Sinopfbd, 2025; 10(1): 289-318 https://doi.org/10.33484/sinopfbd.1618268

Sinop Üniversitesi Fen Bilimleri Dergisi
Sinop Uni J Nat Sci
E-ISSN: 2564-7873

https://dergipark.org.tr/tr/pub/sinopfbd

A Novel Convolutional Neural Network Architecture for the Classification of Binary Images

Yasin ÖZKAN

How to cite: Özkan, Y. (2025). A novel convolutional neural network architecture for the classification of binary images. *Sinop Üniversitesi Fen Bilimleri Dergisi, 10*(1), 289-318. https://doi.org/10.33484/sinopfbd.1618268

Research Article

Corresponding Author
Yasin ÖZKAN
yasin.ozkan@beun.edu.tr

ORCID of Author
Y.Ö: 0000-0002-2029-0856

Received: 12.01.2025
Accepted: 26.06.2025

Abstract

While numerous studies in the literature focus on the classification of color images using deep learning algorithms, there is a notable gap in research dedicated to the classification of binary images. Although Convolutional Neural Networks designed for binary images tend to exhibit lower performance compared to those for color images, their processing speed is significantly faster, as the input data for binary images is reduced by a factor of 24 compared with the 8-bit color images. This study aims to develop network architectures that operate with high efficiency in applications requiring only binary images, such as signature recognition, barcode reading, QR code scanning, and handwriting analysis. For this purpose, a new Bi-CNN(Binary image-CNN) network architecture was designed using existing layers. Then, a special loss function was used to improve the performance of this architecture. By integrating the classification layer called Si-CL(Signature-Classification) into Bi-CNN, a new architecture called Bi-CL-CNN emerged. Both Bi-CNN and Bi-CL-CNN were trained on two datasets. The first dataset, Shape-DU, was specifically created for testing these networks. The second dataset, MPEG-7, serves as a benchmark dataset. The performance of the trained networks is compared with three previously trained networks, namely GoogleNet, ResNet50 and DenseNet201. The empirical evaluation demonstrated that the Bi-CL-CNN network significantly outperformed the other models in both accuracy and computational speed. These findings underscore the robustness and efficiency of the proposed models in handling binary image datasets.

Keywords: Convolutional neural network, deep neural network, image classification, binary image, loss function

İkili Görüntülerin Sınıflandırılması için Yeni Bir Evrişimsel Sinir Ağı Mimarisi

Zonguldak Bülent Ecevit University, Department of Computer Technologies, Zonguldak, Türkiye

Öz

Literatürdeki pek çok çalışma, derin öğrenme algoritmalarını kullanarak renkli görüntülerin sınıflandırılmasına odaklanırken, ikili görüntülerin sınıflandırılmasına yönelik araştırmaların sınırlı olduğu gözlemlenmektedir. İkili görüntüler için tasarlanan Evrişimli Sinir Ağ mimarileri, renkli görüntülere kıyasla genellikle daha düşük performans sergilemekle birlikte, ikili görüntülerdeki giriş verilerinin 8 bitlik renkli görüntülere göre 24 kat daha az bilgi içeriyor olması, işlem hızlarının önemli ölçüde artmasına neden olmaktadır. Bu çalışmanın amacı, yalnızca ikili görüntüler gerektiren uygulamalar —örneğin, imza tanıma, barkod okuma, QR kod tarama ve el yazısı analizi— için yüksek verimlilikle çalışan ağ mimarileri geliştirmektir. Bu hedef doğrultusunda, mevcut katmanlardan yararlanarak yeni bir Bi-CNN ağ mimarisi tasarlanmıştır.

Ardından, bu mimarinin performansını artırmaya yönelik özel bir kayıp fonksiyonu geliştirilmiş ve Si-CL(İmza-Sınıflandırma) adı verilen sınıflandırma katmanı, Bi-CNN(İkili CNN)'e entegre edilerek Bi-CL-CNN olarak adlandırılan yeni bir mimari ortaya çıkmıştır. Hem Bi-CNN hem de Bi-CL-CNN, iki farklı veri kümesi üzerinde eğitilmiştir. İlk veri kümesi olan Shape-DU, bu ağları test etmek amacıyla özel olarak oluşturulmuştur. İkinci veri kümesi olan MPEG-7 ise kıyaslama amacıyla kullanılmıştır. Eğitilen ağların performansı, GoogleNet, ResNet50 ve DenseNet201 gibi daha önce eğitilmiş üç ağ ile karşılaştırılmıştır. Deneysel sonuçlar, Bi-CL-CNN ağının, doğruluk ve hesaplama hızı açısından diğer modellerden anlamlı derecede daha iyi performans gösterdiğini ortaya koymuştur. Bu bulgular, ikili görüntü veri kümelerinin işlenmesinde önerilen modellerin sağlamlık ve verimliliğini vurgulamaktadır.

Anahtar Kelimeler: Evrişimli sinir ağı, derin sinir ağı, görüntü sınıflandırma, ikili görüntü, kayıp fonksiyonu

Introduction

Binary images are images that carry only two color values (black and white) and are ideal for fast processing due to their simple, low-dimensional structure, which makes them widely used in applications as diverse as optical character recognition (OCR), handwriting analysis, document scanning, and object recognition. However, this simple structure makes it possible for deep learning-based models such as convolutional neural networks (CNNs) to more effectively learn complex patterns and features in binary images, while traditional image processing methods are limited. CNNs have demonstrated exceptional performance across a variety of computer vision tasks, including image classification [1], image splicing [2], image segmentation [3], and object recognition [4]. Deep learning, which involves mining high-dimensional data for potential features through model training [5], has proven particularly effective in these applications. Several CNN architectures, such as LeNet [6], Xception [7], and VGGNet [8], have gained popularity due to their ability to address specific problems through various design choices and extensive parameterization. Consequently, the performance of a CNN model is influenced by both its architectural design and the specific problem it is intended to solve. The architecture and configuration of CNN models, including layer types and hyperparameter settings, have a direct impact on their efficacy [9]. Building an effective CNN model for a given problem often involves a trial-and-error process, which can be computationally intensive and time-consuming. Therefore, designing a suitable CNN model for tasks such as binary image classification (e.g., medical imaging, QR codes, and document analysis) is particularly challenging due to the complexities involved in hyperparameter tuning and architectural design.

Image classification involves processing an image to extract probabilities for the classes that best describe it. Computers interpret images as arrays of pixel values, typically ranging from 0 to 255, depending on the image's resolution and size. These pixel values serve as inputs for classification algorithms, which then output the probability of the image belonging to a particular class. CNNs achieve classification by isolating low-level features, such as edges and curves, through a series of convolutional

layers. This process allows CNNs to identify and leverage both low-level features (e.g., lines, colors, dots) and high-level features, which are used to characterize more complex patterns and objects within an image [10]. CNNs are trained on large datasets to effectively recognize and extract these high-level features [11].

When dealing with color images, which consist of three RGB channels, each pixel is represented by a vector of red, green, and blue values [12]. This increases the data dimensionality that CNNs must process, resulting in more parameters and longer training times. The increased complexity of color images can make it challenging for CNNs to identify key features and patterns. For example, distinguishing subtle details in images such as flowers can be difficult. Nevertheless, CNNs have been successfully applied to various tasks involving color images, including object detection, image segmentation, and classification. Techniques such as image normalization, data augmentation, and transfer learning have been developed to enhance CNN performance on color images. These methods help CNNs handle the complexities associated with color data, improving their effectiveness and efficiency.

Binary images, while useful for segmentation algorithms, contain fewer pixel-related features compared to color images [13]. The simplicity of binary images—storing only binary values—makes them computationally efficient but often results in lower CNN performance compared to color images due to the absence of color information. Despite these challenges, binary images are widely used in various applications, including optical character recognition (OCR) systems for handwriting, license plate recognition, and bank check processing [14]. Finding and creating datasets of binary images is challenging, and CNN performance on such images generally falls short compared to color images. To address these issues, have developed the "Shape-DU" binary image dataset specifically for this study and proposed two CNN models, Bi-CNN and Bi-CL-CNN, to enhance binary image classification performance. The Bi-CNN model, developed as a standard CNN model, was evaluated using cross-entropy loss. For the Bi-CL-CNN model, which employs a specialized Si-CL loss function designed for classifying handwritten signatures, we compared the performance of this classification layer against Bi-CNN and pre-trained models. The proposed models are intentionally designed to be lightweight to ensure fast classification even on devices with limited processing power, such as portable computers. Compared the classification performance of these new models against a range of state-of-the-art classifiers, including deep neural networks.

Although extensive research has been conducted on image classification, a notable gap exists in studies focused specifically on binary image classification. This work specifically addresses the challenge of binary image classification, a critical task with important practical applications in fields such as medical imaging, document analysis, and automatic recognition systems. Despite the simplicity of binary images, their effective classification presents unique challenges, primarily due to the lack of rich pixel-level information compared to color images. The central objective of this study is to enhance the efficacy

of CNNs in binary image classification tasks. This endeavor involves the proposal of specialized models that can effectively address the inherent challenges posed by binary data. This work aims to contribute to the field in several key areas:

- Proposal of Two CNN Models: Propose two novel CNN models for binary image classification. Unlike existing methods, these models have been developed with special configurations that can work more efficiently with binary data.
- End-to-End Solution: By utilizing deep neural networks, our models eliminate the need for manual feature selection and extraction, offering a comprehensive end-to-end solution. This aims to solve some previously unsolved challenges, especially considering the simple structure of binary images.
- Introduction of the Shape-DU Dataset: In order to contribute to the research community working in this area, we introduce the Shape-DU dataset, which consists of 20 classes and is widely available. This dataset will be an important resource for researchers in the field of binary image classification.
- Hyperparameter Optimization: Performed extensive hyperparameter optimization to identify the optimal settings for models, resulting in the selection of the model with the highest classification accuracy. This optimization process enables to make specific adjustments for better performance on binary images.
- Specialized Loss Function: The Bi-CL-CNN model uses a loss function specifically developed for binary image classification. This loss function is designed to improve the classification accuracy of the models and offers a significant innovation compared to other works in the field.

