ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Baseline Performance of Pre-trained Models on Movie Genre Classification from Spectrograms

Porawat Visutsak[1], Kavin Treeraphapkajondet[2], Visaroot Sakphet[3], Wachirawit Nitinuntatip[4], Pawwinkan Satthong[5], Tanajak Tongbai[6] , Duongduen Ongrungruaeng[7], Atiwitch Juntra[8], Watcharaporn Aiamlamai[9], Issares Sungwanna[10], Prapaporn Phetrak[11], Ponrudee Netisopakul[12] and Keun Ho Ryu[13]

ABSTRACT

This study investigates the use of deep learning for classifying movie genres based on audio spectrograms. We construct a dataset of movie trailers, transform them into spectrograms, and label them by genre. Then, we utilize MATLAB's pre-trained convolutional neural networks (CNNs) for classification, comparing the performance of 9 different architectures, including MobileNet-v2, RestNet-18, DenseNet-201, Places365-GoogLeNet, VGG-16, VGG-19, Inception-RestNet-v2, Inception-v3, and NASANet-Mobile. We evaluated all models based on their ability to classify movie trailers into five genres: action, romance, drama, comedy, and thriller. Our results, based on accuracy and F1-score across genres, indicate that VGG16 achieves the highest overall performance with an accuracy of 86.27%, an F1-score of 86.69%, a recall of 86.87%, and a precision of 87.28%. This research demonstrates the potential of leveraging pre-trained CNNs, particularly VGG-16, for efficient and effective audio-based genre classification in movie trailers.

Article information:

Keywords: Movie Genre Classification, Audio Spectrograms, Deep Learning, Pre-trained CNNs, MATLAB

Article history:
Received: December 6, 2024
Revised: April 11, 2025
Accepted: April 28, 2025
Published: April 30, 2025
(Online)

DOI: 10.37936/ecti-cit.2025192.259990

1. INTRODUCTION

Deep learning has emerged as a powerful tool for audio classification tasks in recent years, with applications ranging from speech recognition to music genre classification. This study explores the potential of deep learning for classifying movie genres based on audio information extracted from movie trailers. Automatically classifying movie genres has several practical applications, including content-based recommendation systems, organization of large movie databases, and targeted advertising.

Traditionally, genre classification has relied on textual information, such as plot summaries, cast lists, and user reviews. However, audio information, particularly the soundtrack and dialogue, can also provide valuable cues for genre classification. For example, action movies often feature fast-paced music and loud sound effects, while romantic movies may have softer music and more melodic dialogue.

This study investigates the potential of leveraging pre-trained deep-learning models for classifying movie genres solely from audio information, specifically using spectrograms derived from movie trailers. We hypothesize that the acoustic characteristics captured in spectrograms offer valuable cues for genre identification. To establish baseline performance for this approach, we evaluate and compare nine distinct pre-trained CNN architectures available in MATLAB—MobileNet-v2, ResNet-18, DenseNet-201, Places365-GoogLeNet, VGG-16, VGG-19, Inception-ResNet-v2, Inception-v3, and NASANet-Mobile—on their ability to classify trailers into five genres: action, romance, drama, comedy, and thriller.

[1,2,3]The authors are with Department of Computer and Information Science, Faculty of Applied Science, KMUTNB, Bangkok, Thailand, Email: porawatv@kmutnb.ac.th, s6604062857069@kmutnb.ac.th and s6604062857077@kmutnb.ac.th

[4,5,6,7,8,9,10,11,12]The authors are with the Faculty of Information Technology, KMITL, Bangkok, Thailand, Email: 66076038@it.kmitl.ac.th, 66076028@it.kmitl.ac.th, 66076019@it.kmitl.ac.th, 66076017@it.kmitl.ac.th, 66076049@it.kmitl.ac.th, 66076040@it.kmitl.ac.th, 65076081@it.kmitl.ac.th, 66076026@it.kmitl.ac.th and ponrudee@it.kmitl.ac.th

[13]The author is with Database/Bioinformatics Laboratory Chungbuk National University, Chungbuk, Republic of Korea, Email: khryu@chungbuk.ac.kr

[1]Corresponding author: porawatv@kmutnb.ac.th

Our findings reveal that this spectrogram-based method is effective. The VGG-16 architecture demonstrated the highest classification accuracy (86.27%) and F1-score (86.69%) among the evaluated models, indicating its particular suitability for this task.

The subsequent sections of this paper are as follows: Section 2 provides the literature review. Section 3 outlines the methodology. Section 4 presents the results and discussion. Finally, section 5 concludes this work by summarizing the key findings, discussing limitations, and proposing future directions for research and development in this area.

2. LITERATURE REVIEW

Despite the progress made in analyzing movie content through text and visuals, the potential of audio information for genre classification remains largely untapped. Existing research has primarily focused on textual analysis of synopses, scripts, reviews, or visual analysis of posters and keyframes. While some studies have explored limited aspects of audio, such as identifying specific sound effects or analyzing dialogue sentiment, a comprehensive exploration of audio features for genre classification is lacking; we further address this gap by proposing a novel approach based on spectrogram analysis of movie trailers in the next section. In this section, we investigate recent advancements in movie genre classification.

In [1], the paper surveys various methods for automatically detecting movie genres from trailers. It discusses low-level features such as color, lighting, and motion and high-level features such as plot and characters. The paper also explores using different machine learning algorithms, including convolutional neural networks (CNNs) and support vector machines (SVMs). The paper concludes that no perfect method for movie genre detection exists, but combining different methods may be the most effective approach.

In [2], the work investigates the application of machine learning to automatically identify movie genres from synopses, a critical task for efficiently conveying a film's essence to potential viewers. Leveraging movie data from Kaggle and Rotten Tomatoes, the work evaluates the performance of two supervised learning models (k-NN and SVM) and two deep learning models (CNN and RNN). The work also further explores the impact of removing proper nouns on classification accuracy. Results demonstrate the superiority of RNN with LSTM layers for analyzing the textual data in movie synopses, achieving an accuracy of 80.5%. The research provides valuable insights into selecting appropriate machine-learning models based on textual movie descriptions for genre classification.

In [3], the study addresses the challenging task of automatically classifying movie genres using deep learning analysis of poster images. The paper explained the importance of genre classification for aiding viewer decision-making; the study proposes a computerized framework that leverages low-level (e.g., color, edge) and high-level (e.g., object detection, shape) image features. A multi-layered convolutional neural network (CNN) is trained on a large dataset of movie posters to extract relevant features and classify genres. The work achieves an accuracy of 91.15%, demonstrating the effectiveness of CNNs for this task. However, further improvements are needed in other performance metrics, such as F1-score, precision, hamming loss, and zero-one loss, which indicate challenges in achieving balanced and precise genre classification.

In [4], the study addresses the challenge of multilabel movie genre classification, where a film can belong to multiple genres simultaneously. The study proposes a novel method combining problem transformation techniques, text vectorization, and traditional machine learning classifiers to improve efficiency and reduce computational resources compared to recent neural network approaches. Specifically, the study also explores binary relevance (BR) and label powerset (LP) for transforming the multilabel problem, Count Vectorizer (CV) and TF-IDF for text representation, and classifiers such as Logistic Regression, Multinomial Naive Bayes, K-Nearest Neighbour, and Support Vector Classifier. Through extensive experiments on an IMDb dataset with 27 movie genres, the study evaluates 16 different combinations using k-fold cross-validation. Results demonstrate that the combination of label powerset, TF-IDF, and Support Vector Classifier achieves the best performance with an accuracy of 0.95 and an F1-score of 0.86, offering a highly effective and efficient solution for movie genre classification.

In [5], the work investigates the effectiveness of transfer learning for multilabel movie genre classification using poster images. Six state-of-the-art pretrained models—VGG16, ResNet, DenseNet, Inception, MobileNet, and ConvNeXt—are employed and fine-tuned for this task. This work obtained movie posters from the Internet Movie Database (IMDB) and divided the dataset using an iterative stratification technique to ensure a balanced representation of genres. The paper also evaluated the performance across various metrics, including accuracy, loss, Hamming loss, F1-score, precision, and AUC. Results indicate that the modified DenseNet architecture achieves the highest accuracy at 90%, closely followed by the ConvNeXt model. The work highlights the potential of transfer learning for multilabel classification and the effectiveness of iterative stratification for handling unbalanced datasets. Future work will explore incorporating natural language processing and ensemble methods to enhance classification performance.

In [6], the study addresses the challenge of reclassifying large, diverse datasets, often obtained through web scraping, to improve the performance of machine learning models. This paper recognized the impact of dataset quality on model effectiveness, and a threshold-based algorithm for effective stop-word removal is proposed. The method utilizes an unsupervised classification technique (K-means) to accurately categorize user reviews from the IMDb dataset into their most suitable genres, creating a well-balanced dataset. The analysis highlights the influence of text vectorization methods on cluster generation and the impact of word embedding and stop-word removal on categorization accuracy. The proposed method analyses the presence and frequency of potential stop words within reviews across different genres, removing those exceeding a predefined threshold. The approach achieves over 80% genre mapping success compared to traditional methods. Employing mini-batch K-means for cluster formation further enhances review reclassification. Combining the proposed stop word removal method with TF-IDF effectively categorizes sparsely labeled data into meaningful clusters. The resulting reclassified and balanced datasets demonstrate significant improvement, achieving 94% accuracy compared to the original dataset.

