big data and cognitive computing

MDPI

Article

Attention Mechanism and Support Vector Machine for Image-Based E-Mail Spam Filtering

Ghizlane Hnini *, Jamal Riffi, Mohamed Adnane Mahraz, Ali Yahyaouy and Hamid Tairi

Laboratory of Computer Science, Signals, Automation and Cognitivism (LISAC), University Sidi Mohamed Ben Abdellah, Fez 30000, Morocco
* Correspondence: ghizlane.hnini@usmba.ac.ma

Abstract: Spammers have created a new kind of electronic mail (e-mail) called image-based spam to bypass text-based spam filters. Unfortunately, these images contain harmful links that can infect the user's computer system and take a long time to be deleted, which can hamper users' productivity and security. In this paper, a hybrid deep neural network architecture is suggested to address this problem. It is based on the convolution neural network (CNN), which has been enhanced with the convolutional block attention module (CBAM). Initially, CNN enhanced with CBAM is used to extract the most crucial information from each image-based e-mail. Then, the generated feature vectors are fed to the support vector machine (SVM) model to classify them as either spam or ham. Four datasets—including Image Spam Hunter (ISH), Annadatha, Chavda Approach 1, and Chavda Approach 2—are used in the experiments. The obtained results demonstrated that in terms of accuracy, our model exceeds the existing state-of-the-art methods.

Keywords: attention mechanism; image spam; MobileNetV2; convolutional block attention module; support vector machine; deep learning

Citation: Hnini, G.; Riffi, J.; Mahraz, M.A.; Yahyaouy, A.; Tairi, H. Attention Mechanism and Support Vector Machine for Image-Based E-Mail Spam Filtering. *Big Data Cogn. Comput.* **2023**, *7*, 87. https://doi.org/10.3390/bdcc7020087

Academic Editors: Miguel Correia and Min Chen

Received: 24 February 2023
Revised: 17 April 2023
Accepted: 25 April 2023
Published: 6 May 2023

1. Introduction

E-mail is a widely used communication tool that offers many benefits for businesses, such as convenience, efficiency, and the ability to collaborate remotely. However, spammers have turned e-mail into a deceptive method of communication, using unsolicited e-mails, commonly known as spam, to send unwanted messages. Spam content can be text-based, image-based, or hybrid-based, with image-based spam being a new type of spam designed to evade text-based spam filters. The new type of spam produced by spammers to get past text-based spam filtering systems is known as image-based spam, which is defined as text embedded within an image. Compared to text-based spam, it is distinguished by its sophisticated structure. So, it is necessary to create a reliable and accurate image-based spam detection system. The main objective of an image-based e-mail spam detection system is to identify and filter undesirable e-mails that contain images. This system analyses the images that are attached to e-mails to assess whether they are spam or ham using a combination of visual analysis, pattern recognition, and content recognition techniques. The presence of image-based spam detection systems is required for several reasons, namely, to protect online security, improve productivity, and protect privacy. Unfortunately, spam can contain viruses, malware, and other threats that can compromise computer security. They can also waste users' time and attention, leading to decreased productivity, and contain personal or sensitive information that can be used for malicious purposes. Pre-processing, feature extraction, and classification are the three main components of the image-based spam detection architecture. Among them, feature extraction is considered the most important component. Hand-created features and deep-learned features are the two feature extraction techniques for spam images. According to the latest work, the use of deep convolutional neural networks (DCNN) to create features from images yields outstanding and significant results compared to other methods.

The authors of [1] introduced the DeepCapture architecture for image-based spam detection using the CNN model and XGBoost classifier. Their architecture [1] performed well on datasets that were made publicly available, outperforming state-of-the-art techniques by 6%. In addition, ref. [2] suggested a multi-modal system that is based on CNN and long short-term memory (LSTM) models; the former is used for the e-mail's image-based content and the latter is used for its text-based content. The two models individually analyse the text and image to produce two classification probabilities, which are then combined to determine whether the e-mail is spam or ham. In addition, ref. [3] suggested a multi-modal system for hybrid-based spam e-mail detection that is based on the Paragraph Vector Distributed Bag of Words (PV-DBOW), and random forest (RF) models. In addition, ref. [4] compared four DCNN pre-trained models, including Densely Connected Convolutional Networks 121 (DenseNet201), Xception, Residual Networks (ResNet152V2), and Visual Geometry Group (VGG19). Their experiment results demonstrate that the VGG19 model produces the best results. In addition, ref. [5] employed various pre-trained deep learning models, namely, InceptionV3, DenseNet121, ResNet50, VGG16, and MobileNetV2, to detect undesirable spam images. To evaluate the effectiveness of the models, the paper employs several widely datasets such as Dredze Dataset and ISH Dataset. Although these methods have achieved good accuracy, they have several limitations. First, the features extracted from the CNN model are not relevant, resulting in low accuracy. Second, the use of a large number of parameters in these DCNN models increases the time required for inference. For example, AlexNet [6] uses 61M, and VGG16 [7] employs 138M parameters, while DenseNet201 [8] uses 20M parameters, as do ResNet50 [9] and Xception [10]. Recently, numerous researchers have demonstrated that the CNN model performs better when the attention mechanism is used [11,12]. The attention mechanism, which draws its inspiration from the human visual system, seeks to concentrate on the crucial details while omitting the unimportant ones. Convolutional block attention module (CBAM), a new efficient attention module, is utilised to feedforward the CNN model and choose the most pertinent features [13]. In this paper, we offer a novel robust and effective architecture for image-based e-mail spam detection, based on CNN enhanced with the CBAM model. It addresses the shortcomings and limitations concerning the feature representation. The architecture is specifically divided into two main components, feature extraction and classification. Then, each image's features are extracted using the CNN model, which further refines features using CBAM blocks. Second, the support vector machine (SVM) model is fed with these vectors to determine if they are spam or ham. Using four public datasets, the suggested technique significantly outperforms the current models.

