IMAGE SPAM CLASSIFICATION USING DEEP LEARNING AND MACHINE LEARNING

ABIJITH	ANIRUDH BHASKAR	D KARTHIK SIVA SAI	M SAI SUHAS
CB.EN.U4AIE19002	CB.EN.U4AIE19007	CB.EN.U4AIE19020	CB.EN.U4AIE19042

Amrita School of Engineering, Amrita Vishwa Vidyapeetham Coimbatore, Tamil Nadu, India.

Deep Learning for Signal and Image Processing - Final Project Report

Course: 21AIE312(21 – 22(Even)

Class: III-B.Tech-CSE-AI (Semester 6)

Batch: 2019 – 2023

May 2022

ABSTRACT:

Nowadays with the enormous increase of the internet, cyberspace is confronting several threats (like spam emails) from hackers and attackers. Over the time attackers moved from text-based email spam to image spam to escape the text-based spam filters. To resolve this issue researchers came up with diverse ML and DL approaches that use few features like colour, shape. In this particular report we are going to explore 2 Deep Conventional Neural Networks (DCNN) models followed by a few pre - trained ImageNet architectures like Xception , ResNet50 which are trained on 3 different spam datasets.

INTRODUCTION:

With the improvement of technology, people are better connected than ever before. There has been a rapid increase in the usage of the internet amongst the worldwide population, and with that usage comes security threats. With many communication channels available nowadays, hackers have found and exploited many security loopholes and bombard normal users with malicious content that can hurt their personal devices as well as lead to irreparable damage. One of the earliest such techniques used by these hackers was sending spam through the mail. Bots can be created in order to send the same spam mail across millions of users in a split second. Spam through email was initially in a text format, and therefore ML models were easily created in order to find feature words and have been successful in correctly identifying and

filtering emails as spam or not. Nowadays, hackers have moved on from just text to making the spam into an image form. These Spam images contain the malicious words embedded onto it, and hence text-based spam filters cannot identify them. The dangerous part of these images is that users might unknowingly click on the texts present in the image which could redirect them to unsafe and malicious websites causing malware-related problems. DL techniques can be applied for image spam detection. We have used Convolutional Neural Networks (CNN) models for spam image classification.

At first we use two Deep Convolutional Neural Networks (DCNN) models and observe their performance in spam detection across our three datasets. We also applied transfer learning approaches across various pre-trained CNN models like Xception, DenseNet, VGG19 etc. We also observe the performance of various ML classifiers trained on the extracted feature layers from the CNN models.

LITERATURE:

Chao Wang proposed a method that collected the low level features of images making the classification process faster. The SVM model proposed by him also proved to get 95% spam detection from the images. In another instance Ajay Pal Singh had proposed four CNNs alongside the pre-trained VGG19 model to detect spam on the current dataset and an improved dataset. He achieved an accuracy of 95-98.78 % for the various datasets like ISH, Improved Dataset and the Dredze dataset taken in respective permutations. Shikar Seth and Sagar Biswas proposed multi-modal architectures which were a combination of image and text classifiers. The proposed models had a significant difference in classification accuracy than the basic CNN which were 96.87% on feature fusion and 98.11% on learned rule. The models were tested on datasets created by them from scratch using e-mail spam images and random ham images from the internet. In another scenario Ahmad Mahdi Salih proposed a CNN based image spam classification using color models evaluated on a public image dataset. From the proposed models the XYZ color model proved to achieve an accuracy of 98.4%.

OBJECTIVES:

The primary purpose of this work is to classify the given image as spam or ham using various deep learning and machine learning techniques and analyse the results of the same to conclude which technique performs better to do these kinds of tasks.

DATASET DESCRIPTION:

The following are the 3 datasets which are used for this project.

1) Image spam Hunter (ISH):

This dataset consists of 2 sets of images; one is spam images followed by ham images. All the images are in JPG format which is gathered from original emails.

Total number of Spam images - 927

Total number of **unique** spam images – 895

Total number of Ham images - 810

Total number of **unique** Ham images – 810

Spam and Ham Image

2) Improved Dataset:

This is a challenging dataset in work, generated by ingraining spam text in ham images to mislead the existing models.