In [7], the paper introduces a novel method for movie genre classification that leverages a diverse set of pre-trained models to extract rich features from movie trailers. These features encompass visual elements (scenery, objects, characters, text) and audio components (speech, music, and sound effects). The paper trained small, efficient classifier models to fuse these pre-trained features intelligently. By employing a transformer model, the approach utilizes all video and audio frames without temporal pooling, effectively capturing the complex relationships between elements and avoiding the limitations of traditional methods that rely on a fixed, small number of frames. This method effectively fuses features from diverse tasks and modalities with varying dimensionalities, temporal lengths, and complex dependencies. It outperforms state-of-the-art movie genre classification models regarding precision, recall, and mean average precision (mAP). The MovieNet dataset's trained features, genre classification code, and models are made publicly available to encourage further research.

In [8], streaming services increasingly rely on automated genre classification to optimize content management and user experiences. The study introduces a novel deep-learning architecture for accurate and efficient movie genre classification. The proposed approach employs an ensemble-gated recurrent unit (ensGRU) neural network to analyze video content's motion, spatial information, and temporal relationships. A sophisticated deep neural network incorporating the ensGRU captures robust video representations for multi-class movie classification. Evaluations on benchmark datasets, including the LMTD dataset, demonstrate the high performance of the ensGRU model, achieving accurate genre classification by effectively extracting and learning motion, spatial, and temporal features. The work performed further validation using an engine block assembly dataset. The enhanced architecture significantly improves movie genre categorization on the LMTD dataset, outperforming existing models while requiring less computational power. The model consistently delivers outstanding results with an F1 score of 0.9102 and an accuracy rate of 94.4%. Comparative evaluations highlight the accuracy and effectiveness of the proposed model in identifying and classifying video genres by extracting contextual information from video descriptors. Integration of edge processing capabilities enables optimal real-time video processing and analysis, enhancing performance in dynamic media environments.

In [9], the study addresses the limitations of existing video-based movie genre classification methods, which often overlook language elements and process entire videos inefficiently. A novel approach, Movie genre Classification based on Language augmentatIon and shot samPling (Movie-CLIP), is proposed. Movie-CLIP incorporates a language augmentation module to recognize language elements from audio, capturing high-level semantics like storylines and context. Additionally, a shot sampling module selects representative shots, avoiding unnecessary processing of the entire video. This approach enhances efficiency in movie genre prediction by focusing on key information. Evaluation on MovieNet and Condensed Movies datasets demonstrates a 6-9% mean Average Precision (mAP) improvement over baseline models. Movie-CLIP also generalizes to scene boundary detection, achieving a 1.1% Average Precision (AP) improvement over the state-of-the-art. This method effectively addresses the challenges of incorporating language elements and efficiently processing video data for accurate movie genre classification. The implementation is publicly available.

In [10], the study explores movie trailer genre classification using machine learning and spectrogram analysis. Spectrograms, which visualize the frequency content of audio signals over time, are extracted from movie trailers. Convolutional Neural Networks (CNNs), effective in image recognition tasks, are employed to analyze these spectrograms and identify genre-specific patterns. The paper compared the performance against a Random Forest model. The paper trained the models on a dataset of movie trailers categorized into five genres: action, romance, drama, comedy, and thriller. Librosa, a Python library, is used for audio pre-processing, and the entire training process is conducted in Python. This research investigates the potential of combining machine learning and spectrogram analysis for accurate and efficient movie trailer genre classification.

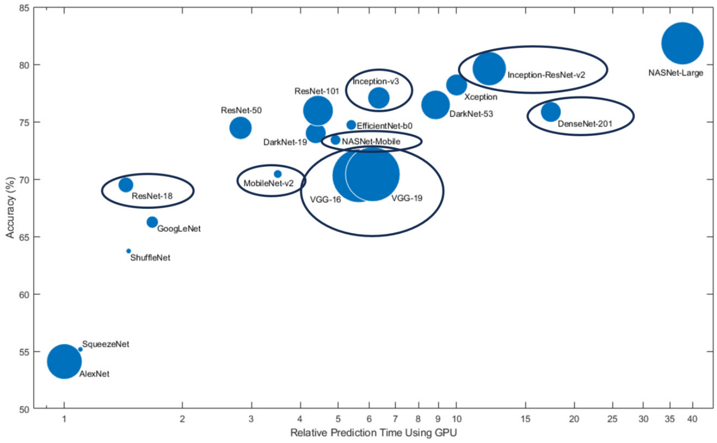


Fig.1: *Comparing model size, speed, and accuracy for popular CNN pre-trained models [11].*

3. METHODS

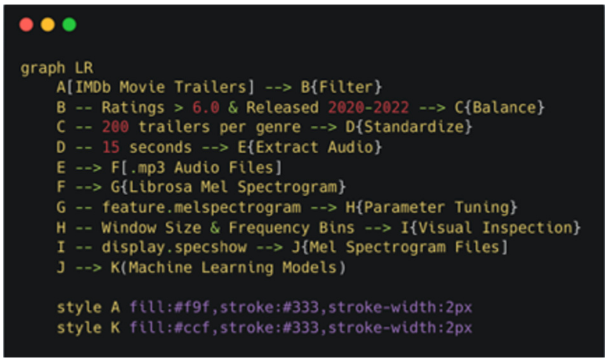
We selected all baseline models for movie genre classification from spectrograms using a comparative analysis of pre-trained convolutional neural networks (CNNs) provided by MathWorks [11] (figure 1). This analysis, depicted in the referenced graph, highlights the performance trade-offs between accuracy and prediction speed for various CNN architectures. Considering the need for accurate genre classification and efficient processing, a balanced approach was adopted to select baseline models that demonstrate strong performance across both metrics. This strategy aimed to identify architectures capable of effectively capturing discriminative features from spectrograms while maintaining reasonable computational efficiency for practical applications.

Considering the trade-off between accuracy and prediction speed, as depicted in the graph referenced, we selected nine distinct CNN architectures for this study: MobileNet-v2, ResNet-18, DenseNet-201, Places365-GoogLeNet, VGG-16, VGG-19, Inception-ResNet-v2, Inception-v3, and NASANet-Mobile.

1) Data preparation:

Movie trailers were sourced from IMDb, specifically targeting trailers released between 2020 and 2022 with ratings above 6.0. The dataset was carefully balanced to include 200 trailers from the five represented genres: action, romance, drama, comedy, and thriller. We standardized each trailer to a length of 15 seconds. The audio tracks were extracted from these trailers and converted into the .mp3 format.

Librosa, a Python audio and music analysis library, was used to generate spectrograms from these audio files—the feature `Mel spectrogram` function in Librosa was employed explicitly for this task. Parameters such as window size and the number of frequency bins were adjusted to capture the most relevant audio characteristics for genre classification. The generated spectrograms were visually inspected using Librosa's display. We used the `specshow` function to ensure quality and gain insights into the audio content. This process transformed the audio data into Mel spectrogram files, visually representing the sound frequencies over time. These spectrograms served as the input features for the machine learning models used in the genre classification task. We show code snippets and data-preparing flowcharts in Figure 2-3, respectively.



```
graph LR
    A[IMDb Movie Trailers] --> B{Filter}
    B -- Ratings > 6.0 & Released 2020-2022 --> C{Balance}
    C -- 200 trailers per genre --> D{Standardize}
    D -- 15 seconds --> E{Extract Audio}
    E --> F[.mp3 Audio Files]
    F --> G{Librosa Mel Spectrogram}
    G -- feature.melspectrogram --> H{Parameter Tuning}
    H -- Window Size & Frequency Bins --> I{Visual Inspection}
    I -- display.specshow --> J{Mel Spectrogram Files]
    J --> K(Machine Learning Models)

    style A fill:#f9f,stroke:#333,stroke-width:2px
    style K fill:#ccf,stroke:#333,stroke-width:2px
```

Fig.2: *The data preparation process for movie genre classification.*

A block diagram illustrating the data preparation process for movie genre classification. The process includes sourcing movie trailers from IMDb, filtering and balancing the dataset, standardizing trailer length, extracting audio, generating Mel spectrograms using Librosa, and finally using the spectrograms as input features for machine learning models. The explanation of the code snippet is as follows:

IMDb Movie Trailers: The starting point represents the raw data source.

• Filter: Trailers are filtered based on the release date (2020-2022) and IMDb rating (> 6.0).

• Balance: The dataset is balanced to ensure an equal number of trailers (200) per genre.

• Standardize: Each trailer is standardized to a length of 15 seconds.

• Extract Audio: The audio track is extracted from each movie trailer.

- .mp3 Audio Files: The extracted audio is saved in .mp3 format.
- Librosa Mel Spectrogram: This represents using the Librosa library to generate Mel spectrograms.
- Parameter Tuning: This includes parameters within the feature `Mel spectrogram` functions (e.g., window size, frequency bins) are adjusted.
- Visual Inspection: Spectrograms are visually inspected using `display.specshow` to ensure quality.
- Mel Spectrogram Files: These are the final outputs of the data preparation process, which are used as input for the machine learning models.
- Machine Learning Models: This is where the prepared data is used for genre classification.

2) Genre Movie Classification using Pre-trained Models in MATLAB:

The next stage focuses on preparing the pre-trained model for the task. Spectrogram images are resized to meet the input requirements of the chosen model. A variety of powerful pre-trained models can be employed, including MobileNet-v2, ResNet-18, DenseNet-201, Places365-GoogLeNet, VGG-16, VGG-19, Inception-ResNet-v2, Inception-v3, and NASANet-Mobile. Having been trained on vast image datasets, these models come equipped with pre-learned features that can be highly beneficial for genre classification. The training process is refined by setting key parameters like the optimization algorithm (often ‘sgdm’), mini-batch size, and learning rate, collectively influencing the model’s learning efficiency and performance.

