i-ETC: ISEL Academic Journal of Electronics,
Telecommunications and Computers
Vol. 6 , n. 1 (2020) ID-3

A Deep Learning Approach to Identify Not Suitable for Work Images

Daniel Bicho[a,b] Artur Ferreira[a,c] Nuno Datia[a,d]
[a]ISEL, Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa
[b]Arquivo.pt
[c]Instituto de Telecomunicações
[d]NovaLincs, FCT, Universidade Nova de Lisboa
daniel.bicho@gmail.com artur.ferreira@isel.pt nuno.datia@isel.pt

Abstract— **Web Archiving (WA) deals with the preservation of portions of the World Wide Web (WWW) allowing their availability for the future. Arquivo.pt is a WA initiative holding a huge amount of content, including image files. However, some of these images contain nudity and pornography, that can be offensive for the users, and thus being Not Suitable For Work (NSFW). This work proposes a solution to classify NSFW images found at Arquivo.pt, with deep neural network approaches. A large dataset of images is built using Arquivo.pt data and two pre-trained neural network models, namely ResNet and SqueezeNet, are evaluated and improved for the NSFW classification task, using the dataset. The evaluation of these models reported an accuracy of 93% and 72%, respectively. After a fine tuning stage, the accuracy of these models improved to 94% and 89%, respectively. The proposed solution is integrated into the Arquivo.pt Image Search System, available at `https://arquivo.pt/images.jsp`.**

Keywords: Deep Learning; Deep Neural Networks; Image Classification; Not Suitable for Work Images; ResNet; SqueezeNet.

I. INTRODUCTION

Web Archiving (WA) is a research field that addresses the problem of collecting portions of the World Wide Web (WWW) to ensure that information is preserved in an archive for researchers, historians, and the general public. Web Archives commonly use Web crawlers, such as Heritrix [1], to collect this information. They automate the process of harvesting Web pages and preserving their contents. These contents include many resource types, such as Hyper-Text Markup Language (HTML) pages, cascading style sheets, JavaScript files, images, and videos, but also metadata related to these resources, resource mime-types, and content length.

The choice of the contents to be preserved is a hard challenge, since it is not possible to assure enough storage space for all the contents and the amount of available data on the Web keeps growing. There are several WA initiatives to preserve the WWW contents. Some Web Archives have narrow scopes, preserving only specific kinds of pages, such as institutional Web pages, as for instance the European Commission Historical Archives[1], while others preserve the entire national top-level domain (UK Web Archive[2]), or the entire Web (Internet Archive[3]).

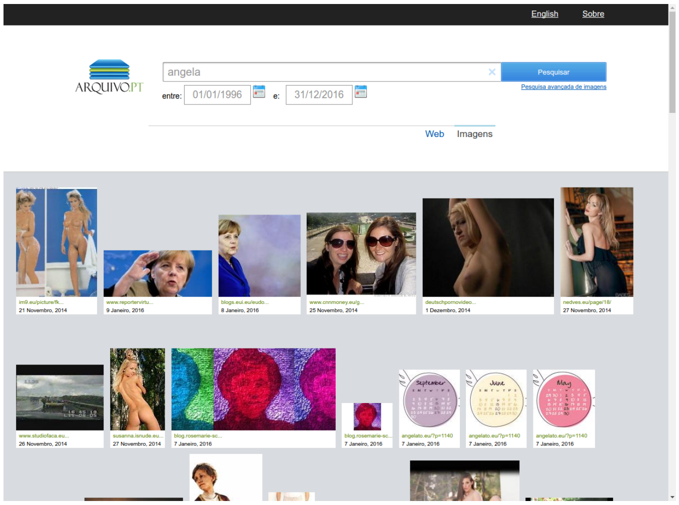
Arquivo.pt[4] is a Portuguese WA initiative to preserve the Portuguese *.pt* top-level domain and all web pages that publish Portuguese related information. It also acts as a research infrastructure, making its contents searchable and publicly available in open access. It is important that this historically valuable information is available.

Without the tools for users to retrieve the desired information, the usefulness of Web Archives is hampered. To accomplish the ability to search, Arquivo.pt provides a full-text search system to all its data. There have been some efforts to improve the Web Archiving Information Retrieval (WAIR) capabilities. For instance, a proposal to improve data search on Web Archives, by exploring their temporal information can be found in [2]. Following this path to provide better searching capabilities to this information, Arquivo.pt is developing an Image Search Service (ISS). This service enables image retrieval capabilities to Arquivo.pt contents, presenting an interface in which users can perform queries in natural language.

