Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 10, No. 3, September 2024, pp. 608-624
ISSN: 2338-3070, DOI: 10.26555/jiteki.v10i3.29540

A Comparative Study of Modern Activation Functions on Multi-Label CNNs to Predict Genres Based on Movie Posters

Ahmad Zein Al Wafi, Anan Nugroho
Universitas Negeri Semarang, Sekaran, Semarang 50229, Indonesia

ARTICLE INFO

Article history:

Received July 23, 2024
Revised August 30, 2024
Published September 26, 2024

Keywords:

Convolutional Neural Network;
Activation Function;
Multi-Label Classification;
Movie Poster Genre

ABSTRACT

Categorization of images based on their visuals into various genres has a crucial role in the recommendation system. However, multilabel classification poses significant challenges due to the complexity of assigning multiple labels to each instance. This study contributes to the understanding of how activation functions influence the efficiency and accuracy of multilabel CNNs and provides practical insights for selecting appropriate functions in movie poster classification tasks. This investigation focused on identifying the activation function that provided the fastest convergence, highest accuracy, and lowest computational cost or training time. The results show that the Leaky ReLU activation function achieved the fastest convergence and highest training accuracy with an top accuracy of 99.7% and GELU demonstrated the highest validation accuracy at 91.5% across the training iteration. Softplus showed convergence characteristics at epoch 14 while other in 30. The computational cost analysis revealed that ReLU was computationally efficient with training time of 1896 seconds. Overall, the Leaky ReLU activation function is identified as the most effective for multilabel CNNs, balancing convergence speed, accuracy, and computational cost.

Corresponding Author:

Anan Nugroho, Universitas Negeri Semarang, Sekaran, Semarang 50229, Indonesia
Email: anannugroho@mail.unnes.ac.id

1. INTRODUCTION

Recommendation systems play a crucial role in enhancing user experience and satisfaction across various platforms, particularly in streaming services like Netflix [1]. Recommender systems are computer programs and algorithms that analyze user habits and characteristics to propose products that are most attractive to consumers based on multiple criteria [2]. One of Netflix's innovative approaches involves generating personalized artwork for its members to help them discover content that matches their interests [3]. This process begins with the creation of static images from source videos, crafted into raw artwork, then ranked based on their aesthetics, creativity, and diversity in objects to accurately represent the content [4]. This ranking ensures that each member receives artwork that is not only relevant but also personalized to their specific preferences.

The importance of genre prediction using image data becomes evident in this context, as it allows for the categorization of images based on their aesthetics into various genres. Movie recommender systems can develop genre similarity and preferred genres by using natural language processing algorithms to extract relevant aspects and expressions [5]. By doing so, Netflix can generate personalized feedback for members, aligning with their recommendations and viewing history [6]. Automatically categorizing movie posters into numerous genres enables efficient categorization and organizing, as well as opportunities for personalized recommendations, genre-focused marketing strategies, and improved user experiences in the film industry [7]. This recommendation system not only enhances the visual appeal but also improves the likelihood of users engaging with the content, thereby fostering higher satisfaction and retention rates [8]. Therefore, the study of information retrieval at visual data is very important for solving this problem.

The study of extracting picture features has received significant attention in recent years due to its potential to enhance numerous tasks, including image retrieval and image captioning [9]. As an emerging trend, deep learning methods have made significant progress in addressing this challenge [10]. While utilizing the visual datasets, Convolutional Neural Networks (CNNs) are widely used algorithms that have proven to be highly effective in analyzing images in various scenarios such as art [11], medicine [12], agriculture [13], transportation [14], Infrastructure [15], remote sensing [16], and biometrics [17]. CNNs have shown remarkable success in single-label image classification proved by various remarkable novel architecture [18]. The ability of CNNs to automatically learn hierarchical feature representations from raw image data has been a significant breakthrough, leading to state-of-the-art performance in numerous computer vision tasks [19]. However, the challenge amplifies in multi-label image classification due to the presence of multiple objects and complex relationships within a single image [20], [21]. Traditional CNNs are primarily designed for single-label classification and may struggle with the intricacies of multi-label data [22]. The main issue lies in the fact that a single image may contain multiple relevant labels, requiring the model to detect and accurately classify each one independently while considering their interdependencies [23]. This task in nature also gives additional complexities such as imbalanced and noisy labels [24].

To address these limitations in multi-class classification tasks, several methods have been proposed to enhance CNNs for multi-label image classification. One approach involves using advanced pooling strategies like Hypotheses-CNN-Pooling (HCP), which aggregates CNN outputs from different object segment hypotheses to produce multi-label predictions [25]. Another approach is the use of Multi-function Convolutional Neural Networks (MCNNs) that apply different activation functions across various neurons, thereby improving classification performance by leveraging diverse activation patterns [26]. Additionally, optimizing loss functions to handle weakly-labeled datasets and label noise has been shown to further improve multi-label classification outcomes [27]. Deep multi-modal CNN (MMCNN) is also being used to decompose the image into a bag of instances shown to be successful even using a pre-trained single-label classification network [28]. Other works for this task are using various training schemes to find the methods for optimal combination of loss weights, mitigate overfitting by enabling early sharing of learnable features, and reconstruct the input accurately [29].

Despite several methods being proposed to enhance CNNs for multi-label image classification, such as advanced pooling strategies like Hypotheses-CNN-Pooling and the use of Multi-function Convolutional Neural Networks, there remain several limitations. This method introduces additional computational complexity and leads to longer training times and increased resource requirements. Implementation of model structures requires complex strategies and extensive experimentation to optimize their parameters effectively. This current approaches primarily focus on specific model structured aspects such as pooling techniques or label noise optimization, without fully addressing how other simple hyperparameter configurations can improve the performance and robustness of multi-label classification models, such as model activation function. Existing studies [25], [26], [27], [28], [29] often lack a comprehensive evaluation of modern activation functions and their potential benefits in multi-label settings, leaving a gap in understanding their impact on model performance. Various modern activation functions offer smoother and more expressive activation curves that might better capture complex patterns but have not been exhaustively compared across various CNN architectures. Additionally, practical implications of these functions in real-world scenarios are often neglected, limiting insights into their effectiveness with complex datasets. Also, there is often a lack of comparative analysis within the same framework, making it challenging to identify the most beneficial activation functions for multi-label tasks. These gaps highlight the need for a detailed investigation into how modern activation functions can impact multi-label classification models' performance.

This paper introduces a new approach that incorporates modern activation functions into CNN architectures to enhance multi-label image classification performance. An activation function is utilized in CNNs to enhance the representation capability through nonlinear operations and achieve state-of-the-art outcomes [30]. Among the different state-of-the-art CNN architecture, ReLU (Rectified Linear Unit) is widely used due to its simplicity and effectiveness [31]. It activates neurons by outputting zero for negative inputs and passing positive inputs unchanged. However, this function has limitations as its gradient becomes zero for negative values, The CNN model may experience a phenomenon called "Dying ReLU" or vanishing gradient problem during the training process [32]. The introduction of novel activation functions has revitalized the scientific community's interest in neural networks, as they play a pivotal role in enhancing the expressive capabilities of artificial neural networks [33]. Various state-of-the-art studies for model architectures introduce various new activation functions that are more powerful than ReLU, such as Leaky ReLU, ELU, Swish, and Mish [34]. For instance, Swish as a modern activation function offers several theoretical advantages over ReLU, including a smoother non-linearity and better gradient flow, which can lead to improved performance

and convergence. Swish addresses some of the limitations of ReLU not causing a vanishing gradient problem and providing a more gradual activation curve that enhances the network's ability to learn more complex patterns [35]. It has been shown to outperform traditional activation functions like ReLU in terms of model accuracy in single-label image benchmark datasets [36]. The choice between these activation functions can significantly impact the model's learning dynamics and overall effectiveness in image classification tasks where nuanced pattern recognition is crucial [37].

This paper aims to fill the literature gap by investigating the impact of various modern activation functions for solving multi-label image classification tasks in movie poster datasets. The study offers a comparative evaluation of several activation functions within CNN models to identify the configurations that perform best for multi-label movie genre classification. This will include insight to improve model performance by identifying which activation functions will enhance accuracy, generalization, and efficiency in multi-label tasks. No other hyperparameters such as loss function, optimization function, and model architecture will be analyzed in this study to show a clearer impact on the activation function. The novel insights will cover how these activation functions influence critical factors such as convergence speed and computational cost to provide a deeper understanding of their effects on model optimization. The findings will deliver practical guidance for selecting activation functions in CNN architectures, contributing valuable knowledge that could shape future research and applications in multi-label classification. This work contributes knowledge about the impact of activation functions to the existing knowledge by systematically evaluating complex problems like multi-label image classification, thus addressing a critical area that has not been extensively explored.