The remainder of this paper is organized as follows: Section 2 provides a brief overview of CNNs and relevant prior work. Section 3 describes the methodology and datasets used. Section 4 presents the experimental results and discussion. Finally, Section 5 offers conclusions and suggestions for future research based on the findings of this study.

Related Studies

Classification involves dividing a dataset into distinct classes based on shared characteristics. Classification algorithms initially learn this distribution from a given training set and subsequently classify test data with unknown class labels. Image classification models, in particular, take an image as input and predict its class membership. With the substantial advancements in deep learning, researchers have significantly contributed to the field of image classification. In the early stages, traditional machine learning algorithms were utilized for image classification. However, with the remarkable progress in deep learning networks, significant advancements have been achieved using CNNs. The primary advantage of CNN architectures over traditional machine learning algorithms is their ability to extract features directly from raw data [15].

Gao et al. proposed a CNN-based algorithm for rotating pixels in binary images and compared its performance with Matlab's rotation algorithm. The results indicated that Matlab's algorithm performed

more effectively on binary images [16]. Mujawar et al. introduced a novel CNN architecture using the MNIST dataset, which comprises handwritten binary images. Their CNN architecture, implemented in Matlab, was compared against Stacked Autoencoder (SAE), Neural Network (NN), and Deep Belief Network (DBN) algorithms. The CNN architecture achieved a commendable performance with a 96% accuracy rate, outperforming the other algorithms [17]. Muscle injuries and fractures are prevalent and can often be overlooked by medical professionals. To address this issue, Chittajallu et al. developed a three-layer CNN architecture designed to identify fractures in X-ray (grayscale) images. Their experimental results demonstrated that the architecture provided accurate predictions [18].

Scanlan evaluated the performance of a supervised binary image classification model on the Kaggle Dogs and Cats dataset. Despite the tendency of CNNs to overfit, the use of dropout layers mitigated this effect, resulting in a training accuracy of 96% [19]. Yang et al. conducted shape classification experiments using the Animal, MPEG-7, Swedish Leaf, and ETH-80 datasets. They achieved accuracies of 88.85% for the Animal dataset, 96.27% for the MPEG-7 dataset, 98.31% for the Swedish Leaf dataset, and 97.93% for the ETH-80 dataset, surpassing the results of previous methods [20].

Patel et al. introduced a technique to analyze shape components and enhance shape-specific data. They utilized two different binary image datasets, Animal and MPEG-7, and compared classification results using two pre-trained CNN models (VGGNet and GoogLeNet) [21].

Early detection of pneumonia, caused by a common virus, is crucial for controlling the disease's progression. Ramya et al. employed a Hybrid Neural Network (HNN) approach to diagnose pneumonia with high accuracy using Chest X-ray (CXR) images [22]. Dermoscopic images provide detailed information for analyzing skin lesions. Jayalakshmi et al. explored the classification of dermoscopic images to determine whether skin lesions are benign or malignant. They proposed a Batch Normalization CNN architecture for lesion identification and image classification, achieving an accuracy improvement to 89.30% [23].

Feature extraction from images can be challenging. Ramanjaneyulu et al. utilized the VGG-16 model in a deep learning framework to extract features from image databases, yielding successful results [24]. Wang et al. developed a system to detect and mitigate traffic congestion. Initially, they tested a traditional feature extraction method on a database with over 30,000 images. Subsequently, they created and tested a CNN-based architecture called TrafficNet, derived from AlexNet and VGGNet, using the same image dataset. The results demonstrated that TrafficNet significantly outperformed the traditional method, achieving 90% accuracy [25].

Hafeez et al. propose a hybrid quantum neural network (H-QNN) model designed for binary image classification. By combining the strengths of quantum computers and classical neural networks, this model is able to compute with high efficiency on noisy intermediate scale quantum (NISQ) devices. The model achieves 90.1% accuracy using a compact quantum circuit of two quantum bits combined with a

classical convolutional architecture. Moreover, the H-QNN model also performs well in image retrieval tasks, solving the problem of overfitting in small data sets [26].

Deepak et al. proposed a novel Hybrid Quantum-Qlassical Neural Network (H-QNN) for binary image classification using MNIST dataset. The integration of quantum technology with traditional neural network structures has provided significant opportunities to improve classification accuracy and efficiency. H-QNN outperformed traditional CNN and QCNN models by achieving 99.7% accuracy [27].

Korkut et al. provided a comparative analysis of CNN architectures for binary image classification and examined the advantages of transfer learning. The performance of leading CNN models such as MobileNetV3, VGG19, ResNet50 and EfficientNetB0 were evaluated for skin cancer classification. The impact of transfer learning on these architectures was investigated and their strengths/weaknesses were identified [28].

Hafez et al. proposed a hybrid quantum neural network (H-QNN) model for binary image classification that combined the advantages of quantum and classical neural networks. By integrating a classical convolutional structure with a compact quantum circuit of two qubits, the model worked efficiently on noisy quantum devices. H-QNN improved classification performance with an accuracy of 90.1% and solved the overfitting problem for small data sets [29].

Bai and Hu proposed a novel quantum neural network (QNN) using alternately controlled gates on MNIST dataset for binary image classification. The proposed QNN exponentially reduced the number of qubits required by encoding image data with quantum superposition using quantum probability image coding (QPIE) [30].

Kiger et al. proposed a new method combining binary Markov images and transfer learning for malware detection. The experiments were successful in terms of accuracy and speed, but at the same time, it might require more computational resources compared to traditional ML approaches [31]. A comparative summary of methods used in binary image studies, the accuracy achieved, and the datasets employed is presented in Table 1.

Table 1. *Comparison of studies on binary images*

Related work	Employed method(s)	Dataset	Accuracy(%)
Mujawar et al. [17]	CNN	MNIST	96
Chittajallu et al. [18]	CNN	Their own dataset	80.45
Scanlan [19]	CNN	Kaggle Dogs vs Cats	96
Yang et al. [20]	SVM	Animals	88.85
		MPEG-7	96.27
Patel et al. [21]	SVM	Animals	91.65
		MPEG-7	98.41
Ramya et al. [22]	HNN	CXR images dataset	98

In existing literature, while the success of deep learning methods in color image classification has progressed significantly, work on binary image classification is more limited. Although binary images

are faster to process than color images, deep learning models for these types of images can often underperform. However, these challenges can provide significant advantages in applications where speed and efficiency are critical, such as signature recognition, barcode reading, QR code scanning and handwriting analysis. In existing literature, CNN-based methods proposed for binary images have been applied with limited success. In this context, this study aims to develop highly efficient network architectures that work only with binary images.

Material and Methods

In the subsections that follow in this part, the models created to benchmark the recommended model, the Bi-CNN and Bi-CL-CNN models, and the dataset structure are all fully detailed.

Datasets Construction

Each image in the dataset was preprocessed to enhance data quality, which significantly impacts the performance of the proposed CNN model, as highlighted by this study [32]. The preprocessing phase involved several key steps: initially, each image was resized to 224 by 224 pixels. Subsequently, all images in the dataset were converted to binary format. The preprocessing tasks were fully automated using a custom script, as illustrated in Figure 1.

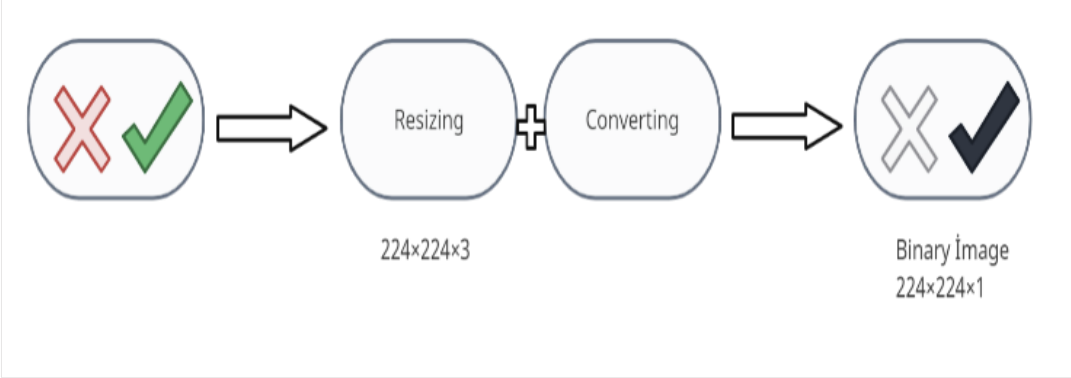

Figure 1. *An example of the pre-processing stage of the suggested model*

In this study, two different datasets were utilized. Specifically, a dataset was tailored for this research, termed the "Shape-DU" dataset, which comprises binary images of 20 distinct shapes, including square, triangle, directional arrow, and others. Each of the 20 classes within the Shape-DU dataset contained 100 binary images. An example of the images representing the classes in the "Shape-DU" dataset is shown in Figure 2.