The following are the primary contributions of the current paper:

(1) We developed a hybrid architecture based on attention mechanism and SVM for extracting features from image-based spam, which is based on CNN enriched with CBAM.
(2) The vector generated from an image using the CNN enriched with CBAM is fed to the SVM model to classify it as either spam or ham.
(3) We also provide a hybrid architecture of MobileNetV2 with SVM in order to compare it with the proposed approach on four datasets.
(4) The experiments are conducted on four datasets and the results show that our method outperforms the state-of-the-art models.

2. Related Work

The classification of an image as spam or ham depends significantly on selecting the appropriate discriminant features, which is a difficult task. The extraction of discriminant features from an image using traditional machine learning has been proposed in a variety of studies. For instance, an image texture analysis-based image spam filtering (ITA-ISE) method was proposed by [14]. With the use of this method, the low-level image properties of an image, such as its histogram, co-occurrence matrix (COM), gradient, run-length matrix (RLM), discrete wavelet transform (DWT), and autoregressive (AR) model, are

extracted. The chosen image features are then fed into different machine learning classifiers in order to categorise the image as spam or ham: SVM, DT (Decision Tree), RF, and naive Bayes (NB). Using the Dredze dataset [15] and the ISH dataset [16], the performance of the ITA-ISF technique, which reached an accuracy of 98.6%, was examined. The gray level co-occurrence matrix (GLCM) was used by [17] to extract 22 features from images, and these features were then classified using the k-nearest neighbor (k-NN) and NB methods. Their suggested approach has shown appreciable accuracy in both datasets. Moreover, the weighted k-NN was employed by [18] to identify the color, texture, and high-level characteristics that were extracted from the images. The approach achieved a 99.36% accuracy on the ISH dataset; however, on their suggested dataset, it only had an 88.6% accuracy [19]. In [20], the authors also utilised low-level image features. They extracted metadata and texture properties from images and used them as input to the SVM classifier. The authors employed the particle swarm optimization (PSO) computational approach to optimize the detection, and they achieved 90% accuracy using the SVM on a publicly available dataset containing 1786 ham images and 3203 spam images. The work in [21] compared file characteristics, RGB histogram, HSV histogram, and a combination of RGB and HSV, which reflect image features. The k-NN classifier was used to categorise an image as spam or ham based on these features. The method suggested by [21] performed well using the combination of RGB and HSV with the k-NN methodology, using the same dataset as in [20]. Ref. [22] proposed two methods for detecting spam images. The first technique is called eigenspam, which is based on principal component analysis (PCA), while the second method consists of extracting 21 image features that are used as input to the SVM classifier. The PCA and SVM methods performed well using the ISH dataset, but they are not accurate on the improved dataset created by [22]. Indeed, they achieved an accuracy of only 70% using the SVM, while the area under the receiver operating characteristic (ROC) curve (AUC) was 0.38 using the eigenspam. The SVM classifier is also used by [19,23]. Ref. [23] used the SVM with Gaussian kernel-based to classify the 12 image features as either spam or ham, whereas, ref. [19] extracted a total of 38 features from images including metadata, color, texture, shape, and noise to classify them using the SVM. The method proposed by [19] achieved an accuracy of 97% and 98% on two publicly available datasets, namely, ISH and Dredze, respectively, but it does not exceed 70% on their proposed challenging dataset [19]. Although the traditional machine learning (ML) methods and hand-crafted features have attained good results, they failed in detecting noisy spam images that are closer to ham ones.

In order to avert these problems, deep learning techniques are used. The CNN is the deep learning technique most used in recent works for image spam detection. For example, a CNN based on deep learning techniques is proposed by [24]. Their proposed model is fine-tuned and optimized for both feature extraction as well as for classification tasks, and they achieved significant results on Chadva and ISH datasets. In addition, CNN enhanced with attention mechanism is used by [25] for image-based spam detection. In addition, the architecture proposed by [26] for image-based spam filtering is based on a CNN method and linear SVM classifier. First, the CNN model, which contains five convolutional layers, is used to generate the feature vector from an image, which is given by the last fully connected layer. Then, instead of using the softmax layer of the CNN model, this image representation is used as the input to a linear SVM. In addition, ref. [27] used the CNN model, which contains four convolutional layers, for image spam detection and their experiments were conducted on a publicly available dataset containing two sets of images, namely, 810 ham and 928 spam. The proposed model achieved a significant accuracy of 91.7%. In the work of [28], the Canny edge detector is used to extract information related to edges in an image, and then it is combined with the raw of the same image to feed it into the CNN model, which contains three convolutional layers. The proposed model is compared to the SVM and multi-layer perceptron (MLP) in refs. [19,22] using the same challenge dataset created by [19,22], and the publicly available ISH, used in the work of [27]. The obtained results demonstrated that the CNN model is efficient and more accurate compared to the