A. The Problem

There is a huge amount of visual content on the Web, stored as graphic files. One part of this visual content is Not Suitable For Work (NSFW) for most users, because it contains offensive or explicit images (naked persons, violence, and pornography, for instance). The exposure to these types of content is particularly critical for children and young persons. The Arquivo.pt ISS retrieves images matching the query against the filename, alternative text and the surrounding text of an image presented on a Web page. Therefore, due to the nature of the Web, there are no guarantees that an apparently not problematic query will not yield results with some offensive content. For instance, a website that got hacked for Web spam can exhibit this unexpected behavior. An example of this problem using the ISS is shown in Figure 1, in which an apparently not offensive query term *angela* was fulfilled on the ISS. The retrieved results have content that can be considered offensive to the users. There are several types of NSFW content, depending on the context of the application. In this work, we aim to filter out

[1] `https://ec.europa.eu/historical_archives`
[2] `https://www.webarchive.org.uk/ukwa`
[3] `https://archive.org`
[4] `https://arquivo.pt`



angela

entre: 01/01/1996 e: 31/12/2016

im9.eu/picture/fk... 21 Novembro, 2014
www.reportervirtu... 9 Janeiro, 2016
blogs.eui.eu/eudo... 8 Janeiro, 2016
www.cnnmoney.eu/g... 25 Novembro, 2014
deutschpornovideo... 1 Dezembro, 2014
nedves.eu/page/18/ 27 Novembro, 2014
www.studiofaca.eu... 26 Novembro, 2014
susanna.isnude.eu... 27 Novembro, 2014
blog.rosemarie-sc... 7 Janeiro, 2016
blog.rosemarie-sc... 7 Janeiro, 2016
angelato.eu/?p=1140 7 Janeiro, 2016
angelato.eu/?p=1140 7 Janeiro, 2016
angelato.eu/?p=1140 7 Janeiro, 2016

Fig. 1. Example of Arquivo.pt problematic content, retrieved with the ISS using the query term *angela*.

nudity/pornography content from the Arquivo.pt resources, by performing a binary classification task. The detection of NSFW contents on the archived Web pages from the Internet is challenging due to both the scale (billions of images) and the diversity (small to very large images, computer graphic images, natural images, among others) of image contents.

B. Organization of the article

The remainder of this article is organized as follows. Section II describes related work for classifying NSFW contents. On Section III, we present the main characteristics of Arquivo.pt. The proposed NSFW classification solution is addressed in Section IV. The experimental evaluation of our solution is reported in Section V. Finally, Section VI ends the article with some concluding remarks and directions for future work.

II. BACKGROUND AND RELATED WORK

The problem of automatically identify NSFW content from images and multimedia content is well studied [3], [4], [5], [6]. There has been a significant amount of research to provide and to improve methods to identify these contents. In this Section, we address an overview of the techniques employed for the task of NSFW content classification. Section II-A presents the first methods of image classification based on skin detection. Another type of techniques, based on Bag-of-Visual-Words (BoVW) is presented in Section II-B. Finally, Section II-C addresses the use of Neural Networks (NN) and Deep Learning (DL) to handle these tasks.

A. Skin Detection Methods

In 1999, an automatic system to detect if human nudes were present in an image was proposed [4]. It uses methods to mark skin-like pixels combined with color and texture properties. These marked regions are then analyzed by a specialized grouper, which attempts to group a human figure using geometric constraints on the human structure. Based on that, the system decides if a human is present or not.

The first methods [6] to address this problem were based on skin-detection algorithms to identify regions of interest, and then they analyzed the features of these skin regions to decide whether they were pornographic or not. As an example of these methods, we have the POESIA filter [7], an open source implementation of a skin-color-based filter. The performance of these methods relies on the accuracy of the skin detection algorithm and the extracted features, which are usually hand made. They present high false positive rates in images related to beach and sports activities.

B. Bag of Visual Words Methods

Another type of techniques that showed adequate image classification results was through BoVW [3]. These techniques extract, from an image, a set of visual features represented as words, similar to the Bag-of-Words (BOW) for text document classification [8], building a vocabulary vector with the number of occurrences of these visual words representing local image features. Those features usually are derived from detecting keypoints or local descriptors, such as Scale-Invariant Feature Transform (SIFT) [5] variations. A classifier that uses these representations is then trained to classify the image content as pornographic or not.