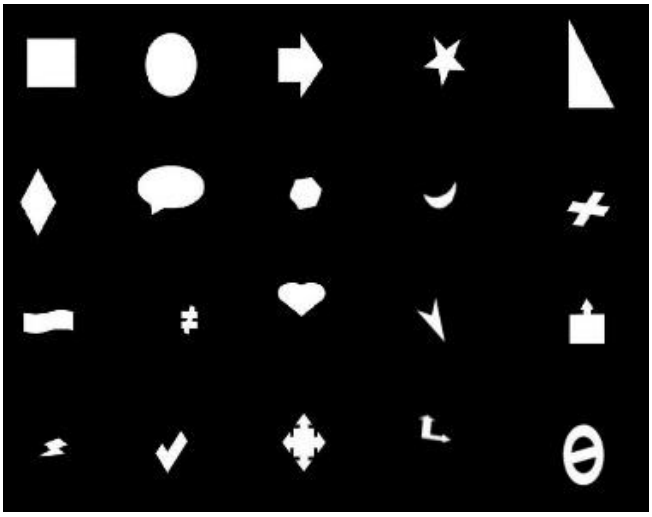

Figure 2. *"Shape-DU" Dataset Overview*

The second dataset utilized in this study was the MPEG-7 dataset [33], which contained binary images representing 70 distinct animal species, including birds, elephants, dogs, cats, and others. Each of the 70 classes within this dataset comprised 20 binary images. An example image illustrating the classes in the MPEG-7 dataset is presented in Figure 3.

Figure 3. *MPEG-7 Dataset Overview*

The performance of the proposed CNN models was influenced by the preprocessing stages applied to the dataset. To enhance the performance of the Bi-CNN model, every binary image in the dataset was preprocessed. Specifically, each image was resized to 224 by 224 pixels.

The proposed models present an effective approach for classifying binary images. An overview of the proposed methodology is illustrated in Figure 4. This approach consists of three key components: preprocessing, feature extraction, and classification. Each component plays a crucial role in the overall process, contributing to the efficacy of the image classification task.

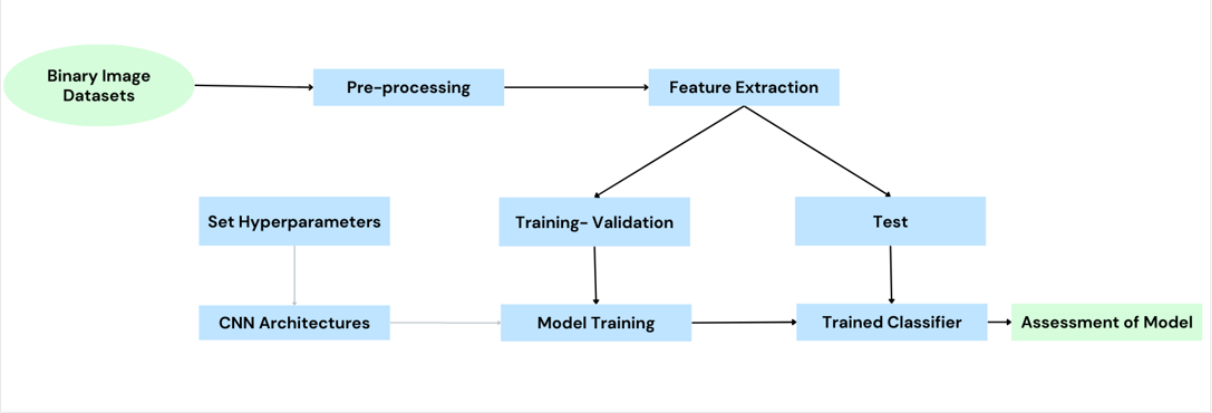

Figure 4. *Block schematic illustrating the suggested approach.*

Background

Classification of binary images has an important place in the field of computer vision (CV), and CNNs play a critical role in this process. The architecture of CNNs consists of a multilayer structure designed to extract features from binary images. These layers make an important contribution to the classification of binary images by providing the ability to identify local features of images and combine them in more complex ways. For example, as shown in Krizhevsky et al. [35], deep network structures provide higher accuracy rates in object recognition and segmentation tasks in binary images compared to traditional machine learning methods.

In binary images, feature extraction is divided into two main categories: low-level and high-level features. Low-level features refer to basic elements such as edges, texture and contrast, while high-level features cover more complex tasks such as object recognition and classification. By processing these two types of features together, CNNs can extract meaningful inferences from binary image data. For example, the first convolution layers are used to recognize edges and basic patterns in binary images, while subsequent layers combine these simple features and move on to more complex object recognition processes. This progressive structure allows CNNs to achieve higher accuracy rates in the classification of binary images [36].

Machine learning involves a set of methods and algorithms that allow the model to learn from the data. CNNs, as a sub-branch of machine learning, stand out with their ability to optimize learning processes in binary images. These models greatly automate the process of learning from binary image data, while reducing the need for hyperparameter tuning and feature engineering. As the depth and complexity of CNNs increase, the overall performance and accuracy of binary images also increases. For example, research by He et al [37] demonstrates the effectiveness of deep networks in binary image classification tasks. In this context, the learning capabilities of CNNs on binary images increase the applicability of innovations in machine learning.

The choice of the loss function in CNNs is crucial for classification tasks. The loss function measures the difference between the model's predictions and the actual labels and plays a fundamental role in

guiding the learning process. The Binary Cross-Entropy loss used in binary classification tasks is a particularly common choice in this field. This loss function helps to minimize the misclassifications of the model and directly affects the performance of the model [38]. An incorrect choice of loss function, on the other hand, can negatively affect the learning process of the model, leading to incorrect predictions and poor performance. Therefore, correctly determining the loss function is critical to improve the overall classification performance of binary images.

The modular design of CNNs allows for the addition of such different layers, providing flexibility in image classification and facilitating the development of application-oriented solutions. Each layer fulfills a specific function, giving it the ability to adapt to different sets of binary images and tasks. This modular structure allows users to experiment with different layers and combinations of structures during the research and development process, helping to achieve better performance and results. In particular, Szegedy et al. in [39] investigated the effects of different layer combinations on performance in binary images. Therefore, choosing CNNs for binary image classification offers a great advantage both theoretically and practically.

Optimization algorithms also play a critical role in the learning process of CNNs. In binary images, algorithms such as Stochastic Gradient Descent (SGD), Adam and RMSprop are used to update the weights and improve the performance of the model. These optimization processes are performed with continuously updated parameters to minimize the loss function of the model. Since the effective use of loss functions directly affects the success of the model, it is one of the critical elements that should be considered together with optimization algorithms [40].

As a result, CNNs have become a preferred method for binary image classification due to their high performance, modular structure and efficient feature extraction processes. The integration of low- and high-level features, loss function and the effective use of optimization algorithms increase the power of these models and have an important place in the field of binary image classification. The success of CNNs in this field is further strengthened by the combination of deep learning and machine learning methods.

In terms of feature extraction techniques, both Bi-CNN and Bi-CL-CNN models employ a set of complex methods designed to extract relevant features from binary images. The convolutional layers in these models are responsible for capturing basic low-level features such as edges, textures and basic patterns, which are essential for the initial stages of image classification. As the network deepens, these basic features are combined to create increasingly complex, high-level features such as object recognition and classification. Furthermore, batch normalization is applied after the convolutional layers to standardize activations, thus improving model convergence and reducing training time. This combination of convolutional layers, ensemble normalization, and pooling operations allows models to effectively extract both low-level and high-level features, which is critical for achieving optimal classification performance in binary image tasks.

Bi-CNN And Bi-Cl-CNN Models Configuration

This paper presents two different CNN models for the classification of binary images: Bi-CNN and Bi-CL-CNN. Both models consist of 44 layers in total, differing only in the classification layers. The remaining 43 layers are the same in both models, including the ranking and hyperparameter settings. The process starts with the input layer, followed by a convolution layer with 8 filters and the application of the ReLU activation function. After the initial convolution and activation stages, an additional convolution layer and activation function are applied to further refine the features. The model is then supplemented with batch normalization, maximum pooling and dropout layers.

The reason for using batch normalization after convolution and the ReLU activation function is that it improves the overall performance of the model and reduces the training time. This layer corrects the distribution of activations by calculating the mean and standard deviation of each mini-batch. Thus, it enables the network to learn faster while reducing the risk of overlearning [41]. Especially for deep networks, batch normalization helps to optimize the model faster by using higher learning rates. This is recognized as an important development in the field of deep learning.

The max pooling layer minimizes information loss by reducing the image size, while at the same time reducing computational costs. By reducing the output of each convolution layer to a certain size, it preserves important features and allows the model to generalize better. Research shows that maximum pooling increases the ability to preserve local variations of features [42]. This is critical in ensuring that salient objects and details are preserved, especially in binary images.

The dropout layer is a technique used to prevent the model from overlearning. It increases the generalization ability of the model by randomly turning off neurons at certain rates during the training process. Dropout has been proven by many studies to improve the accuracy of the model and provide a more robust learning process [43]. Therefore, dropout layers in Bi-CNN and Bi-CL-CNN models contribute to make the model more robust during training.

To extract image features, this study applied four iterations of a sequence of Convolution, ReLU activation function, Convolution, ReLU activation function, Batch Normalization, Maximum Pooling and Dropout layers. During these iterations, the number of convolution filters was systematically increased by a factor of 8, 16, 32 and 64. This structure increases the learning capacity of the model and allows for richer feature extraction. Consistent with the convolution layers, a Fully Connected (FC) layer with 128 units is used by applying the ReLU activation function. After the FC layer, the Batch Normalization and Dropout layers come into play again. The Fully Connected layers, together with the combination of Batch Normalization and Dropout, are repeated with a staggered number of units: 128, 64, 32.

The output layer of the Bi-CNN model classifies the images into 20 different classes using a fully connected layer of 20 units with a softmax activation function. Cross-entropy is used as the loss function

in this model. On the other hand, the Bi-CL-CNN model uses Si-CL [44], a special loss function developed for handwritten signature images. This special loss function was chosen to evaluate the effectiveness of the model on binary images and to investigate its impact on different datasets.

