DOI: 10.15849/IJASCA.220328.15

Int. J. Advance Soft Compu. Appl, Vol. 14, No. 1, March 2022
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Image SPAM Detection Using ML and DL Techniques

Nawal Nayl Abuzaid[1], Huthaifa Zaid Abuhammad[2]

Department of Data Science and Artificial intelligence Science- Isra University-Jordan.
[1]e-mail: Nabuza1d1@yahoo.com.
[2]e-mail: H.abuhammad@iu.edu.jo

Abstract

Since e-mail is one of the most common places to send messages, spammers have, in recent years, targeted it as a preferred way of distributing undesired messages (spam) to several users to spread viruses, cause destruction, and obtain user's information. Spam images are considered one of the known spam types. The spammer processes images and changes their characteristics, especially background colour, font type, or adding artefacts to the images to spread spam. In this paper, we proposed a spam detection model using Several ML (Random-Forest (RF), Decision-Tree (DT), K-Nearest Neighbor (KNN), Support-Vector Machine (SVM), Naïve-Bays (NB), and Convolutional Neural Network (CNN)). Several experiments evaluate the efficiency and performance of the (ML) algorithms for spam detection. Using the Image Spam Hunter Dataset extracted from real spam e-mails, the proposed model achieved over 99% accuracy on spam image detection.

Keywords: *SPAM, Machine Learning, Image Classification, Feature Extraction, Deep Learning.*

1 Introduction

In the last years, E-mail services have become one of the most common means of communication. However, the number of spammers has increased. The term spam covers all undesired messages which are sent to several users to spread viruses, intruding on users' privacy, and as a means of obtaining people's information. Info Week published about e-mails and admitted that 94% of all messages were spam. This makes it crucial for users to filter spam to continue using e-mail. The receiving party uses automated spam filters to classify messages into spam or ham mail. Spam refers to an unwanted e-mail sent to groups. Ham refers to e-mail messages that the user wants to receive [1].

In recent times, the volume of spam has increased and now represents 80% of all messages on the Internet. Spam images are a significant cause of the spread of spam. Image spam is an unwanted message inserted into images sent to the

Received 8 November 2021; Accepted 28 February 2022

recipient without verifiable permission. Spammers employ images to avoid text-based filters. HAM e-mails are another form of spam message, but they are unwanted and harmless. Image spam is constantly being modified over the years, leading to its enormous expansion. Whereas many users can detect spam e-mails by subject, sender, and e-mail content, it is not simple to detect image spam, especially when attackers insert text within the images [2].

Many techniques have been developed to differentiate between spam and normal e-mail, particularly deep-learning techniques. This approach in the research focuses on the implementation of CNN [3].

Ligament or Spam content is one of the challenges facing e-mail service providers and Internet Service Providers (ISPs). It is used to spread unwanted content, inflict attacks on users, increase system correspondence, and consume space. It also destroys user data and obtains personal information such as account numbers, passwords, etc. Spam is one of the most common attacks users face, despite the many security technologies available in message sending. These technologies have helped researchers focus on this area by discovering new ways to classify e-mails. Spam messages cause significant losses to organizations and increase mail server space, network bandwidth, spam filtering, and mail server processing. Spam forces users to expend extra time and effort deleting and cleaning the bin and discard these unsolicited message contents. Finding a way to successfully filter spam before it reaches the user should increase users' productivity who depend on e-mail for personal and business correspondence [4].

Those who send spam messages use many techniques to process images and change the characteristics of the message, whether by altering background colours or font type or adding artefacts to the images that are sent. These technologies pose a significant challenge when filtering out unwanted messages attached to images sent via e-mail, as existing technologies struggle to detect them. Images are used to deceive spam filtering methods, as they are sent via e-mail as a picture or drawing of characters, appearing to the recipient as text. Upon receiving the message, the recipient opens it, and the image is downloaded automatically.

As far back as 1996, the Internet community began to study a solution to this problem. Software engineers Dave Rand and Paul Vixie founded a non-profit organization called Mail Abuse Prevention System (MAPS) to trace the IP addresses of spammers in a bid to counter the rapid increase in the number of spam e-mails. Now, spam detection is marked as necessary regard for security in all domains of the Internet, especially websites and e-mails servers.

The organization of this paper shall be as follows: The Literature Review and Related Work will be presented in Section II. Section III is the Methodology, while Section IV illustrates the research results. Finally, Section V concludes the findings and recommends future work.

2 Related Work

Conventional approaches for spam filtering, such as malicious and E-mail content machine-learning approaches, are considered simple techniques in this field. Current methods are well performed on e-mail messages, but recently, spam content detection raised difficulties when dealing with short and noisy SMS, Tweets, and comments on social media platforms. Fast-moving platforms that rely on short messages are complex and make it challenging to develop and deploy a

reliable E-mail spam content detection and prevention model. Deep Learning (DL) is the most recent growth in the field of classification Technologies of natural language processing (NLP) [5]. This section shows a study that implements a CNN algorithm to differentiate spam from non-spam messages on Tiago's Dataset.

Several steps were taken to collect the Dataset to get a good model. The first step included preprocessing text by decreasing lower case, tokenization, stemming, and stop word removal. Finally, text was transformed into a matrix of term frequency & inverse document frequency (TF-IDF) features. Results showed that CNN could perform spam classification with an accuracy rate of 95.5% [6].

The study, as mentioned above, uses a CNN to detect spam in social media text in the SMS Spam dataset (UCI repository) and Tweet datasets. This approach is a semantic co-evolutionary neural network (SCNNWordNet), and the term net model is applied to locate the similarities between terms and interpreted as a vector. Results were on the SMS spam dataset up to 98.65%, and on the social media (Twitter), Dataset was up to 94.40%, which is higher than the state-of-the-art results[7].