2. METHODS

2.1. Dataset

The dataset of this work is taken from a study [9] which facilitates movie poster analysis. The movie poster data are obtained from the IMDB website as well as the associated metadata like movie genre. The dataset includes one poster for each Hollywood movie released from 1980 to 2015, resulting in a total collection of 8,191 poster images. The resolutions of these poster images vary from 89 × 132 to 300 × 581. The study collects genre information for each movie, which is evaluated across 25 different classes. The visual distribution of the dataset is shown in three categories, Fig. 1 shows the frequency of genre appearances per movie, Fig. 2 shows the distribution of all classes evaluated, and Fig. 3 shows the top 10 genre combinations in the dataset. The graph indicates that the dataset is unbalanced, with certain genres being overrepresented while others are underrepresented.

In analyzing the genre distribution and combinations within the movie poster dataset, several insights emerge that could significantly impact the development and performance of multi-label CNN models. The genre distribution reveals a predominant presence of Drama (3619 occurrences) and Comedy (2900 occurrences), with Action, Romance, and Crime following. This skewed distribution suggests that models trained on this dataset should be particularly adept at identifying these more common genres to achieve higher accuracy and relevance in genre classification. The frequency of genre combinations highlights that Drama and Comedy are the most frequently co-occurring genres, with combinations such as "Comedy, Drama" and "Comedy, Drama, Romance" being prominent. This expects that multi-label CNN models should be equipped to handle common genre pairs and triples effectively.

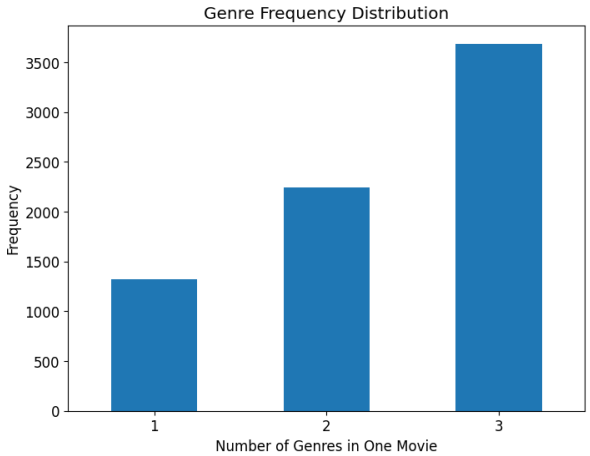

Fig. 1. Total genre in the movie frequency distribution

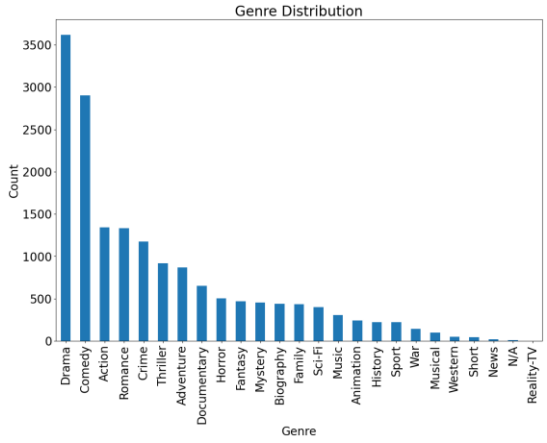

Fig. 2. Genre distribution

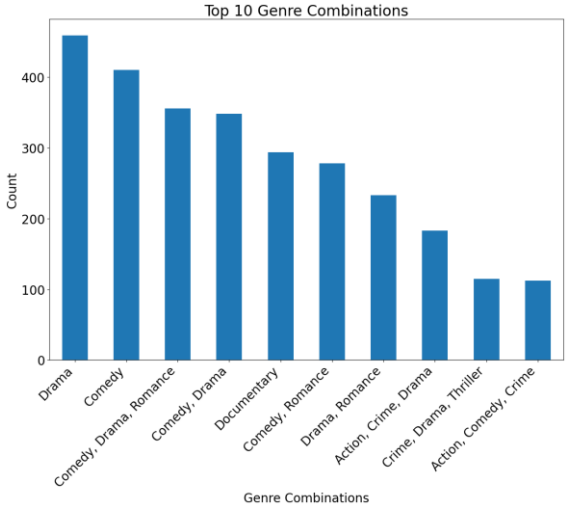

Fig. 3. Top 10 genres in one movie

2.2. Research Design

The research design for this study comprised several key stages to systematically investigate the comparative performance of different activation functions on multilabel CNNs. The flow of the research is illustrated in the block diagram of Fig. 4.

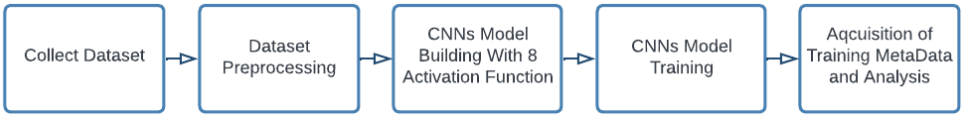

Fig. 4. Research flow

Initially, the dataset was obtained and subjected to a preprocessing phase to prepare it for model training. Afterward, a CNN model was constructed and trained using eight different activation functions. During the training, the data was continuously acquired to monitor performance metrics. The collected data was then analyzed to assess and compare the impact of each activation function based on the loss, accuracy, speed of convergence, best performance, and overall training time. This structured approach enabled a comprehensive comparison of the activation functions to identify which most effective and efficient one.

2.3. Data Preprocessing

Several preprocessing steps were attempted to ensure consistency and compatibility with CNN architectures. The movie poster images, originally varying in resolution from 89 × 132 to 300 × 581 pixels, were resized to a uniform dimension of 224 × 298 pixels. This resizing step was essential to standardize the

input size for the CNN models, facilitating effective training and evaluation across different architectures. Normalization was applied to the dataset to standardize the pixel values and aid the network in learning patterns more efficiently. This step helps in maintaining numerical stability during training and improves convergence rates by ensuring that the input data has consistent statistical properties.

The decision was made to exclude data augmentation in this study. While data augmentation is a widely used technique to enhance model generalization by artificially increasing the diversity of the training dataset, it was excluded to evaluate the raw performance of the models and activation functions without any artificial modifications. By avoiding augmentation, the study aims to assess the models' capabilities to handle unaltered, real-world data. This approach is crucial for understanding baseline performance and limitations, providing insights into how well the models and activation functions perform with the original data distribution.

The potential impact of excluding data augmentation on the study's results includes the possibility of reduced model generalization, as augmentation typically helps in improving robustness and adaptability to varied conditions. However, the focus on unaltered data allows for a clearer evaluation of the intrinsic strengths and weaknesses of the activation functions to ensure that performance metrics reflect the true capabilities of the models in handling real-world data scenarios. This understanding is essential for applications such as personalized artwork generation in recommendation systems, where real-world data performance is critical.

2.4. CNN Architecture

The model architecture employed for the multi-label genre prediction task is a traditional CNN architecture with 25 output neurons representing the class target. This model is designed to effectively extract and learn features from movie posters by utilizing multiple convolutional layers, batch normalization layers, pooling layers, and dropout layers for regularization. Each layer plays a crucial role in enhancing the model's ability to accurately predict multiple genres from a single image. The details of the architecture are shown in Table 1.

Table 1. Layer configurations of CNN models

Layer Order	Layers Type	Hyperparameter	Activation Function
1	Conv2D	Filter: 16, Kernel Size: (3,3)	Adjustable
2	BatchNormalization	-	-
3	MaxPooling2D	Pool Size: (2,2)	-
4	Dropout	Rate: 0.2	-
5	Conv2D	Filter: 32, Kernel Size: (3,3)	Adjustable
6	BatchNormalization	-	-
7	MaxPooling2D	Pool Size: (2,2)	-
8	Dropout	Rate: 0.2	-
9	Conv2D	Filter: 64, Kernel Size: (3,3)	Adjustable
10	BatchNormalization	-	-
11	MaxPooling2D	Pool Size: (2,2)	-
12	Dropout	Rate: 0.2	-
13	Flatten	-	-
14	Dense	Units: 128	Adjustable
15	BatchNormalization	-	-
16	Dropout	Rate: 0.5	-
17	Dense	Units: 25	Sigmoid

The architecture begins with an initial convolutional layer followed by batch normalization and max pooling, which helps in capturing basic spatial features while reducing the dimensionality of the data. This sequence of layers is the most common in various state-of-the-art CNN architecture [38]. The choice of filter sizes in the convolutional layers—16, 32, and 64—reflects a common practice in CNN design where smaller filters are used initially to capture basic features and larger filters are employed in subsequent layers to detect more complex patterns [39], [40]. Specifically, the initial layer with 16 filters focuses on basic textures and edges, while the later layers with 32 and 64 filters progressively capture more abstract and intricate features from the images. This hierarchical approach helps the network build a robust representation of the visual data [41].