The main difference between the Bi-CNN and Bi-CL-CNN models lies in the change in the classification layers. All other layers and hyperparameter settings are the same between the two models. Therefore, only the classification layers are changed in order to evaluate the impact of the custom loss function on training time and classification accuracy. The only modification made to the Bi-CNN architecture to create the Bi-CL-CNN architecture is the replacement of the classification layer with the Si-CL layer. This change is considered as an important step to improve the classification accuracy of the model. Figure 5 shows the layer layouts of the Bi-CNN architecture.

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K}(T_{ni}\ln(Y_{ni}) + (1-T_{ni})\ln(1-Y_{ni})) - \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{K}(Y_{ni}-T_{ni})^4 \quad (1)$$

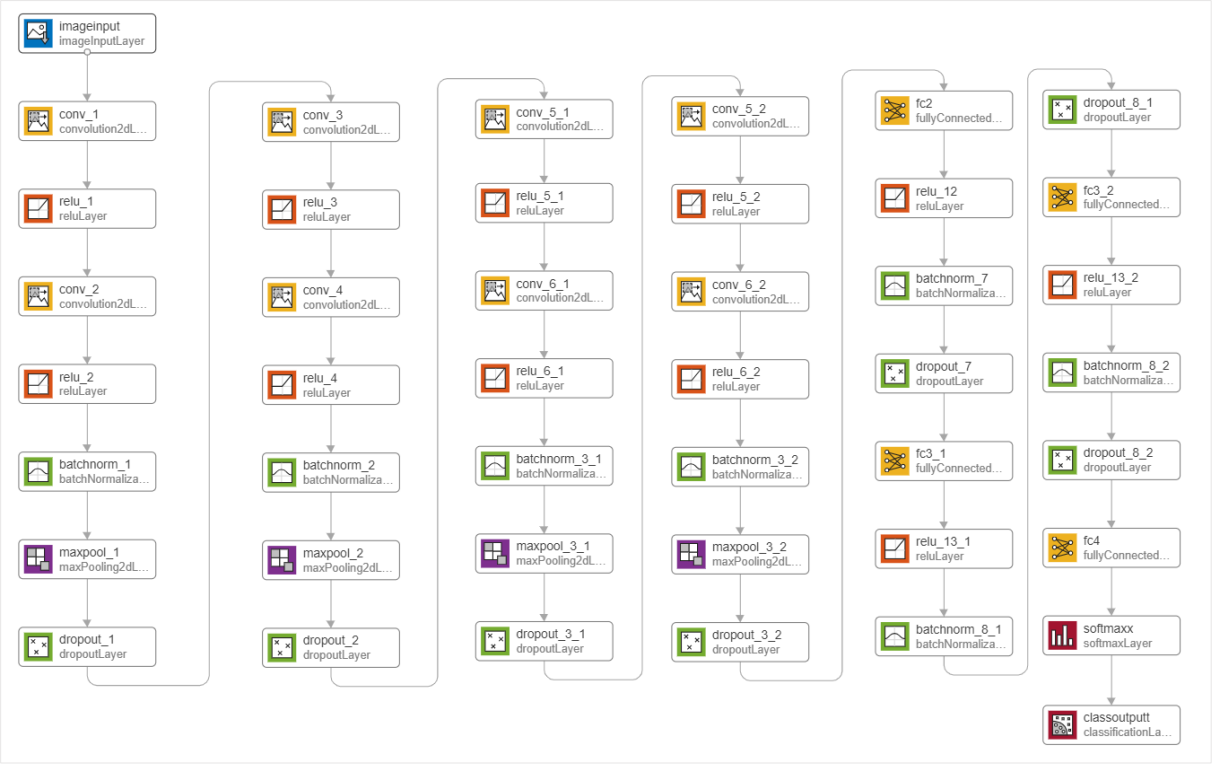

Figure 5. *Graph representing the suggested Bi-CNN model*

Models Training

The training process of the model was carried out with an optimized strategy to improve its performance. In this context, the Adam optimization algorithm was used. Adam is an extension of Stochastic Gradient Descent (SGD), which automatically adjusts the learning rate for each parameter. This allows for a faster and more efficient learning process, especially when working with large datasets and deep networks [45]. The Adam algorithm provides adaptive learning rates for each parameter by combining the first

moment (mean) and second moment (variance) estimates. In this way, the updates of the model become more stable and converge faster [46].

Hyperparameter optimizations also play an important role in model training. Hyperparameters are values that directly affect the training process and results of the model. In this study, various hyperparameter values were tested and the best-performing ones were determined. The hyperparameters include critical parameters such as learning rate, batch size, dropout rate and number of layers.

The learning rate also plays an important role in the hyperparameter settings. The learning rate determines the magnitude of the model's weight updates. A too low learning rate slows down the training process, while too high a learning rate may cause the model's learning process to become unstable [47]. Therefore, different values were tested to determine an optimal learning rate. Batch size refers to the number of samples that the model will use in each update step. Small batch sizes provide more frequent updates, but require more computation time, while large batch sizes provide fewer updates, but use more memory for each update [48]. In this study, different batch sizes were experimented with and the best results were obtained.

Dropout rates are another important step in hyperparameter optimizations. Dropout rate is a technique used to prevent overfitting. This rate determines the probability of randomly turning off neurons during training. Choosing an appropriate dropout rate is critical in increasing the generalization ability of the model. Also, the number of layers is a factor that determines the depth of the model. More layers can increase the capacity of the model to learn more complex relationships; however, excessive deepening can lead to difficult learning [49].

The hyperparameter values expressed in bold in Table 2 produced the best results in the experiments. These values significantly improved the overall performance of the model and optimized the efficiency of the training process. Moreover, this optimization process enabled a better success rate in the classification of binary images.

In conclusion, the Adam optimization algorithm and a systematic hyperparameter optimization process played a critical role in the training phase of the model. The careful selection of the hyperparameters improved the overall performance of the model and provided an effective solution for binary image classification tasks.

Table 2. *Tested values and hyperparameters (the optimal value for every parameter is bolded.)*

Hyper-parameter	**Experimented values**
Dropout rate of Conv layers	0.2, **0.3**, 0.4, 0.5, 0.6, 0.7
Dropout rate of the final fully connected layer	0.2, **0.3**, 0.4, 0.5, 0.6, 0.7
Learning rate	**1xe−3**, $1xe{-}4$, $1xe{-}5$, $1xe{-}6$, $1xe{-}7$
Batch size	8,16, **32**, 64, 128
Activation function	**ReLU**, eLU, $PReLU$, $Leaky\ ReLU$, $tanh$, $softmax$
Optimization algorithms	***Adam***, $RMSprop$, Adadelta, SGD

CNNs offer a highly effective architecture in the field of deep learning, finding a wide range of applications in image processing and classification tasks. These networks are designed with a modular structure. This means that each component can be independently optimized and replaced as needed. The modular structure is characterized by the arrangement of layers in a specific order and sequence of operations. A typical CNN architecture consists of convolution layers, activation functions, pooling layers and fully connected layers. Each module fulfills a specific function. Convolution layers are used to extract features from images, while activation functions come into play to speed up the learning process and increase the non-linearity of the network. Pooling layers reduce the computational burden of the model by summarizing the information, while fully connected layers perform classification. This modular structure increases the ability of researchers to modify and optimize specific components, which makes it possible to develop more flexible and efficient models [50].

In addition to modular structures, the use of specialized loss functions also plays a critical role in the success of CNNs. Instead of traditional loss functions, this study uses a custom loss function designed for a specific application. Specifically developed for binary classification tasks such as handwritten signature images, the Si-CL loss function optimizes the model's learning process and increases sensitivity to certain types of errors. This special loss function improves the classification accuracy of the model, leading to a better performance on binary images [44]. Consequently, careful design and implementation of special loss functions is an important strategy to improve the overall performance of CNNs.

The dataset used in this study was created with a structure in which 80% of the total data was used for training and 20% was reserved for testing. Another 20% of the training set is reserved for the validation process of the model. This approach is critical to evaluate the performance of the model during training and to avoid overfitting. While the training data is actively used in the learning process of the model, the validation set is used to monitor the overall performance of the model and perform parameter adjustments. The test data is used to evaluate the real-world performance of the model after the training process is completed.

During the training process, the early stopping method was also used to reduce overfitting. This method involves monitoring the performance of the model on the training data as well as on the validation data at each epoch. If the model does not improve its performance on the validation set for a certain number of consecutive epochs, the early stopping mechanism is activated. This prevents overfitting of the model and helps to ensure more general and robust learning [51]. Thus, the efficiency of the training process is increased and the overall performance of the model is optimized. Early stopping also allows the training time to be managed efficiently, thus saving time and resources.

The "Shape-DU" and "MPEG-7" datasets were employed to train the Bi-CNN and Bi-CL-CNN models, as well as the benchmark models discussed in the subsequent sections. Consequently, the results across

two distinct datasets were assessed. Moreover, the performance of both the benchmark models and the proposed Bi-CNN and Bi-CL-CNN models was evaluated separately for each dataset.

Benchmarking Models

In this study, popular transfer learning architectures such as GoogleNet, DenseNet201 and ResNet50 are used to compare the classification performance of two different proposed models. Transfer learning is a technique based on the reuse of pre-trained models for a similar task. This method accelerates the learning process of the model and improves the overall performance, especially when the dataset is limited [52]. Transfer learning is a powerful tool for maximizing the ability of deep learning models to learn complex features.

GoogleNet is characterized by its multi-layered architecture and "Inception" modules. This architecture allows the model to learn features at various scales by using filters of different sizes simultaneously. Thus, it can capture different details in images more effectively [53]. The depth and complexity of GoogleNet improve its overall performance and accelerates the learning process.