Five CNN networks and one feature-based spam detection model on Twitter were used on 1KS10KN and spam data sets. Many word embeddings (Glove, Word2vec) and CNN are commonly used in model training. Feature-based paradigm uses (content, user)-based features. The result shows that CNN will surpass the researcher's approaches using 1KS10KN spam datasets for spam detection with an accuracy level of 97.5% [8].

A further study proposed a CNN algorithm on the social media platform (Instagram) to detect and recognize possible image spam posts. Many architectures of CNN were applied to analyze the achievement of all structures, N-layers, VGG16, and Alex Net approaches used to detect image spam on Instagram posts. The model dataset between training and testing used around 8000 images obtained from applying a web crawler on Instagram posts. The findings show that the VGG16 architecture achieved the highest Accuracy, with Accuracy equal to 0.842 [9].

Other researchers applied the Convolutional Neural Network (CNN-DL) model. Some pre-trained Image.Net architectures such as VGG19 and Alex Net were trained on the span Hunter Dataset (ISH), Improved Dataset, and Dredze Image Spam datasets to classify the image spam. Findings show a high accuracy reached 99% in the best run [10].

Kim, Abuadbba, and Kim[11] used SVM and CNN-DL techniques supported with multilayer perceptron's (MLP) to detect image spam. Raw images and canny images features were tested in both models. The CNN results achieved the best Accuracy of 88%.

We summarize some of the studies that implement an image spam hunter as follows:

A machine learning classifier is implemented to classify spam images in e-mails. This method excludes images used in the training phase to recognize spam images. Spam images are collected from the actual e-mail, besides random natural images. A probabilistic boosting tree classifier was implemented to classify

images into spam or not spam. The result showed a perfect achievement when applying the data set [12].

In this model, a probabilistic boosting tree was suggested to define if the received images contain spam or not, according to colour and gradient histograms. This model proved its ability to detect spam without applying OCR. The results showed that the model identifies 90% of classified spam images besides 86% of unlabeled non-spam images as spam. This model uses file properties, histogram, and Hough transform to detect spam images into ham or spam images. The suggested approach has the classification ability. The results showed that this approach applies file properties and removes 80% of the spam images, and applying a histogram removes 84% of the spam messages.

In a study carried out by [13], CNN was used to classify image spam. The model learned the labelled data and experimented with the CNN test data. CNN used a dataset containing both spam images and natural images. Results reached 91.7% accuracy by enhancing ML and IP processes.

[14] introduced a help vector machine algorithm and an extraction function to detect spam e-mails. This method implemented Apache Public corpus dataset and eliminated all redundant symbols and alphabets and URL and HTML tags. The results revealed that the SVM algorithm had achieved 98% precision in the framework.

An additional research work saw the implementation of the hybrid classifiers, Vector Machine Help and the Spam E-mail Filter Decision Tree [14,15,16].

3 Research Methodology

This research aims to increase model accuracy in detecting and preventing spam images using the latest machine learning technology. This chapter explains how and what the classifiers will do. The six classifiers include deep learning CNN will be used to classify the same features with the same designed software in Python. In the results chapter, the comparison between the classifiers will be presented.
This thesis will use the features and information that had been extracted from the CNN as a 4D array with converting the array to 2D to match machine learning inputs with extract features and best-selected features from real images, thus feeding the used classifiers in our proposed method such as CNN, KNN, RF, DT, SVM, and NB.

4 Dataset

Image Hunter dataset is considered one of the best databases containing spam images and natural images as most studies detected spam images that use data spam only. The Image Hunter dataset was used in 75 studies as the database of natural images is characterized by its comprehensiveness and spam images. The Dataset contains 1,725, 928 of which are spam images extracted from real spam e-mails as the spam sample set. These images are part of the image spam we received in the last six months, excluding animation. The normal images group contains 810 images randomly downloaded from Flickr.com and 20 scanned

documents. The images have been resized and rescaled to 224*224 and been augmented to obtain the best shape for each classifier.

5 The Proposed Method

Our proposed model, shown in Figure 1 below, uses VGG-16 (16 weight layers) for feature extraction and selection to train and test the image dataset. To validate the eleven images in appendix A, the model will use the pre-trained model saved into npy array as 4D array resulting from feature extraction of the images with label each image based on the facade conditions. VGG is a CNN model for recognition images developed at the University of Oxford by the Visual Geometry Group.

The cov1 layer as the input layer has a regular RGB image size (224 x 224). The processed images weighted by a stack of CNN layers with a minimal receptive field were used to the filters with 3×3. The input channels as a linear transform use a 1×1 convolution filter. Spatial pooling layer used and done by six max-pools.
Four Fully Connected layers use CNN inner layers (with varying depths in various architectures): the first two have 4096 channels each. In contrast, the third uses a 1000-way ILSVRC grouping and therefore has 1000 channels. The Soft-max layer is the final one. For all networks, the ultimately linked layers have the same architecture.
Non-linearity correction is applied to both hidden layers (ReLU). Local Response Normalization (LRN) is also included in none of the networks (except one), which does not improve the IL-SVRC data collecting efficiency but leads to higher Memory and computational use.