Total number of Improved Spam images - 1030

Total number of **unique** Improved spam images - 975

Total number of Ham images - 810

Total number of **unique** Ham images - 810

Improved Spam Images

3) Dredze ImageSpam Dataset:

This dataset contains 3 sets of images; Personal Ham(PHam), Personal Spam(PSpam) and Spam Archive(SpamArch). All the images are of various formats like JPG, PNG, and most of them are in GIF format.

Total number of PSpam images - 3298
Total number of **unique** PSpam images - 1266
Total number of PHam images - 2021
Total number of **unique** PHam images - 1519
Total number of SpamArch images - 16028
Total number of **unique** SpamArch images - 3941

METHODOLOGY:

The four major steps in methodology are Preprocessing, Deep Convolutional Networks(DCNN) models, Cost-sensitive and Transfer Learning Models, Statistical Metrics.

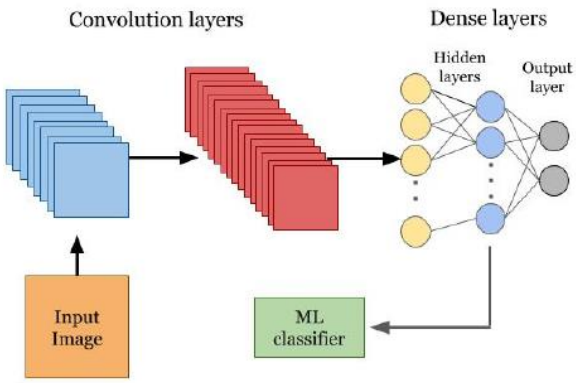
A. Pre-Processing :

As mentioned above in the Dataset description, we use three different benchmark datasets and these datasets will have many duplicate images and improper files such as corrupted files. The first step in pre-processing to exclude the corrupted files and to remove duplicate files we use hashing technique. In this image hashing the algorithm converts the image to a unique hash value. We basically make a list and append these hash values in it, when a duplicate image arrives it checks the list, as that hash value will be already present in the list, it avoids taking that value. At last, all unique images are normalized and resized into essential sizes.This is what happens in pre-processing.

B. Deep Convolutional Networks (DCNN) Models :

In this project, we have designed two DCNN models. The first CNN1 model will have 3 convolutional layers with different filter sizes like 32, 64 and 128. These convolutional layers will be followed by ReLU activation and max pooling layer of size 2. After convolution layers, we use dropout regularization and output is flattened and passed into a Dense layer which contains 128 neurons. Further, this layer is followed by ReLU activation and dropout regularization. In the end, a dense layer of a single neuron is used with a sigmoid activation

function. Features are extracted from the last hidden layer of CNN1 model and passed on to many machie learning classifiers as shown in below figure.



WorkFlow

In this project, we use Machine learning (ML) classifiers such as Logistic Regression, Random Forest, Decision Tree, AdaBoost, K Nearest Neighbour (KNN) , Naive Bayes, Linear SVM, Reduced SVM.

Moreover, the second CNN2 model also has 3 convolutional layers of filter sizes 128, 128, and 256. These convolutional layers will be promptly followed by the ReLU activation and max pooling of pool sizes 4 and 3. Rest of the process is similar to what we did in CNN1 model. The major difference between these two CNN ‘s is, we update the class weights in second model using first model weights using a cost-sensitive learning technique.

C. Cost-Sensitive Learning and Transfer Learning Models

The class imbalanced datasets will occur in many real-world applications where the class distributions of data are highly imbalanced. Cost-sensitive learning is a common approach to solve this problem. In this approach, balanced class weights are calculated and passed to the model while the fitting process so that the model will penalize the prediction mistakes of minority class proportionally based on how underrepresented it is.

In this model we have also used transfer learning using pre-trained models such as VGG19, DenseNet201, ResNet50, Xception. Basically what we do in this transfer learning is the last dense layer is excluded and remaining layers are kept frozen to assist transfer learning.