DenseNet201 is notable for its dense connections between layers. This architecture increases the information flow and strengthens the learning ability of the model by transmitting the output of each layer to the next layer. Thus, higher accuracy is achieved with fewer parameters and the risk of overlearning is reduced [54]. These features of DenseNet201 make a significant contribution to improve the overall performance of the model.

ResNet50 is an architecture built on the concept of "Residual Learning". It allows deeper networks to be trained efficiently by using skip connections to solve the problem of gradient loss during the training of deep networks [55]. In this way, the learning process of deep structures becomes more stable and the overall accuracy of the model is increased.

The selection of these three models was based on their success in the field of deep learning and their flexibility in various tasks. Each model can be adapted quickly and efficiently through transfer learning, thus improving the classification performance of binary images. In this paper, present the performance of the two proposed models and compare them with these pre-trained models. Table 3 illustrates the comparative performance of the proposed models against these pre-trained models. Additionally, Table 4 provides a comparison of the trainable parameters of the benchmark models used in this study, employing the transfer learning technique.

In this study, the final classification layers of these pre-trained models were excluded from the newly generated models. As illustrated in Figure 6, a fully connected layer was incorporated into the model following the foundational layers, substituting the initial classification layers. These custom models were then employed to classify binary images across two distinct datasets.

Table 3. *Comparison of the proposed models with pre-trained models*

Pre-Trained Models	Top-1 Accuracy	Depth
GoogleNet	66.5	22
Densenet201	77.3	201
Resnet50	74.9	50
Bi-CNN	71.3	29
Bi-CL-CNN	72.4	29

Top-1 accuracy is a metric that measures the success of the classification model in predicting the correct class with the highest probability and refers to the proportion of correctly predicted instances. This metric checks whether the true class matches the highest probability predicted by the model and is widely used as a measure of classification success [56].

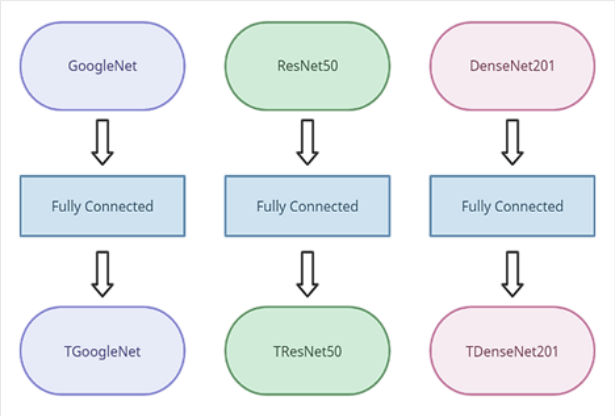

Figure 6. *An example of how benchmarking models are built*

Table 4. *An analysis comparing the number of trainable parameters in each deep neural network's applied transfer learning version and original version.*

Models	Total number of parameters trained without freezing layers	Total number of parameters trained with freezing layers
Googlenet	5.6M	7.1K
Resnet50	25.6M	14.3K
Densenet201	20.2M	3.8K
Bi-CNN	2.1M	-
Bi-CL-CNN	2.1M	-

Experimental Results

The developed models and transfer learning approaches are tested on MPEG7 and our own Shape-DU binary image datasets. These datasets represent different application domains, allowing a comprehensive evaluation of the performance of the model under various scenarios. Various evaluation metrics such as accuracy, precision, recall and F1 score were used to measure the classification performance.

Accuracy is a measure of how many of the model's total predictions are correct. This metric plays an important role in understanding the overall success of the model; however, it can be misleading in

imbalanced datasets. Accuracy is calculated in terms of true positives (TP) and true negatives (TN). True positives are instances that the model correctly classifies as positive, while true negatives are instances that the model correctly classifies as negative. However, false positives (FP), i.e. instances that the model incorrectly classifies as positive, and false negatives (FN), i.e. true positive instances that the model incorrectly classifies as negative, should also be considered in this measure.

Precision assesses how many of the instances that the model classifies as positive are actually positive. That is, it shows the accuracy of the instances where the model says "yes". This metric is critical in situations where false positives are important. Precision is defined as the ratio of true positives (TP) to total positive predictions (TP + FP). Recall measures how well the model detects true positive samples. That is, it is calculated as the ratio of true positives (TP) to total true positives (TP + FN). This metric shows how many true positive examples the model correctly classifies and focuses on reducing false negatives. The F1 score provides a combined assessment of the two metrics, showing the balance between precision and recall. Especially in imbalanced data sets, the F1 score more accurately reflects the overall performance of the model. Therefore, it is important to use these metrics together to ensure an accurate assessment. Equation 2 shows accuracy, equation 3 shows precision, equation 4 shows recall, and equation 5 contains the formulas for F1-score.

$$Accuracy = \frac{(\mathrm{TP} + \mathrm{TN})}{(\mathrm{TP+FP+TN+FN})} \quad (2)$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FP})} \quad (3)$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FN})} \quad (4)$$

$$\mathrm{F1\ Score} = 2 \times \frac{\mathrm{Precision \times Recall}}{(\mathrm{Precision} + \mathrm{Recall})} \quad (5)$$

In addition to the aforementioned metrics, training time is also considered as a performance benchmark for deep learning (DL) platforms. In machine learning processes, datasets are usually divided into three different subsets: (1) a training set for model training, (2) a validation set to evaluate the training effects, and (3) a test set to evaluate the final accuracy of the model. This structure allows the evaluation of the effectiveness of the learning process and the overall performance of the model.

In this study, 20% of the total dataset is designated as the test set. The remaining data is further partitioned by allocating 80% to the training set and 20% to the validation set. As a result, the dataset is structured such that 64% is used for training, 16% for validation and 20% for testing. These proportions are critical for the model to train correctly and improve its performance. The training set is actively used in the learning process of the model, while the validation set is used to evaluate the overall performance of the model and to perform hyperparameter adjustments.

The confusion matrix, which evaluates the class separation ability of a classifier, is a standard tool used to analyze the performance of the model in detail. The elements of the confusion matrix help us to understand the success of the model by showing the correctly and misclassified examples. In this

context, the effectiveness of the proposed models in classifying binary images is evaluated using the confusion matrix.

Furthermore, Table 5 contains the details of the datasets used in this study and the number of instances within each set, clearly showing which data was used in the training and testing processes of the model. This structure increases the reliability of the results obtained when evaluating the overall performance of the model.

Table 5. *Summary of the datasets utilized in this research*

Datasets	Training set	Validation set	Test set	Number of classes	Number of samples in each class	Total images
Shape-DU	1280	320	400	20	100	2000
MPEG-7	896	224	280	70	20	1400

Experiments With the Shape-DU Dataset

A dataset of binary images, designated as "Shape-DU," was utilized to train and evaluate both the benchmark and proposed models. Table 6 summarizes the accuracy of these models assessed when using the "Shape-DU" dataset, based on experimental results. Among the five models tested, the Bi-CL-CNN model demonstrated the highest performance, achieving an accuracy of 93.18%.

Figure 7 and Figure 8 present the accuracy graphs computed during the training phases of the Bi-CNN and Bi-CL-CNN models. The accuracy results obtained from the "Shape-DU" dataset are as follows: the Bi-CNN model achieved an accuracy of 90.63%, while the benchmark models performed as follows: DenseNet201 attained 84.38%, GoogleNet achieved 78.15%, and ResNet50 reached 78.13%. Additionally, Figure 9 illustrates the accuracy values of the benchmark models obtained from the "Shape-DU" dataset.

Table 6. *Classification performances of the models after training and assessment on the "Shape-Du" dataset*

CNN models	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
TGoogleNet	78.15	78.23	78.15	78.18
TDenseNet201	84.38	84.42	84.38	84.40
TResNet50	78.13	78.10	78.13	78.11
Bi-CNN	90.63	90.59	90.63	90.60
Bi-CL-CNN	**93.18**	**93.02**	**93.18**	**93.09**

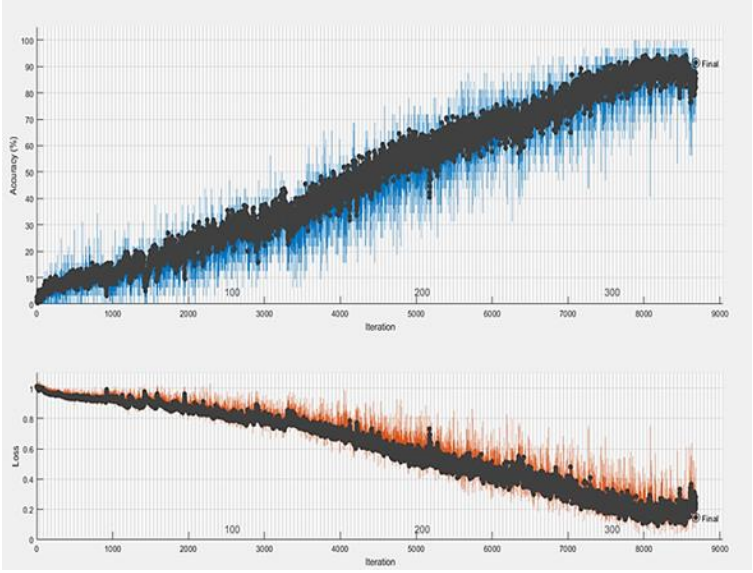

Figure 7. *Accuracy and loss graphs of the Bi-CNN model for the training and validation sets obtained during training with the "Shape-DU" dataset*