With the data prepared, we divided the dataset before the training phase commenced. Specifically, we use the MATLAB function `splitEachLabel(imds, 0.7, 0.1, ‘‘randomized’’);` to partition the spectrogram dataset (`imds`) randomly into three distinct subsets: 70% for training (`imdsTrain`), 10% for validation (`imdsVal`), and 20% for testing (`imdsTest`). The model then learns from the designated training data (`imdsTrain`), and its performance is continuously validated using the validation set (`imdsVal`). This iterative process allows for monitoring and addressing potential issues like overfitting, where the model performs well on training data but poorly on unseen validation data, or slow convergence, where the model takes an excessively long time to learn. After training, the separate testing set (`imdsTest`) is reserved for final evaluation.

Once trained, the model’s classification ability is tested. A sample image is classified, and the model’s accuracy is rigorously evaluated using the testing set. This provides a crucial estimate of how well the model is expected to perform on new, unseen spectrograms.

Finally, we generated a confusion matrix to understand the model’s performance better. This visualization tool clearly shows how accurately the model classifies different genres and, importantly, reveals any

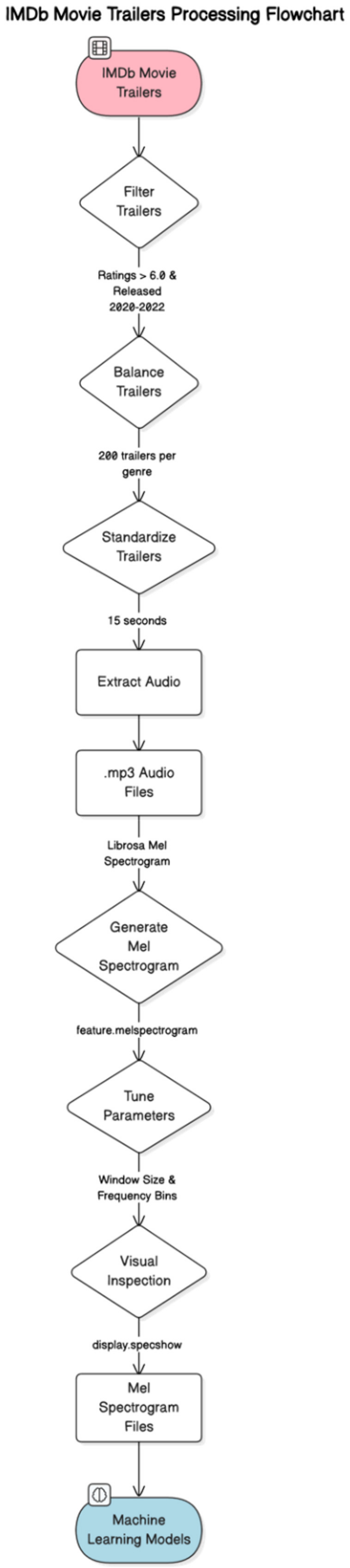


Fig.3: *Flowchart illustrating the data preparation process for movie genre classification.*

Table 1: *Comparison of 9 MATLAB pretrained models.*

Model	**Strengths**	**Weaknesses**	**Detailed comparison (please refer to the numerical results in Table 2 and the accuracy and loss functions in the Appendix)**
MobileNetV2	Highly efficient, suitable for mobile devices	May sacrifice some accuracy for efficiency	Achieved moderate accuracy (54.55%). Showed good learning and generalization during training but might benefit from further training. Its general efficiency comes with mid-range accuracy in this task.
ResNet18	Simpler and faster than ResNet50	May have lower accuracy on complex datasets	Showed lower accuracy (48.37%) but demonstrated effective learning and good generalization during training. Its relative simplicity did not capture the spectrogram features as effectively as deeper models.
DenseNet201	Strong performance, especially for image classification	It can be computationally expensive	Yielded moderate accuracy (56.86%) with good generalization observed during training. Its known strength for image classification resulted in a decent performance.
Places365-GoogLeNet	A variant of the GoogLeNet architecture, making it particularly good at recognizing different environments and places	Limit its performance on tasks like object detection or image classification of objects, which is computationally intensive.	Achieved high accuracy (81.70%), indicating it learned and generalized well, with the potential for more improvement suggested by the learning curves. Its suitability for scene recognition translates well to spectrogram patterns.
VGG16	Good balance of accuracy and efficiency	Less complex than VGG19, potentially less powerful	Delivered the highest accuracy (86.27%) in this study, showing effective learning and good generalization. Its architecture provided the best balance for capturing relevant features in this spectrogram dataset.
VGG19	Powerful feature extraction	Computationally expensive	Performed surprisingly poorly in accuracy (49.35%) despite showing effective learning curves. Its greater complexity did not yield better results here, potentially due to overfitting or the added layers not capturing more relevant features for this specific task.
Inception-ResNet-v2	Combines Inception and ResNet for better performance	More complex than InceptionV3	Resulted in moderate accuracy (54.25%), with training curves indicating effective learning and generalization. Its sophisticated architecture performed similarly to MobileNet-v2 and DenseNet-201.
Inception-V3	Excellent performance on large datasets	Complex architecture	Had lower accuracy (47.60%) despite effective learning and good generalization during training. Its known suitability for large datasets did not translate to top performance here.
NASANet-Mobile	Efficient and accurate	May struggle with complex patterns	Achieved the lowest accuracy (45.10%) and showed clear signs of overfitting during training (poor validation accuracy/loss compared to training), indicating it failed to generalize well on this specific dataset.

confusion between genres. For instance, the confusion matrix might highlight if the model frequently misclassifies Action movies as Thriller movies or vice versa, offering valuable insights for further improvements and refinements to the classification process.

4. RESULTS AND DISCUSSION

This study investigated the effectiveness of utilizing spectrograms, visual representations of audio frequencies, in training deep-learning models to classify movies into different genres. We employed a dataset of 1000 spectrograms categorized into five distinct genres: Action, Comedy, Drama, Romance, and Thriller. The spectrograms were treated as images, and a series of 9 pre-trained deep learning models, specifically MobileNet-v2, ResNet-18, DenseNet-201, Places365-GoogLeNet, VGG-16, VGG-19, Inception-ResNet-v2, Inception-v3, and NASANet-Mobile, were trained and evaluated for their genre classification capabilities.

Selecting the right pre-trained model for spectrogram-based movie genre classification requires careful consideration of several factors. A crucial aspect is balancing the desired accuracy with computational efficiency. For example, if an application demands a model that can run smoothly on devices with limited processing power or memory, MobileNetV2, known for its efficiency, might be a suitable choice. However, if achieving the highest possible accuracy is the primary goal and computational resources are readily available, more complex and powerful models like VGG19 or InceptionV3 could be preferred.

Furthermore, the size of our dataset plays a significant role in model selection. For large datasets, models like InceptionV3 or DenseNet-201 outperform other models with their greater capacity to learn complex patterns. Conversely, if an application works with a smaller dataset, employing simpler models such as NASANet-Mobile or ResNet18 might be sufficient to avoid overfitting and achieve good performance.

Finally, the decision should be guided by the complexity of the classification task. If the genres exhibit subtle differences in their spectrograms, requiring the model to discern intricate patterns, VGG19 or InceptionResNet might be better equipped for the challenge with their strong feature extraction capabilities. For simpler tasks where the distinctions between genres are more apparent, less complex models could prove adequate, offering a good balance of performance and efficiency. Table 1 compares the strengths and weaknesses of our study's nine pretrained MATLAB models. The accuracy and loss functions for the 9 MATLAB pre-trained models are presented in Figures A1 through A9 of the Appendix.

Figure A1 shows the accuracy and loss functions of MobileNet-v2. The training progress graph indicates a successful learning process. Both training and validation accuracy increased over time, while training and validation loss decreased. This suggests that the model effectively learns and generalizes to new data without overfitting. Although the model shows improvement, further training might be beneficial as the performance curves have not completely flattened out. The x-axis of the graph represents the number of training iterations, and we trained the model for 30 epochs, where each epoch is a complete pass through the training dataset. We also employed a constant learning rate throughout the training process.

Figure A2 shows the accuracy and loss functions of ResNet-18. The training progress graph shows a model that is learning effectively. Training and unseen validation data accuracy generally increase while the corresponding loss values decrease. This indicates good generalization, as the model's performance on new data closely mirrors its performance on the training data, suggesting it is not simply memorizing the training examples. The x-axis of the graph represents training iterations, which are grouped into five epochs, each marking a complete pass through the dataset. A constant learning rate was used throughout the training process, meaning the model adjusted its internal parameters steadily.

Figure A3 displays the training progress of DenseNet-201 over five epochs, with each epoch representing a complete pass through the training dataset. The blue line illustrates the model's accuracy on the training data, showing a general upward trend with some fluctuations. The gray line represents the validation accuracy, which measures performance on unseen data. It closely follows the training accuracy, indicating good generalization and suggesting that the model does not overfit the training data. The orange line depicts the training loss, quantifying the model's prediction error. As expected, the loss decreases over time, reflecting the model's learning and improvement. The black dashed line shows the validation loss, which also decreases and mirrors the training loss, further supporting the observation of good generalization. The x-axis of the graph represents the training iterations, organized into five epochs marked by vertical lines. A constant learning rate was used throughout the training process, meaning the model adjusted its internal parameters consistently.