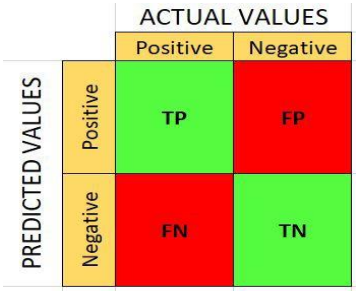
Model	Number of parameters
CNN	Trainable: 25,013,569
CS-CNN	Trainable: 5,984,514
VGG19	Trainable: 10,488,065 Non-trainable: 10,585,152
DenseNet201	Trainable: 4,215,041 Non-trainable: 18,039,360
Xception	Trainable: 13,342,241 Non-trainable: 14,073,096
ResNet50	Trainable: 7,078,657 Non-trainable: 23,062,912

TRAINABLE AND NON-TRAINABLE PARAMETERS OF THE PROPOSED MODELS

From the above table, we can observe that most of the image net models have lesser number of trainable parameters, because we are training only the last layer and comparatively Non-trainable parameters will more.

D. Statistical Metrics:

After making the model we need to know how well is performing. so we consider few metrics to measure model performance such as accuracy, precision, recall, f1-score, False Positive Rate(FPR), False Negative Rate(FNR). These metrics are calculated using terms such as True Positive (TP), False Negative (FN), False Positive (FP), True Negative (TN) that are obtained from the confusion matrix.



TP indicates the number of spam images that are accurately predicted as spam.

FN indicates the number of spam images that are wrongly predicted as normal

FP indicates the number of normal images that are wrongly predicted as spam.

TN indicates the number of normal images that are accurately predicted as spam.

EXPERIMENTAL RESULTS:

PERFORMANCE OF CNN1, CS-CNN1 AND HYBRID MODELS

Model	Accuracy	Precision	Recall	F1-score	TP	FN	FP	TN	FNR	FPR
Image spam hunter dataset										
CNN	0.978	0.981	0.977	0.979	238	6	5	263	0.024	0.018
CS - CNN	0.951	0.917	0.996	0.955	219	1	24	268	0.004	0.082
CNN - LR	0.978	0.978	0.978	0.978	238	6	5	263	0.024	0.018
CS- CNN -LR	0.982	0.98	0.98	0.98	239	6	4	263	0.024	0.014
CNN -RF	0.972	0.972	0.972	0.972	236	7	7	262	0.028	0.026
CS-CNN-RF	0.982	0.982	0.982	0.982	241	6	2	262	0.024	0.007
CNN-DT	0.976	0.976	0.976	0.976	232	3	11	262	0.012	0.04
CS-CNN-DT	0.982	0.982	0.982	0.982	238	4	5	265	0.016	0.018
CNN -KNN	0.972	0.973	0.972	0.972	265	5	6	266	0.018	0.022
CS-CNN-KNN	0.974	0.973	0.974	0.974	265	5	6	266	0.018	0.022
CNN-GNB	0.962	0.963	0.962	0.962	228	4	15	265	0.017	0.053
CS-CNN-GNB	0.962	0.963	0.962	0.962	228	4	15	265	0.017	0.053
CNN-AB	0.968	0.968	0.968	0.968	232	5	11	264	0.021	0.04
CS-CNN-AB	0.971	0.971	0.971	0.971	234	5	9	265	0.02	0.032
CNN-LSVM	0.972	0.973	0.972	0.972	232	3	11	266	0.012	0.039
CS-CNN-LSVM	0.978	0.978	0.978	0.978	238	6	5	263	0.024	0.018
CNN -RSVM	0.964	0.965	0.964	0.964	232	3	11	266	0.012	0.039
CS-CNN-RSVM	0.967	0.967	0.967	0.967	234	3	10	266	0.012	0.036
Image spam hunter Ham + Improved spam dataset										
CNN	0.994	1	0.989	0.994	243	3	0	290	0.012	0
CS-CNN	0.996	1	0.993	0.996	243	2	0	291	0.008	0
CNN - LR	0.994	0.994	0.994	0.994	243	3	0	290	0.012	0
CS-CNN - LR	0.996	0.996	0.996	0.996	243	2	0	290	0.008	0
CSS-RF	0.986	0.996	0.996	0.996	242	1	1	292	0.004	0.003
CS-CNN-RF	0.996	0.996	0.996	0.996	242	1	1	292	0.004	0.003
CNN-DT	0.986	0.986	0.986	0.986	242	6	1	287	0.24	0.003
CS-CNN-DT	0.994	0.994	0.994	0.994	243	3	1	290	0.012	0.003
CNN- KNN	0.996	0.996	0.996	0.996	243	2	0	291	0.008	0
CS-CNN-KNN	0.996	0.996	0.996	0.996	243	2	0	291	0.008	0
CNN-GNB	0.988	0.988	0.988	0.988	237	0	6	293	0	0.02
CS-CNN-GNB	0.99	0.99	0.99	0.99	240	2	3	291	0.008	0.01
CNN-AB	0.994	0.994	0.994	0.994	241	1	2	292	0.004	0.006
CS-CNN-AB	0.994	0.994	0.994	0.994	241	1	2	292	0.004	0.006
CNN-LSVM	0.994	0.994	0.994	0.994	243	3	0	290	0.012	0
CS-CNN-LSVM	0.996	0.996	0.996	0.996	243	2	0	291	0.008	0
CNN-RSVM	0.996	0.996	0.996	0.996	243	2	0	291	0.008	0
CS-CNN-RSVM	0.996	0.996	0.996	0.996	243	2	0	291	0.008	0

