[1]Himani Jain

[2]Amit Dixit

Multilayer Perceptron's Neural Network Based Image Spam Detection on Social Media

Abstract: - Spam, typically unwanted material, can manifest in various forms, including images. While numerous machine learning techniques excel in detecting textual spam, they often falter when it comes to identifying image-based spam. This paper introduces a novel framework designed specifically for identifying image spams. Images are categorized into two groups: spam images, containing undesirable material, and ham images, encompassing everything else. In this paper, a novel technique based on CNN and gated recurrent unit (GRU) for image spam detection has been proposed. Our proposed methodology hinges on the utilization of diverse pre-trained deep learning models, such as InceptionV3, DenseNet121 (Densely Connected Convolutional Networks 121), ResNet50 (Residual Networks), VGG16 (Visual Geometry Group), and MobileNetV2, to effectively filter out unwanted spam images. We evaluate the performance of our approach using different Dataset. Additionally, we address the challenge of limited labeled data by leveraging transfer learning and employing data augmentation techniques. Experimental results demonstrate the efficacy of our proposed model, achieving impressive accuracy levels while maintaining computational efficiency, with testing times ranging from one to two seconds for the challenge dataset.

Keywords: Deep Learning Framework, Data Augmentation, Pre-trained Models, Transfer Learning

1. Introduction

In the contemporary digital landscape, the practice of sharing moments via photos and videos on social media platforms has experienced an unprecedented surge in popularity. However, this widespread adoption and utilization of social media have also attracted individuals who exploit these platforms for personal gain, often through the dissemination of spam, including advertising content. Consequently, there exists a critical need for the development of robust systems capable of detecting and filtering out such spam, thereby ensuring that users can access authentic and meaningful information. While previous studies in the realm of image spam detection have predominantly concentrated on employing traditional classification methods to filter out inappropriate content, recent advancements have ushered in the era of Deep Convolutional Neural Networks (DCNN). This innovative technique has demonstrated superior accuracy in image classification tasks, obviating the necessity for manual feature extraction processes. The advent of deep learning technologies in the domain of image analysis promises a novel approach to security applications. By harnessing the potential of CNNs, raw data inputs—such as the image itself—can be processed, thereby automatically extracting crucial low-level features. However, it has been observed that the detection accuracy of existing CNN-based image spam detection models may significantly degrade when confronted with new and unseen instances of image spam.Previously, image spam detection primarily revolved around correctly identifying objects within images, often relying on various machine learning algorithms and handcrafted feature extraction methods. These methods would extract features from images, which could be local, global, or a combination of both, followed by the application of single or ensemble machine learning classification algorithms based on attributes like color, shape, or texture. In the contemporary landscape, the paradigm shift towards deep learning has yielded remarkable results across various computer vision applications, including image classification, object detection, security, and image processing. Deep learning, a subset of machine learning, automates both feature extraction and classification tasks, eliminating the need for manual intervention. The proliferation of image spam, characterized by the embedding of spam text within images, poses a significant challenge, as it enables spammers to circumvent text-based spam filters. The term "ham" is used to differentiate legitimate messages from spam, highlighting the core challenge of distinguishing between genuine

[1] [1] Quantum University, Roorkee, Uttarakhand Ph.D. Scholar, Department of MCA, ABES Engineering College, Ghaziabad, Uttar Pradesh, INDIA Email: Himanijain1987ap@gmail.com

ORCID: 0000-0001-5562-373X

[2]Dean Research Quantum University,Roorkee, Uttarakhand Email: dixitamit777@gmail.com

ORCID: 0000-0003-2697-4279

and spam images. However, one notable drawback of employing deep learning techniques is the requisite for extensive datasets for training, along with significant computational resources for model refinement. Deep learning necessitates substantial memory capacity, particularly during the feature extraction phase, and entails considerable computation time, often spanning several hours or days. Moreover, deep learning algorithms mandate specialized hardware, such as GPUs and TPUs, which can be prohibitively expensive and inaccessible to many. Despite advancements in data augmentation techniques, which aim to augment datasets to enhance model performance, challenges persist in assembling sufficiently large and diverse datasets representative of real-world scenarios. While data augmentation has mitigated some of these challenges, it also engenders increased storage requirements and computational overhead. So, we worked on a pre-trained model with deep learning. In summary, the primary contributions of our work include:

- Leveraging a Deep Learning approach for automatic feature extraction
- Utilizing pre-trained models
- Fine-tuning deep learning models to enhance classification accuracy

The subsequent sections of this paper are organized as follows: Section 2 reviews related works, Section

3 provides an overview of the dataset, Section 4 outlines our proposed methodologies, Section 5 presents experimental results and discussions, and finally, Section 6 concludes the paper.

2. Related Works

Mahmood et al.[1] ‘introduced a ‘hybrid methodology’ for image classification. Their approach utilized the Res Net model to extract features from images, followed by fine-tuning these features using PCA-SVM for the classification task. They conducted experiments on four datasets: MIT-67, MLC, Caltech-101, and Caltech-256. Training the model involved utilizing 30 images from each class, resulting in superior performance compared to alternative methodologies.

Kataoka et al.[2] Published research on evaluating the differences between deep learning methods for object recognition and exploration. Their experiments show that VGGN et architecture is better than AlexNet architecture. Th ey also performed regression analysis by combining s ome criteria of two factories and applied principal co mponent analysis (PCA) to them. They used pedestria n data from Caltech101 and Daimler in the experimen t and achieved 91.8% accuracy..

Ensemble approach that merges local and deep features to classify images. They evaluated multiple pre-trained convolutional neural networks to extract features and compared their performance. Their approach involved combining features extracted from Scale-Invariant Feature Transform (SIFT) with those from various pre-trained neural networks. The model was trained using an SVM classifier and further enhanced with a majority voting scheme for image recognition. Evaluation was performed on the CIFAR- 10 dataset, resulting in an accuracy of 91.8%.[3]’

Fusion method that combines the VGG19 deep learning model for extracting features with the ‘support vector machine (SVM)’ for classifying images. They compared various neural models, including AlexNet, VGG16, and VGG19, for feature extraction. These models were fine-tuned using the GHIM10K and Caltech256 datasets for image classification tasks. Their results revealed that the VGG19 architecture surpassed both AlexNet and VGG16. Evaluation was conducted based on precision, recall, and F-score, illustrating VGG19's superior performance across these metrics[4]’.

Kumar V et al [5] conducted an analysis focusing on the performance variations between ‘Deep Learning (DL) and Classical Machine Learning (CML)’ classifiers. They explored different feature vector representations and proposed an ensemble approach that combines DL and CML for classification tasks. The primary objective of their experiment was to enhance the performance of individual models by leveraging the strengths of both DL and CML techniques.

Several ‘pre-trained Convolutional Neural Network (CNN)’ models with fine-tuning for the purpose of detecting and classifying invasion ‘ductal carcinoma’. The models evaluated included ‘VGG16, VGG19, ResNet50, DenseNet, MobileNet, and Efficient Net’. Among these models, fine-tuned VGG19 demonstrated the most promising results, achieving a sensitivity of 93.05% and precision of 94.46%, which surpassed the performance of the other models tested. The experiment encompassed approximately 90,000 images[6]

Kumaresan et al [7]proposed a technique for ‘detecting image spam based on color features’, employing the k-nearest neighbor (k-NN) algorithm. Their approach relied on RGB and HSV histograms as features. Through their research, they found that a simple k-NN classifier achieved an accuracy of 0.945 in detecting image spam.