Figure A4 illustrates the training progress of a Place365-GoogLeNet model. The top graph shows accuracy, with the blue line representing training accuracy and the grey line representing validation accuracy on unseen data. Both generally increase, indicating the model is learning and generalizing well. The bottom graph shows loss, with the orange line representing training loss and the black dashed line representing validation loss. Both decreases, suggesting the model is improving and not just memorizing the training data. The x-axis represents training iterations, with vertical grid lines likely separating epochs. While the model progresses, the lack of complete flattening in the accuracy and loss curves suggests potential for further improvement with more training.

Figure A5 illustrates the training progress of a VGG-16 model, likely over 5 epochs. The blue line, representing training accuracy, shows a general upward trend with some fluctuations, while the grey line, depicting validation accuracy, closely follows it. This suggests the model is learning effectively and generalizing well to unseen data. The orange line, representing training loss, consistently decreases, indicating that the model is improving its predictions. The validation loss, shown by the dashed black line, mirrors this trend, further supporting the notion of good generalization. Although the model demonstrates progress, the accuracy and loss curves have not completely flattened out, indicating potential for further improvement with continued training.

Figure A6 shows the training progress of a VGG-19 model, a deep convolutional neural network with 19 layers. The training and validation accuracy increase over time, while the training and validation loss decrease. This indicates that the model learns effectively and generalizes well to unseen data. Although there are fluctuations in accuracy, the overall trend is positive. VGG-19, with its 19 layers, is a more complex model than the similar VGG-16 and often achieves slightly higher accuracy, though it requires more computational resources. The graph suggests that the VGG-19 model is suitable for the task, demonstrating good performance and generalization.

Figure A7 depicts the successful training of an Inception-ResNet-v2 model, a sophisticated neural network suitable for image recognition tasks. It demonstrates the model's increasing accuracy and decreasing loss over time, indicating effective learning and generalization to new data. While computationally demanding, Inception-ResNet-v2's combination of inception blocks and residual connections allows for high accuracy and efficient processing, making it a powerful tool for image-related tasks. The graph showcases this potential, with promising results that suggest further improvement is possible with continued training.

Figure A8 shows the training progress of an Inception-v3 model, likely over three epochs. The blue line, representing training accuracy, shows a clear upward trend with some fluctuations, indicating that the model is learning and improving its performance on the training data. The grey line, depicting validation accuracy, closely follows this trend, suggesting the model is generalizing well to unseen data and not overfitting. The orange line, representing training loss, steadily decreases, reflecting the model's increasing ability to make accurate predictions. The black dashed line, showing validation loss, mirrors this decrease, further supporting the observation of good generalization. Overall, the graph illustrates a successful training process where the model is learning effectively and demonstrating promising performance on both seen and unseen data.

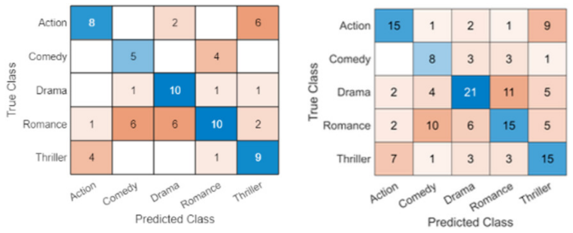
Figure A9 shows the training progress of a NASANet-Mobile model. The blue line, representing training accuracy, shows a general upward trend, indicating the model is learning and improving its performance on the training data. However, the grey line, which represents validation accuracy (performance on unseen data), remains relatively flat, hovering around 40%. This discrepancy suggests that the model might be overfitting to the training data, meaning it's memorizing the training examples instead of learning generalizable patterns. The orange line, depicting training loss, decreases as expected, but the black dashed line shows validation loss, plateaus, and a slight increase. This further process supports the possibility of overfitting. While the model shows improvement in the training data, its inability to generalize to new data indicates a need for adjustments to the training process, such as using regularization techniques or increasing the size of the training dataset. Table 2 shows the average computational times of 9 Pre-trained models, and Table 3 shows the results of genre classification from spectrograms. Figures 4 to 8 show the confusion matrices of each model.

Table 2: *Average Computational Times for 9 Pre-trained models.*

Pre-trained Model	**Average Computational Times (MATLAB Single CPU)**
MobileNet-v2	380ms
ResNet-18	4.8ms
DenseNet-201	6 seconds
Places365-GoogLeNet	3 seconds
VGG-16	906ms
VGG-19	260ms
Inception-ResNet-v2	795ms
Inception-v3	553ms
NASANet-Mobile	5 seconds

Table 3: *Evaluation of Pre-trained Models for Genre Classification from Spectrograms.*

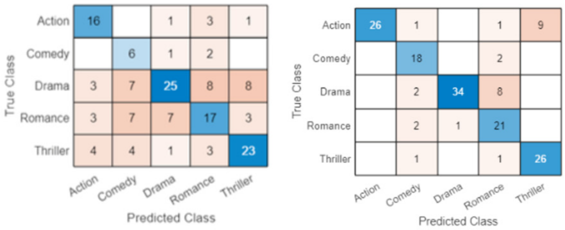
MobileNet-v2			
Class	**Precision**	**Recall**	**F1-score**
Action	0.5	0.61538	0.55172
Comedy	0.55556	0.41667	0.47619
Drama	0.76923	0.55556	0.64516
Romance	0.4	0.625	0.4878
Thriller	0.64286	0.5	0.5625
Accuracy			0.5455
ResNet-18			
Class	**Precision**	**Recall**	**F1-score**
Action	0.5357	0.5769	0.5556
Comedy	0.5333	0.3333	0.4103
Drama	0.4884	0.6	0.5385
Romance	0.3947	0.4545	0.4225
Thriller	0.5172	0.4286	0.4688
Accuracy			0.4837
DenseNet-201			
Class	**Precision**	**Recall**	**F1-score**
Action	0.7619	0.61538	0.68085
Comedy	0.66667	0.25	0.36364
Drama	0.4902	0.71429	0.5814
Romance	0.45946	0.51515	0.48571
Thriller	0.65714	0.65714	0.65714
Accuracy			0.5686
Places365-GoogLeNet			
Class	**Precision**	**Recall**	**F1-score**
Action	0.7	1.0	0.83
Comedy	0.9	0.75	0.82
Drama	0.77	0.97	0.86
Romance	0.88	0.64	0.74
Thriller	0.93	0.74	0.83
Accuracy			0.8170
VGG-16			
Class	**Precision**	**Recall**	**F1-score**
Action	0.7931	0.88462	0.83636
Comedy	0.92	0.95833	0.93878
Drama	0.93939	0.88571	0.91176
Romance	0.96154	0.75758	0.84746
Thriller	0.75	0.85714	0.80
Accuracy			0.8627
VGG-19			
Class	**Precision**	**Recall**	**F1-score**
Action	0.8571	0.4615	0.6
Comedy	0.2222	0.5	0.3077
Drama	0.6316	0.6667	0.6486
Romance	0.5	0.1250	0.2
Thriller	0.6	0.6667	0.6316
Accuracy			0.4935
Inception-ResNet-v2			
Class	**Precision**	**Recall**	**F1-score**
Action	0.56154	0.5615	0.56154
Comedy	0.24286	0.18333	0.20526
Drama	0.55	0.61111	0.5789
Romance	0.4125	0.4125	0.4125
Thriller	0.52857	0.6	0.56154
Accuracy			0.5425
Inception-v3			
Class	**Precision**	**Recall**	**F1-score**
Action	0.53	0.62	0.57
Comedy	0.46	0.225	0.32
Drama	0.47	0.77	0.58
Romance	0.44	0.27	0.34
Thriller	0.53	0.49	0.51
Accuracy			0.476
NASANet-Mobile			
Class	**Precision**	**Recall**	**F1-score**
Action	0.54	0.54	0.54
Comedy	0.35	0.25	0.29
Drama	0.48	0.63	0.54
Romance	0.44	0.45	0.45
Thriller	0.4	0.34	0.37
Accuracy			0.451



	Action	Comedy	Drama	Romance	Thriller
Action	8		2		6
Comedy		5		4	
Drama		1	10	1	1
Romance	1	6	6	10	2
Thriller	4			1	9

	Action	Comedy	Drama	Romance	Thriller
Action	15	1	2	1	9
Comedy		8	3	3	1
Drama	2	4	21	11	5
Romance	2	10	6	15	5
Thriller	7	1	3	3	15

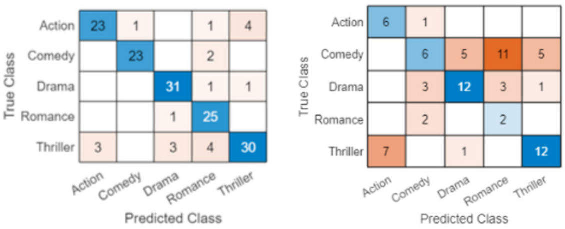
Fig.4: *Confusion matrices of MobileNet-v2 and ResNet-18.*



	Action	Comedy	Drama	Romance	Thriller
Action	16		1	3	1
Comedy		6	1	2	
Drama	3	7	25	8	8
Romance	3	7	7	17	3
Thriller	4	4	1	3	23

	Action	Comedy	Drama	Romance	Thriller
Action	26	1		1	9
Comedy		18		2	
Drama		2	34	8	
Romance		2	1	21	
Thriller		1		1	26

