

Email Image Spam Classification based on ResNet Convolutional Neural Network

Vít Listík¹^a, Jan Šedivý²^b and Václav Hlaváč²^c

¹*Czech Technical University in Prague Faculty of Electrical Engineering, Department of Cybernetics, Prague 6, Technická 2, Czech Republic*

²*Czech Institute of Informatics, Robotics and Cybernetics, Prague 6, Jugoslávských partyzánů 1580/3, Czech Republic*

Keywords: Spam, Email, ResNet, Image, Classification, Convolutional Neural Network.

Abstract: The problem with email image spam classification is known from the year 2005. There are several approaches to this task. Lately, those approaches use convolutional neural networks (CNN). We propose a novel approach to the image spam classification task. Our approach is based on CNN and transfer learning, namely Resnet v1 used for semantic feature extraction and one layer Feedforward Neural Network for classification. We have shown that this approach can achieve state-of-the-art performance on publicly available datasets. 99% F1-score on two datasets (Dredze et al., 2007), Princeton and 96% F1-score on the combination of these datasets. Due to the availability of GPUs, this approach may be used for just-in-time classification in anti-spam systems handling huge amounts of emails. We have observed also that mentioned publicly available datasets are no longer representative. We overcame this limitation by using a much richer dataset from a one-week long real traffic of the freemail provider Email.cz. The training data annotation was created by user labeling of the emails. The image spam (and image ham even more) tackles privacy issues. We overcame it by publishing extracted feature vectors with associated classes (instead of images itself). This data does not violate privacy issues. We have published Email.cz image spam dataset v1 via the AcademicTorrents platform and propose a system, which achieves up to 96% F1-score with presented model architecture on this novel dataset. Providing our dataset to the community may help others with solving similar tasks.


1 INTRODUCTION


Spam emails (unconsolidated bulk messages) costs email providers and users a huge amount of time and money. Anti-spam systems are trying to lower those losses by separating the email traffic to wanted and unwanted (commonly called ham and spam). Anti-spam techniques have to evolve because methods used by spammers are evolving too. One of the examples of a relatively recent technique is the image spam appearing in email communication (email image spam - referred also as image spam throughout this work)


The image spam problem in emails was specified in (Aradhye et al., 2005), (Wu et al., 2005). The problem was defined as an email content obfuscation method for the anti-spam filters because they did not use information from images attached to emails. At that time, it was hard to process information contained

in images mainly because of the processing power and insufficient algorithms. The problem of image spam in its original form is not that urgent as before because anti-spam solutions are now able to detect it¹, therefore making it less efficient for attackers.

However, it keeps being an interesting research domain because the task is getting harder (Carpinteiro et al., 2017), (Shang and Zhang, 2016). The reason why it is getting harder is that emails containing images are now much more common for both legitimate emails and unfortunately for spam emails too. Another reason is that the available processing power is now much bigger. Consequently, more complex techniques may be used on both sides of the barricade. The final reason is that most of the email providers are not classifying only to spam and ham classes anymore. They use more classes, i.e. Newsletter or Forums. Those conditions are suggesting that image information may be helpful for those emerging tasks.

^a  <https://orcid.org/0000-0002-1907-6334>

^b  <https://orcid.org/0000-0003-0626-2303>

^c  <https://orcid.org/0000-0002-8472-3147>

¹<https://wiki.apache.org/spamassassin/FuzzyOcrPlugin>

1.1 Public Datasets

As we have stated above, characteristics of the image spam and email communication have changed significantly. However, the only commonly used publicly available dataset is over a decade old (Dredze et al., 2007). The biggest problem for publishing the image spam dataset is the private nature of email contents. Other datasets used in publications are listed in Sec. 3.1. Those datasets are commonly not available. It is not because they were not published, but because the publication medium (most commonly server of the publisher) is no longer available.

2 THE APPROACHED TASK

In this contribution, we approach the email image spam classification. The image spam may hide obfuscated text or URL in images instead of plain-text (which is common for regular spam). Such harmful images may contain inappropriate content, unsolicited marketing campaigns, and text. The format of HTML emails containing only embedded images and no text is very common for newsletters², which may be either legitimate or unsolicited. When those emails do not contain any other information other than the images and traffic information (headers, sender IP), we suppose that the information extracted from images may improve anti-spam solutions significantly.