‘Support Vector Machines (SVM)’ to a set of 21 image features. By employing feature selection techniques based on linear SVM weights, they achieved an impressive accuracy rate of 97% using a relatively small subset of features. Furthermore, the authors introduced a challenge dataset designed to mimic image spam, which proved to be significantly more challenging to detect compared to real-world image spam instances[8].

Chavda et al [9]performed two sets of experiments utilizing Support Vector Machines (SVM) and image processing techniques. They utilized a comprehensive set of 41 image features and achieved impressive accuracy rates of 97% and 98% on two publicly available datasets. Additionally, the authors introduced a challenge dataset, demonstrating that it posed an even greater difficulty in detection compared to the dataset developed in the study by Annadatha and Stamp (2018).

Fusion model is used to filter spam emails. Their approach involved processing the image and text components separately using a Convolutional Neural Network (CNN) for images and a ‘Long Short- Term Memory’ (LSTM) network for text. Subsequently, they combined the resulting classification probabilities from both models to determine whether the email should be classified as spam or not. This fusion technique allowed for a comprehensive analysis of both image and text content within emails to improve spam detection accuracy[10].

3. Material and Methods

This section elaborates on the datasets utilized for conducting the diverse experiments, as well as the array of deep learning models employed.

3.1 Image spam datasets

The specifics of the datasets employed in the experiment are presented in Table 1.

Table 1: DATASETS USED IN THE EXPERIMENT [22]

Sl No	Dataset Name	Total Used	Total Non-Spam used	Remark
1	Dredze [11]	1085	1029	This dataset comprises 2551 images classified as non-spam, along with 3238images classified as spam, and 9502 images sourced from the Spam Archive..
2	Image Spam Hunter [12]	926	811	This dataset contains 811images categorized as non- spam and 926 images categorized as spam.

3	Improved [13]	1027	811	This dataset includes 811non-spam images and 1027 handcrafted 'improved spam images'.
4	Challenge A [14]	811	812	This dataset comprises 810 non-spam images and 812 handcrafted 'challenge spam dataset images'.
5	Challenge B [14]	812	811	This dataset contains 811 non-spam images and 812 handcrafted 'challenge spam dataset images'.

3.2 Performance Measure

To gauge the efficacy of the proposed approach, we employed various evaluation metrics, including Accuracy, Recall, Precision, and F1-score. False Positive (FP) signifies the number of legitimate images misclassified, False Negative (FN) represents the misclassified spam count, True Positive (TP) denotes the correctly classified spam instances, and True Negative (TN) indicates the correct classification of legitimate emails [22]. The confusion matrix, which defines the 'performance of the classification algorithm', is provided below in Table 2.

TABLE 2. CONFUSION MATRIX [22]

PREDICATED	ACUTAL	
	SPAM	HAM
SPAM	TP	FN
HAM	FP	TN

4. Proposed Model

The architecture of the proposed system is illustrated in Fig. 1. Our method is grounded in a fusion of deep learning features and transfer learning feature extraction algorithms. We utilized four datasets, and following augmentation techniques, we employed a custom CNN with GNU. However, due to excessive processing time, we found CNN alone insufficient. Hence, we applied transfer learning on a different dataset. After employing a pre-trained model, we optimized our approach by adjusting various parameters such as learning rate and batch size, and subsequently, we froze the top layers. Adam optimizer was employed in constructing our model. The proposed model works in two phases: feature extraction and image classification. The first phase CNN and pre-trained model task is performed. During the second phase, we evaluate the performance of the model by applying various machine learning classification algorithms, namely 'LR' (Logistic Regression), 'RF' (Random Forest), 'DT' (Decision Tree), KNN (K-Nearest Neighbors), 'GNB' (Gaussian Naive Bayes), 'AB' (AdaBoost), 'LSVM' (Linear Support Vector Machine), and 'RSVM' (Radial Support Vector Machine). In this process, we utilize a standard data partitioning strategy, allocating '70%' of the images from each class for training purposes and reserving the remaining '30%' for testing the model's recognition capabilities. Subsequently, we predict the performance of the model on the test dataset.

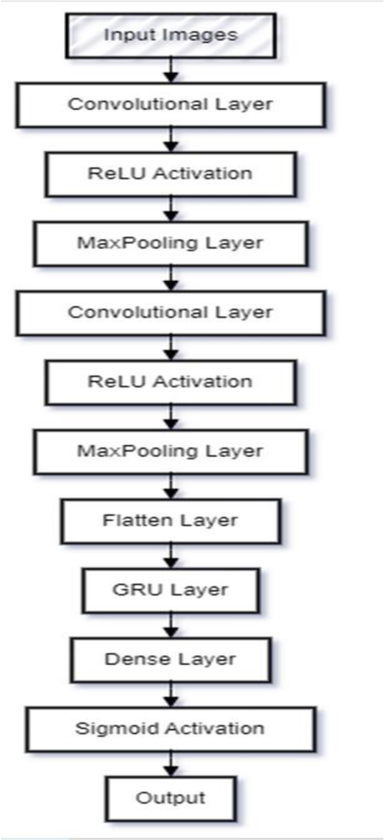


FIGURE 1ARCHITECTURE OF THE PROPOSED SYSTEM

4.1 . Custom CNN Architecture

In this section, we present the methodology employed for the design and evaluation of a ‘custom Convolutional Neural Network (CNN) architecture along with GRU’ for image classification, specifically tailored for spam detection. The entire process involves data preparation, model architecture design, training, evaluation, and subsequent transfer learning for additional analysis.

4.2 Model Architecture

The “Convolutional Neural Network (CNN)” architectures employed in this research follow a sequential design, incorporating convolutional and max-pooling layers for feature extraction and spatial down sampling [25]. The first CNN model consists of four convolutional layers with increasing filter sizes (32, 64, 128, and 256), each followed by ReLU activation and max-pooling operations [23]. Flattening is performed to convert the output into a one- dimensional array, followed by two dense layers (256 and 128 neurons) with ReLU activation and dropout regularization to prevent overfitting. The final layer uses a sigmoid activation function for binary classification. The second model shares a similar architecture but integrates class weights during training to address potential imbalances in the dataset. The models are characterized by a hierarchical arrangement of convolutional operations, enabling them to automatically learn relevant features from input images. This architecture demonstrates the power of deep learning in automatically extracting hierarchical representations, contributing to the models' effectiveness in discriminating between spam and non-spam images. Fig 2 shows the CNN architecture.