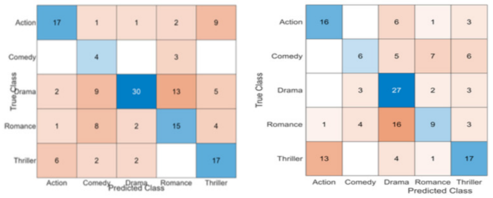
Fig.5: *Confusion matrices of DenseNet-201 and Places365-GoogLeNet..*



	Action	Comedy	Drama	Romance	Thriller
Action	23	1		1	4
Comedy		23		2	
Drama			31	1	1
Romance			1	25	
Thriller	3		3	4	30

	Action	Comedy	Drama	Romance	Thriller
Action	6	1			
Comedy		6	5	11	5
Drama		3	12	3	1
Romance		2		2	
Thriller	7		1		12

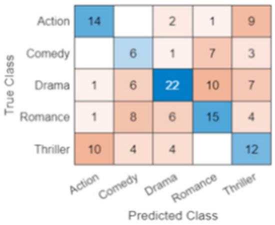
Fig.6: *Confusion matrices of VGG-16 and VGG-19.*



	Action	Comedy	Drama	Romance	Thriller
Action	17	1	1	2	9
Comedy		4		3	
Drama	2	9	30	13	5
Romance	1	8	2	15	4
Thriller	6	2	2		17

	Action	Comedy	Drama	Romance	Thriller
Action	16		6	1	3
Comedy		6	5	7	6
Drama		3	27	2	3
Romance	1	4	16	9	3
Thriller	13		4	1	17

Fig.7: *Confusion matrices of Inception-ResNet-v2 and Inception-v3.*



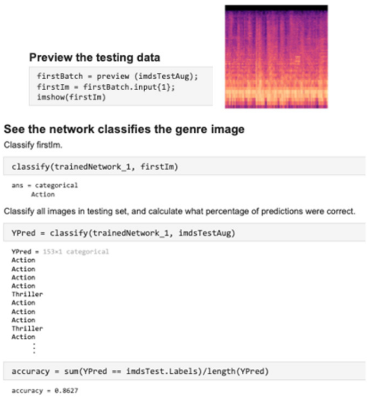
	Action	Comedy	Drama	Romance	Thriller
Action	14		2	1	9
Comedy		6	1	7	3
Drama	1	6	22	10	7
Romance	1	8	6	15	4
Thriller	10	4	4		12

Fig.8: *Confusion matric of NASANET-Mobile.*

While VGG-16 achieved the best classification accuracy among the tested models, the comparison with VGG-19 reveals interesting points about complexity, performance, and computational time. Although VGG-19 is generally considered a more complex model with more layers than VGG-16, its classification accuracy was surprisingly lower in this study. Several factors can cause it. VGG-19's increased complexity might have led to overfitting on this specific dataset, where the model memorized training data features instead of generalizing well to unseen spectrograms. Alternatively, the additional layers in VGG-19 might not have extracted more discriminative features for this particular genre classification task, or its hyperparameters might not have been optimally tuned.

Moreover, the provided computational times (Table 1) indicate that VGG-19 had a faster average inference time (260ms) compared to VGG-16 (906ms) under the tested "MATLAB Single CPU" conditions. This result highlights that assumptions about computational cost based solely on model depth are not always straightforward and can depend on the specific execution environment and implementation. Ultimately, the results underscore that model complexity is not the sole predictor of performance; factors like generalization ability, feature relevance, hyperparameter optimization, and even measured computational speed play crucial roles in determining the best model for a task.

Figure 9 illustrates an example of generating a genre spectrogram classification test result using MATLAB commands.



Preview the testing data

```
firstBatch = preview (imdsTestAug);
firstIm = firstBatch.input{1};
imshow(firstIm)
```

See the network classifies the genre image

Classify firstIm.

```
classify(trainedNetwork_1, firstIm)
```

```
ans = categorical
     Action
```

Classify all images in testing set, and calculate what percentage of predictions were correct.

```
YPred = classify(trainedNetwork_1, imdsTestAug)
```

```
YPred = 153×1 categorical
Action
Action
Action
Action
Thriller
Action
Action
Action
Thriller
Action
      ⋮
```

```
accuracy = sum(YPred == imdsTest.Labels)/length(YPred)
```

```
accuracy = 0.8627
```

Fig.9: *Testing result.*

We explain the code by showing the code breakdown:

1. Preview the testing data
    - `firstBatch = preview(imdsTestAug);`: This line uses the `preview` function to get a small batch of data from the augmented test image datastore (`imdsTestAug`). This datastore likely contains spectrograms of movie audio that have been augmented (e.g., rotated, flipped, etc.) to

improve model generalization.

- `firstIm = firstBatch.input{1};`: This extracts the first image from the batch. `.input{1}` accesses the first image within the input field of the `firstBatch` structure.
- `imshow(firstIm)`: This displays the extracted image, allowing us to visualize the data used for testing.

2. Our network classifies the genre image

- `classify(trainedNetwork_1, firstIm)`: This line uses a pre-trained neural network (`trainedNetwork_1`) to classify the `firstIm` image. The output (`ans`) is a categorical variable indicating the predicted genre (e.g., "Action").

3. Classify all images and calculate accuracy

- `YPred = classify(trainedNetwork_1, imdsTestAug)`: This applies the trained network to classify all images in the augmented test dataset (`imdsTestAug`). The result (`YPred`) is a categorical array containing the predicted genre for each test image.
- `accuracy = sum(YPred == imdsTest.Labels)/ length(YPred)`: This line calculates the classification accuracy. It compares the predicted labels (`YPred`) with the true labels (`imdsTest.Labels`) from the original test dataset (without augmentation). `sum(YPred == imdsTest.Labels)` counts the number of correct predictions, and dividing by `length(YPred)` (the total number of predictions) gives the accuracy as a percentage.
- `accuracy = 0.8627`: This shows that the trained network achieved an accuracy of 86.27% on the test set.

5. CONCLUSION

This study investigated the effectiveness of using spectrograms, visual representations of audio frequencies, to train deep learning models for classifying movies into different genres. The research used a dataset of 1000 spectrograms across five genres: Action, Comedy, Drama, Romance, and Thriller. 9 pre-trained MATLAB deep learning models were trained and evaluated, including MobileNet-v2, ResNet-18, DenseNet-201, Places365-GoogLeNet, VGG-16, VGG-19, Inception-ResNet-v2, Inception-v3, and NASANet-Mobile.

The results showed that VGG-16 was the most effective model for this task, achieving the highest accuracy at 86.27%. This suggests that pre-trained CNNs, particularly VGG-16, can be effectively used for audio-based genre classification in movie trailers.

- Expanding the dataset to include a broader range of genres and a larger number of samples to improve model generalization and performance.
- We are investigating other pre-trained models, including more complex and recently developed models, to explore their effectiveness for this task.
- Developing ensemble methods that combine multiple models' predictions to improve overall classification accuracy.

ACKNOWLEDGEMENT

"This research was funded by King Mongkut's University of Technology North Bangkok, Contract no. KMUTNB-68-BASIC-31".

AUTHOR CONTRIBUTIONS

Conceptualization, P. Visutsak; methodology, P. Visutsak; software, K. Treeraphapkajondet, V. Sakphet, W. Nitinuntatip, P. Satthong, T. Tongbai, D. Ongrungruaeng, A. Juntra, W. Aiamlamai, I. Sungwanna, and P. Phetrak; validation, P. Visutsak, P. Netisopakul, and K. H. Ryu; formal analysis, P. Netisopakul and K. H. Ryu; investigation, P. Netisopakul, and K. H. Ryu; data curation, K. Treeraphapkajondet, V. Sakphet, W. Nitinuntatip, P. Satthong, T. Tongbai, D. Ongrungruaeng, A. Juntra, W. Aiamlamai, I. Sungwanna and P. Phetrak; writing—original draft preparation, P. Visutsak; writing—review and editing, P. Visutsak; visualization, K. Treeraphapkajondet, V. Sakphet, W. Nitinuntatip, P. Satthong, T. Tongbai, D. Ongrungruaeng, A. Juntra, W. Aiamlamai, I. Sungwanna and P. Phetrak; supervision, P. Visutsak; funding acquisition, P. Visutsak. All authors have read and agreed to the published version of the manuscript.

References

[1] P. G. Shambharkar, A. Anand and A. Kumar, "A Survey Paper on Movie Trailer Genre Detection," *2020 International Conference on Computing and Data Science (CDS)*, Stanford, CA, USA, pp. 238-244, 2020.

[2] J. Wang, "Using Machine Learning to Identify Movie Genres through Online Movie Synopses," *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, Guangzhou, China, pp. 1-6, 2020.

[3] N. Hossain, M. M. Ahamad, S. Aktar and M. A. Moni, "Movie Genre Classification with Deep Neural Network using Poster Images," *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, Dhaka, Bangladesh, pp. 195-199, 2021.

[4] S. Kumar, N. Kumar, A. Dev and S. Naorem, "Movie genre classification using binary relevance, label powerset, and machine learning classifiers," *Multimedia Tools and Applications* , vol. 82, pp. 945–968, 2023.

[5] F. Z. Unal, M. S. Guzel, E. Bostanci, K. Acici and T. Asuroglu, "Multilabel Genre Prediction

Using Deep-Learning Frameworks," *Applied Sciences*, vol. 13, no. 15, p. 8665, 2023.

[6] F. González, M. Torres-Ruiz, G. Rivera-Torruco, L. Chonona-Hernández and R. Quintero, "A Natural-Language-Processing-Based Method for the Clustering and Analysis of Movie Reviews and Classification by Genre," *Mathematics*, vol. 11, no. 23, p. 4735, 2023.