Spam emails make 45% of email traffic globally. At Email.cz, it was measured that 25% of emails are spam. This is caused by the fact that Email.cz classifies newsletter emails into a separate class. Spam and newsletter categories combined constitute up to 75% of the traffic. It was also measured at Email.cz that 75% of emails contain images embedded in HTML and 13% of emails contain image attachments. This means that 88% of emails contain at least one image.

Measurements for all the categories are shown in Table 1. These measurements demonstrate that spam and newsletter emails contain even more images than personal emails.

2.1 Dataset

Due to the lack of representable dataset (problems with datasets are described in Sec. 1.1 and available datasets are described in Sec. 3.1) for this task we decided to create a novel dataset as part of this work.

We established the following requirements for the dataset:

²Emails containing advertisements commonly sent in bulks

Table 1: Statistics of emails containing images measured at Email.cz. Image means that email contained at least one image. HTML means images linked from HTML content of the email. Attach means images in attachments. News means newsletter. Categories are based on currently used anti-spam filter at Email.cz.

	All	Spam	News	Inbox
Emails	100.00%	25.10%	52.59%	22.60%
Image	88.15%	95.81%	98.99%	54.37%
HTML	75.39%	65.46%	95.65%	39.17%
Attach	12.76%	30.35%	3.33%	15.21%

- The resulting dataset should be recent and based on a real email traffic
- Images have to be labeled into following classes {spam, ham, advertisement}
- The data have to be anonymized because of their private nature
- It should contain enough samples for training Neural Networks (NN) or other statistical models
- Reliable platform for the publication

3 RELATED WORK

In this section, we describe publicly available datasets and commonly used methods for email image spam classification.

3.1 Publicly Available Datasets

Based on (Biggio et al., 2011), which is an overview of used datasets and methods until 2011, the commonly used and still available dataset is (Dredze et al., 2007). The other commonly used available dataset is (Cormack and Lynam, 2005). A combination of more public datasets is also used. A different approach is to use a custom dataset (not public).

We studied approaches that appeared after 2011. They follow the pattern described in (Biggio et al., 2011). Most of the approaches use custom datasets and (Dredze et al., 2007).

Dredze 2007

(Dredze et al., 2007) is the most commonly used publicly available dataset for image spam classification. This dataset is unique because it contains both spam and ham samples. Dredze 2007 dataset is separated into three parts described in Table 2.

This corpus was created by image extraction from emails. Only images attached to emails were used.

Table 2: Numbers based on (Dredze et al., 2007) in which the dataset was published. Those numbers are correct, but some of the images are damaged and cannot be used for training.

Corpus name	# of images
Personal Ham	2550
Personal Spam	3239
SpamArchive Spam	9503

Images were detected based on the file extension. SpamArchive is still available³. One part (called public) contains spam only. The authors also collected personal emails from 10 email accounts from 10 different domains over one month and extracted both spam and ham images. This part is called private or personal.

Princeton Spam Image Benchmark

Princeton Spam Image Benchmark was published in 2007. It contains 1071 images in 178 groups. It is accessible⁴.

The first issue with Dredze 2007 and Princeton datasets is that they contain only a few thousands of samples. Neural networks need more samples for training. We are not training the CNN because it comes already pre-trained on a huge dataset. Still, neural nets perform better on more samples.

TREC

TREC dataset⁵ is commonly used for benchmarking. It was also created in 2007 and contains only emails (Cormack and Lynam, 2005). Images need to be extracted from the emails. Because this dataset contains around 7,000 images and the extracted version is not publicly available we decided not to use it. It solves neither the issue with old data nor dataset size issue.

3.2 Statistical Models

Email image spam task has been approached many times by the community (Biggio et al., 2011). Between the years 2005 and 2015, the most common approach to this task was to use low-level image feature extraction and SVM classifier. Lately, it has been more common to use neural networks for this

³The site was available at <http://www.spamarchive.org/> earlier. It may be found at <http://untroubled.org/spam/> now.