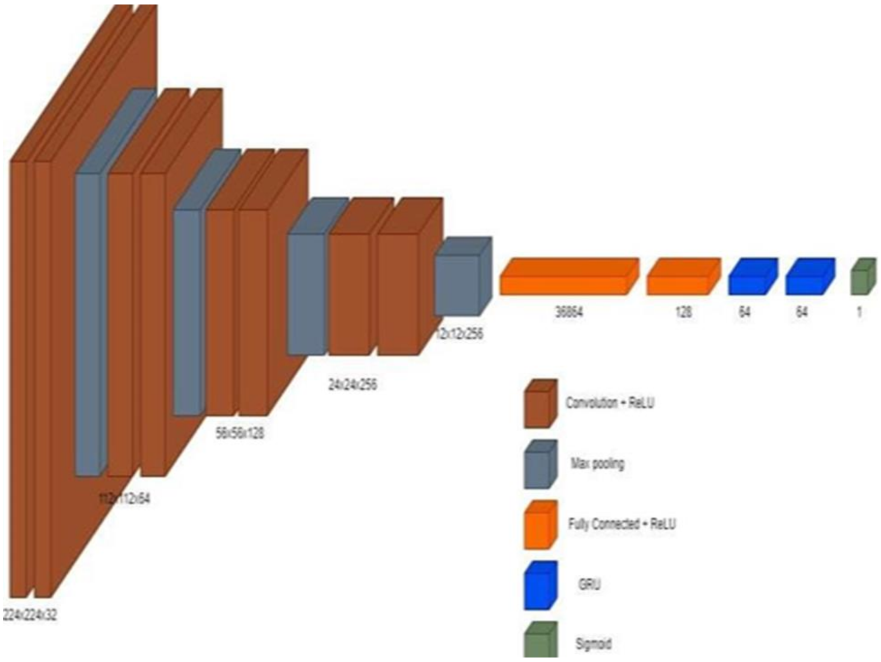


FIGURE 2 CUSTOM CNN WITH GRU ARCHITECTURE

4.3 *Model Training*

The models are trained on a labeled dataset using the Adam optimizer with a binary cross-entropy loss function. The training process spans 50 epochs, and during this period, callbacks such as ModelCheckpoint and ReduceLROnPlateau are implemented to save the best-performing model and dynamically adjust the learning rates based on validation performance.

4.4 Model Evaluation

After training the models, we perform a thorough evaluation on a test dataset. We calculate various performance metrics such as accuracy, precision, recall, and F1-score to assess the models' efficacy in distinguishing between spam and non-spam images. Furthermore, we generate confusion matrices and classification reports to offer a comprehensive breakdown of true positive, true negative, false positive, and false negative predictions, thereby providing detailed insights into the models' classification performance..

4.5Classifiers and Ensemble Models.

In the paper, we examined the outcomes of image classification employing various renowned classification techniques, including Logistic Regression (LR), Random Forest (RF), Decision Trees (DT), k-Nearest Neighbors (KNN), Gaussian Naive Bayes (GNB), AdaBoost (AB), Linear Support Vector Machine (LSVM), and Radial Support Vector Machine (RSVM) [24]. Each of these methods has its own strengths and weaknesses, with some emphasizing speed while others prioritize accuracy. This section outlines the diverse set of classifiers and ensemble models employed in the spam detection research. Each model is discussed with its underlying theory, application, and relevant formulas, followed by a comprehensive presentation of the obtained results. Fig 3-18 shows the confusion matrix of these classifiers and table 3 show the performance of these classifiers.

4.5.1 *Logistic Regression (LR)*

Logistic Regression, a linear model primarily utilized for binary classification tasks, estimates the probability that an instance belongs to a specific class. It achieves this by employing the logistic function, which maps the output to a value between 0 and 1, representing the probability of belonging to the positive class.

4.5.2 Random Forest(RF)

The Random Forest classifier was trained using features extracted from the CNN models, demonstrating accuracy rates of 86.44% and 86.50% in cost-sensitive and cost-insensitive scenarios, respectively. The model's ability to capture complex relationships in the feature space contributed to its competitive performance.

4.5.3Decision Tree (DT)

Employing Decision Tree classification, this model exhibited accuracies of 86.26% and 86.24% in cost- sensitive and cost-insensitive scenarios, respectively. Decision Trees provided insights into feature importance, contributing to the overall interpretability of the classification process.

4.5.4 K-Nearest Neighbors (KNN)

KNN classification, leveraging the extracted features, yielded accuracy rates of 86.46% and 86.40% in cost-sensitive and cost-insensitive contexts. The model's reliance on proximity in feature space enabled it to discern patterns and achieve competitive results.

4.5.5Gaussian Naive Bayes (GNB)

The Gaussian Naive Bayes classifier, applied to the extracted features, achieved accuracies of 86.26% and 86.46% in cost-sensitive and cost-insensitive scenarios, respectively. Its probabilistic approach proved effective in handling the inherent uncertainty in spam detection.

4.5.6AdaBoost (AB)

The AdaBoost classifier, incorporating features from both CNN models, demonstrated robust performance with accuracies of 86.46% and 86.50% in cost- sensitive and cost-insensitive scenarios. AdaBoost's ensemble learning strategy effectively combined weak learners to enhance overall classification accuracy.

4.5.7Linear Support Vector Machine (LSVM)

A Linear Support Vector Machine classifier, trained on features from CNNs, achieved accuracies of 86.80% and 86.44% in cost-sensitive and cost- insensitive scenarios. Its ability to create optimal hyperplanes contributed to its discriminative power in spam detection.

4.5.8Radial Support Vector Machine (RSVM) Model

The Radial Support Vector Machine, deployed with features from CNNs, showcased accuracies of 86.50% and 86.44% in cost-sensitive and cost-insensitive contexts. Its non-linear decision boundaries proved beneficial in capturing complex relationships within the feature space.

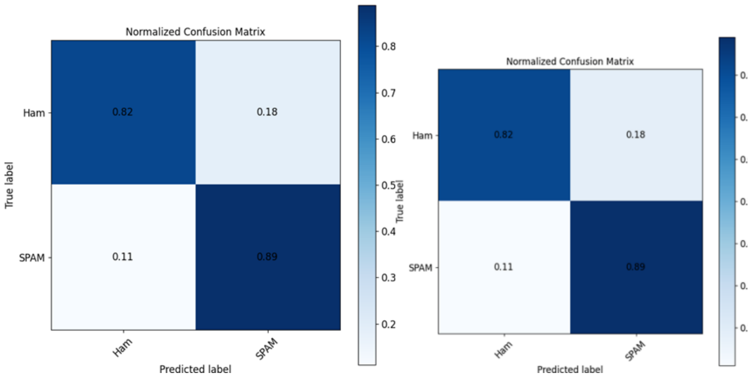


FIG 3 -4 CONFUSION MATRIX OF LOGISTIC REGRESSION (LR)