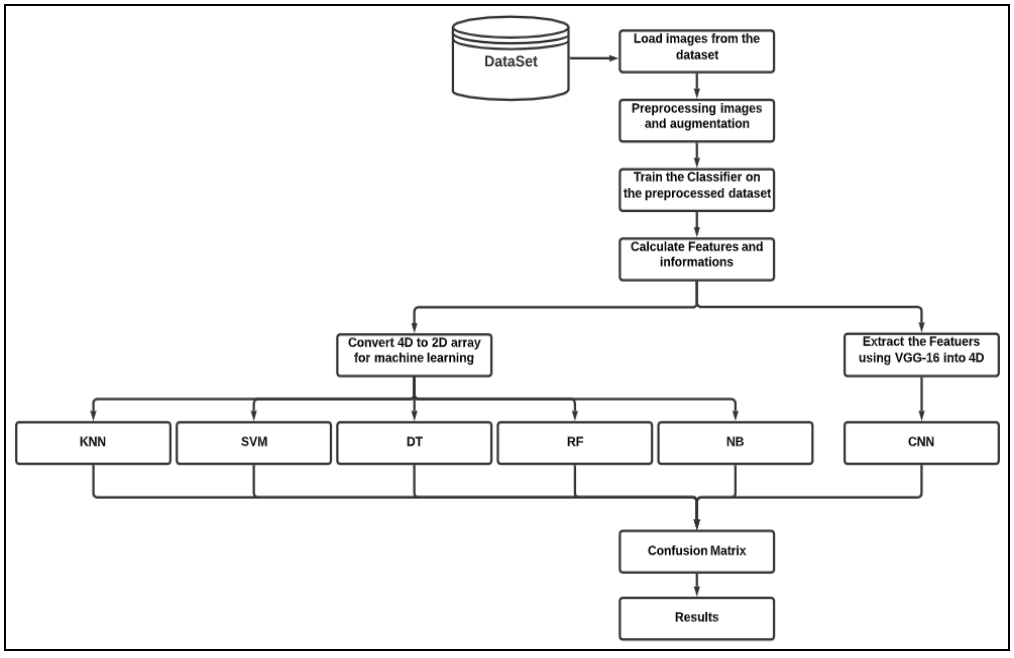


Figure1: The Proposed Method Flowchart

5.1 Experimental Setup

5.1.1 Summarizes table

The following table is a summary of the variable, sets, and other information that has been used in the proposed method,

Table 1: Summarizes Table

Variable, Function or Abbrivation	Explanation
SMS	Short Message Service
OCR	Optical Character Recognition
IDLM	Image-Based Deep Learning Model
CNN	Convolutional Neural Network
FP	False Positive
TP	True Positive
FN	False Negative
TN	True Negative
ML	Machine Learning
SVM	Support Vector Machine
DT	Decision Tree Classifier
kNN	k nearest Neighbor
NLP	Natural Language Processing
TF-IDF	Term-Frequency Inverse Document
S(CNN)	Semantic Convolutional Neural Network
ISH	Image Span Hunter Dataset
MLP	Multilayer Perceptron's
RF	Random Forest
ROC	Receiver Operating Characteristic Curve
AUC	The area under The Curve
ARFF file	Attribute-Relation File Format
NBC	Naive Bayes Classifier

5.1.1 Hardware

Much computational power is needed for deep learning, and the other five classifications need the data set used in the experiment because the algorithms are compatible. Creating deep convolution neural systems in a grammar processing unit (GPU) can be difficult in parallel because of robust repetitive calculations, such as convolution and backpropagation.

A lot of basic parallel matrix calculations are required for machine programming. Tests were carried out using an Intel Core i7-7700HQ2.80GHz central processing unit (CPU), 16 GB RAM, and 4 GB Nvidia GeForce 1060MX GPU display memory. The display adapters support parallel Nvidia CUDA systems and speed GPU calculations by making the library Nvidia CUDA Deep Neural Network (cuDNN).

5.1.2 Software

Anaconda Python 3.7.: The machine-learning architecture for the sponsored version of GPU was chosen to be Keras (Tensorflow). Keras is based on Tensorflow. KERAS has added to its success and broad support for various learning styles, design features, and hypermeters. Libraries such as Panda's data storage, Numpy for multidimensional arrays, Scikit Learn for data analysis were enabled. The other classifiers have been trained, tested, and classified via the Sklearn machine libraries.

6 Results, Analysis and Discussions

This section describes most of the experiments that were applied, in addition to the results that these experiments produced. The results are presented based on five runs and both datasets; the number of runs came to verify our model Overstretching and get a standard AUC and a second AUC based on X train predictions (so on the same set we used fit the algorithm). we could overfat, but the precise outcomes on the training set and try to apply it to the test) if values are widely apart. With and without black images. Four types of performance measures were used: Accuracy, precision, recall, and f-score, in addition to time. We used in our discussion five runs for each classifier and the average.

6.1 Naive Bayes:

Table.2 shows the results for each naïve Bayes classifier run. Naive Bayes classifier achieved the same Accuracy for each run resulting in 84.36%. The naïve Bayes has been trained on data split based on 70% training and 30% testing data. The same Accuracy was based on the naïve Bayes statistical method calculating the features with the same results for our balanced Dataset. The non-spam precision and spam recall and the inability of a classifier to decide the results made the results to be under 90%.

Table 3 shows the results with the black images, where we can see the drop of Accuracy for our model to 80.53% because the classifier cannot figure out the feature for the black images, thus affecting the results.

Table 2: Results of Naive Bayes Classifier without Black Images

Run	Accuracy	Precision	Recall	F-score	Time
1	84.36	NSPAM 0.74 SPAM 1.00	NSPAM 1.00 SPAM 0.72	NSPAM 0.85 SPAM 0.84	00.512656
2	84.36	NSPAM 0.74 SPAM 1.00	NSPAM 1.00 SPAM 0.72	NSPAM 0.85 SPAM 0.84	00.538416
3	84.36	NSPAM 0.74 SPAM 1.00	NSPAM 1.00 SPAM 0.72	NSPAM 0.85 SPAM 0.84	00.530576
4	84.36	NSPAM 0.74 SPAM 1.00	NSPAM 1.00 SPAM 0.72	NSPAM 0.85 SPAM 0.84	00.521637
5	84.36	NSPAM 0.74 SPAM 1.00	NSPAM 1.00 SPAM 0.72	NSPAM 0.85 SPAM 0.84	00.603419
Avg. (Accuracy)		84.36			