state-of-the-art ML techniques. It achieved an accuracy of 99.02%, 83.13%, and 71.83%. Furthermore, ref. [29] developed the 123DNet architecture for image spam detection using two embedded convolutional layers and three neural network layers. The accuracy attained was, respectively, 95%, 90%, and 88% on three datasets: the Dredze dataset, ISH dataset, and a personally generated dataset.

3. Materials and Methods

3.1. MobileNetV2

The MobileNetV2 (MNV2), which is a DCNN, was developed by [30]. The MNV2 is based on an inverted residual and linear bottleneck and is considered the most efficient and lightweight CNN. In addition, it is pre-trained on a large dataset which is called ImageNet. The use of the pre-trained MNV2 as feature extraction or classification model has several advantages, including the preservation of previously learned features and biased weights, as well as the avoidance of unnecessary computational costs. The MNV2 contains two blocks that are repeated multiple times: the first block is a residual, in which stride is equal to 1, which has a 1×1 convolution layer with ReLu6 function, a 3×3 depthwise convolution layer, and a 1×1 convolution layer without any non-linear function. The second block has the same layers as the first one with a stride that is equal to 2.

3.2. Convolutional Block Attention Module

Recently, the attention mechanism has been incorporated in various works [13,31–33] to enhance the performance of CNN. In this paper, we used the CBAM that is proposed by [13]. As shown in Figure 1, the CBAM architecture is based on two sequential attention modules, namely, channel and spatial, which indicate 'what' and 'where' to focus on, for improving the representation of interests, selecting the most meaningful features, and removing those that are unnecessary.

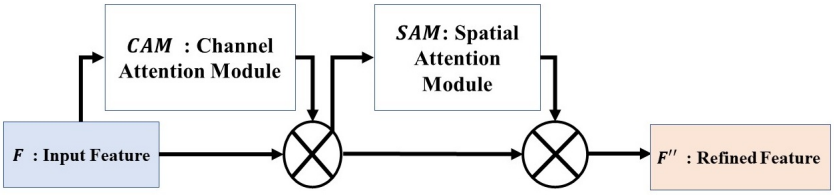


Figure 1. The CBAM [13] architecture.

The channel attention module (CAM) specifies 'what' to focus on, while the spatial attention module (SAM) indicates 'where' to emphasise. The CAM and SAM modules are sequentially used to obtain the final refined feature map F'' as follows:

$$F' = M_c(F) \otimes F \tag{1}$$

$$F'' = M_s(F') \otimes F' \tag{2}$$

where $\otimes$ refers to the element-wise multiplication.

Firstly, the max-pooled features F^c_{max} and average-pooled features F^c_{avg} are generated from an input feature map $F \in \mathbf{R}^{C \times H \times W}$. These descriptors are fed to MLP with one hidden layer to obtain the channel attention $M_c(F) \in \mathbf{R}^{C \times 1 \times 1}$ as follows:

$$M_c(F) = \sigma\left(W_1\left(W_0\left(F^c_{avg}\right)\right) + W_1(W_0(F^c_{max}))\right) \tag{3}$$

where W_0 and W_1 denote the weights of the shared MLP which use the ReLu activation function, while the σ represents the sigmoid function.

Secondly, the average-pooled $F^s_{avg} \in \mathbf{R}^{1\times H\times W}$ and max-pooled $F^s_{max} \in \mathbf{R}^{1\times H\times W}$ are generated from the channel information and then concatenated to obtain the spatial attention map $M_s \in \mathbf{R}^{1\times H\times W}$as follows:

$$M_s(F) = \sigma\left(f^{7\times 7}\left(\left[F^s_{avg}; F^s_{max}\right]\right)\right) \quad (4)$$

The sigmoid function is referred to as σ, while $f^{7\times 7}$ denotes the convolution operation with a size of 7×7.