C. Neural Network Based Methods

An artificial NN is a computing system inspired by biological neural networks, found in animals and humans, being composed by simple structures that map an input value to an output value. Networks with a significant number of neurons and layers are named Deep Neural Networks (DNN).

For image classification, the input data to the network is given by the image pixels. Depending on the architecture and its goal, each NN inputs a fixed size data. A common input data size value for a NN to classify color images is $3 \times 256 \times 256$ array, such that each array unit corresponds to a pixel value from a specific RGB channel. Each array value can be seen as a *feature*, from a *feature vector* with 196 608

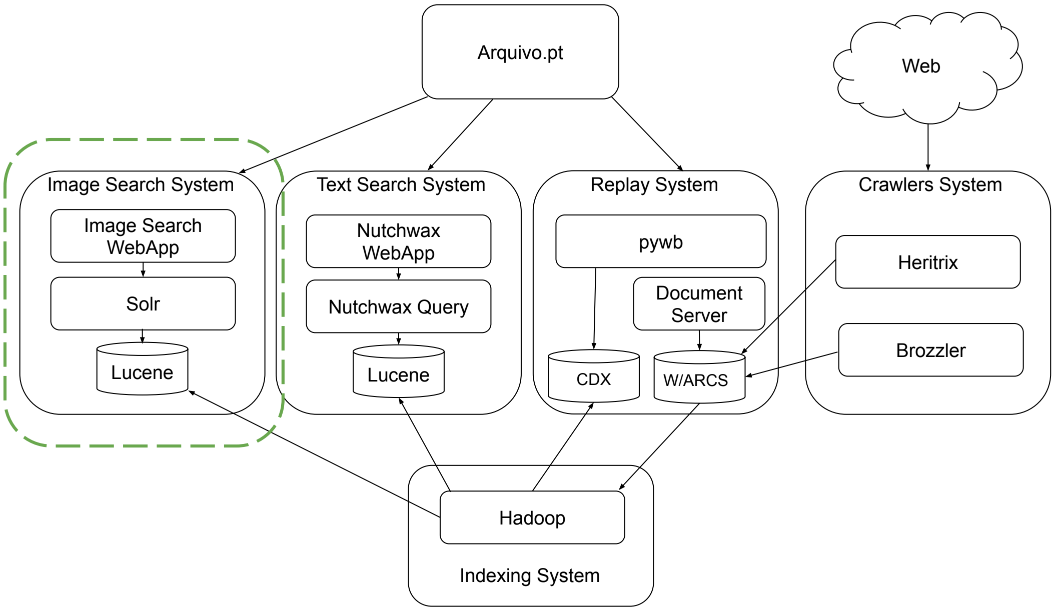


Fig. 2. Arquivo.pt Infrastructure and its subsystems. The Image Search System is highlighted since our proposal addresses this system.

($= 3 \times 256 \times 256$) dimensions. Based on these *features*, the NN assigns a class label to a given input image.

DNN are the latest research development on neural networks. It learns representations from data that puts an emphasis on learning successive layers of increasingly meaningful representations. It often involves tens or hundreds of successive layers of representations. These networks are characterized by the number of parameters, layers (depth) and neurons they have. There are several types, but more specifically for image recognition problems there has been a huge success with the Convolutional Neural Networks (CNN) [9]. The *depth* of a DNN represents how many layers contribute to model the data. The mechanism used to propagate the loss score and to update each layer weights is called back-propagation.

Recently, DNN showed state of the art results in many tasks of image recognition. For instance, CNN have been widely used on image recognition tasks [10], [11], for NSFW image classification [12], [13]. Many different CNN architectures have been published with improved accuracy on the standard ImageNet [14] classification challenge, namely:

- ALexNet (2012) [10] - 16.4%;
- ZF Net (2013) [15] - 11.7%;
- GoogLeNet (2014) [16] - 6.7%;
- Residual Networks (2015) [17] - 3.6%;

The use of DL combines both feature extraction and classification, so there is less involvement of a designer in terms of selecting the features or the classifier, as compared to a standard pattern recognition approach [18], [19]. The downside of these techniques is the amount of training data as well as the infrastructure that they require. Nowadays, improved training datasets such as ImageNet are available. Pre-trained models have been published openly to be used by people without the need to train a NN from scratch. An example of this kind of initiatives is the OpenNSFW, by Yahoo!, an open source model that identifies NSFW images, specifically, pornographic images [20]. The model is publicly available to be used for classification by developers, but the training dataset is not available, due to the nature of its content.