Model	Accuracy	Precision	Recall	F1-score	TP	FN	FP	TN	FNR	FPR
Dredze PSpam and PHam										
CNN	0.965	0.98	0.942	0.961	449	22	7	358	0.046	0.019
CS-CNN	0.964	0.96	0.96	0.96	441	15	15	365	0.032	0.039
Dredze SpamArch and PHam										
CNN	0.944	0.964	0.963	0.963	402	56	54	1469	0.122	0.035
CS-CNN	0.942	0.97	0.954	0.962	411	69	45	1456	0.143	0.029
Dredze PSpam, PHam and SpamArch										
CNN	0.902	0.91	0.902	0.906	404	52	92	1470	0.114	0.058
CS-CNN	0.894	0.904	0.894	0.899	381	75	79	1483	0.164	0.05

PERFORMANCE OF PRE-TRAINED CNN ARCHITECTURES

Model	Accuracy	Precision	Recall	F1-score	TP	FN	FP	TN	FNR	FPR
DenseNet201	0.988	0.992	0.985	0.988	241	4	2	265	0.016	0.007
Xception	0.98	0.974	0.988	0.981	236	3	7	266	0.012	0.025
ResNet50	0.87	0.8	1	0.89	177	0	66	269	0	0.197
VGG19	0.988	0.988	0.988	0.988	240	3	3	266	0.012	0.011

OBSERVATIONS:

After a comparison of the models we have concluded that the CS CNN (Cost Sensitive) model is better due to the consideration of class weights. Although the difference between the accuracy of CS CNN and CNN may not be a lot, we have experimentally proven that CS CNN provides a better accuracy.

We have also used Transfer Learning Models and from the experimentally tested models we have concluded that the DenseNet201and VGG19 models are more precise and provide better accuracy than the ResNet50 and Xception models.

CONCLUSION:

Using three separate datasets, the effectiveness of one Deep Convolutional Neural Networks and hybrid models for image spam categorization is investigated. The impacts of cost-sensitive learning are investigated using balanced class weights, and transfer learning is investigated using multiple pre-trained CNN architectures such as VGG19, Xception, and others. Some of the proposed models outperformed earlier studies, while others failed to do so. It may be concluded that in order to create a better picture spam classifier, extra information such as metadata should be included in the model training. The impacts of adversarial samples, which

are capable of deceiving the model into making an inaccurate prediction, can be examined in future publications.

REFERENCES:

1) **Image spam classification based on low-level image features** - Chao Wang; Fengli Zhang; Fagen Li; Available at : https://ieeexplore.ieee.org/abstract/document/5581998/
2) **Image spam classification using Deep learning** - Ajay Pal Singh ; Available at : https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1634&context=etd_projects
3) **Color Model Based Convolutional Neural Network for Image Spam Classification** - Ahmad Mahdi Salih; Available at : https://mail.anjs.edu.iq/index.php/anjs/article/view/2313/1820
4) **Multimodal Spam Classification Using Deep Learning Techniques** - Seth and Biswas ; Available at : https://ieeexplore.ieee.org/abstract/document/8334769