[7] S. Sulun, P. Viana and M. E. P. Davies, "Movie trailer genre classification using multimodal pre-trained features," *Expert Systems with Applications*, vol. 258, p. 125209, 2024.

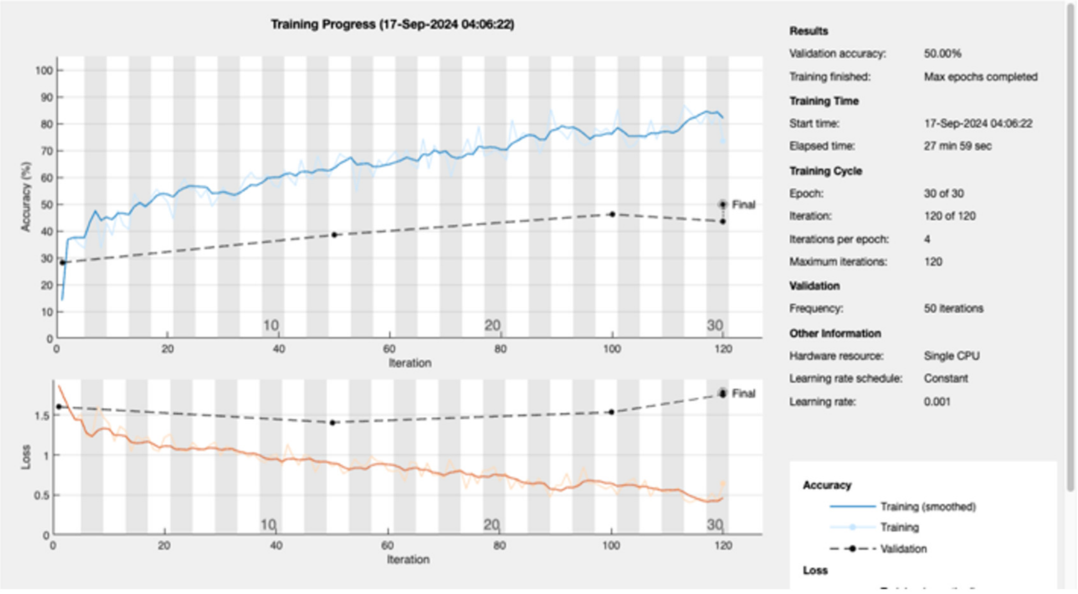
[8] Y. Shao and N. Guo, "Recognizing online video genres using ensemble deep convolutional learning for digital media service management," *Journal of Cloud Computing*, vol. 13, p. 102, 2024.

[9] Z. Zhang, Y. Gu, B. A. Plummer, X. Miao, J. Liu and H. Wang, "Movie genre classification by language augmentation and shot sampling," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7260–7270, 2024.

[10] P. Visutsak *et al.*, "Genre Classification of Movie Trailers using Spectrogram Analysis and Machine Learning," *2024 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, Tbilisi, Georgia, pp. 324-327, 2024.

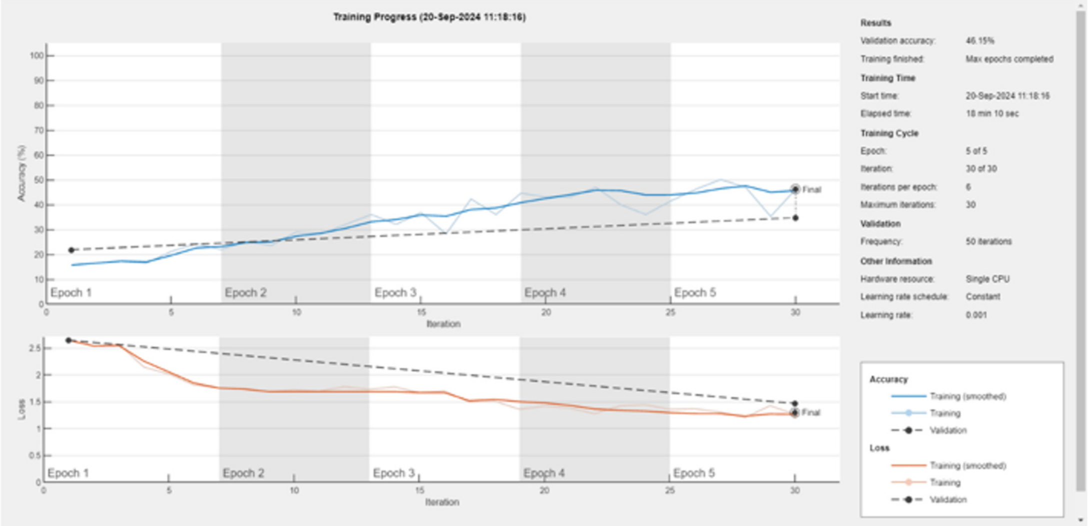
[11] "Transfer Learning," MathWorks, 2024. [Online]. Available: `https://mathworks.com/discovery/transfer-learning.html`. [Accessed: Dec. 2, 2024].

Appendix: The accuracy and loss functions for the 9 MATLAB pre-trained models.



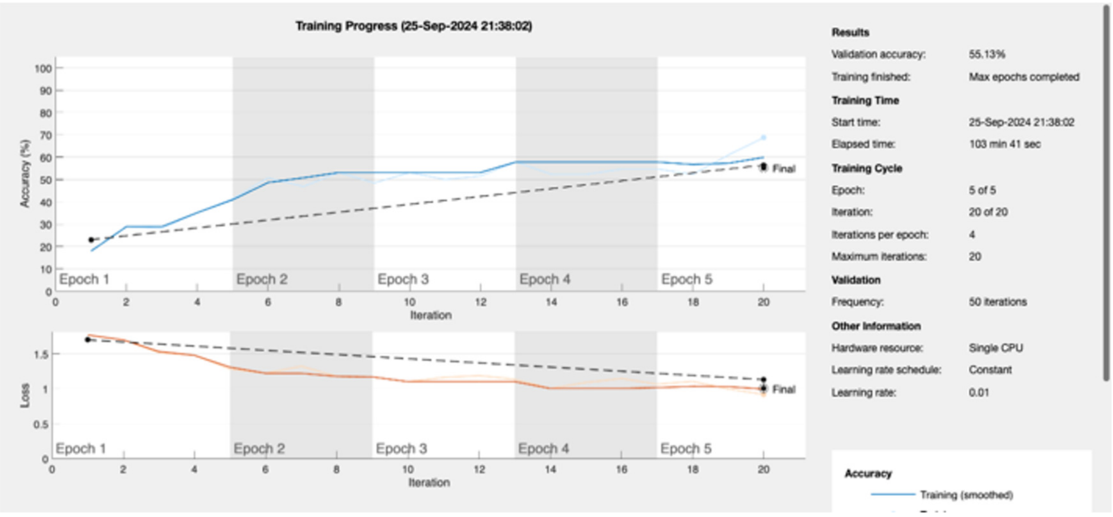
Results
Validation accuracy: 50.00%
Training finished: Max epochs completed
Training Time
Start time: 17-Sep-2024 04:06:22
Elapsed time: 27 min 59 sec
Training Cycle
Epoch: 30 of 30
Iteration: 120 of 120
Iterations per epoch: 4
Maximum iterations: 120
Validation
Frequency: 50 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.001

Fig.A1:*Accuracy and loss functions of MobileNet-v2.*



Results
Validation accuracy: 46.15%
Training finished: Max epochs completed
Training Time
Start time: 20-Sep-2024 11:18:16
Elapsed time: 18 min 10 sec
Training Cycle
Epoch: 5 of 5
Iteration: 30 of 30
Iterations per epoch: 6
Maximum iterations: 30
Validation
Frequency: 50 iterations
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.001

Fig.A2: *Accuracy and loss functions of ResNet-18.*



Results
Validation accuracy: 55.13%
Training finished: Max epochs completed

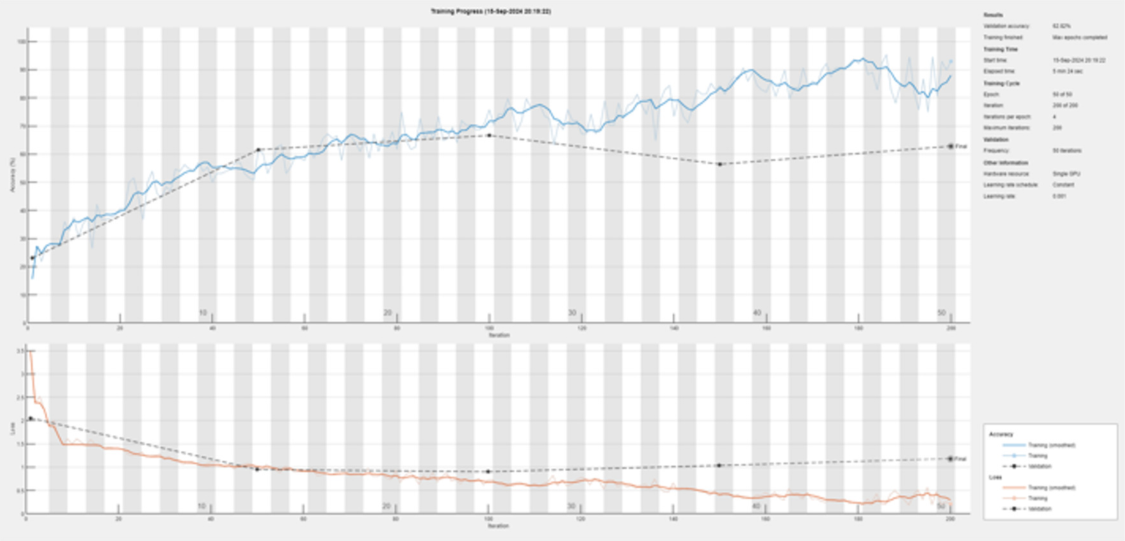
Training Time
Start time: 25-Sep-2024 21:38:02
Elapsed time: 103 min 41 sec