Table 3: Results of Naive Bayes Classifier with Black Images

Run	Accuracy	Precision	Recall	F-score	Time
1	80.53	NSPAM 0.70 SPAM 1.00	NSPAM 1.00 SPAM 0.64	NSPAM 0.82 SPAM 0.78	00.402400
2	80.53	NSPAM 0.70 SPAM 1.00	NSPAM 1.00 SPAM 0.64	NSPAM 0.82 SPAM 0.78	00.399899
3	80.53	NSPAM 0.70 SPAM 1.00	NSPAM 1.00 SPAM 0.64	NSPAM 0.82 SPAM 0.78	00.393980
4	80.53	NSPAM 0.70 SPAM 1.00	NSPAM 1.00 SPAM 0.64	NSPAM 0.82 SPAM 0.78	00.388919
5	80.53	NSPAM 0.70 SPAM 1.00	NSPAM 1.00 SPAM 0.64	NSPAM 0.82 SPAM 0.78	00.404918
Avg. (Accuracy)		80.53			

6.2 KNN:

Table 4 represents the results and efficiency of the proposed model performance with the KNN classifier. As for the accuracy scale, the model achieved the best performance with 98.55; the time taken in the classification process was (32.214631 seconds) on average. The interpretation of the accuracy results obtained through the training and testing process appears, where we notice a stable accuracy result. This appeared in all five runs when we used Kfold=10 and defined the classes assigning the labels for each class. KNN is a slow, non-parametric learning technique. It predicts the classification of a new sample point based on data from many classes. KNN is non-parametric since it makes no assumptions about the data under consideration, i.e., the model is created from the data. Because most data do not adhere to standard theoretical assumptions, such as applying a linear regression model, KNN is crucial for analyzing data with little or no prior information. Table 5 shows the results with the black images. With the black images, we can see the drop of Accuracy for our model to 97.41% because

the classifier cannot figure out the feature for the black images, thus affecting the results.

Table 4: Results of KNN Classifier without Black Images

Run	Accuracy	Precision	Recall	F-score	Time
1	98.55	NSPAM 0.98 SPAM 0.99	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	32.304398
2	98.55	NSPAM 0.98 SPAM 0.99	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	32.539341
3	98.55	NSPAM 0.98 SPAM 0.99	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	32.126861
4	98.55	NSPAM 0.98 SPAM 0.99	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	32.214631
5	98.55	NSPAM 0.98 SPAM 0.99	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	32.869269
Avg. (Accuracy)		98.55			

Table 5: Results of KNN Classifier with Black Images

Run	Accuracy	Precision	Recall	F-score	Time
1	97.41	NSPAM 0.96 SPAM 0.93	NSPAM 0.98 SPAM 0.99	NSPAM 0.97 SPAM 0.95	15.090925
2	97.41	NSPAM 0.96 SPAM 0.93	NSPAM 0.98 SPAM 0.99	NSPAM 0.97 SPAM 0.95	15.484741
3	97.41	NSPAM 0.96 SPAM 0.93	NSPAM 0.98 SPAM 0.99	NSPAM 0.97 SPAM 0.95	16.005532
4	97.41	NSPAM 0.96 SPAM 0.93	NSPAM 0.98 SPAM 0.99	NSPAM 0.97 SPAM 0.95	16.111429
5	97.41	NSPAM 0.96 SPAM 0.93	NSPAM 0.98 SPAM 0.99	NSPAM 0.97 SPAM 0.95	15.979068
Avg. (Accuracy)		98.55			

6.3 SVM:

Table 6 presents our proposed model performance findings and efficiency results with the SVM classifier. It also achieved a stable accuracy rate for every five runs with 98.76% accuracy. Regarding time, it achieved 07.010768 seconds on average for all runs. The interpretation of the accuracy results obtained through the training and testing process appears. We notice a stable accuracy result when the hyper-parameter used 10 Kfold for each run.

Because of the ideal margin gap between separating hyper-planes, SVM is more accurate and resilient than its rivals, and it can make superior predictions on test data. It is also more computationally efficient since it employs the Kernel technique in a dual problem. As a result, it is faster to train, more accurate and has more stability/robustness.

Table 7 shows the results with the black images, where we can see the drop of Accuracy for our model to 97.52 because the classifier cannot figure out the feature for the black images, thus affecting the results.

'

Table 6: Results of an SVM Classifier Without Black Image

Table 7: Results of an SVM Classifier with Black Images

Run	Accuracy	Precision	Recall	F-score	Time
1	98.76	NSPAM 0.98 SPAM 0.99	NSPAM 0.99 SPAM 0.99	NSPAM 0.99 SPAM 0.99	07.009903
2	98.76	NSPAM 0.98 SPAM 0.99	NSPAM 0.99 SPAM 0.99	NSPAM 0.99 SPAM 0.99	07.010768
3	98.76	NSPAM 0.98 SPAM 0.99	NSPAM 0.99 SPAM 0.99	NSPAM 0.99 SPAM 0.99	07.122561
4	98.76	NSPAM 0.98 SPAM 0.99	NSPAM 0.99 SPAM 0.99	NSPAM 0.99 SPAM 0.99	06.685158
5	98.76	NSPAM 0.98 SPAM 0.99	NSPAM 0.99 SPAM 0.99	NSPAM 0.99 SPAM 0.99	06.930724
Avg. (Accuracy)	98.76				

Run	Accuracy	Precision	Recall	F-score	Time
1	97.52	NSPAM 0.96 SPAM 0.92	NSPAM 0.98 SPAM 0.95	NSPAM 0.97 SPAM 0.93	04.841315
2	97.52	NSPAM 0.96 SPAM 0.92	NSPAM 0.98 SPAM 0.95	NSPAM 0.97 SPAM 0.93	04.439815
3	97.52	NSPAM 0.96 SPAM 0.92	NSPAM 0.98 SPAM 0.95	NSPAM 0.97 SPAM 0.93	04.691754
4	97.52	NSPAM 0.96 SPAM 0.92	NSPAM 0.98 SPAM 0.95	NSPAM 0.97 SPAM 0.93	04.486370
5	97.52	NSPAM 0.96 SPAM 0.92	NSPAM 0.98 SPAM 0.95	NSPAM 0.97 SPAM 0.93	04.470860
Avg. (Accuracy)	98.76				

6.4 Decision Tree:

Table 8 represents the results and efficiency of the proposed model performance with the Decision Tree classifier. As for the accuracy scale, our proposed method achieved the best results and performance of 70%; it achieved 97.32, the time taken in the classification process was 03.723518 seconds.