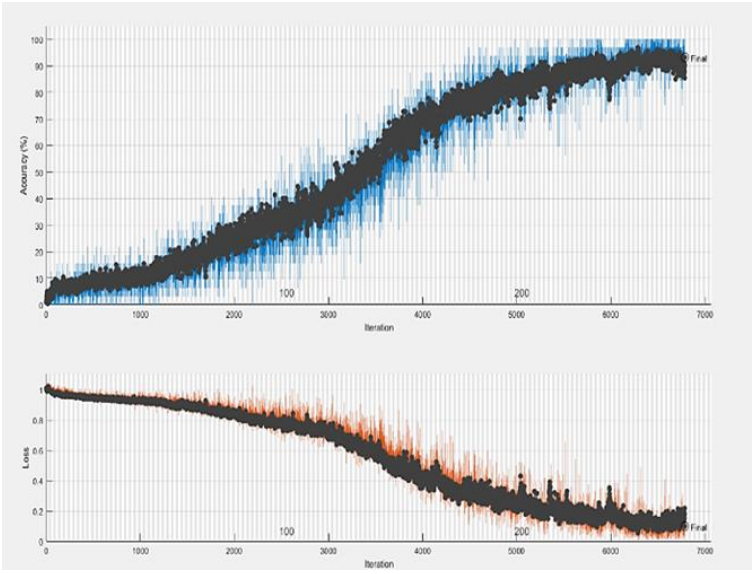

Figure 8. *Accuracy and loss graphs of the Bi-CL-CNN model for the training and validation sets obtained during training with the "Shape-DU" dataset*

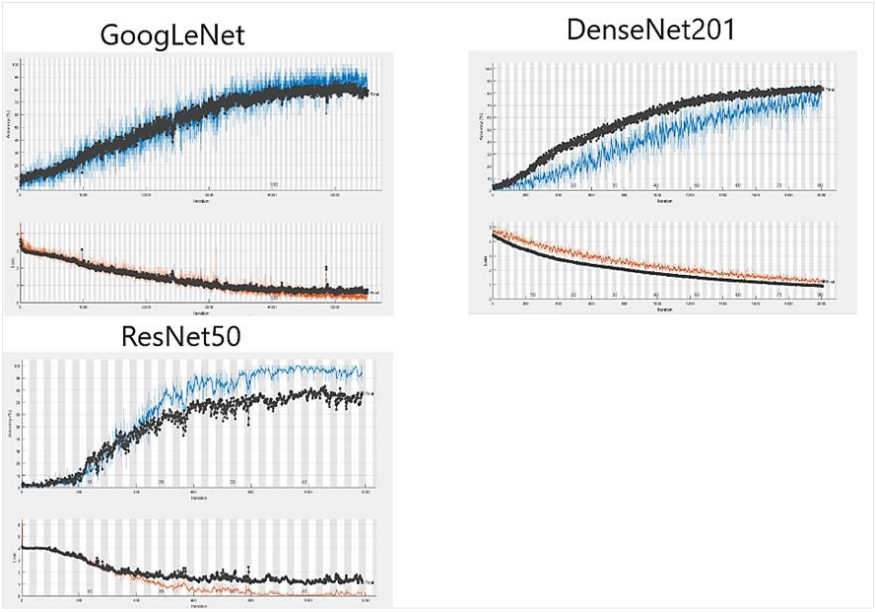

Figure 9. *Accuracy and loss graphs of the benchmark model for the training and validation sets obtained during training with the "Shape-DU" dataset*

Due to their intricate and deep architectures, extensive data processing requirements, and numerous parameters, CNN models necessitate considerable training time, even on high-performance computing systems with advanced processing speeds and large memory capacities. Accordingly, Table 7 presents the training times for each CNN model when evaluated using the "Shape-DU" dataset.

Table 7. *Training times for the models used on the "Shape-DU" dataset*

CNN Models	Training Time
TGoogleNet	92 min. 15 sec.
TDenseNet201	129 min. 45 sec.
TResNet50	75 min. 35 sec.
Bi-CNN	87 min. 22 sec.
Bi-CL-CNN	**81 min. 33 sec.**

The experimental results reveal that the ResNet50 model was the fastest, completing its training in 4535 seconds. The Bi-CL-CNN and Bi-CNN models followed, with training durations of 4893 seconds and 5242 seconds, respectively. Among the benchmark models, DenseNet201 exhibited the greatest depth, resulting in the longest training time. The training times shown in the table correspond to the fine tuning process only. These times reflect the process of making the pre-trained weights more appropriate for the "Shape-DU" dataset. This refers to the process of adapting the model to the new data and differs from training from scratch or inference time, as this process requires less time and computational resources. When comparing the training times of the Bi-CNN and Bi-CL-CNN models, differences are noted, attributable to the distinct loss functions employed. Specifically, the loss function utilized in the Bi-CL-CNN model accelerates the training process by providing more accurate error feedback. This custom loss function allows the model weights to be optimized more quickly during the tuning phase, resulting

in faster training time. Consequently, significant variations in network performance, in terms of both speed and accuracy, are observed due to the application of the specialized loss function. These differences emphasize the importance of carefully choosing the loss function, as it can significantly affect both the computational efficiency of the model and its success in achieving high accuracy.

Experiments With the MPEG-7 Dataset

The "MPEG-7" binary image dataset was utilized to train and evaluate the Bi-CNN and Bi-CL-CNN models, along with other benchmark models. Table 8 summarizes the accuracies of the models based on the "MPEG-7" dataset, as determined from the experimental results. The Bi-CL-CNN model exhibited the highest performance among the five models, achieving an accuracy of 99.18%. Figures 10 and 11 display the accuracy graphs for the Bi-CNN and Bi-CL-CNN models, respectively, as recorded during their training phases.

The accuracy results for the "MPEG-7" dataset are as follows: the Bi-CNN model attained an accuracy of 95.91%, DenseNet201 achieved 95.16%, GoogleNet reached 91.50%, and ResNet50 recorded an accuracy of 89.50%. Figure 12 illustrates the accuracy values for the benchmark models derived from the "MPEG-7" dataset.

Table 8. *Classification performances of the models after training and assessment on the "MPEG-7" dataset*

CNN models	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
TGoogleNet	91.50	91.29	91.52	91.40
TDenseNet201	95.16	94.92	95.05	94.98
TResNet50	89.50	89.12	89.54	89.33
Bi-CNN	95.91	95.73	95.80	95.76
Bi-CL-CNN	**99.18**	**99.02**	**99.22**	**99.12**

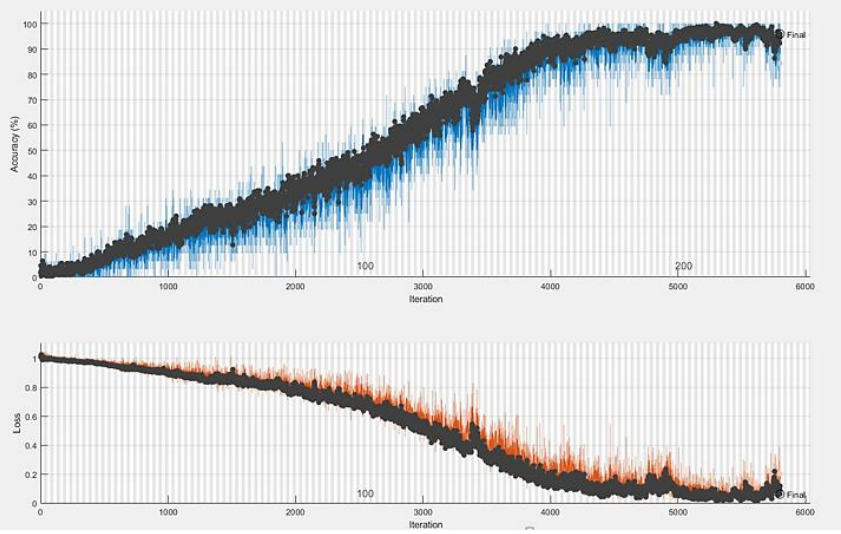

Figure 10. *Accuracy and loss graphs of the Bi-CNN model for the training and validation sets obtained during training with the "MPEG-7" dataset*

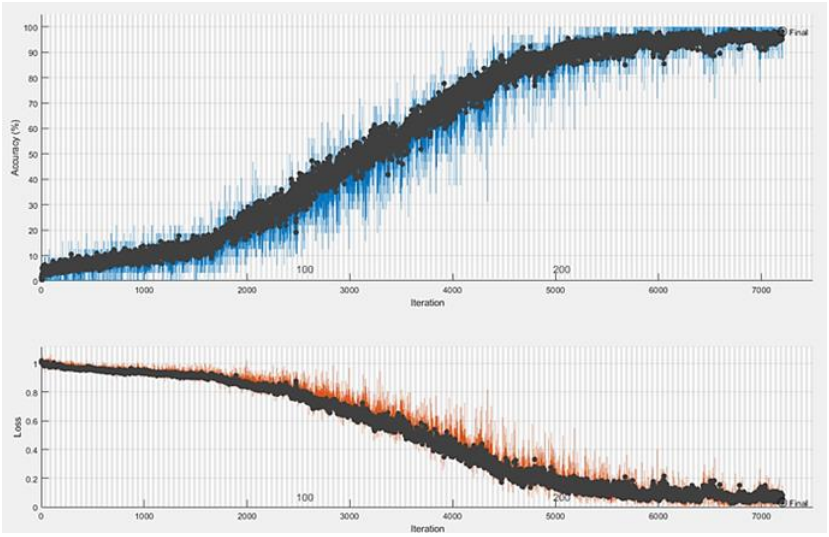

Figure 11. *Accuracy and loss graphs of the Bi-CL-CNN model for the training and validation sets obtained during training with the "MPEG-7" dataset*