Training Cycle
Epoch: 5 of 5
Iteration: 20 of 20
Iterations per epoch: 4
Maximum iterations: 20

Validation
Frequency: 50 iterations

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

Fig. A3: *Accuracy and loss functions of DenseNet-201.*



Results
Validation accuracy: 62.82%
Training finished: Max epochs completed

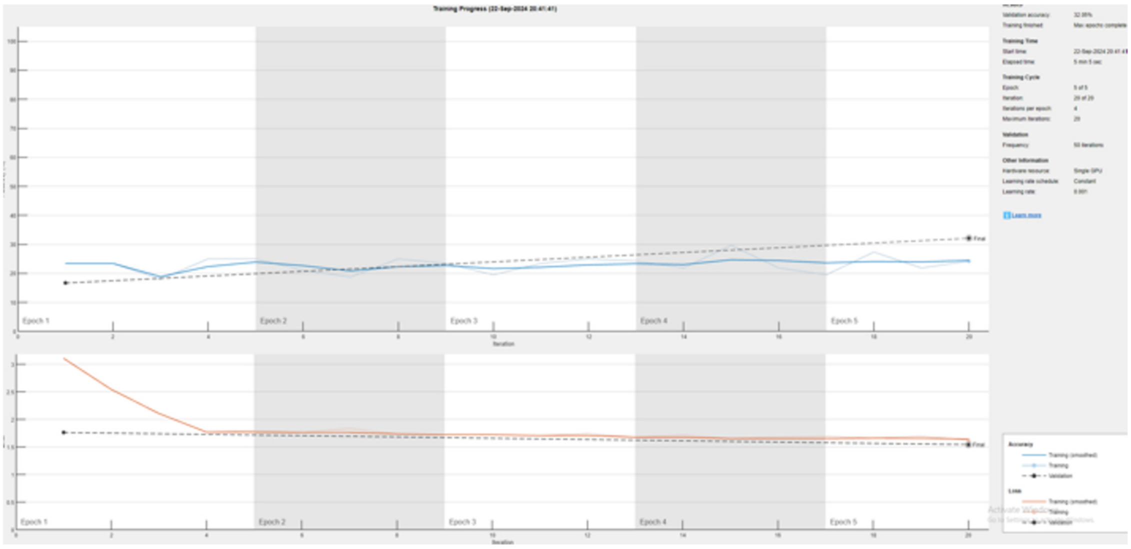
Training Time
Start time: 15-Sep-2024 20:19:22
Elapsed time: 5 min 24 sec

Training Cycle
Epoch: 50 of 50
Iteration: 200 of 200
Iterations per epoch: 4
Maximum iterations: 200

Validation
Frequency: 50 iterations

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.001

Fig. A4: *Accuracy and loss functions of Places365-GoogLeNet.*



Results
Validation accuracy: 32.35%
Training finished: Max epochs complete

Training Time
Start time: 22-Sep-2024 20:41:41
Elapsed time: 5 min 5 sec

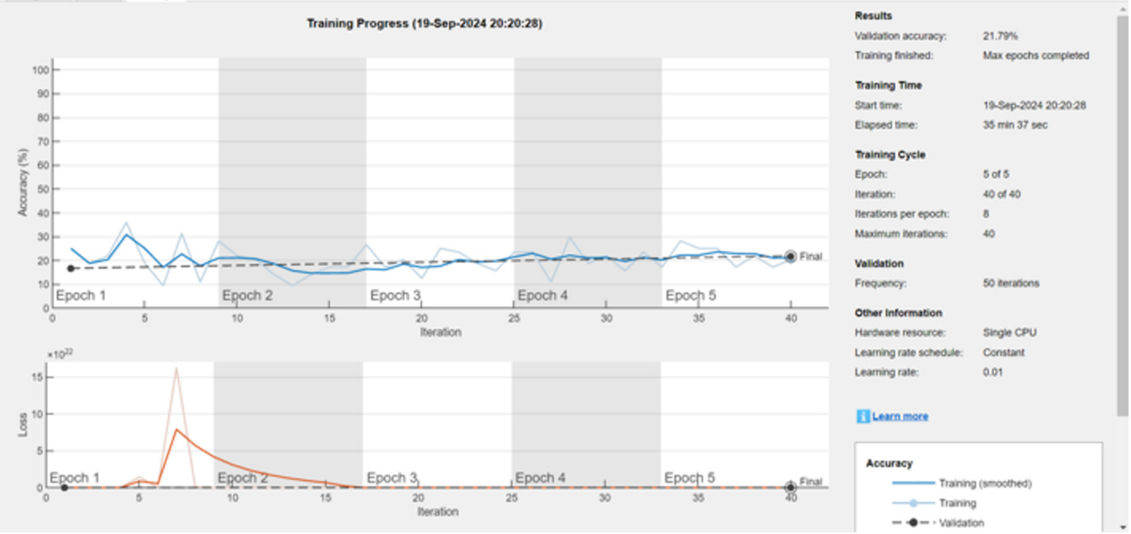
Training Cycle
Epoch: 5 of 5
Iteration: 20 of 20
Iterations per epoch: 4
Maximum iterations: 20

Validation
Frequency: 50 iterations

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.001

Learn more

Fig. A5: *Accuracy and loss functions of VGG-16.*



Results
Validation accuracy: 21.79%
Training finished: Max epochs completed

Training Time
Start time: 19-Sep-2024 20:20:28
Elapsed time: 35 min 37 sec

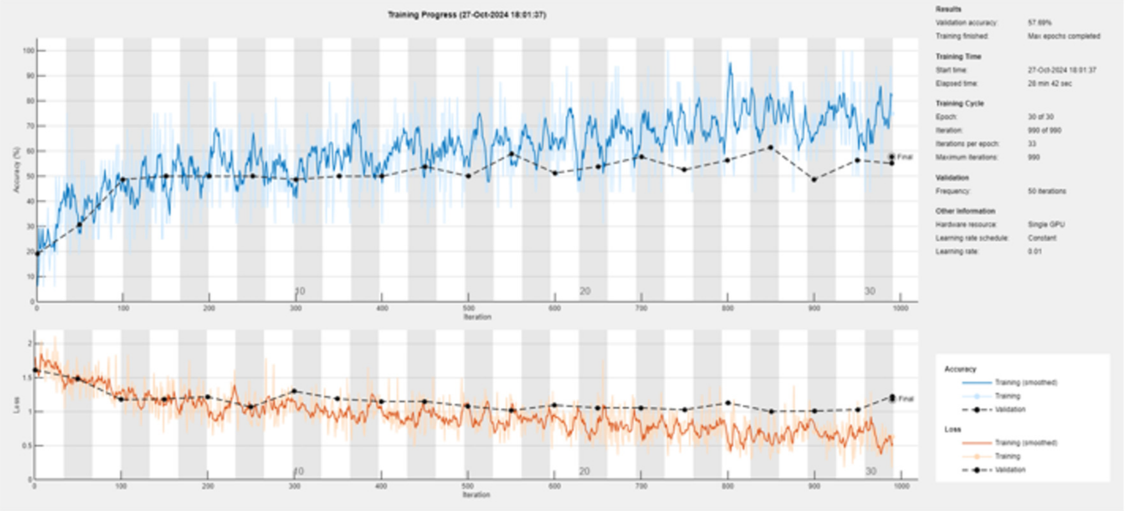
Training Cycle
Epoch: 5 of 5
Iteration: 40 of 40
Iterations per epoch: 8
Maximum iterations: 40

Validation
Frequency: 50 iterations

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.01

Learn more

Fig.A6: *Accuracy and loss functions of VGG-19.*



Results
Validation accuracy: 57.69%
Training finished: Max epochs completed

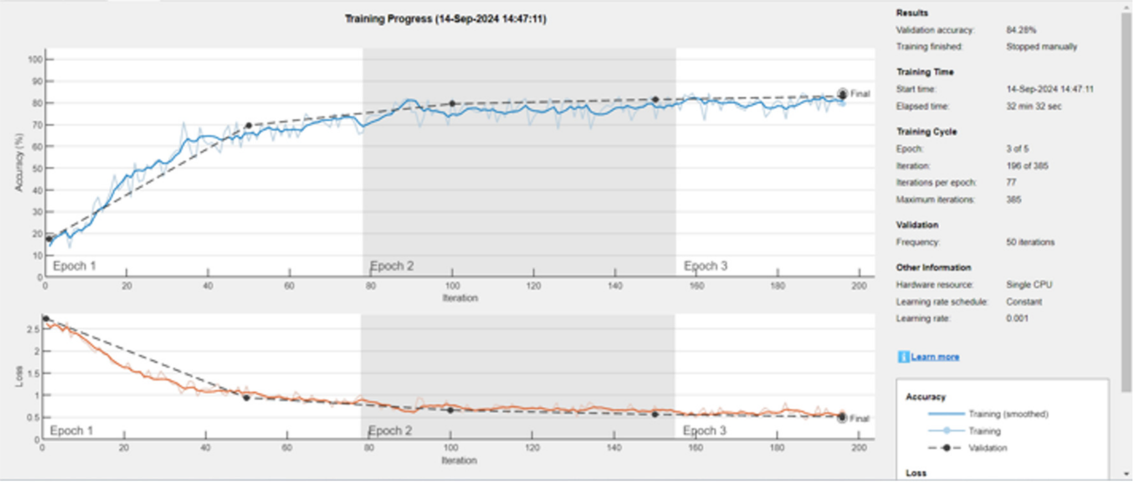
Training Time
Start time: 27-Oct-2024 18:01:37
Elapsed time: 28 min 42 sec