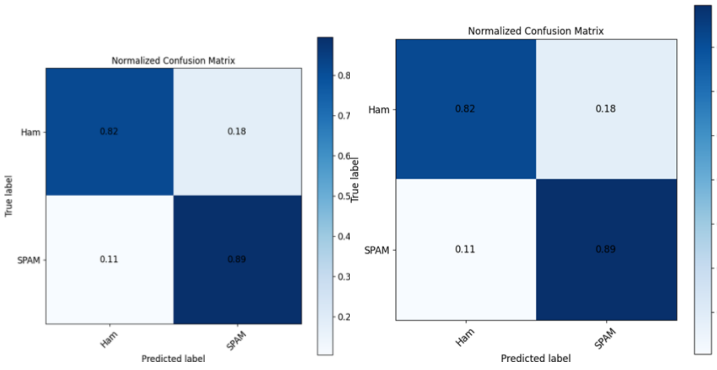


FIGURE 5-6 CONFUSION MATRIX OF RANDOM FOREST

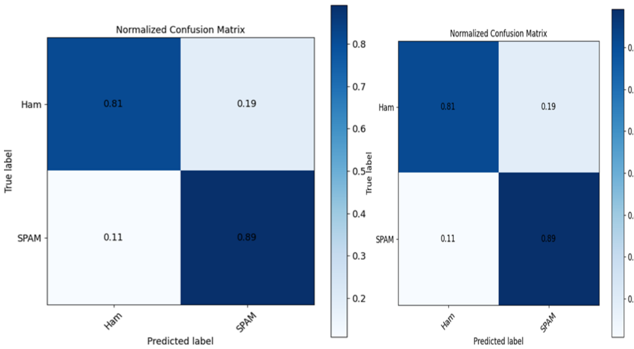


FIG 7-8 CONFUSION MATRIX OF DT

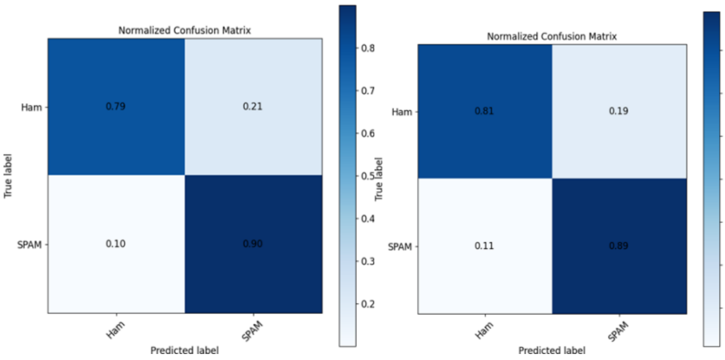


FIGURE 9-10 CONFUSION MATRIX OF KNN

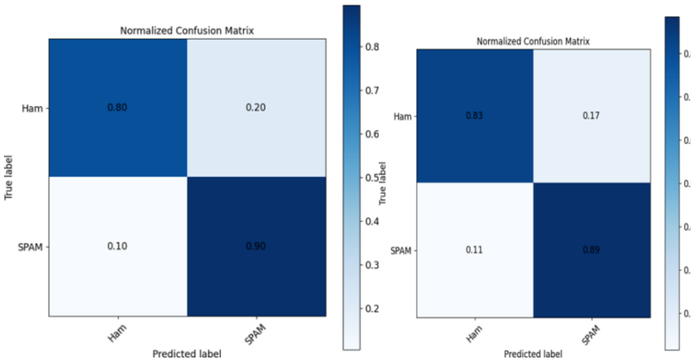


FIGURE 11-12 CONFUSION MATRIX OF GNB

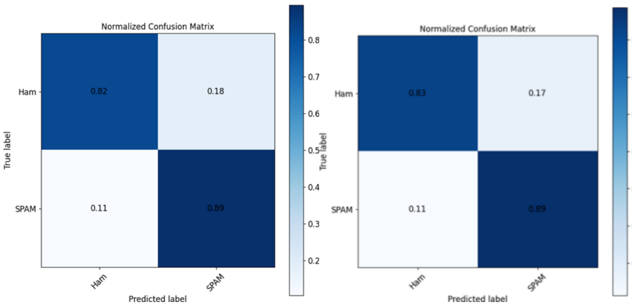


FIGURE 13-14 CONFUSION MATRIX OF AB

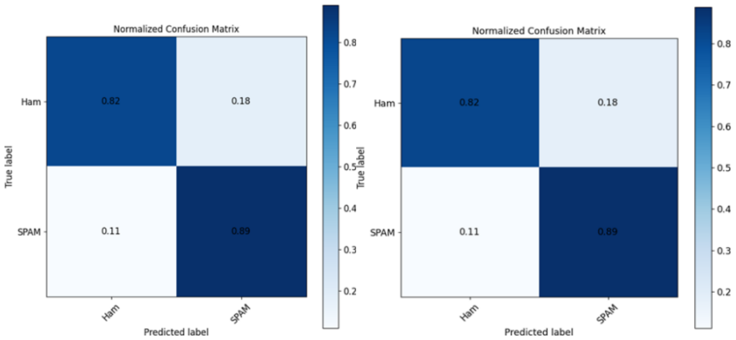


FIG 15-16 CONFUSION MATRIX OF LSVM

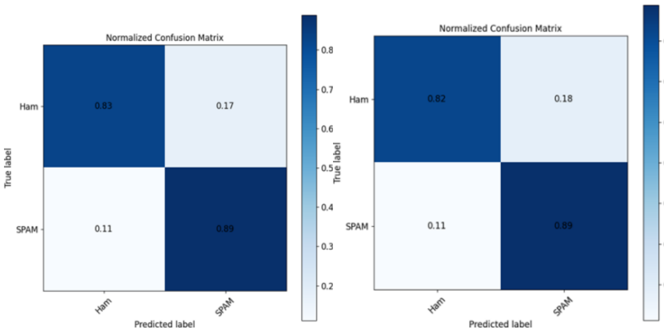


FIGURE 17-18 CONFUSION MATRIX OF RSVM

TABLE 3: PERFORMANCE OF CLASSIFIERS

Model	Scenario	Accuracy	Precision	Recall	F1-Score
CNN	Cost-Insensitive	0.9639	0.9459	0.9483	0.9502
CNN	Cost-Sensitive	0.9616	0.9482	0.9451	0.9646
LR	Cost-Insensitive	0.9639	0.9639	0.9639	0.9639
LR	Cost-Sensitive	0.9605	0.9604	0.9605	0.9604
RF	Cost-Insensitive	0.9442	0.9441	0.9442	0.9441
RF	Cost-Sensitive	0.9535	0.9534	0.9535	0.9535
DT	Cost-Insensitive	0.9442	0.9441	0.9442	0.9441

DT	Cost-Sensitive	0.9535	0.9534	0.9535	0.9535
KNN	Cost-Insensitive	0.9605	0.9605	0.9605	0.9603
KNN	Cost-Sensitive	0.9581	0.9581	0.9581	0.9581
GNB	Cost-Insensitive	0.9639	0.964	0.9639	0.964
GNB	Cost-Sensitive	0.9616	0.9616	0.9616	0.9616
AB	Cost-Insensitive	0.9628	0.9626	0.9628	0.9626
AB	Cost-Sensitive	0.9628	0.9628	0.9628	0.9628
LSVM	Cost-Insensitive	0.9651	0.9651	0.9651	0.9651
LSVM	Cost-Sensitive	0.9628	0.9628	0.9628	0.9628
RSVM	Cost-Insensitive	0.9639	0.9639	0.9639	0.9639
RSVM	Cost-Sensitive	0.9628	0.9628	0.9628	0.9628

4.6 Fine Tuning Pretrained Models

Transfer learning has proven to be a valuable method in machine learning, wherein a pretrained CNN model is repurposed to leverage its learned weights as initialization for a novel ‘CNN model’ tailored to a different task. There are two main approaches to employing ‘transfer learning’ [25]:Utilizing the pretrained model as a ‘feature extractor’ and incorporating a new classifier for the task at hand [26].Employing the pretrained model for fine-tuning (FT), which involves adjusting the parameters of both the new fully connected (FC) layers of the classifier and specific convolutional layers of the CNN through selective unfreezing.