⁴<http://www.cs.princeton.edu/cass/spam/>

⁵Accessible at <http://trec.nist.gov/data/spam.html> when this paper was written.

task (Carpinteiro et al., 2017; Shang and Zhang, 2016).

Convolutional neural networks (CNN) and transfer learning have been used in computer vision and decision-making tasks recently. In this work, we use CNN for classifying image spam in emails based on (Shang and Zhang, 2016). They have used a non-public image spam dataset. Their approach is to classify images extracted from emails to seven classes using CNN and SVM. They do not mention how they are using the result of the classification for an anti-spam solution.

The closest work to ours is (Shang and Zhang, 2016) because of the use of CNN. The difference is that we are using a pre-trained network and they are training the network from scratch. The other difference is that our dataset is publicly available and our classes may be used directly in the anti-spam system.

The second closest work to ours is (Carpinteiro et al., 2017) which compares results on 3 publicly available datasets and one of the models they used is a neural network.

4 OUR APPROACH

We proposed a model architecture for this task based on CNN and implemented it. This architecture has shown promising results in image classification lately. We have tested it on two above mentioned publicly available datasets and also on Email.cz image spam dataset v1 (described in Sec. 4.2). In the following sections, we describe the proposed architecture and the dataset we published as part of this work.

4.1 Model Architecture

We propose using CNN for the image spam classification task. Namely a pre-trained ResNet v1 for semantic feature extraction (He et al., 2016). We are using the extracted features as an input to a single hidden layer fully connected neural network. We tested this approach on publicly available datasets. Unsatisfied with its ability to benchmark the image spam classification task, we decided to use the CNN model for the creation of our dataset.

Transfer learning (sometimes called fine-tuning) is a process of training a model trained for another task previously. This technique is used with CNN commonly. Sometimes this process is performed on one model. The CNN part of the model is frozen (the learning rate for that part is very small or zero) and the fully connected layers are trained with new data. This method is mostly used when the dataset size or the

computational power is insufficient for training the entire CNN. We split the network into two separate parts. It made it possible to store the results of the CNN evaluation for further evaluation and learning. Otherwise, the process stayed the same.

We have used ResNet v1 for feature extraction (He et al., 2016). More precisely we are using a fully connected layer of the ResNet, which consists of 2048 output neurons as the output of the feature-extractor. Consequently, the used feature vector contains 2048 float entries. The first reason to use the ResNet model was that it achieved state-of-the-art results on the ImageNet classification task (Russakovsky et al., 2015). The second reason is that it was tested in-house in Email.cz for a similar task and achieved the best results. That resulted in a prepared and tested implementation and acceptance at Email.cz.

Resnet is a novel architecture designed for the ImageNet classification task. It consists of 152 layers. It overcomes the problem of vanishing gradients by using residual connections (skip connections over a group of layers). This architecture won ILSVRC 2015 classification challenge (He et al., 2016).

Our proposed model using features extracted by ResNet v1 consists of one fully connected hidden layer with 2048 neurons and ReLu activations. For the output layer, we are using Softmax activation. The implementation of this model is described in Sec. 5.2.

4.2 Email.cz Image Spam Dataset v1

As stated in Sec. 2.1 we decided to create our own dataset and to publish it to the scientific community. We did it in cooperation with Email.cz, which is the largest freemail provider in the Czech Republic. We publish the data in an anonymized format having minimal information loss in mind. Our proposed format is to publish features extracted from the pre-trained model instead of the images itself. We named the dataset Email.cz image spam dataset v1.

At Email.cz, each image attachment is sent for evaluation. Embedded HTML images are identified by the URL. Not all images may be downloaded because some images are used for tracking the open rate. The image is identified with a composite key containing, e.g. the email and image identifiers. Then the number of occurrences of this composite key is calculated and when it hits a defined threshold the image is sent for analysis and cache the result. The results of the images evaluations are stored for future processing. One part of the image evaluation is also the extraction of a ResNet feature vector.

