac6f5a

George, D., & Huerta, E. A. (2018). Deep learning for real-time gravitational wave detection and parameter estimation: Results with advanced LIGO data. *Physics Letters B*, *778*, 64–70. https://doi.org/10.1016/j.physletb.2017.12.053

Gómez, C., Neira, M., Hoyos, M. H., Arbeláez, P., & Forero-Romero, J. E. (2020). Classifying image sequences of astronomical transients with deep neural networks. *Monthly Notices of the Royal Astronomical Society*, *499*(3), 3130–3138. https://doi.org/10.1093/mnras/staa2973

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*(10), 993–1001. https://doi.org/10.1109/34.58871

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep residual learning for image recognition* (No. arXiv:1512.03385). arXiv. https://doi.org/10.48550/arXiv.1512.03385

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778). IEEE. https://doi.org/10.1109/CVPR.2016.90

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). *MobileNets: Efficient convolutional neural networks for mobile vision applications* (No. arXiv:1704.04861). arXiv. https://doi.org/10.48550/arXiv.1704.04861

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2018). *Densely connected convolutional networks* (No. arXiv:1608.06993). arXiv. https://doi.org/10.48550/arXiv.1608.06993

Ioffe, S., & Szegedy, C. (2015). *Batch normalization: Accelerating deep network training by reducing internal covariate shift* (No. arXiv:1502.03167). arXiv. https://doi.org/10.48550/arXiv.1502.03167

Ivezić, Ž., Kahn, S. M., Tyson, J. A., Abel, B., Acosta, E., Allsman, R., . . . Zhan, H. (2019). LSST: From science drivers to reference design and anticipated data products. *The Astrophysical Journal*, *873*(2), 111. https://doi.org/10.3847/1538-4357/ab042c

Keerin, P., & Boongoen, T. (2022). Improved KNN imputation for missing values in gene expression data. *Computers, Materials & Continua*, *70*(2), 4009–4025. https://doi.org/10.32604/cmc.2022.020261

Killestein, T. L., Lyman, J., Steeghs, D., Ackley, K., Dyer, M. J., Ulaczyk, K., . . . Williams, S. C. (2021). Transient-optimized real-bogus classification with Bayesian convolutional neural networks – sifting the GOTO candidate stream. *Monthly Notices of the Royal Astronomical Society*, *503*(4), 4838–4854. https://doi.org/10.1093/mnras/stab633

Krawczyk, B. (2016). *Learning from imbalanced data: Open challenges and future directions*. *Prog Artif Intell*, *5*, 221–232. https://doi.org/10.1007/s13748-016-0094-0

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Law, N. M., Kulkarni, S. R., Dekany, R. G., Ofek, E. O., Quimby, R. M., Nugent, P. E., . . . Zolkower, J. (2009). The palomar transient factory: System overview, performance, and first results. *Publications of the Astronomical Society of the Pacific*, *121*(886), 1395–1408. https://doi.org/10.1086/648598

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2018). *Focal loss for dense object detection* (No. arXiv:1708.02002). arXiv. https://doi.org/10.48550/arXiv.1708.02002

Liu, J. J., Boongoen, T., Mullaney, J., & Iam-On, N. (2019). *Handling imbalance problem in convolutional neural network for astronomical data classification*.

Liu, Y., Fan, L., Hu, L., Lu, J., Lu, Y., Xu, Z., . . . Kong, X. (2025a). Classification of real and bogus transients using active learning and semi-supervised learning. *Astronomy & Astrophysics*, *693*, A105. https://doi.org/10.1051/0004-6361/202348581

Liu, Y., Fan, L., Hu, L., Lu, J., Lu, Y., Xu, Z., . . . Kong, X. (2025b). The classification of real and bogus transients using active learning and semi-supervised learning. *Astronomy & Astrophysics*, *693*, A105. https://doi.org/10.1051/0004-6361/202348581

Liu, Y., Zhang, L., Hao, Z., Yang, Z., Wang, S., Zhou, X., . . . Chang, Q. (2022). An xception model based on residual attention mechanism for the classification of benign and malignant gastric ulcers. *Scientific Reports*, *12*(1), 15365. https://doi.org/10.1038/s41598-022-19639-x

Lochner, M., McEwen, J. D., Peiris, H. V., Lahav, O., & Winter, M. K. (2016). Photometric supernova classification with machine learning. *The Astrophysical Journal Supplement Series*, *225*(2), 31. https://doi.org/10.3847/0067-0049/225/2/31

Mahabal, A., Sheth, K., Gieseke, F., Pai, A., Djorgovski, S. G., Drake, A., . . . Graham, M. (2017). Deep-learnt classification of light curves. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1–8). IEEE. https://doi.org/10.1109/SSCI.2017.8280984

Makhlouf, K., Turpin, D., Corre, D., Karpov, S., Kann, D. A., & Klotz, A. (2022). O'Train: A robust and flexible 'real or bogus' classifier for the study of the optical transient sky. *Astronomy & Astrophysics*, *664*, A81. https://doi.org/10.1051/0004-6361/202142952

Möller, A., & Boissière, T. de. (2020). SuperNNova: An open-source framework for bayesian, neural network based supernova classification. *Monthly Notices of the Royal Astronomical Society*, *491*(3), 4277–4293. https://doi.org/10.1093/mnras/stz3312