The use of max pooling with a pool size of (2,2) reduces the spatial dimensions of the feature maps, which helps in decreasing the computational load and mitigating overfitting by abstracting the feature representation [42]. The dropout rates, set at 0.2 after each pooling layer and 0.5 before the final output, are chosen to strike a balance between regularization and model capacity. Dropout is employed to prevent overfitting by randomly deactivating a fraction of neurons during training, which helps the model generalize better to unseen data [43].

Batch normalization is included to stabilize and accelerate training by normalizing the inputs to each layer. This technique helps in reducing internal covariate shifts and can improve convergence rates, making the network more robust to variations in input data [44]. The model also includes a fully connected dense layer that further processes the flattened feature maps before the final dense layer outputs the genre predictions. The adjustable activation function will be filled with various functions explained in the next section. In general, the matrix results from operations on the network are resized three times using pooling and then flattened as visualized in Fig. 5.

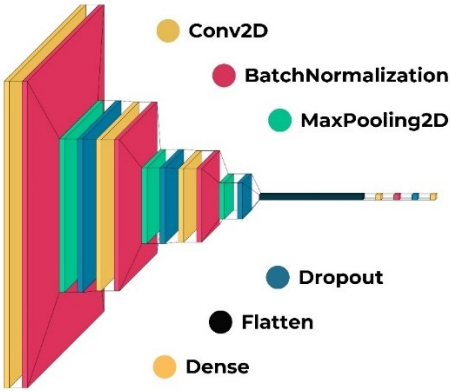

Fig. 5. Visualization Result Each Layer of the CNN model

2.5. Activation Functions

Activation functions play a critical role in the performance of CNNs, especially in complex tasks like multi-label genre prediction from movie posters. They introduce non-linearity into the network, enabling it to learn and model complex data patterns. Below is a brief description of each modern activation function and its relevance to this work:

1. **Exponential Linear Unit (ELU)**: The Exponential Linear Unit (ELU) activation function aims to speed up the learning process by bringing the mean activations closer to zero, which improves the convergence rate. Unlike the ReLU function, which can result in dead neurons (neurons that stop learning completely), ELU has negative values that help to push the mean unit activations closer to zero [45]. This characteristic allows for faster and more efficient learning, especially in deeper neural networks.
2. **Gaussian Error Linear Unit (GELU)**: The Gaussian Error Linear Unit (GELU) activation function introduces smooth non-linearity, combining the benefits of ReLU and dropout regularization [46]. GELU applies a smoother transition for negative values compared to ReLU, which helps in retaining information and avoiding the zero-gradient problem. This smooth and probabilistic approach enhances the model's ability to capture the nuances of data, leading to improved learning dynamics and performance.
3. **Leaky Rectified Linear Unit (Leaky ReLU)**: The Leaky ReLU activation function addresses the "dying ReLU" problem, where neurons can stop learning entirely when the input is negative. Leaky ReLU allows a small, non-zero gradient for negative inputs, which ensures that the information flow remains active across the network, even for neurons that would otherwise be inactive with standard ReLU [47]. This small "leak" helps in maintaining a more robust gradient during backpropagation, aiding the training of deeper networks.
4. **Mish**: Mish is a smooth, non-monotonic activation function that has been shown to outperform traditional activation functions like ReLU in some scenarios [48]. Mish provides a combination of desirable properties from both the smoothness of GELU and the flexibility of Swish. The function is defined as $x \cdot \tanh(\text{softplus}(x))$, where softplus is $\log(1+e^x)$. This formulation helps in retaining information throughout the network, leading to better performance in complex learning tasks.
5. **Rectified Linear Unit (ReLU)**: The Rectified Linear Unit (ReLU) activation function is one of the most widely used activation functions in deep learning due to its simplicity and effectiveness [49]. ReLU introduces non-linearity by outputting zero for all negative input values and passing through positive input values unchanged. This helps in avoiding the vanishing gradient problem, allowing for faster and more efficient training of deep networks. However, it can suffer from the "dying ReLU" problem where neurons can become inactive.
6. **Scaled Exponential Linear Unit (SELU)**: The Scaled Exponential Linear Unit (SELU) is designed to induce self-normalizing properties within the network [50]. This means that during the forward pass, the

mean and variance of each layer's output remain constant, which helps in maintaining stable gradients. SELU achieves this by scaling the outputs of ELU with specific parameters (λ and α), ensuring that the activations are centered around zero with unit variance. This self-normalizing behavior aids in the efficient training of deep ne tworks.

7. **Softplus**: Softplus is a smooth approximation of the ReLU function, providing a continuous and differentiable function that mitigates the issues of zero gradients faced by ReLU. Defined as $\log(1+e^x)$, Softplus ensures that all input values are mapped to positive output values, which helps in maintaining a positive and smooth gradient [51]. This function is particularly useful in scenarios where smoothness and differentiability are crucial for the learning process.
8. **Swish**: The Swish activation function is a smooth, non-monotonic function that has been shown to outperform ReLU in various deep-learning tasks. Defined as $x \cdot \sigma(x)x$, where $\sigma(x)$ is the sigmoid function [52], Swish allows a smooth transition for negative inputs, unlike the sharp cut-off in ReLU. This smoothness helps in retaining information and improving the gradient flow, leading to better performance and faster convergence in training deep neural networks.

2.6. Training Procedure

The training procedure for the multilabel CNNs is conducted using an 80-20 split of the dataset, with 80% allocated for training and 20% for validation, without shuffling the data. The data split is to access overfitting characteristics that are monitored through validation loss and accuracy metrics during training. The model's performance on a separate validation set, distinct from the training data, provided insights into its generalization capabilities and identified if it was learning noise rather than true patterns. The models are optimized using the Adam optimizer was considered to have performed better than others in the study [53] for image classification. In this study, Adam optimizer uses the following hyperparameters: a learning rate of 0.001, $\beta 1$ set to 0.9, $\beta 2$ set to 0.999, and ϵ set to 1×10^{-7}. Binary Crossentropy was used in [40] to calculate the loss between true labels and predicted labels for multilabel classification tasks. Binary Cross entropy is defined in [54] employed as the loss function in this study. This is because multi-class classification tasks are resulting binary classes and so the evaluation will be well-suited using binary [55]. The network will be trained for 100 epochs with a batch size of 32. This training procedure is not doing any hyperparameter tuning process and the parameters are not changed manually during the training process. However, the Adam optimizer has an adaptive learning rate that will be adjustable automatically during the training process [56].

2.7. Performance Metric

The performance of the multi-label CNNs was assessed using Binary Crossentropy loss and Binary Accuracy, which are well-suited for multi-label classification tasks. Binary Crossentropy loss was chosen because it effectively handles the problem of predicting multiple independent labels per instance, calculating the loss for each label independently, and then averaging across all labels. This metric is particularly useful in multi-label scenarios where each label is considered a separate binary classification problem, allowing the model to learn and optimize for each label's presence or absence.

Binary Accuracy is defined as the proportion of correct predictions (true positives and true negatives) out of the total number of predictions [7] are used to measure the model's performance. This metric is appropriate for multi-label classification as it provides a clear indication of how well the model identifies the correct labels for each instance, considering all possible labels. It reflects the model's ability to correctly predict multiple genres from movie posters, which is crucial for evaluating its effectiveness in a real-world application.

Additionally, training time was monitored to assess the computational cost of the model. This includes recording the duration of the overall training process to evaluate the efficiency and practicality of different activation functions and architectural configurations. Training time is a critical factor in understanding the trade-offs between model performance and resource consumption, providing insights into the feasibility of deploying the model in practical scenarios where computational resources may be limited.

2.8. Software and Tool

In this study the multi-label CNNs model will be developed and trained using Python programming language, leveraging TensorFlow framework and Keras API for the implementation. TensorFlow provided the backend computational framework for a heterogeneous development environment [57], while Keras a high-level API facilitated more simple code to construct TensorFlow backend for training the neural network model. The development and training process are conducted on Google Colab which is a cloud service fully configured with the leading artificial intelligence libraries and Graphical Processing Unit (GPU) accelerating deep learning applications [58].