Unlike the other classifiers, the decision tree shows different results for each run with an average accuracy rate (96.95). The difference came to that decision tree for each run assigned different labels for node and started decisioning the results based on different and very narrow classing. The decision tree with the different results obtained a good result in our proposed model.

A decision tree has the benefit of forcing the examination of all potential choice outcomes and tracing each path to a conclusion. It generates a detailed analysis of the implications along each branch and indicates decision nodes that require more investigation. Decision trees assign each problem, decision path, and the particular result values. The use of monetary values clarifies the costs and advantages. This method highlights the pertinent decision routes, lowers confusion, clarifies ambiguity, and explains the financial implications of

alternative options. Decision trees are simple to use and explain, requiring no complicated formulae. They graphically show all the choice options for rapid comparisons in an easy-to-understand manner with very brief explanations.

Table 9 shows the results with the black images; with the black images, we can see the increase of Accuracy for our model to an average of 97.63 due to the classifier figuring the feature for the black images affecting the results.

Table 8 Result of a Decision Tree Classifier Without Black Images

Run	Accuracy	Precision	Recall	F-score	Time
1	96.5	NSPAM 0.98 SPAM 0.96	NSPAM 0.94 SPAM 0.98	NSPAM 0.96 SPAM 0.97	04.010087
2	96.91	NSPAM 0.98 SPAM 0.96	NSPAM 0.95 SPAM 0.98	NSPAM 0.96 SPAM 0.97	04.256231
3	97.11	NSPAM 0.97 SPAM 0.97	NSPAM 0.96 SPAM 0.98	NSPAM 0.97 SPAM 0.97	03.817902
4	97.32	NSPAM 0.98 SPAM 0.97	NSPAM 0.96 SPAM 0.98	NSPAM 0.97 SPAM 0.98	03.723518
5	96.91	NSPAM 0.97 SPAM 0.97	NSPAM 0.96 SPAM 0.98	NSPAM 0.96 SPAM 0.97	04.263265
Avg. (Accuracy)		96.95			

Table 9 Result of a Decision Tree Classifier with Black Images

Run	Accuracy	Precision	Recall	F-score	Time
1	97.34	NSPAM 0.98 SPAM 0.97	NSPAM 0.96 SPAM 0.98	NSPAM 0.97 SPAM 0.98	02.767157
2	97.93	NSPAM 0.99 SPAM 0.97	NSPAM 0.97 SPAM 0.99	NSPAM 0.98 SPAM 0.98	02.633753
3	97.64	NSPAM 0.97 SPAM 0.97	NSPAM 0.96 SPAM 0.98	NSPAM 0.97 SPAM 0.97	02.790724
4	97.05	NSPAM 0.98 SPAM 0.97	NSPAM 0.96 SPAM 0.98	NSPAM 0.97 SPAM 0.98	02.620806
5	98.23	NSPAM 0.97 SPAM 0.97	NSPAM 0.96 SPAM 0.98	NSPAM 0.96 SPAM 0.97	02.770994
Avg. (Accuracy)		97.638			

6.5 Random Forest:

Table 10 represents the results and efficiency of the proposed model performance with the Adaboost Classifier. As for the accuracy scale, the model achieved the best performance at all runs. It achieved 98.7, the time taken in the classification process was 02:44.64 seconds which is considered the most running time for all classifiers.

The interpretation of the accuracy results obtained through the training and testing process appears. We notice high accuracy results due to the random forest

building the best tree for the final results with taking all possible labels and decisions.

Random Forests are ideal for improving decision tree performance on binary classification tasks. Random Forests was the name given to the game when it first came out. Freund and Schapire, the creators of the method M1. Because it is used for classification rather than regression, it has lately been referred to as discrete Random Forests. Any machine learning algorithm can benefit from the usage of Random Forests. It is excellent for students who are not as bright as the rest of the class.

On a classification issue, these models reach Accuracy slightly above random chance. Decision trees with one level are the most suitable and commonly used method with Random Forests. These trees are known as decision stumps because they are so short and only have one categorization decision.

Table 11 shows the results with the black images; with the black images, we can see the drop of Accuracy for our model to 98.5 due to the classifier cannot figure out the feature for the black images affecting the results.

Table 10 Results of Random Forest Classifier Without Black Images

Run	Accuracy	Precision	Recall	F-score	Time
1	98.7	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	NSPAM 0.98 SPAM 0.99	02:50.38
2	98.7	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	NSPAM 0.98 SPAM 0.99	02:44.64
3	98.7	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	NSPAM 0.98 SPAM 0.99	02:49.62
4	98.7	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	NSPAM 0.98 SPAM 0.99	02:41.38
5	98.7	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	NSPAM 0.98 SPAM 0.99	02:44.59
Avg. (Accuracy)		98.7			

Table 11 Results of Random Forest Classifier with Black Images

Run	Accuracy	Precision	Recall	F-score	Time
1	98.5	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	NSPAM 0.98 SPAM 0.99	01:56.312556
2	98.5	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	NSPAM 0.98 SPAM 0.99	01:56.779420
3	98.5	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	NSPAM 0.98 SPAM 0.99	01:56.965603
4	98.5	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	NSPAM 0.98 SPAM 0.99	01:56.215759
5	98.5	NSPAM 0.99 SPAM 0.99	NSPAM 0.98 SPAM 0.99	NSPAM 0.98 SPAM 0.99	01:58.431218
Avg. (Accuracy)		98.5			