4.6.1 *VGG16 Fine-Tuning for Spam Classification*

Network Configuration and Pre-training: Employing the VGG16 architecture, a convolutional neural network pre-trained on ImageNet, this methodology initiates with configuring the model to accept input images of size (224,224,3). The initial layers are retained with frozen weights to preserve generic image features, while subsequent layers undergo fine-tuning to adapt to the specific task. This approach capitalizes on the hierarchical feature learning capabilities encoded in ImageNet pre-trained weights.

Fine-Tuning Layers: The layers of VGG16 are partitioned into two sets, with the first 15 layers frozen and the rest set as trainable. This selective fine-tuning strategy enables the model to specialize in spam and ham image discrimination while retaining previously learned lower-level features.

Model Augmentation: The architecture extends beyond the pre-trained layers, integrating a Flattening layer, a Dense layer with rectified linear unit (ReLU) activation, a Dropout layer for regularization, and a final Dense layer with sigmoid activation for binary classification.

Compilation and Optimization: The model is compiled using binary cross-entropy loss and Stochastic Gradient Descent (SGD) optimizer with a learning rate of 1e-4 and momentum of 0.9. This configuration facilitates effective backpropagation for training the fine-tuned layers.

Evaluation Metrics: The performance of the fine-tuned VGG16 model is comprehensively assessed using various metrics such as accuracy, precision, recall, F1 score, confusion matrix, and a detailed classification report. The achieved results demonstrate the efficacy of the proposed methodology, with an accuracy of 96.98%, precision of 97.33%, recall of 94.19%, and an F1 score of 95.74% on a dataset comprising 860 images. Fig 19 show the VGG16 Fine-Tuning for Spam Classification.

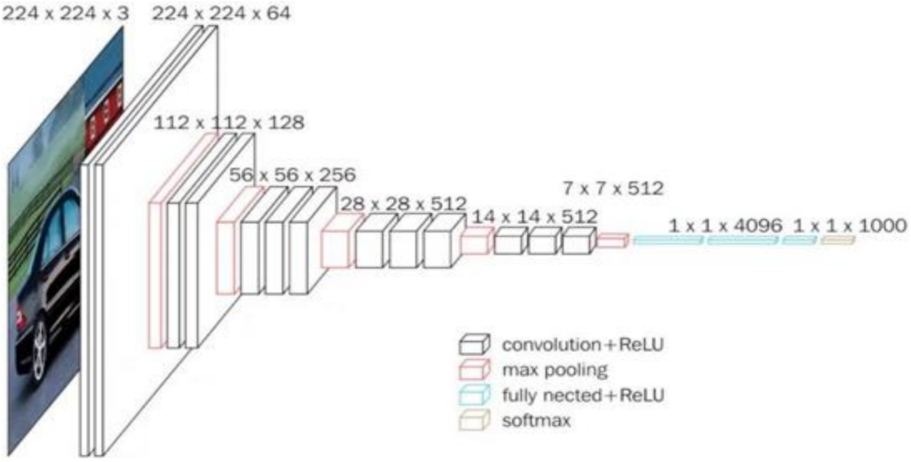


FIGURE 19: VGG16 FINE-TUNING FOR SPAM CLASSIFICATION

4.6.2 *VGG19 Fine-Tuning for Image Classification*

Network Configuration and Pre-training: Leveraging the VGG19 architecture pre-trained on the ImageNet dataset, this methodology focuses on image classification. The model is initialized to accept input images of size (224,224,3), and the initial layers are kept frozen to retain general image features learned from ImageNet. This approach harnesses the hierarchical feature representations captured in the pre-trained weights.

Fine-Tuning Strategy: The VGG19 model is modified by introducing additional layers for specialized classification. The initial layers of VGG19 are set as non-trainable, and a custom classifier is appended, comprising a Flattening layer, two Dense layers with rectified linear unit (ReLU) activation, Batch Normalization for regularization, Dropout layers for preventing overfitting, and a final Dense layer with softmax activation for binary classification.

Compilation and Optimization: The model is compiled using sparse categorical cross-entropy loss and Stochastic Gradient Descent (SGD) optimizer with a learning rate of 1e-4 and momentum of 0.9. This facilitates efficient optimization during the fine-tuning process.

Evaluation Metrics: A comprehensive evaluation is conducted using metrics such as accuracy, precision, recall, F1 score, confusion matrix, and a detailed classification report. The achieved results demonstrate the efficacy of the proposed methodology, with an accuracy of 97.44%, precision of 97.06%, recall of 95.81%, and an F1 score of 96.43% on a dataset comprising 860 images. The confusion matrix further highlights the model's ability to discriminate between the two classes, yielding promising outcomes for image classification tasks. Figure 20 show the VGG19 Fine-Tuning for Image Classification.

4.6.3 *MobileNetV2 Fine-Tuning with Enhanced Convolutional Layers*

Network Architecture and Pre-training: This methodology leverages the MobileNetV2 architecture pre-trained on ImageNet, emphasizing image classification tasks. The initial layers of the model are frozen to retain generic features learned from ImageNet. The base model is loaded with weights from 'imagenet' and configured to accept input images of size (224,224,3).

Selective Fine-Tuning: Fine-tuning is strategically applied to the MobileNetV2 model by freezing layers until the 'block_16_expand' layer, enabling the retention of learned features while allowing further adaptation to the target task[28].

Enhanced Convolutional Layers: Additional convolutional layers are introduced to the model, including a layer with 256 filters and a (3,3) kernel, followed by max-pooling. Optionally, if the output shape permits, another convolutional layer with 512 filters and max-pooling is appended. These enhancements aim to capture and amplify discriminative features relevant to the specific image classification requirements.

Global Average Pooling and Regularization: The model incorporates global average pooling for effective feature summarization. Dropout regularization with a rate of 0.5 is applied to mitigate overfitting during training.

Binary Classification Head: A dense layer with a sigmoid activation function serves as the final layer for binary classification. The model is compiled using binary cross-entropy loss and an Adam optimizer with a reduced learning rate of 0.00001, optimizing for efficient adaptation to the target task.

Comprehensive Evaluation: The methodology is rigorously evaluated using key metrics such as accuracy, precision, recall, F1 score, confusion matrix, and a detailed classification report. The achieved results demonstrate the effectiveness of the proposed approach, with an accuracy of 97.67%, precision of 97.70%, recall of 95.81%, and an F1 score of 96.74% on a dataset comprising 860 images. The robustness and discriminatory power of the model are evident in the detailed classification report and confusion matrix, affirming its suitability for image classification tasks. Figure 21 show the MobileNetV2 Fine-Tuning with Enhanced Convolutional Layers.

4.6.4 Xception Fine-Tuning with Global Average Pooling

Architectural Foundation and Pre-training: Employing the Xception model, initially pre-trained on the ImageNet dataset, this methodology focuses on feature extraction and classification for image-based tasks. The Xception architecture, known for its depth and efficiency, is loaded with 'imagenet' weights, and the first layers are frozen to retain foundational features.

Strategic Fine-Tuning: The fine-tuning strategy involves selectively freezing the majority of the initial layers, leaving the last few layers trainable for adaptation to the specific classification task. This selective fine-tuning ensures the preservation of high- level features while allowing model specialization.

Model Construction for Classification: A new sequential model is constructed by adding the pre- trained Xception base model, followed by global average pooling for effective feature summarization. Additional layers, including a dense layer with 512 units and ReLU activation, a dropout layer with a rate of 0.5 for regularization, and a final dense layer with a sigmoid activation for binary classification, are appended.