The computational environment utilized in this study included 56GB of RAM and an NVIDIA Tesla T4 GPU. This computational resource has high performance in deep learning tasks by enabling parallel processing and efficient handling of large-scale computations [59]. The hardware setup supports higher batches of data and higher training speed when dealing with large datasets and complex models. However, the use of cloud-based services like Google Colab also introduces potential limitations, such as occasional resource sharing from server cluster management might impact the consistency and reliability of computational resources [60].

3. RESULTS AND DISCUSSION

This section presents the findings of the investigation performance from various activation functions used in the training of multilabel CNNs. The analysis focuses on three key areas: the speed of convergence for each activation function, the best performance achieved in terms of accuracy, and the computational cost associated with training each network. The analysis results obtained are compared with other studies and in the discussion several recommendations will be given to determine a good activation function according to needs.

3.1. Speed of Convergence

To determine which activation function reached convergence first, the training process is monitored across 100 epochs for each of the eight activation functions. Convergence is defined as the point where the loss function stabilized and further training yielded minimal improvements. Fig. 6 illustrates all epochs on the CNNs model, Fig. 7 presents more zoom in binary accuracy score to see more details.

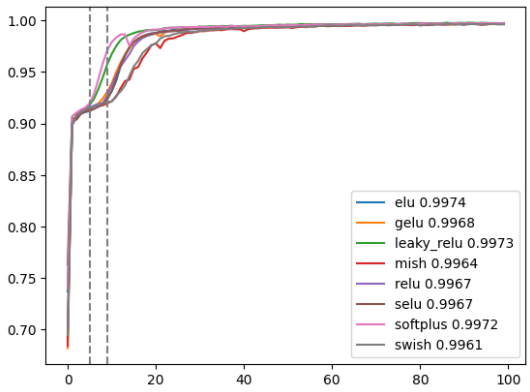

Fig. 6. Chart of binary accuracy score in the training set

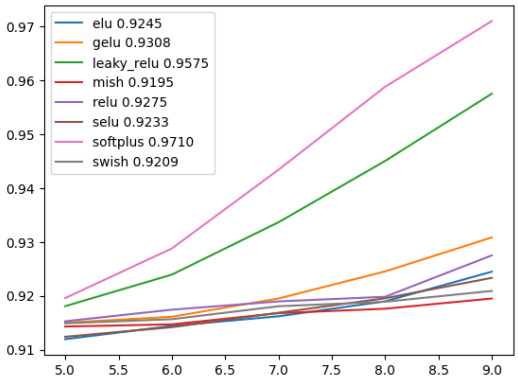

Fig. 7. Detail binary accuracy score in training set at epoch 5 - 9

Fig. 6 suggests that the Softplus activation function exhibits the fastest convergence optimizing the network parameters. The rapid convergence observed with Softplus could initially indicate superior optimization capabilities. However, the convergence trends shown in Fig. 7 suggest that after 30 epochs, the differences in convergence speed among the activation functions diminish. The initial differences in convergence speed might be influenced by factors such as the specific inherent characteristics of the activation functions, rather than representing substantial performance advantages.

The observed similarity in convergence after 30 epochs suggests that while Softplus may offer quicker initial convergence, the ultimate performance across activation functions tends to equalize over time. This implies that the choice of activation function may influence the rate of convergence, but its impact on overall model performance may be less pronounced in the longer term. For more detailed expanations, Fig. 8 provided result of ANOVA test applied to the binary accuracy over 100 epochs of training, presenting both the F-statistic and p-value across epochs.

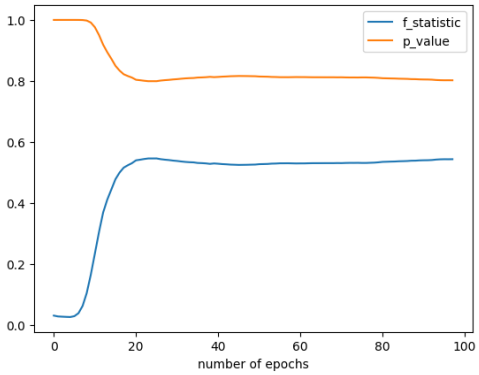

Fig. 8. ANOVA test of binary accuracy

The F-statistic (blue line) initially increases sharply and remainly same around 30 epochs indicating that variability among accuracy measurements becomes more distinct early in the training process. After this point, the F-statistic remains relatively stable, suggesting consistent model performance over subsequent epochs. The p-value (orange line) decreases rapidly from a value close to 1 at the start, reaching a stable level around 30 epochs as well, indicating statistical significance in the differences observed among the groups. This stability in both metrics suggests that after 30 epoch the model has sufficiently converged and further training may yield diminishing returns in terms of accuracy improvement.

3.2. Accuracy Performance

The evaluation of accuracy for each activation function was carried out by examining the binary accuracy metric on both the training set and validation set. Comprehensive tables that detail the performance of each activation function in terms of accuracy across different segments of the training process are presented below. Specifically, the table shows the percentage of accuracy at various stages: the bottom 10%, 25%, and 50% of epochs, as well as the top 10%, 25%, and 50% of epochs, for both the training set and validation set.

Table 2. Performance Statistics of Binary Accuracy in Training Set

index	elu	gelu	leaky relu	mish	relu	selu	softplus	swish
buttom10	0.89478	0.89191	0.90003	0.88979	0.89303	0.89794	0.91072	0.89145
buttom25	0.9432	0.94248	0.95201	0.93098	0.94197	0.94462	0.95658	0.93297
buttom50	0.96819	0.96766	0.9732	0.96074	0.9673	0.96874	0.97535	0.96218
top10	0.99188	0.99184	0.99447	0.9878	0.99146	0.9917	0.99425	0.9885
top25	0.99516	0.99497	0.99603	0.99367	0.99484	0.99489	0.99567	0.99406
top50	0.99616	0.99603	0.99685	0.99525	0.99593	0.9959	0.99644	0.9954
top1	0.99742	0.99726	0.99777	0.99689	0.99715	0.99702	0.99736	0.9965

Table 3. Performance Statistics of Binary Accuracy in Validation Set

index	elu	gelu	leaky relu	mish	relu	selu	softplus	swish
buttom10	0.88463	0.90528	0.89971	0.90209	0.90471	0.8444	0.83086	0.90232
buttom25	0.89891	0.90692	0.90216	0.90436	0.90565	0.88086	0.87551	0.90383
buttom50	0.90407	0.90779	0.90364	0.90543	0.90638	0.89387	0.89096	0.90479
top10	0.91005	0.90937	0.90611	0.90774	0.90793	0.90789	0.90714	0.90664
top25	0.91037	0.90964	0.90657	0.90811	0.90826	0.90843	0.90752	0.907
top50	0.91094	0.91012	0.9073	0.90893	0.90883	0.90921	0.90807	0.90763
top1	0.91545	0.91556	0.91504	0.91487	0.91336	0.91597	0.91314	0.91303

To gain deeper insights into performance analysis, the table data was visualized to highlight the accuracy percentages at various stages of the training process. Graphically representing the bottom 10%, 25%, and 50%,

as well as the top 10%, 25%, and 50% accuracy metrics for both the training and validation sets, allowing for clearer identification of trends and patterns. Fig. 9 and Fig. 10 facilitated a more intuitive comparison of the activation functions, aiming to observe their performance dynamics and stability throughout the training epochs.