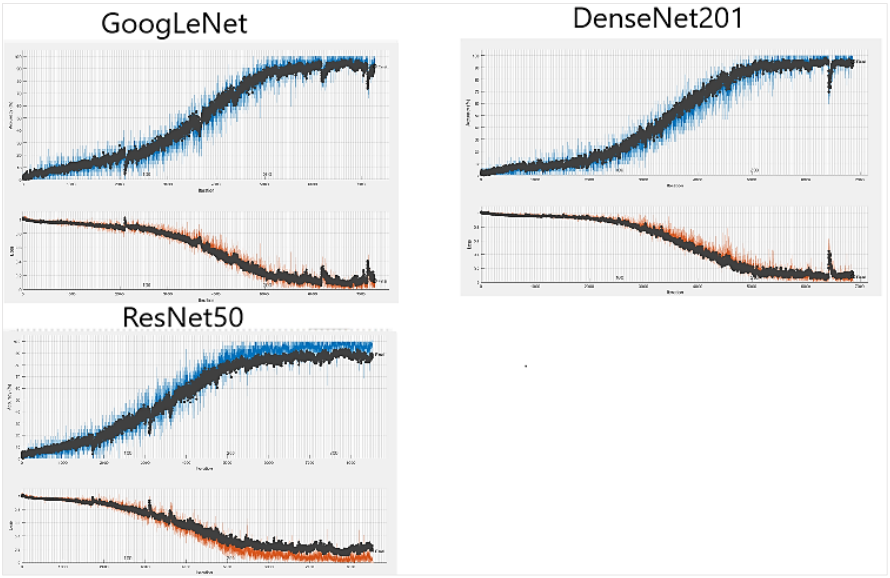

Figure 12. *Accuracy and loss graphs of the benchmark model for the training and validation sets obtained during training with the "MPEG-7" dataset*

The analysis of the experimental results using the "MPEG-7" dataset and the training times for each CNN model are presented in Table 9.

Table 9. *Training times for the models used on the "MPEG-7" dataset*

CNN models	**Training Time**
TGoogleNet	79 min. 30 sec.
TDenseNet201	127 min. 17 sec.
TResNet50	76 min. 25 sec.
Bi-CNN	72 min. 15 sec.
Bi-CL-CNN	**65 min. 16 sec.**

Table 9 shows that the Bi-CL-CNN model is the fastest model, completing the training in 3916 seconds.

It was followed by the Bi-CNN and ResNet50 models, which completed the trainings in 4335 and 4585 seconds, respectively. Consistent with the results from the Shape-DU dataset, DenseNet201 had the longest training time.

The overarching conclusion from these results is that training times increase with the depth of the neural network. A comparative analysis of training times and accuracy rates for the Bi-CNN and Bi-CL-CNN models underscores the significant role of the classification layer. The use of the specialized loss function in the Bi-CL-CNN model facilitates faster error feedback, thereby reducing overall training time. This indicates that the choice of loss function has a substantial impact on the network's performance, with implications for both speed and accuracy.

Discussion

In evaluating the performance of the proposed models on the "Shape-DU" dataset, which was specifically developed for this study, the models demonstrated favorable outcomes. To ensure a rigorous and unbiased comparison, it is crucial to test the models using established datasets commonly utilized in prior research. Consequently, the "MPEG-7" dataset was employed alongside the "Shape-DU" dataset for comparative analysis. The results of the assessments using the "MPEG-7" dataset are summarized in Table 10, providing a detailed overview of the findings.

Table 10. *Comparing similar studies that were trained and assessed using the "MPEG-7" dataset*

References	Method used	Accuracy (%)
Yang et al. [20]	SVM	96.27
Patel et al. [21]	SVM	98.41
Bicego et al. [57]	CNN+SVM	98.10
Govindaraj et al. [58]	k-NN	95.09
Jayasumana et al [59]	SVM	96.57
Bi-CNN	CNN	95.91
Bi-CL-CNN	CNN	**99.18**

The evaluation of the MPEG-7 dataset reveals that Support Vector Machines (SVM) and their hybridization with CNN yield high accuracy rates. Specifically, Patel et al. [21] achieved the highest accuracy of 98.41% using SVM alone. Similarly, Bicego et al. [57] demonstrated competitive performance with a CNN-SVM hybrid approach, attaining an accuracy of 98.10%.

In contrast, our proposed models, Bi-CNN and Bi-CL-CNN, exhibit notable advancements. The Bi-CNN model achieved an accuracy of 95.91%, while the Bi-CL-CNN model surpassed all other methods with an impressive accuracy of 99.18%. This substantial improvement highlights the Bi-CL-CNN model's superior capability in classifying binary images from the MPEG-7 dataset.

The enhanced performance of the Bi-CL-CNN model can be attributed to its advanced architecture, which significantly improves feature extraction and classification accuracy. The integration of the Class-Level

(CL) component within the Bi-CL-CNN model likely provides additional contextual information, thereby facilitating more precise classification.

Overall, these results underscore the efficacy of our proposed models, particularly the Bi-CL-CNN, in elevating classification accuracy for binary images in the MPEG-7 dataset. This suggests that our approach has the potential to establish new benchmarks for classification performance in this domain.

The superior performance of the Bi-CL-CNN model can be attributed to its innovative architecture, specifically the incorporation of the Class-Level (CL) component. This enhancement likely improves the model's ability to capture detailed features and contextual information, contributing to its increased accuracy. Due to the different loss functions in the proposed models, Table 11 presents a comparison of training times.

Table 11. *Comparison of training times of Bi-CNN and Bi-CL-CNN models.*

Datasets	Bi-CNN	Bi-CL-CNN	Ratio of Bi-CL-CNN to Bi-CNN
Shape-DU	87 min. 22 sec.	81 min. 33 sec.	1.071
MPEG-7	72 min 15 sec.	65 min 16 sec.	1.107

In addition, this study compares the performance of Bi-CNN and Bi-CL-CNN models and the computation time of each model. The results show that a significant speed difference occurs between the models, which can be attributed to the nature of the loss functions used.

For the Shape-DU dataset, the Bi-CNN model took 87 minutes and 22 seconds to process, whereas the Bi-CL-CNN model completed the task in 81 minutes and 33 seconds, demonstrating approximately 7.1% faster performance. In the MPEG-7 dataset, the Bi-CNN model's processing time was 72 minutes and 15 seconds, while the Bi-CL-CNN model reduced this to 65 minutes and 16 seconds, resulting in a 10.7% increase in speed.

The primary reason for these differences lies in the structural variations of the loss functions used. The Bi-CNN model employs the classic cross-entropy loss function, which is widely utilized to minimize the discrepancy between the model output and the target class. However, this loss function may extend processing time, especially when dealing with high-dimensional datasets and multi-class problems. Conversely, the Bi-CL-CNN model utilizes the Si-CL function, which more effectively models the similarities and differences between data points, resulting in a faster learning process. The Si-CL loss function is particularly beneficial in complex tasks, where it enhances both the speed and accuracy of the model.

In conclusion, the Si-CL loss function employed in the Bi-CL-CNN model provides a more efficient learning process compared to the cross-entropy loss function used in the Bi-CNN model. This efficiency is reflected in the significant reductions in processing times observed. Particularly in large and complex datasets, the speed advantage conferred by the Si-CL loss function makes the Bi-CL-CNN model a more suitable option for applications where processing time is critical. This study underscores the effectiveness

of the Si-CL loss function in enhancing the performance of deep learning models. The results confirm that this loss function is effective not only for its intended purpose but also in improving the classification performance of binary images, validating its broader applicability in image classification tasks.

The Bi-CNN and Bi-CL-CNN architectures offer notable advantages over other models. A significant benefit is their capacity to achieve high-precision classification with a relatively small number of training examples, thanks to the robust feature extraction capabilities inherent in CNNs. However, a key disadvantage is the substantial training time required due to the large number of parameters to be processed. Despite this limitation, the results indicate that these models are both highly effective and satisfactory for binary image classification. Consequently, they hold promise as valuable tools in applications such as barcode reading, QR code scanning, and handwriting analysis. Their precision and feature extraction capabilities make them particularly suited for these tasks. Future research will focus on further optimizing these models and assessing their performance on additional datasets and more complex classification tasks.

Conclusion

The convenience and benefits of technology have made the identification and detection of objects in digital photographs increasingly important. Although many studies have been carried out in the fields of machine learning and image processing, there is a lack of adequate methods for categorizing binary images. In this paper, we introduce two innovative CNN-based models, Bi-CNN and Bi-CL-CNN, for categorizing binary images. Two different datasets are used to evaluate the effectiveness of Bi-CNN and Bi-CL-CNN models. First, the "Shape-DU" dataset is used to train and evaluate these models. The experimental results show that the Bi-CNN model performs well with an accuracy of 93.18%, while the Bi-CL-CNN model achieves 90.63% accuracy. Three different CNN models were used to compare the effectiveness of Bi-CNN and Bi-CL-CNN models. The Bi-CL-CNN model performed the best among the benchmark models with an accuracy of 99.18%. Moreover, the training time of the Bi-CL-CNN model was found to be shorter since it provides a more accurate error transformation of the loss function in the classification layer.

Future studies aim to develop various strategies to improve the performance of the proposed Bi-CNN and Bi-CL-CNN models. First, diversifying the datasets will be a critical step to strengthen the generalization capability of the model; in this regard, the integration of larger and heterogeneous datasets for different application domains will allow for evaluating the effectiveness of the model under various conditions. Furthermore, the application of data augmentation techniques and regularization methods to address the problem of overlearning can increase the robustness of the model. Optimization of model architectures can be achieved by adapting pre-trained networks through transfer learning, which has the potential to offer performance improvements on more complex data sets. The study of different loss functions and optimization algorithms can also make the model learning processes more efficient. The integration of advances in this field into industrial applications will increase the practical utility of these models in real-

world scenarios, and working on potential use cases in various fields such as health, safety and automation will expand the impact of the research. In this regard, the innovative approaches offered by the proposed models will continue to make significant contributions both to the academic literature and to the applied fields.