3.3. The Proposed Architecture

The image spam detection performance is generally affected by the quality of the extracted features from the images. These features are classified into two categories: hand-crafted and deep-learned. Hand-crafted features are time-consuming, do not perform well, and do not consider the semantics compared to the deep-learned features, which are extracted using different deep learning models. In this paper, the CNN based on the CBAM is used to extract discriminative features from images. Figure 2 shows the proposed model for detecting spam image-based e-mail that has been confirmed using four datasets to demonstrate its efficiency. The following paragraphs describe the proposed model that involves three important steps:

The pre-processing step involves resizing and normalisation of the images. The datasets used in the experiments contain RGB spam and ham images of various sizes. For this reason, the images are resized to 128×128 pixels and normalised. **The feature extraction** step involves generating vectors from images using the CNN with CBAM as follows: The images are fed into the CNN-CBAM model, as shown in Table 1, which contains three blocks. In the first and second blocks, there are two convolutional layers with 32 and 64 filters followed by a batch normalisation layer, the CBAM model, and a max-pooling layer in each. The third block contains two convolutional layers of 128 filters, the first of which is followed by a batch normalisation layer and the second by a CBAM model. Finally, a global average pooling is added. In all convolutional layers, the Exponential Linear Unit (Elu) function is employed as an activation function [34], which works as demonstrated in Equation (5); all layers have the same kernel size of 3.

$$f(x) = \begin{cases} x & if\ x > 0 \\ \alpha(exp(x) - 1) & if\ x \leq 0 \end{cases} \quad (5)$$

The classification step aims at feeding the SVM model. Thus, the obtained vector is characterised by its richness and highly semantic representation. This vector is then fed to the SVM model for classifying the image as spam or ham. To train the CNN-CBAM model, Adam function is used as the optimizer. In addition, the learning rate is set to 0.001.

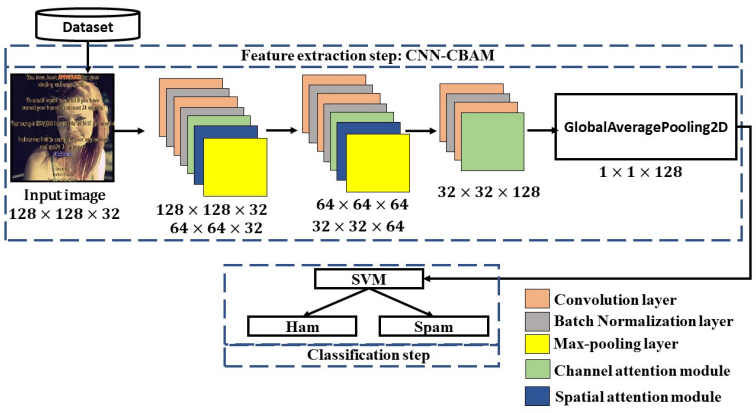


Figure 2. The proposed architecture.

Table 1. Summary of CNN-CBAM model.

Layer (Type)	Output Shape	Number of Param
InputLayer	(None, 128, 128, 3)	0
Conv2D	(None, 128, 128, 32)	896
Batch Normalization	(None, 128, 128, 32)	128
Conv2D	(None, 128, 128, 32)	9248
Batch Normalization	(None, 128, 128, 32)	128
Channel_attention	(None, 128, 128, 32)	0
Spatial_attention	(None, 128, 128, 32)	0
MaxPooling2D	(None, 64, 64, 32)	0
Conv2D	(None, 64, 64, 64)	18,496
Batch Normalization	(None, 64, 64, 64)	256
Conv2D	(None, 64, 64, 64)	36,928
Batch Normalization	(None, 64, 64, 64)	256
Channel attention	(None, 64, 64, 64)	0
Spatial attention	(None, 64, 64, 64)	0
MaxPooling2D	(None, 32, 32, 64)	0
Conv2D	(None, 32, 32, 128)	73,856
Batch Normalization	(None, 32, 32, 128)	512
Conv2D	(None, 32, 32, 128)	147,584
Channel attention	(None, 32, 32, 128)	0
Spatial attention	(None, 32, 32, 128)	0
GlobalAveragePooling2D	(None, 128)	0
Dense	(None, 1)	129

4. Experimental Results and Analysis

4.1. Datasets

The first publicly available dataset is called ISH [16]. It contains, as shown in Figure 3, two sets of real images in JPEG format. The first sample set contains 810 ham images, see Figure 3a, that are collected and downloaded randomly from social networking sites such as Flickr. The second sample set includes 928 spam images, see Figure 3b, that are from real spam e-mails, noting that eight spam images are excluded because they were corrupted from the spam image set. Therefore, after cleaning the data, the experiments provided below were conducted on 810 ham and 920 spam images.

(**a**) Ham images from ISH dataset

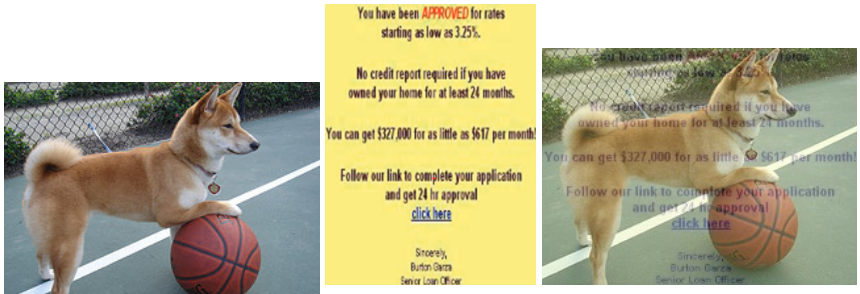
(**b**) Spam images from ISH dataset

Figure 3. Some examples of images from ISH dataset.