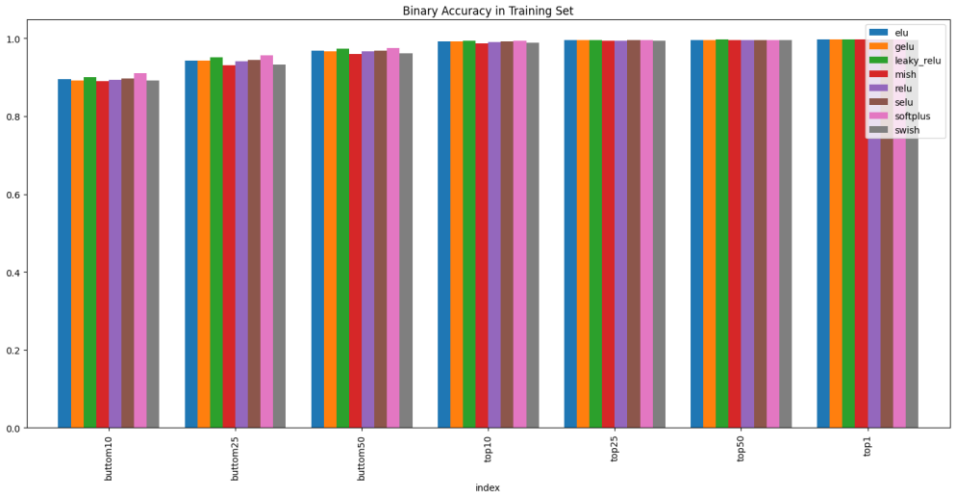

Fig. 9. Performance statistics of the training set

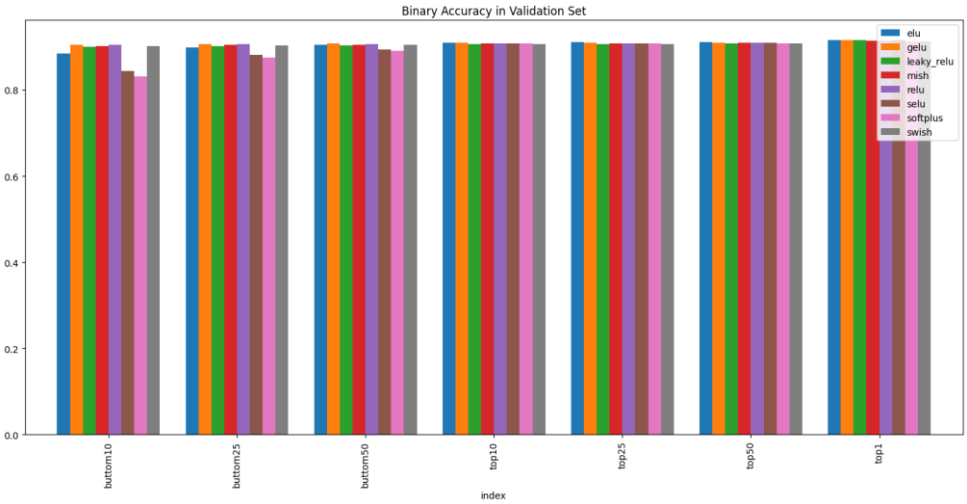

Fig. 10. Performance statistics of the validation set

From the performance statistics Table 2 and Table 3 also the accompanying visualization of the training and validation set accuracy metrics, several insights can be drawn regarding the performance of the different activation functions. In the training set, Leaky ReLU consistently outperformed other activation functions across all stages, including the bottom and top percentages of epochs, indicating strong and stable performance throughout the training process. It achieved the highest overall accuracy at 0.99777. Softplus also showed strong performance, particularly in the bottom 10%, 25%, and 50% epochs, indicating its effectiveness during the initial stages of training. ELU and GELU demonstrated competitive performance, especially in the higher percentages of epochs, suggesting good convergence properties.

In the validation set, Softplus showed the lowest accuracy in the bottom 10%, 25%, and 50% epochs, indicating it might be less effective during the early training stages of the validation set. ELU, GELU, and Leaky ReLU maintained relatively high accuracy across all validation stages, with GELU achieving the highest overall validation accuracy at 0.91556. Mish and ReLU provided consistent and competitive performance but slightly trailed behind the top-performing functions in overall validation accuracy. Overall, Leaky ReLU emerged as the best-performing activation function in terms of training accuracy, while GELU was slightly

better on the validation set. To see more detail about the indication of overfitting, Fig. 11 shows the performance difference between the training set and the validation set.

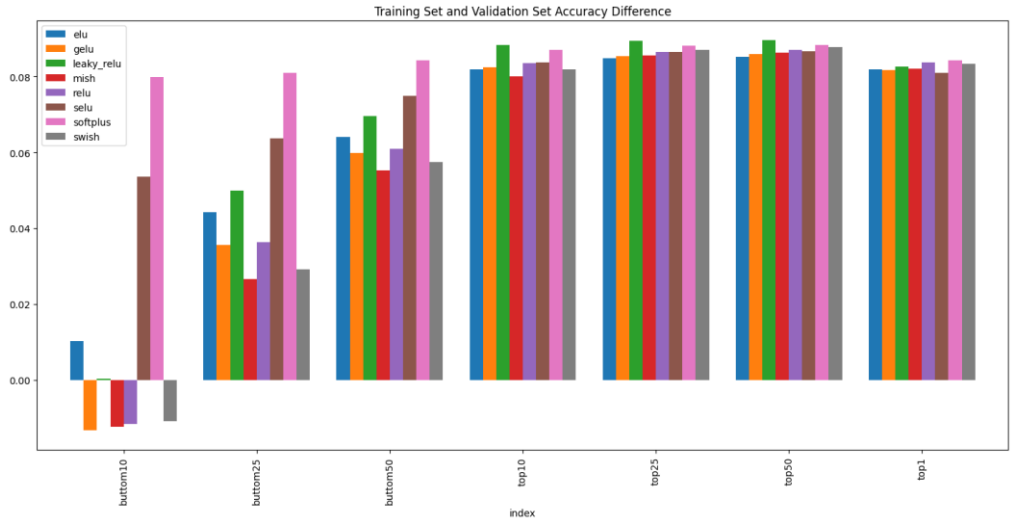

Fig. 11. Performance statistics of the validation set

The analysis of the delta or difference between training accuracy and validation accuracy reflects the difference between performance on the training set and the validation set and highlights how well the model generalizes to unseen data. The data reveals varying levels of overfitting across different activation functions. Notably, activation functions like ELU and Mish exhibit smaller delta values in the top-performing categories (e.g., top 10, top 25, top 50), suggesting that they maintain a more balanced performance between training and validation sets. This indicates that these activation functions are less prone to overfitting and generalize better, as their performance on the validation set closely matches their performance on the training set.

In contrast, activation functions such as SELU and Softplus show larger delta values, particularly in the lower-performing categories (e.g., buttom10, buttom25). These larger differences suggest that models using these activation functions may be experiencing overfitting, as they achieve higher accuracy on the training set but perform comparatively worse on the validation set. This disparity could imply that these functions lead to models that fit the training data too closely, capturing noise rather than generalizable patterns. The GELU and Swish functions show moderate delta values across different performance levels, suggesting a balanced approach to overfitting. They offer a reasonable trade-off between training and validation performance, but still, their behavior in specific scenarios should be monitored to ensure they do not overfit or underfit the data excessively.

3.3. Computational Cost

The computational cost of training the network with each activation function was measured in terms of total training time. From the hardware, computational cost analysis revealed that ReLU required the least amount of training time, completing the training process in ReLU hours. This indicates that ReLU is computationally efficient, minimizing the time and resources needed for model training. Detailed computational costs for each activation function are presented in Table 4 and visualized in Fig. 12.

Table 4. Models training time

Model's Activation Function	Training Time (seconds)
ReLU	1896
ELU	1976
Leaky ReLU	1986
SELU	1991
SoftPlus	2044
Swish	2534
Mish	2689
GELU	3052

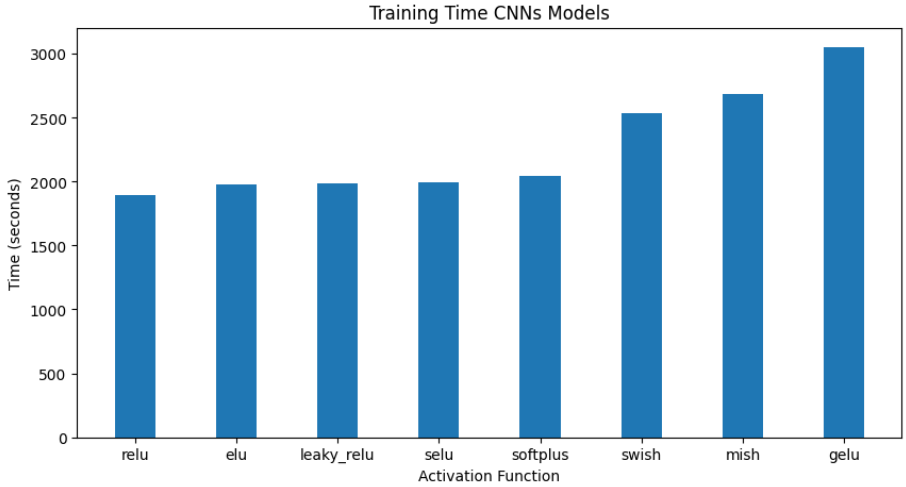

Fig. 12. Visualization of CNNs models training time

3.4. Recommendations and Limitations

These findings highlight the importance of selecting appropriate activation functions based on the specific needs and stages of training for multilabel CNN models on movie poster analysis cases. As already mentioned, SoftPlus emerges as the top performer in terms of convergence speed, closely followed by leaky ReLU. However, SoftPlus and SELU exhibited signs of potential overfitting to the validation dataset due to significant differences between training set accuracy and validation set accuracy. To make safer activation function choices, consideration should be focused be a that has>90% validation accuracy in the bottom 50% of the training process as it shows effectiveness in maintaining generalization. Conversely to SoftPlus, GELU could offer better performance by adapting to varied data distributions. The discrepancies between training and validation performances, particularly with SoftPlus, suggest that this activation function might not generalize as effectively as GELU. Such discrepancies can impact practical applications where model robustness and generalization to new, unseen data are critical. This highlights the need for careful selection of activation functions, taking into account both convergence characteristics and generalization performance.