6.6 CNN:

Convolutional Neural Networks (CNN) is a type of feed-forward neural network that is highly complicated. Because of its great Accuracy, CNNs are utilized for picture categorization and identification. Yann LeCun, a computer scientist, proposed it in the late 1990s after being inspired by the human visual perception of object recognition. The CNN uses a hierarchical model that builds a network in the shape of a funnel and then outputs a fully connected layer where all the neurons are linked, and the output is processed. From Table 12, the results show that CNN had the best results for all classifiers in our proposed model.
Table 13 shows the findings with using the black images in the classifications; with the black images, we can see the drop of Accuracy for our model to 99.048 because the classifier cannot figure out the feature for the black images with affecting the results.

Table 12 Results of CNN Algorithm without Black Images

Run	Accuracy	Precision	Recall	F-score	Time
1	99.52	NSPAM 1.00 SPAM 0.99	NSPAM 0.99 SPAM 1.00	NSPAM 0.99 SPAM 1.00	01.939677
2	99.36	NSPAM 0.99 SPAM 0.99	NSPAM 0.99 SPAM 0.99	NSPAM 0.99 SPAM 0.99	01.835872
3	99.63	NSPAM 0.99 SPAM 1.00	NSPAM 1.00 SPAM 0.99	NSPAM 0.99 SPAM 1.00	01.983839
4	99.52	NSPAM 1.00 SPAM 0.99	NSPAM 0.99 SPAM 1.00	NSPAM 0.99 SPAM 1.00	01.773001
5	99.26	NSPAM 0.99 SPAM 1.00	NSPAM 1.00 SPAM 0.99	NSPAM 0.99 SPAM 0.99	02.123340
Avg. (Accuracy)		99.458			

Table 13 Results of CNN Algorithm with Black Images

Run	Accuracy	Precision	Recall	F-score	Time
1	98.97	NSPAM 0.99 SPAM 0.99	NSPAM 0.99 SPAM 1.00	NSPAM 0.99 SPAM 1.00	10.846783
2	99.05	NSPAM 0.99 SPAM 0.99	NSPAM 0.99 SPAM 0.99	NSPAM 0.99 SPAM 0.99	10.239096
3	98.96	NSPAM 0.99 SPAM 1.00	NSPAM 1.00 SPAM 0.99	NSPAM 0.99 SPAM 1.00	10.239096
4	99.11	NSPAM 1.00 SPAM 0.99	NSPAM 0.99 SPAM 1.00	NSPAM 0.99 SPAM 1.00	10.343774
5	99.15	NSPAM 0.99 SPAM 1.00	NSPAM 1.00 SPAM 0.99	NSPAM 0.99 SPAM 0.99	10.222532
Avg. (Accuracy)		99.048			

6.7 Classifiers Comparison

The comparison will be based on the accuracy rate in total and for each training split. Also, the performance and time will be considered to understand the best and well-performed classifiers that have been implemented in this work.

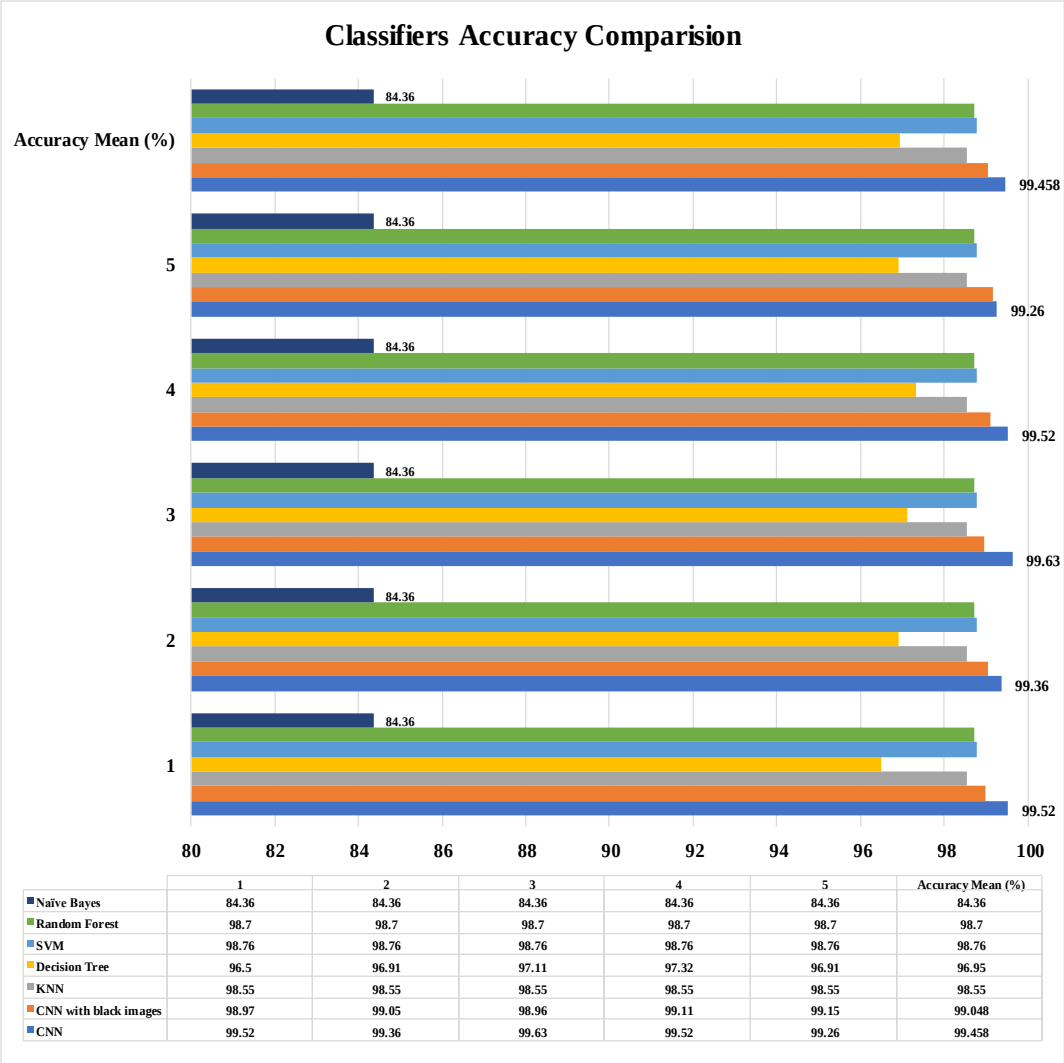
Figure 2 shows a comparison between the classifiers that have been implemented in the proposed method. In Figure 2, the classifier's accuracy rate for each data split, CNN had the better accuracy rate in each split data amount. CNN classifier is the best classifier to classify the images and data on it in this proposed method due to the following:

- Parameters: The number of parameters in a neural network expands fast as the number of layers increases. This can make model training computationally intensive (and sometimes not feasible). Tuning so many settings may be a big undertaking.
- Network: CNNs are feed-forward neural networks that are entirely linked. CNNs are particularly good at decreasing the number of parameters without sacrificing model quality. Images have a high dimensionality, which complements CNN's capabilities. CNN was also created with pictures in mind, but it has also set standards in text processing.

Because the idea of dimensionality reduction matches many factors in an image, CNNs are useful for image classification. This article scratches the surface of CNNs, yet it gives a fundamental understanding of the fact mentioned above.

In Figure 3, the running time and performance for the classifiers in the proposed method have been calculated and monitored CNN; due to the algorithm's complexity and the huge amount of data that has been broken into, it shows the most running time and lowest performance efficiency. The SVM and AdaBoost classifiers show the lowest running time and the most efficient performance for classifying the images.

Despite the results, CNN is still the best classifier. There are no significant changes in the running time and the performance. Also, the classifier always had the best results in all data split amounts in the proposed method.
Table 14. shows the mean, standard deviation, and variance for the Accuracy results without using the black images. The results show a difference in CNN and DT results during the runtimes. The other models maintained the same accuracy results with zero in standard deviation and variance. CNN based on images classification with the VGG-16 show different classification and some FP in some cases, and despite that, CNN achieved the highest Accuracy among all classifiers.



	1	2	3	4	5	Accuracy Mean (%)
Naïve Bayes	84.36	84.36	84.36	84.36	84.36	84.36
Random Forest	98.7	98.7	98.7	98.7	98.7	98.7
SVM	98.76	98.76	98.76	98.76	98.76	98.76
Decision Tree	96.5	96.91	97.11	97.32	96.91	96.95
KNN	98.55	98.55	98.55	98.55	98.55	98.55
CNN with black images	98.97	99.05	98.96	99.11	99.15	99.048
CNN	99.52	99.36	99.63	99.52	99.26	99.458

Figure-2: Classifier's Accuracy Results Based on Data Split

Table 14 K-Fold mean and SD for each classifier accuracy (without black images)

	CNN	KNN	RF	DT	SVM	NB
Mean	99.048	98.55	98.7	96.95	98.76	84.36
Standard deviation	0.075	0	0	0.2714	0	0
Variance	0.0056	0	0	0.07364	0	0