Email.cz provides its users a possibility to express misclassified emails, which is a standard in this in-

dustry. User reactions are stored which allows us to map the reactions to all images contained in the email. Apache Spark is used to connect email reactions to the images contained in those emails (Zaharia et al., 2016). The result is the feature vector of the analyzed image and all corresponding user reactions.

In our case, the anti-spam system classifies emails into three classes.

1. Ham emails - mostly personal communication and should end up in the user's inbox.
2. Newsletter emails - mostly messages containing advertisements.
3. Spam emails - unsolicited messages.

When the classification is incorrect, the user may change the label for which results in one of the actions.

- USER_MARK_SPAM: From anywhere to spam.
- USER_UNMARK_SPAM: From spam to anywhere else.

The newsletters are treated analogously.

Email corporas are difficult to build and publish due to the private nature of email communications. The same applies to image datasets. We suggest a new approach to this task, namely publishing only an extracted feature vectors, which are representative enough for the classification task (as shown in Sec. 7). It shouldn't be possible to reconstruct the personal data from it (Listik, 2018).

We are following the approach of (Dredze et al., 2007) for attaching labels to images contained in emails. If the email is classified spam, all images inside are classified as spam too, same apply to all the labels.

5 IMPLEMENTATION

We implemented the proposed algorithm and tested the algorithm on the publicly available datasets and our newly gathered dataset. In the following section, we describe the process of data gathering and model training.

5.1 Data Gathering

For feature extraction from the public datasets, we are using the open-source implementation of ResNet v1 without any modifications. In the following sections, we focus on the details of the creation of our published dataset called Email.cz image spam dataset v1.

As described in Sec. 4.2 only some images are sent for analysis. When the image is chosen to be

analyzed, it is put into the queue. A server (an instance of a ResNet model) takes it out from the queue and analyzes it. We use open-source implementation⁶ of ResNet using Tensorflow for feature extraction (Abadi et al., 2015; He et al., 2016). The result is stored in two locations. First, it is stored in the key-value store as a cache for the analysis of emails containing the same image. Second, it is stored in the Hadoop Distributed File System (HDFS) for future use. We also store user reactions including information which images were contained in each email on HDFS too. Hence we may connect user reactions to images with a Spark job. This cannot be done without storing the image analysis data because the user reaction comes after the email delivery (after the analysis).

The Spark job filters out all emails without images. First, it finds all reactions to the email. Second, it also filters out emails without any reactions, groups them by users who reacted to them. Third, finds all the images contained in emails. Fourth and finally, it extracts the image vector from the image representation and separates those vectors to single records with the corresponding reactions (Zaharia et al., 2016). The output of this job is the dataset described and used in this paper.

Dataset Format

The format of the data is JSON⁷ structure stored separately on each line which supports efficient loading in many programming languages.

Dataset Name

We decided to call this dataset Email.cz image spam dataset v1.

Data Time Span

The images were gathered in the period June 12-18, 2017.

Dataset Download

URL of the dataset may be found at Github.com⁸.

⁶https://github.com/tensorflow/tensorflow/blob/master/tensorflow/contrib/slim/python/slim/nets/resnet_v1.py

⁷Described in RFC 7159 <https://tools.ietf.org/html/rfc7159>

⁸<https://github.com/tivvit/image-spam-cnn-classifier/blob/master/dataset.yaml> - We chose this way because repository may be updated but the URL in the paper cannot. The URL also contains the actual process of how to download the dataset.

Our dataset is published via AcademicTorrents⁹. We chose it because of the distributed fashion of the peer2peer network. The main reason was that we encountered a lot of problems with missing or moved datasets (as described in Sec. 1.1) for this task on the Internet.

Format Specification

[[[user_reactions], [user_reactions], ...], [resnet_vector]], where

- Resnet_vector is 2048 long vector of floats.
- User_reactions are all reactions for one image from one user defined in user_reaction.

User_reaction is one of:

- USER_MESSAGE_MARK_SPAM
- USER_MESSAGE_UNMARK_SPAM
- USER_MESSAGE_MARK_AD
- USER_MESSAGE_MARK_NONAD

5.2 Statistical Model

We are using Keras with Tensorflow backend for the implementation of model described in Sec. 4.1 (Chollet et al., 2015; Abadi et al., 2015). This implementation is open-source and may be found at Github.com¹⁰. All the results in this paper may be reproduced with that repository in favor of reproducible research.