The analysis of training times reveals notable differences in computational expense among the various activation functions, with some functions like GELU and Mish being more computationally intensive compared to others such as ReLU and ELU. However, after a very extensive and time-consuming training process, the experiment shows there are no significant final performance differences among diverse activation functions, as also mentioned in the study [61]. In the top 1, all of the activations have very similar results, but the difference in the training time can be up to 50% than the efficient one. In detail, while having very high computational resources, GELU and Mish may offer superior performance in terms of accuracy or convergence speed, their higher computational cost could pose challenges in scenarios requiring rapid model training or deployment on resource-constrained environments. While discussing cost-benefits, activation functions like ReLU demonstrate lower training times that provide a more cost-effective solution but very slightly sacrifice performance advantages offered by more complex functions.

Due to significant differences in computational resources, with the complexity of the multi-label classification task, it is suggested to use the Leaky ReLU for the first experiment, as also advised in the study [41]. The concept of ReLU in the Leaky ReLU function strikes a balance between convergence speed and accuracy performance. Moreover, the simple rectifying concept requires very light computational cost [42] shows this function typically requires the least amount of training time compared to more complex activation functions. Other studies have also shown that Leaky ReLU outperforms complex activation functions in terms of accuracy and training speed [43]. More extended concept ReLU that have adjustable parameters make Leaky ReLU solve vanishing gradient problem and have better performance result [44]. However, no activation function consistently outperforms the others [45], choice of activation function should be guided by the specific characteristics of the dataset, including the degree of similarity between training and validation data, as well as considerations of training speed and model performance metrics.

However, the practical implication might not represent other broader cases than movie posters considering that the dataset consists solely of movie posters without any augmentation. Addressing how training procedure variability such as adjustments to learning rates or different batch sizes might influence the performance of activation functions is important. Variations in these parameters can significantly impact model training

dynamics and overall performance [66], [67]. For instance, higher learning rates might accelerate convergence but could also lead to instability, while different batch sizes can affect gradient estimates and training stability [68], [69]. The data augmentation methods and the preprocessing methods applied can significantly influence the outcomes of machine learning models [70]-[76]. A more detailed exploration of how these factors interact with different activation functions could enhance the robustness of the comparison and improve the reliability of the findings.

Future research should explore the impact of augmented data, which introduces variations and enriches the dataset, potentially improving model robustness and generalization. Investigating a broader range of training configurations, including various architectures, hyperparameters, and optimization techniques, could provide deeper insights into activation function performance and stability in multi-label genre prediction tasks. This comprehensive approach would enable a more thorough understanding of how these factors interact and affect model performance across diverse datasets and scenarios

4. CONCLUSION

This study underscores the importance of selecting appropriate activation functions tailored to the specific needs and training stages of multilabel Convolutional Neural Networks (CNNs). Our findings reveal that activation function have significant different characteristics in few iteration such as 15 epoch, but with exhaustive training iteration such as 100 epoch a slight differences are expected. Function activation SoftPlus exhibits superior convergence speed, it is prone to overfitting in the absence of data augmentation. In contrast, Leaky ReLU offers a more robust alternative, particularly for training and validating identical datasets due to balanced performance across various stages of training. However, GELU demonstrates superior adaptability to diverse data characteristics, suggesting its potential for applications involving varied data distributions. Notably, the Leaky ReLU activation function is recommended for initial investigations due to its advantageous balance between convergence speed, accuracy, and cost-benefit analysis. While computational resources are not the issue, other functions such as GELU are recommended for adding more performance to the multi-label CNNs model.

The analysis of SoftPlus indicates a risk of overfitting but does not thoroughly discuss the impact of overfitting on model performance. As this work does not utilize any data augmentation, future work should delve deeper into overfitting issues and mitigation strategies to enhance model generalization. This includes exploring various augmentation methods to enrich the dataset and improve model robustness, like geometric transformation, color space adjustment, random noise injection, and more modern methods like synthetic data generation. The discussion also highlights the necessity of considering other factors that are not explored in this work such as network architecture, hyperparameter tuning, and optimization techniques. Variations in learning rates, batch sizes, and dropout rates might significantly impact the effectiveness of activation functions and overall model performance. Future studies that include a comprehensive analysis of these factors could provide a more nuanced understanding of their interplay with activation functions.

The study's findings contribute valuable insights into the field of CNN research, offering practical guidelines for activation function selection and emphasizing the dynamic relationship between activation functions and multi-label classification tasks. However, the applicability of these findings is limited by the specific dataset used—movie posters—and may not be generalized to other contexts without further investigation. While the study provides a foundational understanding of activation functions in multilabel CNNs, it also calls for further research to address unresolved questions and explore additional factors influencing model performance. Future work should focus on detailed experimental designs and hypotheses to advance the field and refine activation function usage in diverse scenarios.

REFERENCES

[1] K. van Es, "Netflix & Big Data: The Strategic Ambivalence of an Entertainment Company," *Television and New Media*, vol. 24, no. 6, pp. 656–672, Sep. 2023, https://doi.org/10.1177/15274764221125745.
[2] A. Nilla and E. B. Setiawan, "Film Recommendation System Using Content-Based Filtering and the Convolutional Neural Network (CNN) Classification Methods," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 10, no. 1, p. 17, Feb. 2024, https://doi.org/10.26555/jiteki.v9i4.28113.
[3] N. Pajkovic, "Algorithms and taste-making: Exposing the Netflix Recommender System's operational logics," *Convergence*, vol. 28, no. 1, pp. 214–235, Feb. 2022, https://doi.org/10.1177/13548565211014464.
[4] K. Kundalia, Y. Patel, and M. Shah, "Multi-label Movie Genre Detection from a Movie Poster Using Knowledge Transfer Learning," *Augmented Human Research*, vol. 5, no. 1, Dec. 2020, https://doi.org/10.1007/s41133-019-0029-y.
[5] N. K. Rajput and B. A. Grover, "A multi-label movie genre classification scheme based on the movie's subtitles," *Multimed Tools Appl*, vol. 81, no. 22, pp. 32469–32490, Sep. 2022, https://doi.org/10.1007/s11042-022-12961-6.