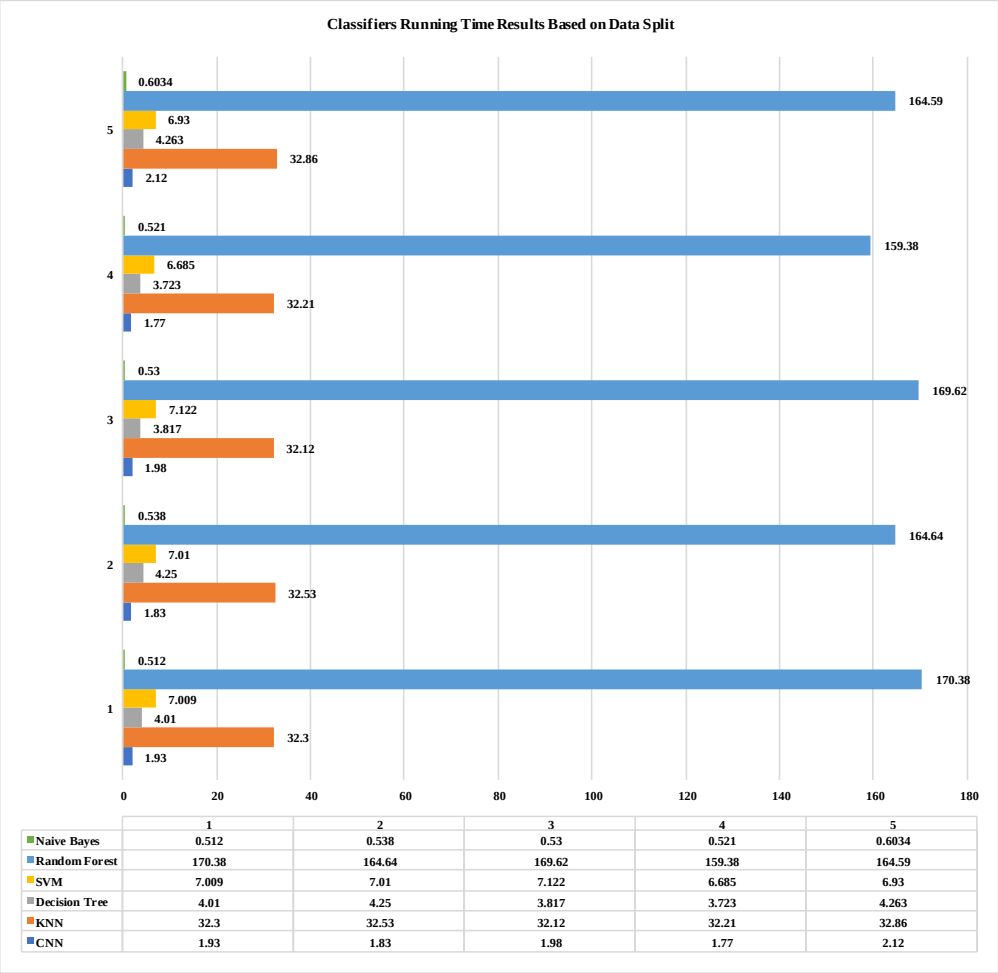


Figure.3: Classifiers Running Time Results Based on Data Split

7 conclusion and future work

This research and proposed model work goals to gain a deeper understanding of classifying images based on extracting their features and then training the models to classify the images into two categories original and spam images. Identifying spam images is one of the inherently tricky problems. In this research work, six machine learning algorithms were tested: CNN, KNN, SVM, NB, RF, and DT. The evaluation metrics were based on Classification Report (ACC, Rcl, Pre, and F1). Datasets have been preprocessed, and the selected images have been augmented for best feature extraction and selection; the training method was

based on splitting the data into 70-30% with Hayperparameters and Kfold=10 to eliminate bias and obtain the best features for both training and testing. CNN showed the best Accuracy among all the classifiers with and without the black images on the Dataset with reasonable running time. RF shows the highest run time among all classifiers, near 2 minutes for each run and while Naive Bayes was the fastest classifier in the proposed method. With a designed system to identify the SPAM and non-SPAM images in real-time using the saved weights, the findings show that our proposed method has detected all images correctly with certainty reaching 100% for each image. In the future work, we will focus on adding and applying additional deep learning techniques to focus on tests as RNN and LSTM.

References

[1] Gao, Y., Yang, M., Zhao, X., Pardo, B., Wu, Y., Pappas, T. N., & Choudhary, A. (2008, March). Image spam hunter. In 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 1765-1768). IEEE.

[2] Harisinghaney, Anirudh, et al. "Text and image-based spam e-mail classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm." 2014 International Conference on Reliability Optimization and Information Technology (ICROIT). IEEE, 2014.

[3] Win, Z. M., & Aye, N. (2014). Detecting image spam based on file properties, histogram and Hough transform. Journal of Advances in Computer Networks, 2(4), 287-292

[4] Kumar, A. D., & KP, S. (2018). Deepimagespam: Deep learning-based image spam detection. arXiv preprint arXiv:1810.03977.

[5] Popovac, M., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (2018, November). Convolutional neural network-based SMS spam detection. In 2018 26th Telecommunications Forum (TELFOR) (pp. 1-4). IEEE.

[6] Jain, G., Sharma, M., & Agarwal, B. (2018). Spam detection on social media using semantic convolutional neural network. International Journal of Knowledge Discovery in Bioinformatics (IJKDB), 8(1), 12-26.

[7] Madisetty, S., & Desarkar, M. S. (2018). A neural network-based ensemble approach for spam detection in Twitter. IEEE Transactions on Computational Social Systems, 5(4), 973-984

[8] Fatichah, C., Lazuardi, W. F., Navastara, D. A., Suciati, N., & Munif, A. (2019). Image spam detection on Instagram using convolutional neural network. In Intelligent and Interactive Computing (pp. 295-303). Springer, Singapore.

[9] Ako, R. T., Upadhyay, A., Withayachumnankul, W., Bhaskaran, M., & Sriram, S. (2020). Dielectrics for Terahertz Metasurfaces: Material Selection and Fabrication Techniques. Advanced Optical Materials, 8(3), 1900750.

[10] Kim, B., Abuadbba, S., & Kim, H. (2020, November). Deep Capture: Image Spam Detection Using Deep Learning and Data Augmentation. In Australasian Conference on Information Security and Privacy (pp. 461-475).

Springer, Cham.

[11] Gao, Y., Yang, M., Zhao, X., Pardo, B., Wu, Y., Pappas, T. N., & Choudhary, A. (2008, March). Image spam hunter. In 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 1765-1768). IEEE.

[12] Sharmin, T., Di Troia, F., Potika, K., & Stamp, M. (2020). Convolutional neural networks for image spam detection. Information Security Journal: A Global Perspective, 29(3), 103-117.

[13] Verma, T. (2017). E-Mail Spam Detection and Classification Using SVM and Feature Extraction.

[14] Yüksel, A. S., Cankaya, S. F., & Üncü, İ. S. (2017). Design of a Machine Learning-Based Predictive Analytics System for Spam Problem.Acta Physica Polonica, A.,132(3).

[15] Younis, Mohammed Chauhan, and Huthaifa Abuhammad. "A hybrid fusion framework to multi-modal biometric identification." Multimedia Tools and Applications 80.17 (2021): 25799-25822.

[16] Abuhammad, Huthaifa, and Richard Everson. "Emotional Faces in the Wild: Feature Descriptors for Emotion Classification." International Conference Image Analysis and Recognition. Springer, Cham, 2018.

Notes on contributors

Nawal Nayl Abuzaid has received a Masters's degree in computer science, from Mutah University, with an excellent grade in 2021. And she has received a school administration diploma, from Yarmouk university, with a grade of excellent, in 2014

Huthiafa Ziad Abuhammad has received a PhD degree from the College of Engineering, Mathematics, and Physical Sciences, University of Exeter, in 2019. He is currently an Assistant Professor with the Department of Data Science and Artificial intelligence Science – Isra University, Jordan. He has been working in machine learning, data science and orifical intelligence applications.