6 TEST DATASET PROPERTIES

Our dataset consists of 778,768 lines, where each line represents one image with its reactions. The dataset is distributed in gzip format and its size is 2.5 GB (16 GB after extraction). For this number of images, we have collected 10,623,635 reactions. That is 13.64 reactions per image on average. The reaction distribution is shown in Table 3.

The category distribution is shown in Table 4.

We have implemented filters that helps the model to use only consistent reactions. Implementations of those filters are also included in the repository referenced in Sec. 5.2.

First, we have filtered inconsistent reactions of a single user to one image. Those reactions are probably misclassifications. They make only 2.34% of the data which is not significant. It simplifies the reaction structure to a flat array. We call remaining reactions

⁹<http://academictorrents.com/>

¹⁰<https://github.com/tivvit/image-spam-cnn-classifier>

Table 3: The reaction distribution. Correlation between minimal number of reactions and number of sample images.

Reactions to one image	#image samples
>0	778768
>1	422541
>2	318234
>3	263478
>4	228131
>5	201003
>6	182244
>7	167866
>8	155431
>9	143242

Table 4: Distribution of the classes, based on user reaction distribution.

Category	Samples
Spam	52.92%
Advertisements	13.94%
Non-spam	6.43%
Non-ad	26.71%

valid and use them in all the following tests and measurements.

We have also merged unmark user reactions (USER_MESSAGE_UNMARK_SPAM and USER_MESSAGE_MARK_NONAD) to one class because we do not need to use the information where the email was delivered before, but only the information where it belongs to. We may suppose it belongs to the inbox, which is the class name for this merged group. This also simplifies the usage of classification result in the anti-spam filter.

We are merging all user's reactions to one reaction because reactions are very noisy. When the reaction is inconsistent the image is not used because the image itself probably does not contain any information which may be used for the classification (e.g. emoji).

We have defined reaction consistency as

$$\frac{\#cr}{\#r} * 100, \quad (1)$$

where #cr is the number of reactions for most common reaction group for a given image and #r is the number of all reactions for one image.

The average consistency of all user reactions for one image is following $86.21\% \pm 20.96$. When we consider only images with more than one reaction we get to $74.71\% \pm 22.69$.

Table 5: Performance on publicly available datasets. Where Dr means Dredze and PR spam means Princeton spam.

Dataset	Precision	Recall	F1-score
Dr personal	99%	99%	99%
Dr public	95%	95%	95%
Dr combined	96%	96%	96%
Dr ham, PR spam	99%	99%	99%
All combined	96%	97%	96%

Table 6: Shows the number of samples used for testing (15%) for 100,000 sample subset for a different number of reactions (rows) and consistencies (columns).

	0	0.5	0.6	0.75
>0	14,897	13,840	12,177	10,894
>1	8,276	7,219	5,556	4,273
>2	6,299	5,242	4,280	2,997
>3	5,204	4,216	3,275	2,424
>4	4,504	3,534	2,720	1,888
>5	3,953	3,039	2,255	1,523
>7	3,382	2,513	1,862	1,246
>10	2,663	1,877	1,387	922
>20	1,779	1,192	839	544

7 EXPERIMENTAL RESULTS

All models were trained for 80 epochs with weighted classes. We are using a 75:25 train/test split for public datasets and 85:15 train/test split for our dataset. For all the models we are using Adam optimizer (Kingma and Ba, 2014). For other details please consult Sec. 5.

7.1 Public Datasets

In Table 5, we present the performance of our solution on public datasets.

7.2 Our Dataset

In Table 6, we present how minimal consistency and the minimal number of reactions correspond to the number of samples.

Table 7 shows the results of our architecture for various consistencies and minimal reaction counts. Classifiers were trained on a subset of our dataset (100,000 samples). We can see that when we filter out the reactions supported by more users and those reactions are consistent we will get better results.