[6] J. Kim and J. Lee, "Between Familiarity and Unfamiliarity: Users' Perception and Intention of Watching Netflix Artwork," *Archives of Design Research*, vol. 34, no. 4, pp. 23–37, 2021, https://doi.org/10.15187/adr.2021.11.34.4.23.
[7] F. Z. Unal, M. S. Guzel, E. Bostanci, K. Acici, and T. Asuroglu, "Multilabel Genre Prediction Using Deep-Learning Frameworks," *Applied Sciences (Switzerland)*, vol. 13, no. 15, Aug. 2023, https://doi.org/10.3390/app13158665.
[8] S. Periaiya and A. T. Nandukrishna, "What Drives User Stickiness and Satisfaction in OTT Video Streaming Platforms? A Mixed-Method Exploration," *Int J Hum Comput Interact*, vol. 40, no. 9, pp. 2326–2342, 2024, https://doi.org/10.1080/10447318.2022.2160224.
[9] W. T. Chu and H. J. Guo, "Movie genre classification based on poster images with deep neural networks," in *MUSA2 2017 - Proceedings of the Workshop on Multimodal Understanding of Social, Affective and Subjective Attributes, co-located with MM 2017*, pp. 39–45, 2017, https://doi.org/10.1145/3132515.3132516.
[10] J. Chai, H. Zeng, A. Li, and E. W. T. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios," *Machine Learning with Applications*, vol. 6, p. 100134, 2021, https://doi.org/10.24433/CO.0411648.v1.
[11] T. Dobbs, A. A. R. Nayeem, I. Cho, and Z. Ras, "Contemporary Art Authentication with Large-Scale Classification," *Big Data and Cognitive Computing*, vol. 7, no. 4, Dec. 2023, https://doi.org/10.3390/bdcc7040162.
[12] N. G. Cholli, "Early Identification of Alzheimer's Disease Using Medical Imaging: A Review From a Machine Learning Approach Perspective," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 9, no. 3, pp. 708–719, 2023, https://doi.org/10.26555/jiteki.v9i3.25148.
[13] B. D. Satoto, R. T. Wahyuningrum, and B. K. Khotimah, "Classification of Corn Seed Quality Using Convolutional Neural Network with Region Proposal and Data Augmentation," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 9, no. 2, pp. 348–362, 2023, https://doi.org/10.26555/jiteki.v9i2.26222.
[14] I. V. Pustokhina *et al.*, "Automatic Vehicle License Plate Recognition Using Optimal K-Means with Convolutional Neural Network for Intelligent Transportation Systems," *IEEE Access*, vol. 8, pp. 92907–92917, 2020, https://doi.org/10.1109/ACCESS.2020.2993008.
[15] H. S. Munawar, F. Ullah, D. Shahzad, A. Heravi, S. Qayyum, and J. Akram, "Civil Infrastructure Damage and Corrosion Detection: An Application of Machine Learning," *Buildings*, vol. 12, no. 2, Feb. 2022, https://doi.org/10.3390/buildings12020156.
[16] Y. Bazi, L. Bashmal, M. M. Al Rahhal, R. Al Dayil, and N. Al Ajlan, "Vision transformers for remote sensing image classification," *Remote Sens (Basel)*, vol. 13, no. 3, pp. 1–20, Feb. 2021, https://doi.org/10.3390/rs13030516.
[17] T. Chai, J. Li, S. Prasad, Q. Lu, and Z. Zhang, "Shape-driven lightweight CNN for finger-vein biometrics," *Journal of Information Security and Applications*, vol. 67, Jun. 2022, https://doi.org/10.1016/j.jisa.2022.103211.
[18] H. Yu, L. T. Yang, Q. Zhang, D. Armstrong, and M. J. Deen, "Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives," *Neurocomputing*, vol. 444, pp. 92–110, Jul. 2021, https://doi.org/10.1016/j.neucom.2020.04.157.
[19] T. Georgiou, Y. Liu, W. Chen, and M. Lew, "A survey of traditional and deep learning-based feature descriptors for high dimensional data in computer vision," *Int J Multimed Inf Retr*, vol. 9, no. 3, pp. 135–170, Sep. 2020, https://doi.org/10.1007/s13735-019-00183-w.
[20] J. Wu *et al.*, "Multi-Label Active Learning Algorithms for Image Classification: Overview and Future Promise," *Overview and future promise. ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1-35, 2020, https://doi.org/10.1145/3379504.
[21] T. Chen, M. Xu, X. Hui, H. Wu, and L. Lin, "Learning Semantic-Specific Graph Representation for Multi-Label Image Recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 522–533, 2019, [Online]. Available: https://github.com/HCPLab-SYSU/SSGRL.
[22] R. Wu *et al.*, "An Efficient Multi-Label Classification-Based Municipal Waste Image Identification," *Processes*, vol. 12, no. 6, p. 1075, May 2024, https://doi.org/10.3390/pr12061075.
[23] J. Lanchantin, T. Wang, V. Ordonez, and Y. Qi, "General Multi-label Image Classification with Transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16478–16488, 2021, https://openaccess.thecvf.com/content/CVPR2021/html/Lanchantin_General_Multi-Label_Image_Classification_With_Transformers_CVPR_2021_paper.html.
[24] A. N. Tarekegn, M. Giacobini, and K. Michalak, "A review of methods for imbalanced multi-label classification," *Pattern Recognit*, vol. 118, Oct. 2021, https://doi.org/10.1016/j.patcog.2021.107965.
[25] Y. Wei *et al.*, "HCP: A flexible CNN framework for multi-label image classification," *IEEE Trans Pattern Anal Mach Intell*, vol. 38, no. 9, pp. 1901–1907, Sep. 2016, https://doi.org/10.1109/TPAMI.2015.2491929.
[26] L. M. Zhang, "Multi-function Convolutional Neural Networks for Improving Image Classification Performance," *arXiv preprint arXiv:1805.11788.*, May 2018, [Online]. Available: http://arxiv.org/abs/1805.11788.
[27] D. Liu, G. Yang, J. Wu, J. Zhao, and F. Lv, "Robust binary loss for multi-category classification with label noise," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 1700–1704, 2021, https://doi.org/10.1109/ICASSP39728.2021.9414493.
[28] L. Song *et al.*, "A Deep Multi-Modal CNN for Multi-Instance Multi-Label Image Classification," *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 6025–6038, Dec. 2018, https://doi.org/10.1109/TIP.2018.2864920.
[29] S. Haidar and J. Oramas, "Training Methods of Multi-Label Prediction Classifiers for Hyperspectral Remote Sensing Images," *Remote Sens (Basel)*, vol. 15, no. 24, Dec. 2023, https://doi.org/10.3390/rs15245656.
[30] X. Wang, H. Ren, and A. Wang, "Smish: A Novel Activation Function for Deep Learning Methods," *Electronics (Switzerland)*, vol. 11, no. 4, Feb. 2022, https://doi.org/10.3390/electronics11040540.

[31] M. Zhu, W. Min, Q. Wang, S. Zou, and X. Chen, "PFLU and FPFLU: Two novel non-monotonic activation functions in convolutional neural networks," *Neurocomputing*, vol. 429, pp. 110–117, Mar. 2021, https://doi.org/10.1016/j.neucom.2020.11.068.
[32] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition," *Applied Sciences (Switzerland)*, vol. 10, no. 5, Mar. 2020, https://doi.org/10.3390/app10051897.
[33] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete, "A survey on modern trainable activation functions," *Neural Networks*, vol. 138, pp. 14–32, Jun. 2021, https://doi.org/10.1016/j.neunet.2021.01.026.
[34] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Trans Neural Netw Learn Syst*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, https://doi.org/10.1109/TNNLS.2021.3084827.
[35] A. M. Alhassan and W. M. N. W. Zainon, "Brain tumor classification in magnetic resonance image using hard swish-based RELU activation function-convolutional neural network," *Neural Comput Appl*, vol. 33, no. 15, pp. 9075–9087, Aug. 2021, https://doi.org/10.1007/s00521-020-05671-3.
[36] S. Verma, A. Chug, and A. P. Singh, "Revisiting activation functions: empirical evaluation for image understanding and classification," *Multimed Tools Appl*, vol. 83, no. 6, pp. 18497–18536, Feb. 2024, https://doi.org/10.1007/s11042-023-16159-2.
[37] M. Awal Kassim, H. Viktor, and W. Michalowski, "Multi-Label Lifelong Machine Learning: A Scoping Review of Algorithms, Techniques, and Applications," *IEEE Access*, vol. 12, pp. 74539–74557, 2024, https://doi.org/10.1109/ACCESS.2024.3403569.
[38] M. M. Taye, "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions," *Computation*, vol. 11, no. 3, 52, 2023, https://doi.org/10.3390/computation11030052.
[39] S. P. Singh, L. Wang, S. Gupta, B. Gulyas, and P. Padmanabhan, "Shallow 3D CNN for Detecting Acute Brain Hemorrhage from Medical Imaging Sensors," *IEEE Sens J*, vol. 21, no. 13, pp. 14290–14299, Jul. 2021, https://doi.org/10.1109/JSEN.2020.3023471.
[40] M. Umer, S. Sadiq, M. Ahmad, S. Ullah, G. S. Choi, and A. Mehmood, "A novel stacked CNN for malarial parasite detection in thin blood smear images," *IEEE Access*, vol. 8, pp. 93782–93792, 2020, https://doi.org/10.1109/ACCESS.2020.2994810.
[41] D. Bhatt *et al.*, "CNN variants for computer vision: History, architecture, application, challenges and future scope," *Electronics*, vol. 10, no. 20, pp. 2470, 2021, https://doi.org/10.3390/electronics10202470.
[42] P. Singh, P. Raj, and V. P. Namboodiri, "EDS pooling layer," *Image Vis Comput*, vol. 98, Jun. 2020, https://doi.org/10.1016/j.imavis.2020.103923.
[43] I. Salehin and D. K. Kang, "A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain," *Electronics,* vol. 12, no. 14, p. 3106, 2023, https://doi.org/10.3390/electronics12143106.
[44] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization: an empirical study of their impact to deep learning," *Multimed Tools Appl*, vol. 79, no. 19–20, pp. 12777–12815, May 2020, https://doi.org/10.1007/s11042-019-08453-9.
[45] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, vol. 503, Elsevier B.V., pp. 92–108, Sep. 07, 2022, https://doi.org/10.1016/j.neucom.2022.06.111.
[46] M. Lee, "Mathematical Analysis and Performance Evaluation of the GELU Activation Function in Deep Learning," *Journal of Mathematics*, p. 4229924, https://doi.org/10.1155/2023/4229924.
[47] A. Bhusal, A. Alsadoon, P. W. C. Prasad, N. Alsalami, and T. A. Rashid, "Deep learning for sleep stages classification: modified rectified linear unit activation function and modified orthogonal weight initialisation," *Multimed Tools Appl*, vol. 81, no. 7, pp. 9855–9874, Mar. 2022, https://doi.org/10.1007/s11042-022-12372-7.
[48] H. Abdel-Nabi, G. Al-Naymat, M. Z. Ali, and A. Awajan, "HcLSH: A Novel Non-Linear Monotonic Activation Function for Deep Learning Methods," *IEEE Access*, vol. 11, pp. 47794–47815, 2023, https://doi.org/10.1109/ACCESS.2023.3276298.
[49] A. A. Alkhouly, A. Mohammed, and H. A. Hefny, "Improving the Performance of Deep Neural Networks Using Two Proposed Activation Functions," *IEEE Access*, vol. 9, pp. 82249–82271, 2021, https://doi.org/10.1109/ACCESS.2021.3085855.
[50] D. Kim, J. Kim, and J. Kim, "Elastic exponential linear units for convolutional neural networks," *Neurocomputing*, vol. 406, pp. 253–266, Sep. 2020, https://doi.org/10.1016/j.neucom.2020.03.051.
[51] A. Ciuparu, A. Nagy-Dăbâcan, and R. C. Mureşan, "Soft++, a multi-parametric non-saturating non-linearity that improves convergence in deep neural architectures," *Neurocomputing*, vol. 384, pp. 376–388, Apr. 2020, https://doi.org/10.1016/j.neucom.2019.12.014.
[52] E. C. Seyrek and M. Uysal, "A comparative analysis of various activation functions and optimizers in a convolutional neural network for hyperspectral image classification," *Multimed Tools Appl*, vol. 83, no. 18, pp. 53785–53816, May 2024, https://doi.org/10.1007/s11042-023-17546-5.
[53] S. Bera and V. K. Shrivastava, "Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification," *Int J Remote Sens*, vol. 41, no. 7, pp. 2664–2683, Apr. 2020, https://doi.org/10.1080/01431161.2019.1694725.