Optimization and Training Configuration: The model is compiled using stochastic gradient descent (SGD) as the optimizer with a learning rate of 1e-4 and momentum of 0.9. The binary cross-entropy loss function is employed, and accuracy is chosen as the evaluation metric. This configuration aims to strike a balance between efficient convergence and fine- grained learning[30].

Comprehensive Evaluation: Evaluation metrics such as accuracy, precision, recall, F1 score, confusion matrix, and a detailed classification report are employed to assess the model's performance. The results demonstrate the efficacy of the approach, achieving an accuracy of 98.37%, precision of 99.33%, recall of 96.13%, and an F1 score of 97.70% on a dataset comprising 860 images. The model's proficiency in distinguishing between classes is further elucidated through the detailed classification report and confusion matrix, reinforcing its suitability for image classification tasks. Figure 22 show the Xception Fine-Tuning with Global Average Pooling.

4.6.5 ResNet50 with Fine-Tuning and Custom Classification Layers

Architectural Foundation and Pre-training: Leveraging the powerful ResNet50 model pre-trained on ImageNet, this methodology employs a hierarchical feature extraction approach. The ResNet50 architecture, known for its residual connections, is initially loaded with 'imagenet' weights, allowing the model to capture intricate hierarchical features[27].Strategic Fine-Tuning: To adapt the pre-trained ResNet50 to the specific classification task, a strategic fine-tuning approach is employed. The first layers, except for the last five, are frozen to preserve low-level features, while the remaining layers are unfrozen for specialized learning. This dual-phase fine-tuning strikes a balance between feature retention and task- specific adaptation.Custom Classification Layers: Custom layers are added atop the pre-trained ResNet50 architecture to tailor it for binary classification. These include a flattening layer, a densely connected layer with 512 units and ReLU activation for feature transformation, a dropout layer with a rate of 0.5 for regularization, and a final dense layer with a sigmoid activation for binary classification[29].

Optimization and Training Configuration: The model is compiled using the Adam optimizer with a reduced learning rate of 1e-5 to facilitate nuanced learning during fine-tuning. The binary cross-entropy loss function is utilized, and accuracy is chosen as the evaluation metric.

Comprehensive Evaluation: The model's performance is thoroughly assessed using key metrics such as accuracy, precision, recall, F1 score, confusion matrix, and a detailed classification report. Achieving an accuracy of 96.05%, precision of 94.52%, recall of 94.52%, and an F1 score of 94.52% on an 860-image dataset, this methodology demonstrates its effectiveness in binary image classification tasks. The detailed evaluation metrics provide insights into the model's ability to discern between classes, reinforcing its suitability for diverse image classification applications. Figure 23 show the ResNet50 with Fine- Tuning and Custom Classification Layers.

4.6.6 InceptionV3 with Feature Extraction and Custom Classification Layers

Architectural Foundation and Pre-training: Employing the InceptionV3 architecture, this methodology taps into the richness of hierarchical feature extraction. The InceptionV3 model, pre-trained on ImageNet, serves as a potent feature extractor, capturing intricate patterns in the input images.Comprehensive Evaluation: A thorough evaluation of the model is conducted, revealing its prowess with an accuracy of 97.67%, precision of 98.33%, recall of 95.16%, and an F1 score of 96.72% on an 860-image dataset. The confusion matrix and detailed classification report provide insights into the model's ability to discriminate between classes, showcasing its efficacy in binary image classification tasks. The high precision and recall values underscore the model's suitability for applications demanding discerning classification capabilities. Figure 24 show the InceptionV3 with Feature Extraction and Custom Classification Layers.

5. EXPERIMENT AND RESULTS

In this part, we elaborate on the shown experiments, detailing the implementation specifics across different datasets. This encompasses explaining the experimental framework utilized, along with presenting the validation and test results obtained. Image preprocessing techniques were executed in ‘Python 3.6’ using ‘OpenCV’ as the primary image processing library. The experimentation was carried out on an Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz with 32 GB of RAM. Keras facilitated the implementation of the transfer learning model. ‘Our proposed CNN model’ achieved an accuracy close to 98% on the enhanced dataset. The performance of PRE-TRAINED CNN ARCHITECTURES is shown in tables 5-9. The accuracy attained by our proposed model significantly exceeds that achieved by the respective authors using the ‘LSVM classifier’, as depicted in Table 3.

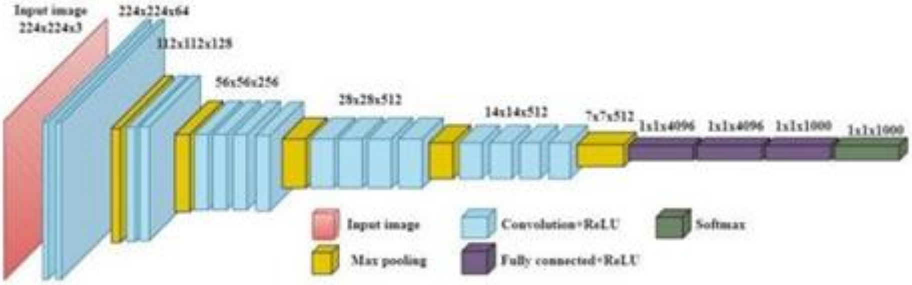


FIGURE 20-21 VGG16 FINE-TUNING FOR SPAM CLASSIFICATION

TABLE 4. NETWORK HYPER-PARAMETERS

‘Batch-Size’	‘Learning Rate’	‘No.Epoc ’	‘Optimizer’	‘Loss-Function’
32	1* ×10-3	50	Adam, SGD	binary_crossentropy

TABLE 5 : PERFORMANCE OF PRE-TRAINED “CNN ARCHITECTURES” WITH ISH DATASET

“Model”	‘Accuracy %’	‘Precision %’	‘Recall %’	F1-Score’ %’
VGG16	96.98	97.33	94.19	95.74
VGG19	97.44	97.06	95.81	96.43
MobileNet V2	96.98	97.33	94.19	95.74
Resnet50	98.37	99.33	96.13	97.70
Xception	96.05	94.52	94.52	94.52

TABLE 6 : PERFORMANCE OF PRE-TRAINED “CNN ARCHITECTURES WITH IMPROVED DATASET.

“Model”	‘Accuracy %’	‘Precision %’	‘Recall %’	F1-Score’ %’
VGG16	94.98	96.33	95.19	95.74
VGG19	93.44	97.06	94.81	94.43
MobileNet V2	94.98	97.33	94.19	96.74
Resnet50	97.37	98.33	97.17	97.77
Xception	96.05	94.52	97.10	93.52
InceptionV 3	93.67	96.33	93.16	97.72

TABLE 7 : PERFORMANCE OF PRE-TRAINED “CNN ARCHITECTURES WITH CHALLENGE-A DATASET

“Model”	‘Accuracy %’	‘Precision %’	‘Recall %’	F1-Score’ %’
VGG16	97.98	97.10	93.19	95.74
VGG19	97.44	93.16	95.81	96.43
MobileNet V2	94.98	97.33	94.19	95.74
Resnet50	97.38	99.33	97.17	97.70
Xception	96.05	94.52	97.10	94.52
InceptionV 3	97.20	98.33	93.16	96.72

TABLE 8 : PERFORMANCE OF PRE-TRAINED “CNN ARCHITECTURES WITH DREDZE DATASET.