Table 8 shows the performance on our dataset for some chosen consistencies and sample counts.

Table 7: Average F1-scores for test set (15%) of 100,000 sample subset for different number of reactions (rows) and consistencies (columns). Bold records were tested further.

	0	0.5	0.6	0.75
>0	0.88	0.89	0.88	0.89
>1	0.9	0.89	0.93	0.94
>2	0.91	0.91	0.91	0.94
>3	0.91	0.92	0.92	0.93
>4	0.9	0.91	0.93	0.94
>5	0.92	0.92	0.93	0.94
>7	0.92	0.93	0.93	0.95
>10	0.92	0.93	0.94	0.95
>20	0.93	0.94	0.96	0.97

Table 8: Model F1-scores for chosen consistencies and min reaction counts.

Min reactions	consistency	samples	F1-score
0	0	649815	87%
3	0.6	140522	93%
5	0.6	98551	93%
10	0.6	59828	95%
20	0.6	34933	96%
3	0.75	102837	93%
5	0.75	66220	95%
10	0.75	39690	95%
20	0.75	22245	96%

8 CONCLUSIONS AND FUTURE WORK

The proposed CNN architecture for the email image spam classification task can achieve state-of-the-art performance on publicly available datasets. 99% F1-score on (Dredze et al., 2007) and Princeton datasets and 96% F1-score on combination of the datasets. It also achieves up to 96% F1-score on the presented Email.cz image spam v1 dataset.

Email.cz image spam v1 dataset is published as part of this work. This dataset focuses on being recent and it is based on real email traffic. Due to this fact, the data have to be anonymized which is done by publishing only features extracted from the images. Those features are extracted by CNN (ResNet v1). The dataset is published via Academic Torrents platform which is distributed in its nature, that should ensure that the data will be available for others in the future. We were also considering the sufficiency of the anonymization and concluded, that it is maybe possible to partially reconstruct the image data. However, it would be computationally very expensive and the level of detail that is needed for recognizing personal

information is already lost in the feature vector (Lis-tik, 2018).

For future work, we want to gather a dataset in a longer time range, which will contain also images correctly classified by the current anti-spam solution. Thus it will lead to a much bigger dataset. Our other suggestion is to use a more complex model architecture or a more sophisticated reaction filtering technique for higher performance.

ACKNOWLEDGEMENTS

We want to thank Seznam.cz company (Email.cz owner) for providing us the data for the dataset creation, computational power and the time of the Email.cz team.

REFERENCES

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- Aradhye, H. B., Myers, G. K., and Herson, J. A. (2005). Image analysis for efficient categorization of image-based spam e-mail. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 914–918. IEEE.
- Biggio, B., Fumera, G., Pillai, I., and Roli, F. (2011). A survey and experimental evaluation of image spam filtering techniques. *Pattern Recognition Letters*, 32(10):1436–1446.
- Carpinteiro, O. A., Sanches, B. C., and Moreira, E. M. (2017). Detecting image spam with an artificial neural model. *International Journal of Computer Science and Information Security*, 15(1):296.
- Chollet, F. et al. (2015). Keras. <https://github.com/fchollet/keras>.
- Cormack, G. V. and Lynam, T. R. (2005). Trec 2005 spam track overview. In *TREC*, pages 500–274.
- Dredze, M., Gevaryahu, R., and Elias-Bachrach, A. (2007). Learning fast classifiers for image spam. In *CEAS*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

- Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Listik (2018). Image reconstruction from resnet semantic feature vector. In *Poster*.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.
- Shang, E.-X. and Zhang, H.-G. (2016). Image spam classification based on convolutional neural network. In *Machine Learning and Cybernetics (ICMLC), 2016 International Conference on*, pages 398–403. IEEE.
- Wu, C.-T., Cheng, K.-T., Zhu, Q., and Wu, Y.-L. (2005). Using visual features for anti-spam filtering. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III–509. IEEE.
- Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Meng, X., Rosen, J., Venkataraman, S., Franklin, M. J., et al. (2016). Apache spark: A unified engine for big data processing. *Communications of the ACM*, 59(11):56–65.