The second dataset, which we referred to as Annadatha dataset, was developed by [22]. It consists of a set of images, see Figure 4, that are generated using different techniques for creating a new improved and challenging spam image dataset. In order to increase the entropy of the local binary pattern and make the spam images more difficult to detect, the authors of [22] added a background layer and noise to them, and then modified the color elements of the spam images to be closer to ham ones. The total number of spam images that were generated is 1029.

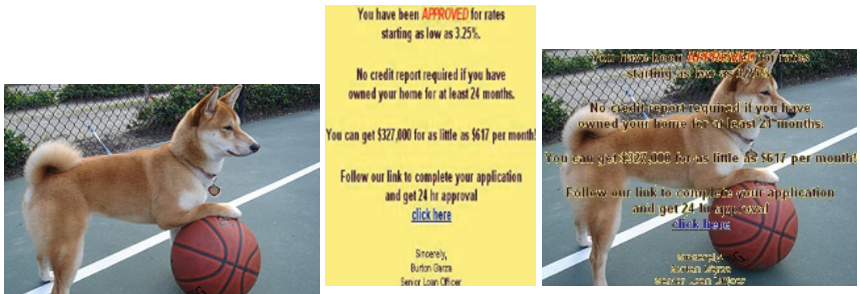
Figure 4. Some examples of images from Annadatha dataset.

As shown in Figures 5c and 6c, the third and fourth datasets, referred to as Chavda Approach 1 and Chavda Approach 2, respectively, were created by [19] to increase the difficulty of detecting image spam compared to the Annadatha dataset. Each dataset contains 810 spam images and 810 ham images, as shown in Table 2. The datasets were generated using both the publicly available Dredze dataset [15] and the ISH dataset. As depicted in Figure 5a, the generation of the datasets involved extracting the content of spam images from the Dredze dataset and overlaying them on ham images from the ISH dataset. The main difference between the third and fourth dataset is the appearance of text on the images.



(**a**) Ham image (**b**) Spam image (**c**) Challenge spam images

Figure 5. Image examples from Chavda approach 1 dataset.



(**a**) Ham image (**b**) Spam image (**c**) Challenge spam images

Figure 6. Image examples from Chavda approach 2 dataset.

Table 2. The summarisation of datasets.

Dataset	Number of Spam	Number of Ham	Total
ISH [16]	920	810	1730
Annadatha [22]	1029	810	1839
Chavda approach 1 [19]	810	810	1620
Chavda approach 2 [19]	810	810	1620

4.2. Experimental Results

The experiments are conducted on four datasets that are divided into two sets, namely, training and testing with a ratio of 70% and 30% consecutively. The metrics used for measuring the performance of the proposed model are as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$f1\_score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{9}$$

where TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively.

Table 3 shows the results of our experiments on four datasets, comparing the performance of two models: MNV2+SVM and our proposed approach. For each dataset, the

table reports the accuracy, precision, recall, and F1 score of each model. The MNV2+SVM model is a pre-trained DCNN model combined with the SVM algorithm as a classifier. Our proposed architecture uses a CNN technique enhanced with the CBAM module, also with the SVM algorithm as a classifier. For the ISH dataset, both MNV2+SVM and the proposed models achieved the same results, with an accuracy of 99.61%, precision of 99.29%, recall of 100%, and F1 score of 99.64%. This indicates that both models are equally effective for this dataset. For the Annadatha dataset, the proposed model performed significantly better than the MNV2+SVM model, with an accuracy of 99.81%, precision of 100%, recall of 99.68%, and F1 score of 99.84%. In contrast, the MNV2+SVM model had an accuracy of 94.92%, precision of 95.31%, recall of 95.91%, and F1 score of 95.61%. For the Chavda approach 1 dataset, the proposed model outperformed the MNV2+SVM model, with an accuracy of 96.29%, precision of 95.25%, recall of 97.57%, and F1 score of 96.40%. In comparison, the MNV2+SVM model had an accuracy of 80.65%, precision of 79.76%, recall of 82.99%, and F1 score of 81.34%. For the Chavda approach 2 dataset, the proposed model again performed better than the MNV2+SVM model, with an accuracy of 98.14%, precision of 98.37%, recall of 97.97%, and F1 score of 98.17%. The MNV2+SVM model had an accuracy of 91.15%, precision of 92.85%, recall of 89.47%, and F1 score of 91.13%. We can conclude that the proposed model outperforms the MNV2+SVM model on all four datasets.

Table 3. The experimental results on four datasets.

Dataset	Model	Accuracy	Precision	Recall	F1_Score
ISH	MNV2+SVM	99.61%	99.29%	100%	99.64%
	Our proposed approach	99.61%	99.29%	100%	99.64%
Annadatha	MNV2+SVM	94.92%	95.31%	95.91%	95.61%
	Our proposed approach	99.81%	100%	99.68%	99.84%
Chavda approach 1	MNV2+SVM	80.65%	79.76%	82.99%	81.34%
	Our proposed approach	96.29%	95.25%	97.57%	96.40%
Chavda approach 2	MNV2+SVM	91.15%	92.85%	89.47%	91.13%
	Our proposed approach	98.14%	98.37%	97.97%	98.17%

In addition, Figure 7a–d presents the confusion matrix of the proposed model on ISH, Annadatha, Chavda approach 1, and Chavda approach 2, respectively.