III. THE ARQUIVO.PT SYSTEM

This Section contains a description of the Arquivo.pt infrastructure, its data and how it is obtained, to provide the system environment context, and data provenance identification. In Section III-A, the Arquivo.pt system components are presented and Section III-B summarizes the type of data that Arquivo.pt contains.

A. Block Diagram

Figure 2 illustrates the logical organization of Arquivo.pt infrastructure, composed by five key system components: Image Search System, Text Search System, Replay System, Crawlers System, and Indexing System, each one with their own functionality and responsibility. The boxes represent a software component and the cylinders the data, needed by the components (inputs) or produced by them (outputs). The arrows represent the input/output relation between the components and the data. For instance, in the Crawlers System, both Heritrix [1] and Brozzler [21] software components produce as output an ARChive (ARC)/Web ARChive (WARC) file format [22], which is used by the Replay System or by the Indexing System as input data to generate derived data such as Capture inDeXes (CDX) [23] and Lucene[5] data sources.

[5] http://lucene.apache.org

B. Data Characterization

At the time of this writing, Arquivo.pt has a total amount of 6 966 589 866 preserved Web files, gathered in 3 967 593 ARC and WARC compressed files, fulfilling 368 Terabytes of disk space storage. The ARC/WARC file formats specify a method for combining multiple digital resources into an aggregate archival file together with metadata information. Each WARC file is a concatenation of one or more WARC records. Each WARC record consists of a header with metadata information, followed by a content block with the corresponding resource, such as images, text documents, or any type of resource found on the Web.

Each crawl has different configurations. For instance, the broad domain crawlers don't download files from the Web with more than 10 MB, but special crawls and high-quality crawls don't have those restrictions. Moreover, they are configured to accept all mime-types found on the Web. For this reason, the type of data that can be found at Arquivo.pt is widespread and heterogeneous.

Table I reports the top 10 mime-types[6] found during a 2017 *.pt* top-level domain crawl. These mime-types are reported by the Web servers using the Hyper-Text Transfer Protocol (HTTP) meta information, when each resource is collected.

Table II displays the top 7 mime-types regarding image content that were found during the last broad domain crawl. The most common image type is the JPEG with 76% of the total images, followed by PNG with 18%. Although only 4% of the images are GIF, this type can present an extra challenge to classify, since the images may have an animation. Since these images are originated from different websites, they can have many different file sizes, different resolutions and any type that is allowed on a Website.

IV. PROPOSED SOLUTION

In this Section, we describe the proposed solution to classify NSFW images. Section IV-A describes the CNN models for this task. Section IV-B details the integration of the proposed solution into the Arquivo.pt infrastructure.

TABLE I
NUMBER OF RESOURCES MIME-TYPES COLLECTED ON ARQUIVO.PT LAST BROAD DOMAIN CRAWL.

% Amount	Number of URL	mime-types
79.60	237 966 251	text/html
10.03	29 997 689	image/jpeg
2.45	7 328 179	image/png
0.77	2 305 834	application/pdf
0.76	2 271 770	application/javascript
0.72	2 145 481	text/xml
0.62	1 869 902	application/rss+xml
0.62	1 861 165	application/json
0.60	1 820 236	image/gif
0.58	1 783 630	text/css
3.25	1 783 630	all others

TABLE II
MIME-TYPES DISTRIBUTION FOR IMAGE TYPE CONTENTS.

% Amount	Number of URL	mime-types
75.59	30 145 538	image/jpeg
18.37	7 328 179	image/png
4.56	1 820 236	image/gif
0.98	389 839	image/svg+xml
0.34	135 720	image/x-icon
0.09	38 577	image/pjpeg
0.06	22 717	image/bmp

A. Deep Neural Network Models

In this work, we made experiments with two different topologies of DNN, namely ResNet [17] and SqueezeNet networks [24]. These networks define a *hypothesis space*, which constrains the space of possibilities to solve a specific problem. It is within this constrained space that we try to find a useful representation of the input data, that will be mapped for the desired output.

DNN are hard to train due to the vanishing and exploding gradient problem [25]. With the gradient value being back-propagated to the first layers, repeated multiplications may make the gradient values infinitely small or large. As a consequence, the network learning tends to halt, specially in the first layers. The same happened in the reverse way, in which the weights get saturated and the network learning also halts. Several approaches were used to address these problems, such as, normalized initialization [25], rectifiers [26], and normalization layers [27]. With the increasing depth of these networks, some network nodes get saturated and the accuracy degrades (not caused by overfitting). The ResNet network model addresses this degradation problem by introducing shortcut connections between the layers, as depicted in Figure 3.