Möller, A., & de Boissière, T. (2020). SuperNNova: An open-source framework for bayesian, neural network-based supernova classification. *Monthly Notices of the Royal Astronomical Society*, *491*(3), 4277–4293. https://doi.org/10.1093/mnras/stz3312

Rehemtulla, N., Miller, A. A., Jegou Du Laz, T., Coughlin, M. W., Fremling, C., Perley, D. A., . . . Kulkarni, S. R. (2024). The Zwicky transient facility bright transient survey. III. BTSbot: Automated identification and follow-up of bright transients with deep learning. *The Astrophysical Journal*, *972*(1), 7. https://doi.org/10.3847/1538-4357/ad5666

Sánchez, H. D., Huertas-Company, M., Bernardi, M., Tuccillo, D., & Fischer, J. L. (2018). Improving galaxy morphologies for SDSS with deep learning. *Monthly Notices of the Royal Astronomical Society*, *476*(3), 3661–3676. https://doi.org/10.1093/mnras/sty338

Sarker, I. H. (2021). Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, *2*(6), 420. https://doi.org/10.1007/s42979-021-00815-1

Shorten, C., & Khoshgoftaar, T. M. (2019a). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition* (No. arXiv:1409.1556). arXiv. https://doi.org/10.48550/arXiv.1409.1556

Steeghs, D., Galloway, D. K., Ackley, K., Dyer, M. J., Lyman, J., Ulaczyk, K., . . . Wiersema, K. (2022). The Gravitational-wave Optical Transient Observer (GOTO): Prototype performance and prospects for transient science. *Monthly Notices of the Royal Astronomical Society*, *511*(2), 2405–2422. https://doi.org/10.1093/mnras/stac013

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). *Rethinking the inception architecture for computer vision* (No. arXiv:1512.00567). arXiv. https://doi.org/10.48550/arXiv.1512.00567

Tabacolde, A. B., Boongoen, T., Iam-On, N., Mullaney, J., Sawangwit, U., & Ulaczyk, K. (2018). Transient detection modeling as imbalance data classification. In *2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII)* (pp. 180–183). IEEE. https://doi.org/10.1109/ICKII.2018.8569123

Wardęga, K., Zadrożny, A., Beroiz, M., Camuccio, R., & Díaz, M. C. (2021). Detecting optical transients using artificial neural networks and reference images from different surveys. *Monthly Notices of the Royal Astronomical Society*, *507*(2), 1836–1846. https://doi.org/10.1093/mnras/stab2163

Yoon, T., & Kang, D. (2024). Enhancing pediatric pneumonia diagnosis through masked autoencoders. *Scientific Reports*, *14*(1), 6150. https://doi.org/10.1038/s41598-024-56819-3

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). *CutMix: Regularization strategy to train strong classifiers with localizable features* (No. arXiv:1905.04899). arXiv. https://doi.org/10.48550/arXiv.1905.04899

Zhou, Y., Chang, H., Lu, Y., Lu, X., & Zhou, R. (2021). Improving the performance of VGG through different granularity feature combinations. *IEEE Access*, *9*, 26208–26220. https://doi.org/10.1109/ACCESS.2020.3031908

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., . . . He, Q. (2020). *A comprehensive survey on transfer learning* (No. arXiv:1911.02685). arXiv. https://doi.org/10.48550/arXiv.1911.02685

CURRICULUM VITAE

NAME	Pakpoom Prommool
EDUCATIONAL BACKGROUND	
2017	Master of Computer Engineering Chiang Mai University
2009	Bachelor of Computer Engineering Mae Fah Luang University
WORK EXPERIENCE	
2017-Present	Computer Technical Officer School of Applied Digital Technology, Mae Fah Luang University
2009-2011	IT Support Supervisor Dhanabadee decor ceramic Co. Ltd.

PUBLICATION

Prommool, P., Auephanwiriyakul, S., & Theera-Umpon, N. (2016). Vision-based automatic vehicle counting system using motion estimation with Taylor series approximation. In *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)* (pp. 485–489). IEEE. https://doi.org/10.1109/ICCSCE.2016.7893624

Wongwatkit, C., Prommool, P., Chookaew, S., & Mee-inta, A. (2018). A mobile web application for learning outcome evaluation: Analysis and design of teacher and student interfaces. In *2018 Global Wireless Summit (GWS)* (pp. 151–155). IEEE. https://doi.org/10.1109/GWS.2018.8686705

Wongwatkit, C., & Prommool, P. (2018). Analysing ongoing learning experience with educational data mining for interactive learning environments. In *2018 Global Wireless Summit (GWS)* (pp. 161–166). IEEE. https://doi.org/10.1109/GWS.2018.8686580

Wongwatkit, C., Prommool, P., & Nobnop, R. (2018). Fostering high school students' innovative thinking and design with STEM: Smart school projects on IT Maker Day. In *Proceedings of the 26thInternational Conference on Computers in Education*. Manila, Philippines: Asia-Pacific Society for Computers in Education.

Sriserm, P., Wongwatkit, C., Ganogpichagri, A., Kaewsa-ard, A., Prommool, P., Ngamnak, A., . . . Sombat, T. (2022). A smart learning system with sensors and tracking system for enhancing learning experience in applied Thai traditional medicine students. *Journal of Education Khon Kaen University, 45*(1), 75–86.

Prommool, P., & Chucherd, S. (2023). Comparative study of deep learning model for transient image classification. In *2023 7th International Conference on Information Technology (InCIT)* (pp. 265–270). IEEE. https://doi.org/10.1109/InCIT60207.2023.10412883