In addition, Figure 8a–d provides the ROC curve on ISH, Annadatha, Chavda approach 1, and Chavda approach 2, respectively. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The TPR represents the proportion of actual positive samples that are correctly identified by the model as positive. The FPR, on the other hand, is defined as the proportion of actual negative samples that are incorrectly identified by the classifier as positive. As shown in Figure 8a–d, the AUC values achieved 1, 1, 0.96, and 0.98 on ISH, Annadatha, Chavda approach 1, and Chavda approach 2, respectively. On the other hand, Table 4 shows the experimental results that are conducted on the merged dataset, which is created by merging the ISH, Annadatha, Dredze, Chavda approach 1, and Chavda approach 2 datasets. The 5-fold cross validation method is applied on the merged dataset which contains 5577 images, and comparing our proposed method with two methods, namely, MNV2+SVM and EfficientV2M+SVM [35]. According to Table 4, we show that our proposed method performs well in terms of accuracy.

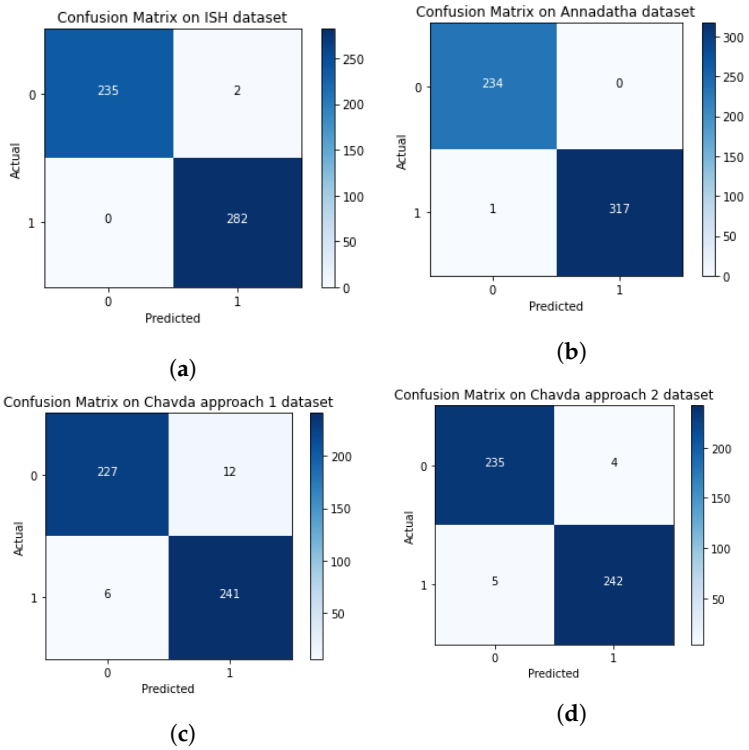


Figure 7. Confusion matrix of our model on four datasets. (**a**) ISH dataset; (**b**) Annadatha dataset; (**c**) Chavda approach 1 dataset; (**d**) Chavda approach 2 dataset.

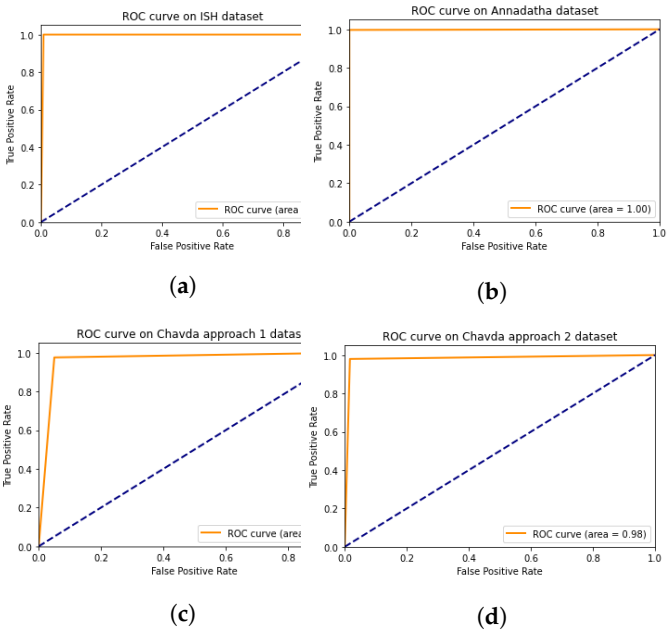


Figure 8. ROC curve. (**a**) ROC curve on ISH dataset; (**b**) ROC curve on Annadatha dataset; (**c**) ROC curve on Chavda approach 1 dataset; (**d**) ROC curve on Chavda approach 2 dataset.

Table 4. The experimental results on merged dataset using 5-fold cross-validation.