“Model”	‘Accuracy %’	‘Precision %’	‘Recall %’	F1-Score’ %’
VGG16	97.98	97.33	94.19	94.74
VGG19	92.44	97.06	95.81	96.43
MobileNet V2	96.98	97.33	94.19	96.74
Resnet50	94.37	99.33	96.13	97.70
Xception	96.05	94.52	94.52	93.52
InceptionV 3	97.67	98.33	95.16	94.72

TABLE 9 : PERFORMANCE OF PRE-TRAINED “CNN ARCHITECTURES WITH CHALLENGE-B DATASET

“Model”	‘Accuracy %’	‘Precision %’	‘Recall %’	F1-Score’ %’
VGG16	96.98	97.33	94.19	95.74
VGG19	94.44	93.06	95.81	96.43
MobileNet V2	97.98	94.33	94.19	98.69
Resnet50	98.37	94.78	95.13	98.70
Xception	97.05	94.52	94.52	94.52
InceptionV 3	90.67	98.33	95.17	96.72

CONCLUSIONS

In this paper, we present a novel framework leveraging multiple “deep learning models”called as combination of convolutional neural networks,gated recurrent units(GRU) (including InceptionV3, DenseNet121, ResNet50, VGG16, VGG19, and MobileNetV2) for the categorization of spam/ham images. Our objective is to enhance accuracy while minimizing computational time. We explore various classification methods, including LR, RF, DT, KNN, GNB, AB, LSVM, and RSVM. Our investigation demonstrates that the performance of the classifiers is enhanced with the use of data augmentation, as evidenced by the obtained results.Among the models tested, the obtained results reveal that the ResNet50 model yields the best performance, achieving an accuracy of 98.99%, precision of 94.78%, recall of 95%, and 98% F1Score in Challenge B Dataset as show in figure 22-26. Our experiments confirm the superiority of our proposed method over those of other researchers as shown in table 10 Additionally, the paper addresses various challenges encountered in the image classification task.In future research endeavors, our intention is to extend the application of our algorithm to additional image datasets, facilitating a comprehensive statistical analysis of its performance.

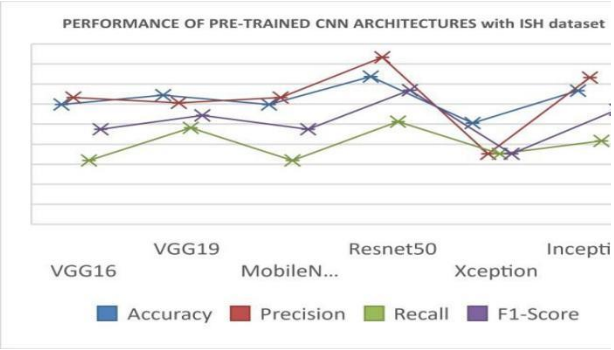


FIGURE 22 PERFORMANCE OF PRE-TRAINED TRAINED

WITH IMPROVED DATASET

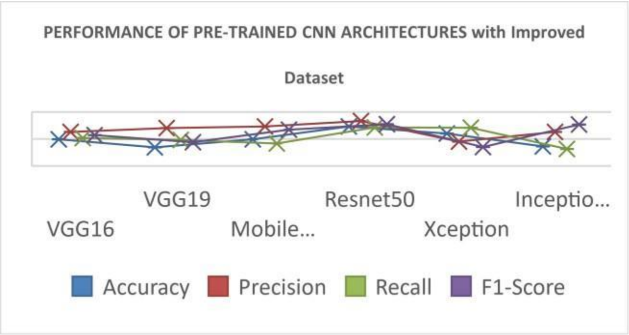
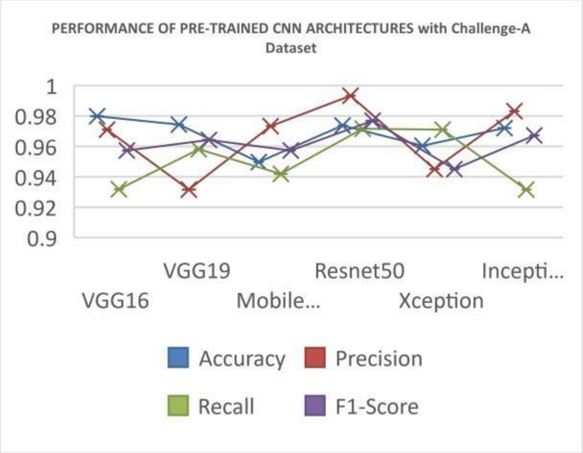


FIGURE 23 PERFORMANCE OF PRE-

WITH ISH DATASET

.



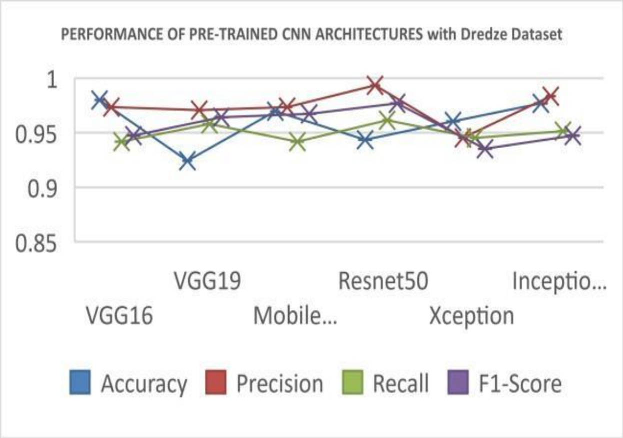


FIGURE 24 PERFORMANCE OF PRE- TRAINED WITH CHALLENGE-A DATASET.

FIGURE 25 PERFORMANCE OF PRE-TRAINED WITH DREDZE DATASET

.

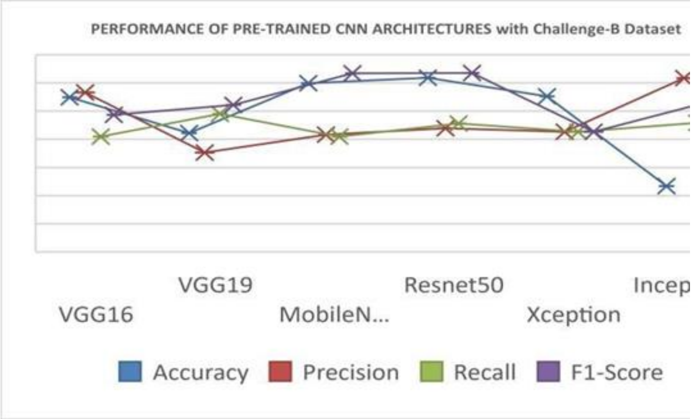


FIGURE 26 PERFORMANCE OF PRE-TRAINED WITH CHALLENGE-B DATASET

TABLE 10 EVALUATION OF THE EFFECTIVENESS OF STATE-OF-THE-ART METHODS.