Acknowledgements -

Funding/Financial Disclosure The author has no received any financial support for the research, author, or publication of this study.

Ethics Committee Approval and Permissions The work does not require ethics committee approval and any private permission.

Conflict of Interests The author stated that there are no conflict of interest in this article.

Authors Contribution Author read and approved the final manuscript.

References

[1] Li, S., Song, W., Fang, L., Chen, Y., Ghamisi, P., & Benediktsson, J. A. (2019). Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing, 57*(9), 6690-6709. https://doi.org/10.1109/TGRS.2019.2907932

[2] Das, D., Naskar, R., & Chakraborty, R. S. (2023). Image splicing detection with principal component analysis generated low-dimensional homogeneous feature set based on local binary pattern and support vector machine. *Multimedia Tools and Applications, 82*, 25847-25864.

[3] Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., & Terzopoulos, D. (2022). Image segmentation using deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(7), 3523-3542.

[4] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International Journal of Computer Vision, 128*(2), 261-318.

[5] Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence, 2*(1), 41-50.

[6] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of IEEE, 86*(11), 2278-2324.

[7] Chollet, F. (2017). Xception: Deep learning with Depthwise separable convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1800-1807.

[8] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *CoRR, abs/1409*, 1-14.

[9] Makwe, A., & Rathore, A. S. (2021). An empirical study of neural network hyperparameters. *Evolution in Computational Intelligence: Frontiers in Intelligent Computing: Theory and Applications (FICTA 2020), 1*, 371-383.

[10] Zhang, R., R., Zheng, Y., Mak, T. W. C., Yu, R., Wong, S. H., Lau, J. Y., & Poon, C. C. (2017). Automatic detection and classification of colorectal polyps by transferring low-level CNN features from nonmedical domain. *IEEE Journal of Biomedical and Health Informatics, 21*(1), 41-47. https://doi.org/10.1109/JBHI.2016.2635662

[11] Ou, X., Wang, H., Liu, X., Zheng, J., Liu, Z., Tan, S., & Zhou, H. (2023). Complex scene segmentation with local to global self-attention module and feature alignment module. *IEEE Access, 11*, 96530-96542. https://doi.org/10.1109/ACCESS.2023.3311264

[12] Jiang, J., Feng, X., Liu, F., Xu, Y., & Huang, H. (2019). Multi-spectral RGB-NIR image classification using double-channel CNN. *IEEE Access, 7*, 20607-20613. https://doi.org/10.1109/ACCESS.2019.2896128

[13] Russ, J. C., Matey, J. R., Mallinckrodt, A. J., & McKay, S. (1994). The image processing handbook. *Computers in Physics, 8*(2), 177-178. https://doi.org/10.1063/1.4823282

[14] Bayram, S., & Barner, K. (2023). A black-box attack on optical character recognition systems. In M. Tistarelli, S. R. Dubey, S. K. Singh, & X. Jiang (Eds.), *Computer Vision and Machine Intelligence* (pp. 18-34). Springer. https://doi.org/10.1007/978-981-19-7867-8_18

[15] Kabakus, A. T., & Erdogmus, P. (2021). A novel handwritten Turkish letter recognition model based on convolutional neural network. *Concurrency and Computation: Practice and Experience, 33*(21), e6429.

[16] Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(7), 971-987. https://doi.org/10.1109/TPAMI.2002.1017623

[17] Mujawar, S., Kiran, D., & Ramasangu, H. (2018). An efficient CNN architecture for image classification on FPGA accelerator. *2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, 1-4.

[18] Chittajallu, S. M., Mandalaneni, N. L. D., Parasa, D., & Bano, S. (2019). Classification of binary fracture using CNN. *2019 Global Conference for Advancement in Technology (GCAT)*, 1-5.

[19] Khandelwal, P., Khandelwal, A., & Agarwal, S. (2020). Using computer vision to enhance safety of workforce in manufacturing in a post-COVID world. *arXiv*. https://doi.org/10.48550/arXiv.2005.05287

[20] Yang, C., Fang, L., & Wei, H. (2020). Learning contour-based mid-level representation for shape classification. *IEEE Access, 8*, 157587-157601. https://doi.org/10.1109/ACCESS.2020.3019800

[21] Patel, V., Mujumdar, N., Balasubramanian, P., Marvaniya, S., & Mittal, A. (2019). Data augmentation using part analysis for shape classification. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1223-1232. https://doi.org/10.1109/WACV.2019.00135

[22] Ramya, N., Om, A. S. B., & Subashini, V. (2022). Detection of pneumonia by binary image classification using hybrid neural networks. *2022 1st International Conference on Computational Science and Technology (ICCST)*, 1-5. https://doi.org/10.1109/ICCST55948.2022.10040322

[23] Jayalakshmi, G. S., & Kumar, V. S. (2019). Convolutional neural network (CNN) based cancerous skin lesion detection system. *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, 1-6. https://doi.org/10.1109/ICCIDS.2019.8862143

[24] Ramanjaneyulu, K., Swamy, K. V., & Rao, C. S. (2018). Novel CBIR system using CNN architecture. *2018 3rd International Conference on Inventive Computation Technologies (ICICT)*, 379-383. https://doi.org/10.1109/ICICT43934.2018.9034389

[25] Wang, P., Li, L., Jin, Y., & Wang, G. (2018). Detection of unwanted traffic congestion based on existing surveillance system using in freeway via a CNN-architecture trafficnet. *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 1134-1139. https://doi.org/10.1109/ICIEA.2018.8397881

[26] Hafeez, M. A., Munir, A., & Ullah, H. (2024). H-QNN: A hybrid quantum–classical neural network for improved binary image classification. *AI, 5*(3), 1462-1481.

[27] Ranga, D., Prajapat, S., Akhtar, Z., Kumar, P., & Vasilakos, A. V. (2024). Hybrid quantum–classical neural networks for efficient MNIST binary image classification. *Mathematics, 12*(23), 1-22.

[28] Korkut, Ş. G., Kocabaş, H., & Kurban, R. (2024). A comparative analysis of convolutional neural network architectures for binary image classification: A case study in skin cancer detection. *Karadeniz Fen Bilimleri Dergisi, 14*(4), 2008-2022.

[29] Hafeez, M. A., Munir, A., & Ullah, H. (2024). H-QNN: A Hybrid Quantum–Classical Neural Network for Improved Binary Image Classification. *AI*, 5(3), 1462-1481.

[30] Bai, Q., & Hu, X. (2023). Quantity study on a novel quantum neural network with alternately controlled gates for binary image classification. *Quantum Information Processing,* 22(5), 184.

[31] Kiger, J., Ho, S. S., & Heydari, V. (2022, March). Malware binary image classification using convolutional neural networks. *In International Conference on Cyber Warfare and Security*, *17*(1), 469-478. Academic Conferences International Limited. https://doi.org/10.34190/iccws.17.1.59

[32] Pal, K. K., & Sudeep, K. S. (2016). Preprocessing for image classification by convolutional neural networks. *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 1778-1781. https://doi.org/10.1109/RTEICT.2016.7808140

[33] Latecki, L. J., Lakamper, R., & Eckhardt, T. (2000). Shape descriptors for non-rigid shapes with a single closed contour. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1, 424-429.

[34] LeCun, Y., Cortes, C., & Burges, C. (2010). MNIST handwritten digit database.

[35] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*.

[36] LeCun, Y., Bottou, L., Orr, G. B., & Müller, K. R. (1998). Gradient-based learning applied to document recognition. *Proceedings of IEEE*.

[37] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[38] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

[39] Szegedy, C., Vanhoucke, V., Ioffe, S., Vinyals, O., & Wu, Y. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[40] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT)*.

[41] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on Machine Learning*.

[42] Wang, Z., Zhang, L., & Wang, F. (2016). Understanding max pooling in CNNs. *arXiv preprint arXiv:1509.02520*.

[43] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*.

[44] Ozkan, Y., & Erdogmus, P. (2024). Evaluation of classification performance of new layered convolutional neural network architecture on offline handwritten signature images. *Symmetry, 16*(6), 649.

[45] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

[46] Kingma, D. P., & Welling, M. (2014). Auto-Encoding variational Bayes. *International Conference on Learning Representations*.

[47] Smith, L. N. (2017). Cyclical learning rates for training neural networks. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*.

[48] Keskar, N. S., Gonzalez, J. R., Dylan, Y., & Bengio, Y. (2016). On how to train a very deep neural network. *Proceedings of the 5th International Conference on Learning Representations (ICLR)*.

[49] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[50] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of IEEE*.

[51] Prechelt, L. (1998). Early stopping – but when? In *Neural Networks: Tricks of the Trade* (pp. 55-69). Springer.

[52] Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*.

[53] Szegedy, C., Vanhoucke, V., Google, L. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[54] Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[55] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[56] Yang, E. H., Amer, H., & Jiang, Y. (2021). Compression helps deep learning in image classification. *Entropy*, 23(7), 881.

[57] Bicego, M., & Lovato, P. (2016). A bioinformatics approach to 2D shape classification. *Computer Vision and Image Understanding, 145*, 59-69.

[58] Govindaraj, P., & Sudhakar, M. S. (2019). A new 2D shape retrieval scheme based on phase congruency and histogram of oriented gradients. *Signal, Image and Video Processing, 13*, 771-778.

[59] Jayasumana, S., Salzmann, M., Li, H., & Harandi, M. (2013). A framework for shape analysis via Hilbert space embedding. *Proceedings of the IEEE International Conference on Computer Vision*, 1249-1256.