5-Fold Cross-Validation	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
Our proposed approach	99.28%	99.01%	99.19%	99.10%	99.37%	99.19%
MNV2+SVM	87.90%	88.26%	89.05%	88.69%	88.25%	88.43%
EfficientNetV2M+SVM	74.56%	69.99%	68.53%	70.50%	70.95%	70.91%

Tables 5–8 provide the comparisons of our proposed model to the other state-of-the-art methods in terms of accuracy that are carried out on four datasets including, ISH, Annadatha, Chavda approach 1, and Chavda approach 2. From Table 7, we can see that the proposed model exceeds the models used by [19,24,28] with an improvement of 3%. In addition, our model outperformed the model used by [5,22,24,28] and it achieved an accuracy of 99.81%. In addition, the proposed model outperforms [19,24,28] with an improvement of 1% in terms of accuracy. In addition, the proposed model is compared with the hybrid MobileNetV2 and SVM model, as shown in Table 3, and our proposed model outperforms on four datasets, namely, ISH, Annadatha, Chavda approach 1, and Chavda approach 2.

Table 5. Comparison results on ISH dataset.

	Method	Accuracy	Precision	Recall	F1_Score
ISH	[22]	97%	-	-	-
	[19]	97%	-	-	-
	[28]	99.02%	-	-	-
	[24]	**99.77%**	-	-	-
	[25]	98.65%	-	-	-
	MNV2+SVM	99.61%	99.25%	100%	99.64%
	Our proposed approach	99.61%	99.29%	100%	99.64%

Table 6. Comparison results on Annadatha dataset.

	Method	Accuracy	Precision	Recall	F1_Score
Annadatha dataset	SVM [22]	70%	-	-	-
	CNN [28]	83.13%	-	-	-
	CNN [24]	99.78%	-	-	-
	ResNet+SVM [5]	95.55%	95.59%	-	95.89%
	Our proposed approach	**99.81%**	**100%**	**99.68%**	**99.84%**

Table 7. Comparison results on Chavda approach 1 dataset.

	Method	Accuracy	Precision	Recall	F1_Score
Chavda approach 1	SVM [19]	68%	-	-	-
	CNN [28]	67.69%	-	-	-
	CNN [24]	93.75%	-	-	-
	Our proposed approach	**96.29%**	**95.98%**	**96.76%**	**96.37%**

Table 8. Comparison results on Chavda approach 2 dataset.

	Method	Accuracy	Precision	Recall	F1_Score
Chavda approach 2	SVM [19]	68%	-	-	-
	CNN [28]	67.69%	-	-	-
	CNN [24]	97.83%	-	-	-
	Our proposed approach	**98.14%**	**98.37%**	**97.97%**	**98.17%**

5. Conclusions

Spammers have developed a new form of unsolicited e-mail called image-based spam to evade text-based spam filters. Determining whether an image is spam or ham requires selecting appropriate distinguishing features, which is a challenging task. Many works have proposed using machine learning and deep learning techniques to extract these features from images. However, recent research has shown that these methods produce insufficient results, especially on improved datasets. Our proposed architecture is based on the CNN technique and is enhanced with the CBAM module, which can reliably detect improved image spam datasets that were previously undetectable using image processing-based features, resulting in a significant increase in detection accuracy. We use the SVM model as a classifier to differentiate spam from ham. We also compared our model to pre-trained DCNN models, such as MobileNetV2, to demonstrate the superiority of our model. In future work, we intend to apply our proposed model to other large datasets with different characteristics.

Author Contributions: Conceptualization, G.H., J.R. and H.T.; methodology, G.H., J.R. and H.T.; software, G.H.; validation, G.H., J.R., M.A.M., A.Y. and H.T.; formal analysis, G.H., J.R. and H.T.; investigation, G.H., J.R. and H.T.; resources, G.H., J.R. and H.T.; data curation, G.H., M.A.M. and A.Y.; writing—original draft preparation, G.H.; writing—review and editing, G.H., J.R., A.Y. and H.T.; visualization, G.H., M.A.M. and A.Y.; supervision, J.R. and H.T.; project administration, J.R. and H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: We used ISH [16], Dredze [15], Annadatha [22] and Chavda [19] datasets in this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kim, B.; Abuadbba, S.; Kim, H. DeepCapture: Image spam detection using deep learning and data augmentation. In *Proceedings of the Australasian Conference on Information Security and Privacy, Perth, WA, Australia, 30 November–2 December 2020*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 461–475.
2. Yang, H.; Liu, Q.; Zhou, S.; Luo, Y. A spam filtering method based on multi-modal fusion. *Appl. Sci.* **2019**, *9*, 1152. [CrossRef]
3. Hnini, G.; Riffi, J.; Mahraz, M.A.; Yahyaouy, A.; Tairi, H. MMPC-RF: A Deep Multimodal Feature-Level Fusion Architecture for Hybrid Spam E-mail Detection. *Appl. Sci.* **2021**, *11*, 11968. [CrossRef]
4. Srinivasan, S.; Ravi, V.; Sowmya, V.; Krichen, M.; Noureddine, D.B.; Anivilla, S.; Soman, K. Deep convolutional neural network based image spam classification. In Proceedings of the 2020 6th Conference on Data Science and Machine Learning Applications (CDMA), Riyadh, Saudi Arabia, 4–5 March 2020; pp. 112–117.
5. Salama, W.M.; Aly, M.H.; Abouelseoud, Y. Deep learning-based spam image filtering. *Alex. Eng. J.* **2023**, *68*, 461–468. [CrossRef]
6. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 84–90. [CrossRef]
7. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
8. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
10. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.