Training Cycle
Epoch: 30 of 30
Iteration: 990 of 990
Iterations per epoch: 33
Maximum iterations: 990

Validation
Frequency: 50 iterations

Other Information
Hardware resource: Single GPU
Learning rate schedule: Constant
Learning rate: 0.01

Fig.A7: *Accuracy and loss functions of Inception-ResNet-v2.*



Results
Validation accuracy: 84.28%
Training finished: Stopped manually

Training Time
Start time: 14-Sep-2024 14:47:11
Elapsed time: 32 min 32 sec

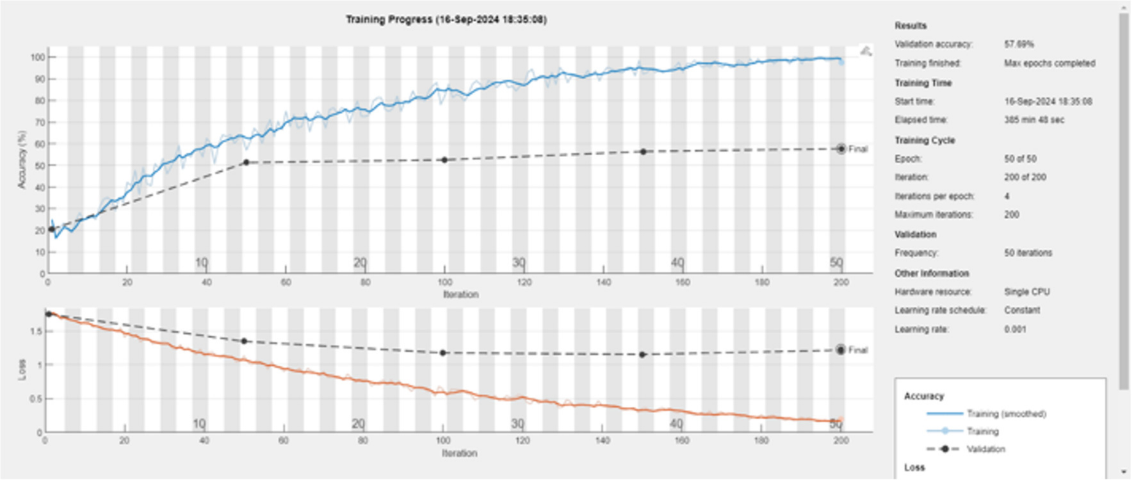
Training Cycle
Epoch: 3 of 5
Iteration: 196 of 385
Iterations per epoch: 77
Maximum iterations: 385

Validation
Frequency: 50 iterations

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.001

Learn more

Fig.A8: *Accuracy and loss functions of Inception-v3.*



Results
Validation accuracy: 57.69%
Training finished: Max epochs completed

Training Time
Start time: 16-Sep-2024 18:35:08
Elapsed time: 385 min 48 sec

Training Cycle
Epoch: 50 of 50
Iteration: 200 of 200
Iterations per epoch: 4
Maximum iterations: 200

Validation
Frequency: 50 iterations

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.001

Accuracy
Training (smoothed)
Training
Validation
Loss

Fig. A9: *Accuracy and loss functions of NASANet-Mobile.*

Porawat Visutsak received PhD in Computer Science from KMITL. He was a Senior Visiting Scholar in the Computer Science and Technology Program, School of Computer Science and Technology, Beijing Institute of Technology, under the China Scholarship Council (CSC). He is now Associate Professor in the Department of Computer and Information Science, Faculty of Applied Science, KMUTNB.

Kavin Treeraphapkajondet graduated with a Bachelor's degree in Computer Engineering from Prince of Songkla University and is currently pursuing a Master's degree in Computer Science at King Mongkut's University of Technology North Bangkok. He now works as a computer academic at the Legal Execution Department, focusing on software development and server administration.

Visaroot Sakphet holds a Bachelor's degree in Mechatronics Engineering from Prince of Songkla University, Thailand. He is currently pursuing a Master's degree in Computer Science at King Mongkut's University of Technology North Bangkok, Thailand. Additionally, he works for a government agency under the Ministry of Public Health, Thailand, where he is responsible for software development, interagency data integration systems, and IT system administration.

Wachirawit Nitinuntatip is an AI and software engineer at Baksters. He graduated with a degree in Computer Engineering from Assumption University and is currently pursuing a Master of Science in Artificial Intelligence for Business Analytics at King Mongkut's Institute of Technology Ladkrabang. He specializes in artificial intelligence, cloud computing, backend development, and computer vision. Wachirawit has experience in designing and deploying intelligent systems that support business decision-making and operational efficiency. He is skilled in developing scalable backend services, integrating AI models into production environments, and leveraging cloud platforms to optimize performance. His research interests include machine learning applications, computer vision technologies, and AI-driven business analytics solutions.

Pawwinkan Satthong is currently pursuing a Master of Science in Artificial Intelligence for Business Analytics at King Mongkut's Institute of Technology Ladkrabang. He now works as a freelance developer.

Tanajak Tongbai is a Senior Developer at Goodwin Corp. He holds a Bachelor's degree in Computer Science from Chulalongkorn University and is currently pursuing a Master's in Artificial Intelligence for Business Analytics at King Mongkut's Institute of Technology Ladkrabang. Tanajak specializes in Python programming, with expertise in API backend development, server DevOps, and deployment processes. He is proficient in cloud technologies, including AWS and Google Cloud services, and skilled in creating scalable backend solutions and implementing efficient deployment workflows. He combines strong technical foundations with emerging AI knowledge to deliver innovative business solutions. Tanajak is passionate about leveraging technology to solve complex business challenges through practical applications of artificial intelligence.

Duongduen Ongrungruaeng graduated with a Bachelor's degree in Information Technology from King Mongkut's University of Technology North Bangkok in 2020. For her graduation project, she created a Machine Learning model designed to manage inventory levels for a company, aiming to reduce warehouse storage costs. She is currently pursuing a Master's degree in Artificial Intelligence and Business Analytics at King Mongkut's Institute of Technology Ladkrabang. Currently, Duongduen works as a Software Engineer at a semiconductor company, where her primary role involves writing software for testing IC parts. She has also developed an AI system to help detect damaged or abnormal parts on the production line.

Atiwitch Juntra is currently pursuing a Master of Science in Artificial Intelligence for Business Analytics at King Mongkut's Institute of Technology Ladkrabang. He graduated with a degree in Accountancy from Chulalongkorn University and formerly worked in business development at Thrive Venture Builder. He specializes in marketing, finance, and design thinking. Atiwitch is skilled in developing and building Thailand's innovation ecosystem by supporting corporates, startups, and entrepreneurs through training programs, mentorship, and venture operations.

Watcharaporn Aiamlamai holds a Bachelor's degree in Electronic Engineering from King Mongkut's Institute of Technology Ladkrabang, Chumphon Campus. Currently, she is pursuing a Master's degree in Artificial Intelligence and Business Analytics at King Mongkut's Institute of Technology Ladkrabang. She is employed as a Supplier Quality Engineer at Western Digital Storage Technologies (Thailand) Ltd., where she collaborates with suppliers on various improvement projects. Her work focuses on AI initiatives, Predictive Quality, digital transformation, smart factory, and smart lab developments. In her role, she leverages her proficiency in Python, machine learning, deep learning, and project management to lead technical initiatives, develop AI-driven solutions, and optimize processes for greater efficiency and innovation across the supply chain.

Issares Sungwanna is currently pursuing a Master of Science in Artificial Intelligence for Business Analytics at King Mongkut's Institute of Technology Ladkrabang. He now works as a freelance developer.

Prapaporn Phetrak graduated with a Bachelor's degree in Statistics from the Faculty of Science and Technology, Thammasat University. Currently, she is pursuing a Master's degree in Artificial Intelligence for Business Analytics at King Mongkut's Institute of Technology Ladkrabang. She works in the customer service sector at Thailand Post Co., Ltd. Her role involves leveraging her expertise in Python for data analysis and developing machine learning models. Prapaporn applies these skills to gain deeper insights into customer behavior and to effectively support business strategy development.

Ponrudee Netisopakul is an Associate Professor at the Faculty of Information Technology at King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand. Her current research interests include Natural Language Processing, Artificial Intelligence, Data Science, Data Mining, and Knowledge Engineering. She is the founder and currently the director of the Knowledge Management and Knowledge Engineering Laboratory (KMaKE). Professor Netisopakul is also a member of renowned academic and research associations, including: IEEE Society, Artificial Intelligence Association of Thailand (AIAT), The Association of Researchers, The Computer Association of Thailand Under the Royal Patronage and The Science Society of Thailand Under the Patronage of His Majesty the King.

Keun Ho Ryu is with Chungbuk National University, Chungbuk, Republic of Korea. He is Adjunct Professor of Faculty of Information Technology, Co-Director of Research Group, Data Science Laboratory, Ton Duc Thang University, Vietnam.