	Improv ed Datase t	Challen ge-A Dataset	Challen ge-B Dataset	Dred ze Datas et	ISH Datas et
Proposed Model	**0.9737**	**0.9738**	**0.9837**	**0.9437**	0.9837
A.Annadatha et al[8]	**70%**				97.00 %
Aneri. et al[14]		**69.32%**	**69.32%**	**98.%**	
Sriram. S et al [15]	**97.0%**			**97.30 %**	99.80 %
Aaisha et al[16]					96.00 %
Dredze et al., 2007[17]				**97%**	
Mehta et al.,					80%

2008[18]				
Soranamage swari et al., 2010 [19]				93%
Kumaresan et al. 2015, [20]				90%
Shang et al.				75.1
2016, [21]				%

Reference

[1] Mahmood A, Bennamoun M, An S, Sohel F (2017) Resfeats: residual network based features for image classification. In: 2017 IEEE International conference on image processing (ICIP). https:// doi.org/ 10. 1109/ icip. 2017. 82965 51.

[2] Kataoka H, Iwata K, Satoh Y (2015) Feature evaluation of deep convolutional neural networks for object recognition and detection.https:// arxiv. org/ abs/ 1509. 07627.

[3] Srivastava S, Mukherjee P, Lall B, Jaiswal K (2017) Object classification using ensemble of local and deep features. In: 2017 ninth international conference on advances in pattern recognition(ICAPR), pp 1–6. IEEE.

[4] Shaha M, Pawar M (2018) Transfer learning for image classification.In: 2018 Second International conference on electronics, communication and aerospace technology (ICECA). https:// doi. org/10. 1109/ iceca. 2018. 84748 02.

[5] Kumar V, Recupero DR, Riboni D, Helaoui R (2021) Ensembling classical machine learning and deep learning approaches for morbidity identification from clinical notes. IEEE Access 9:7107–7126 https:// doi. org/ 10. 1109/ ACCESS. 2020. 30432 21.

[6] Seemendra A, Singh R, Singh S (2021) Breast cancer classification using transfer learning. In: Evolving Technologies for computing,communication and smart world, pp 425–436. Springer

[7] Kumaresan, T., Sanjushree, S., & Palanisamy, C. (2014). Image spam detection using color features and k-nearest neighbor classification.International Journal of Computer, Electrical, Automation, Control and Information Engineering, 8 (10), 1904–1907

[8] Annadatha, A., & Stamp, M. (2018). Image spam analysis and detection.Journal of Computer Virology and Hacking Techniques, 14 (1), 39–52.

[9] Chavda, A., Potika, K., Di Troia, F., & Stamp, M. (2018). Support vector machines for image spam analysis. In Proceedings of the 15th international joint conference on e-business and telecommunications(pp. 597–607).

[10] . H. Yang, Q. Liu, S. Zhou, and Y. Luo, “A spam filtering method based on multi-modal fusion,” Applied Sciences,vol. 9, no. 6, p. 1152, 2019

[11] Dredze. M, Gevaryahu R, Elias-Bachrach A. Learning Fast Classifiers for Image Spam. In CEAS 2007Aug 2 (pp.

[12] 2007487).https://www.cs.jhu.edu/~mdredze/datasets/im age_spam/.

[13] Gao. Y, Yang M, Zhao. X, Pardo B, Wu Y, Pappas TN, Choudhary A. Image spam hunter. In 2008 IEEE international conference on acoustics, speech & signal processing, 2008 March 31 (pp. 1765–1768).IEEE. https://users.cs.northwestern.edu/~yga751/ML/ISH.htm #dataset.

[14] Annadatha A, Stamp M. Image spam analysis and detection. Journal of Computer Virology and Hacking Techniques. 2018 Feb; 14(1):39–52.

[15] Aneri Chavda, Katerina Potika, Fabio Di Troia, and Mark Stamp. Support Vector Machines for Image Spam Analysis. Proceedings of the 15th International Joint Conference on e-Business and Telecommunications— Volume 1:BASS,2018,431–441https://doi.org/10.5220/0006921404310441.

[16] S. Sriram, R. Vinayakumar, V. Sowmya, Moez Krichen, Dhouha Ben Noureddine, A. Shashank, K.P.Soman. Deep Convolutional Neural Networks for Image Spam Classification. 2020. hal-02510594.

[17] Makkar Aaisha, Kumar Neeraj, PROTECTOR: An optimized deep learning-based framework for image spam detection and prevention, Future Generation Computer Systems, Volume 125, 2021, Pages 41–58, ISSN 0167-739X,

[18] https://doi.org/10.1016/j.future.2021.06.026

[19] M. Dredze, R. Gevaryahu, A. Elias-Bachrach, Learning fast classifiers for image spam., in: CEAS, 2007, pp. 2007–2487

[20] B. Mehta, S. Nangia, M. Gupta, W. Nejdl, detecting image spam using visual features and near duplicate detection, in: Proceedings of the 17th International Conference on World Wide Web, ACM, 2008, pp. 497– 506

[21] M. Soranamageswari, C. Meena, Statistical feature extraction for classification of image spam using artificial neural networks, in: 2010 SecondInternational Conference on Machine Learning and Computing, IEEE, 2010,pp. 101–105.

[22] T. Kumaresan, S. Sanjushree, K. Suhasini, C. Palanisamy, Image spam filtering using support vector machine and particle swarm optimization, Int. J. Comput. Appl 1 (2015) 17–21.

[23] E.-X. Shang, H.-G. Zhang, Image spam classification based on convolutional neural network, in: 2016

[24] International Conference on Machine Learning and Cybernetics (ICMLC), 1, IEEE, 2016, pp. 398–403.

[25] Singh AB, Singh KM (2023) Application of error level analysis in image spam classification using deep learning model. PLoS ONE 18(12): e0291037.https://doi.org/10.1371/journal.pone.0291037

[26] A. P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognition,30(7):1145–1159, 1997.

[27] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, Support vector machines, IEEE Intelligent Systems and theirApplications, 13(4):18–28, 1998. West, Jeremy; Ventura, Dan; Warnick, Sean. "Spring Research Presentation: A Theoretical Foundation for Inductive Transfer". Brigham Young University, College of Physical and Mathematical Sciences. Archived from the original on 2007-08-01. Retrieved 2007-08-05. 2007

[28] Maitra D. S.; Bhattacharya U., and Parui S. K. (August 2015). "CNN based common approach to handwritten character recognition of multiple scripts". 2015 13th International Conference on Document Analysis and Recognition (ICDAR): 1021–1025, ISBN 978-1-4799- 1805-8. S2CID 25739012.

[29] Sajja Tulasi Krishna, Hemantha Kumar Kalluri, “Deep learning and transfer learning approaches for image classification”, International Journal of Recent Technology and Engineering (IJRTE), vol.-7, Issue- 5S4,pp. 427-432, 2019.

[30] M. A. Tahoun, K. A. Nagaty, T. I. El-Arief and M. AMegeed",A robust content-based image retrieval system using multiple features representations", Proceedings.2005 IEEE Networking, Sensing and Control, 2005.Tucson, AZ, 2005, pp. 116-122.

[31] Y. Wei et al., "HCP: A Flexible CNN Framework forMulti-Label Image Classification," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no.9, pp. 1901-1907, 1 Sept. 2016

[32] A. P. Singh, “Image spam classification using deep learning,” Master’s Projects. 641. SJSU scholarworks, 2018.

[33] C. Fatichah, W. F. Lazuardi, D. A. Navastara, N. Suciati, and A. Munif,“Image spam detection on instagram using convolutional neural network,”in Intelligent and Interactive Computing. Springer, 2019, pp.295–303