11. Zheng, M.; Xu, J.; Shen, Y.; Tian, C.; Li, J.; Fei, L.; Zong, M.; Liu, X. Attention-based CNNs for image classification: A survey. In Proceedings of the Journal of Physics: Conference Series, Brasilia, Brazil, 23–27 June 2022; IOP Publishing: Bristol, UK; Volume 2171, p. 012068.
12. Xue, Z.; Yu, X.; Liu, B.; Tan, X.; Wei, X. HResNetAM: Hierarchical residual network with attention mechanism for hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 3566–3580. [CrossRef]
13. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
14. Al-Duwairi, B.; Khater, I.; Al-Jarrah, O. Detecting image spam using image texture features. *Int. J. Inf. Secur. Res. (IJISR)* **2012**, *2*, 344–353. [CrossRef]
15. Dredze, M.; Gevaryahu, R.; Elias-Bachrach, A. Learning fast classifiers for image spam. In Proceedings of the CEAS, Rome, Italy, 21–23 May 2007; pp. 2007–2487.
16. Gao, Y.; Yang, M.; Zhao, X.; Pardo, B.; Wu, Y.; Pappas, T.N.; Choudhary, A. Image spam hunter. In Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, USA, 31 March–4 April 2008; pp. 1765–1768.
17. Hosseini, M.; Rahmati, M. A Method for Image Spam Detection Using Texture Features. *Int. Acad. J. Sci. Eng.* **2015**, *2*, 51–58.
18. Salih, A.M.; Dhannoon, B.N. Weighted k-Nearest Neighbour for Image Spam Classification. *Iraqi J. Sci.* **2021**, *62*, 1036–1045. [CrossRef]
19. Chavda, A.; Potika, K.; Troia, F.D.; Stamp, M. Support Vector Machines for Image Spam Analysis. In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications—BASS*; INSTICC, SciTePress: Porto, Portuga, 2018; pp. 431–441. [CrossRef]
20. Kumaresan, T.; Sanjushree, S.; Suhasini, K.; Palanisamy, C. Image spam filtering using support vector machine and particle swarm optimization. *Int. J. Comput. Appl* **2015**, *1*, 17–21.
21. Kumaresan, T.; Sanjushree, S.; Palanisamy, C. Image spam detection using color features and K-Nearest neighbor classification. *Int. J. Comput. Inf. Eng.* **2015**, *8*, 1904–1907.
22. Annadatha, A.; Stamp, M. Image spam analysis and detection. *J. Comput. Virol. Hacking Tech.* **2018**, *14*, 39–52. [CrossRef]
23. Kumar, P.; Biswas, M. SVM with Gaussian kernel-based image spam detection on textual features. In Proceedings of the 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India, 9–10 February 2017; pp. 1–6.
24. Singh, A.B.; Singh, K.M.; Chanu, Y.J.; Thongam, K.; Singh, K.J. An Improved Image Spam Classification Model Based on Deep Learning Techniques. *Secur. Commun. Netw.* **2022**, *2022*, 8905424. [CrossRef]
25. Ghizlane, H.; Jamal, R.; Mahraz, M.A.; Ali, Y.; Hamid, T. Spam image detection based on convolutional block attention module. In Proceedings of the 2022 International Conference on Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 18–20 May 2022; pp. 1–4.
26. Shang, E.X.; Zhang, H.G. Image spam classification based on convolutional neural network. In Proceedings of the 2016 International Conference on Machine Learning and Cybernetics (ICMLC), Jeju, Republic of Korea,10–13 July 2016; Volume 1, pp. 398–403.
27. Kumar, A.D.; Vinayakumar, R.; Soman, K. Deepimagespam: Deep learning based image spam detection. *arXiv* **2018**, arXiv:1810.03977.
28. Sharmin, T.; Di Troia, F.; Potika, K.; Stamp, M. Convolutional neural networks for image spam detection. *Inf. Secur. J. Glob. Perspect.* **2020**, *29*, 103–117. [CrossRef]
29. Onova, C.; Omotehinwa, T.O. Development of a Machine Learning Model for Image-based Email Spam Detection. *FUOYE J. Eng. Technol.* **2021**, *6*, 336. [CrossRef]
30. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
31. Mnih, V.; Heess, N.; Graves, A.; Kavukcuoglu, K. Recurrent models of visual attention. *Adv. Neural Inf. Process. Syst.* **2014**, *27*. Available online: https://proceedings.neurips.cc/paper_files/paper/2014/hash/09c6c3783b4a70054da74f2538ed47c6-Abstract.html (accessed on 24 April 2023).
32. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3156–3164.
33. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.

34. Clevert, D.A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv* **2015**, arXiv:1511.07289.
35. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning. PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.