[54] C. Guo, X. Chen, Y. Chen, and C. Yu, "Multi-Stage Attentive Network for Motion Deblurring via Binary Cross-Entropy Loss," *Entropy*, vol. 24, no. 10, Oct. 2022, https://doi.org/10.3390/e24101414.
[55] S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, and D. Traore, "Deep Convolution Neural Network sharing for the multi-label images classification," *Machine Learning with Applications*, vol. 10, p. 100422, Dec. 2022, https://doi.org/10.1016/j.mlwa.2022.100422.
[56] M. Yaqub *et al.*, "State-of-the-art CNN optimizer for brain tumor segmentation in magnetic resonance images," *Brain Sci*, vol. 10, no. 7, pp. 1–19, Jul. 2020, https://doi.org/10.3390/brainsci10070427.
[57] M. Abadi *et al.*, "TensorFlow: A System for Large-Scale Machine Learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, pp. 256–283, 2016, https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.
[58] T. Carneiro, R. V. M. Da Nobrega, T. Nepomuceno, G. Bin Bian, V. H. C. De Albuquerque, and P. P. R. Filho, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications," *IEEE Access*, vol. 6, pp. 61677–61685, 2018, https://doi.org/10.1109/ACCESS.2018.2874767
[59] R. Gu *et al.*, "Liquid: Intelligent Resource Estimation and Network-Efficient Scheduling for Deep Learning Jobs on Distributed GPU Clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2808–2820, Nov. 2022, https://doi.org/10.1109/TPDS.2021.3138825.
[60] S. Velu, S. S. Gill, S. S. Murugesan, H. Wu, and X. Li, "CloudAIBus: a testbed for AI based cloud computing environments," *Cluster Comput*, pp. 1-29, 2024, https://doi.org/10.1007/s10586-024-04562-9.
[61] V. M. Vargas, P. A. Gutiérrez, J. Barbero-Gómez, and C. Hervás-Martínez, "Activation Functions for Convolutional Neural Networks: Proposals and Experimental Study," *IEEE Trans Neural Netw Learn Syst*, vol. 34, no. 3, pp. 1478–1488, Mar. 2023, https://doi.org/10.1109/TNNLS.2021.3105444.
[62] T. Szandała, "Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks," in *Studies in Computational Intelligence*, vol. 903, 2021, pp. 203–224. https://doi.org/10.1007/978-981-15-5495-7_11.
[63] S.-H. Wang and E. Sakk, "The effect of activation function choice on the performance of convolutional neural networks," *J Emerg Investig*, vol. 6, no. 8, 2023, https://doi.org/10.59720/23-055.
[64] Y. Jiang, J. Xie, and D. Zhang, "An Adaptive Offset Activation Function for CNN Image Classification Tasks," *Electronics (Switzerland)*, vol. 11, no. 22, Nov. 2022, https://doi.org/10.3390/electronics11223799.
[65] G. Maguolo, L. Nanni, and S. Ghidoni, "Ensemble of convolutional neural networks trained with different activation functions," *Expert Syst Appl*, vol. 166, Mar. 2021, https://doi.org/10.1016/j.eswa.2020.114048.
[66] T. Hoefler, D. Alistarh, N. Dryden, and A. Peste, "Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks," *Journal of Machine Learning Research*, vol. 22, no. 241, pp. 1–124, 2021, [Online]. Available: http://jmlr.org/papers/v23/21-0366.html.
[67] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, pp. 312–315, Dec. 2020, https://doi.org/10.1016/j.icte.2020.04.010.
[68] C. Yu, X. Qi, H. Ma, X. He, C. Wang, and Y. Zhao, "LLR: Learning learning rates by LSTM for training neural networks," *Neurocomputing*, vol. 394, pp. 41–50, Jun. 2020, https://doi.org/10.1016/j.neucom.2020.01.106.
[69] Z. Ma, Y. Xu, H. Xu, Z. Meng, L. Huang, and Y. Xue, "Adaptive Batch Size for Federated Learning in Resource-Constrained Edge Computing," *IEEE Trans Mob Comput*, vol. 22, no. 1, pp. 37–53, Jan. 2023, https://doi.org/10.1109/TMC.2021.3075291.
[70] P. Oza, P. Sharma, S. Patel, F. Adedoyin, and A. Bruno, "Image Augmentation Techniques for Mammogram Analysis," *J Imaging*, vol. 8, no. 5, May 2022, https://doi.org/10.3390/jimaging8050141.
[71] M. Nagaraju, P. Chawla, S. Upadhyay, and R. Tiwari, "Convolution network model based leaf disease detection using augmentation techniques," *Expert Syst*, vol. 39, no. 4, May 2022, https://doi.org/10.1111/exsy.12885.
[72] O. N. Oyelade and A. E. Ezugwu, "A deep learning model using data augmentation for detection of architectural distortion in whole and patches of images," *Biomed Signal Process Control*, vol. 65, Mar. 2021, https://doi.org/10.1016/j.bspc.2020.102366.
[73] R. Bravin, L. Nanni, A. Loreggia, S. Brahnam, and M. Paci, "Varied Image Data Augmentation Methods for Building Ensemble," *IEEE Access*, vol. 11, pp. 8810–8823, 2023, https://doi.org/10.1109/ACCESS.2023.3239816.
[74] R. Poojary, R. Raina, and A. K. Mondal, "Effect of data-augmentation on fine-tuned cnn model performance," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 84–92, Mar. 2021, https://doi.org/10.11591/ijai.v10.i1.pp84-92.
[75] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, "A review of medical image data augmentation techniques for deep learning applications," *J Med Imaging Radiat Oncol*, vol. 65, no. 5, pp. 545–563, Aug. 2021, https://doi.org/10.1111/1754-9485.13261.
[76] J. Fonseca and F. Bacao, "Improving Active Learning Performance through the Use of Data Augmentation," *International Journal of Intelligent Systems*, vol. 1, 2023, https://doi.org/10.1155/2023/7941878.

BIOGRAPHY OF AUTHORS

Ahmad Zein Al Wafi, Undergraduate student at Electrical Engineering Department Universitas Negeri Semarang focusing research on applied machine learning and delivering solutions, particularly in digital technology architecture. Email: ahmadzeinalwafi@outlook.com. ORCID: 0009-0002-2738-9661.

Anan Nugroho, Doctor of Electrical & Information Engineering as a lecturer at electrical engineering department Universitas Negeri Semarang with research area IVI (Intelligent Vision & Imaging). IVI covers the development and application of algorithms, models, and systems that enable machines to interpret and understand visual information from environmental objects by adopting human vision capabilities and overcoming their limitations. Email: anannugroho@mail.unnes.ac.id. ORCID: 0000-0002-3844-1405.