The key idea is that building deeper networks by adding more layers should not degrade the network training performance [17]. Stacking identity mappings upon the current network would make the network perform exactly the same. So, this deeper model should not produce a training error higher than its shallower models. Their hypothesis is that letting the stacked layers fit a residual mapping is easier than letting them directly fit the desired underlying mapping. It is shown [17] that these networks are easier to optimize, and achieve a higher accuracy due to their increased depth.

The SqueezeNet [24] network model was designed to

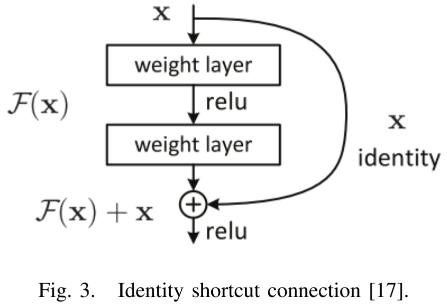

x
weight layer
relu
$\mathcal{F}(\mathbf{x})$
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig. 3. Identity shortcut connection [17].

[6] https://www.iana.org/assignments/media-types/media-types.xhtml

have a smaller architecture, while preserving the accuracy of bigger models. This NN has less parameters and layers and it has multiple advantages:

- More efficient distributed training;
- Less overhead exporting new models;
- Be able to run on Field-Programmable Gate Array (FPGA) and embedded circuits.

The authors of the network model accomplish these goals, with the following strategies:

- Replace 3x3 filters with 1x1 filters;
- Decrease the number of input channels to 3x3 filters;
- Downsample late in the network so that convolution layers have large activation maps.

These strategies reduce the number of parameters (weights) of the NN. With the above strategies in mind, the convolution filters are organized in a Fire Module [24]. The Fire Module, depicted in Figure 4, is composed by a *squeeze* convolution layer which has only 1x1 filters, squeezing the incoming data. It also has an *expand* convolution layer that has a mix of 1x1 and 3x3 convolution filters, expanding the depth of the data again.

There are two key parameters of the DNN that we need to address on our solution. The first one is the loss function and the second one is the optimizer used to train the network. To be able to find an useful representation to solve the problem at hand, we need to somehow get feedback on how well the representation is performing. The loss function, also known as objective function, gives us that feedback. A loss function tells us “how good” our model is at making predictions for a given set of parameters. The choice of a proper loss function is very important, because the networks take any shortcut to minimize the loss. Thus, if the loss function is not well suited for the problem at hand, the network may not achieve the desired results. Since the problem addressed in this work is a binary classification task, we used a Cross-Entropy loss function [28], defined as

$$c = \sum_{i=1}^{n} p_i \log\left(\frac{1}{q_i}\right), \tag{1}$$

where p_i and q_i represent the probabilities/scores of the true model and the estimated model, respectively. Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value (between 0 and 1). Cross-entropy loss increases as the predicted probability diverges from the actual label. A perfect model would have a log loss of 0. Thus, minimizing cross entropy is equivalent to maximizing the log likelihood of our data, which is a direct measure of the predictive power of our model.

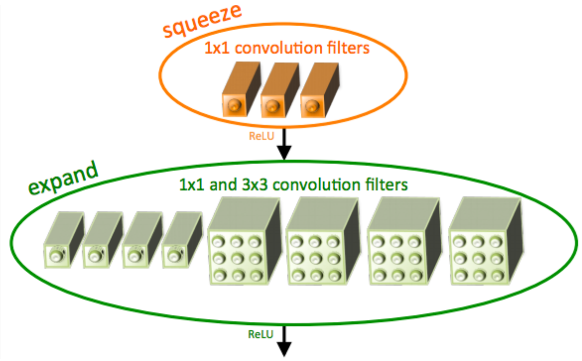


Fig. 4. SqueezeNet Fire Module [24].

The optimizer is the mechanism that updates the network weights based on what the network is predicting and its loss function. Progressively, the incremental updates of the optimizer will lower the loss score, making the network prediction more accurate. The optimizer falls into two types of algorithms: first order and second order. The first order optimization algorithms use the loss function gradient to minimize the network loss. They are known as Gradient Descent (GD) algorithms. The second order algorithms use the second order derivative to minimize the network loss. With second order algorithms, it is possible to know if the first derivative is increasing or decreasing, and with that the function curvature. The second order derivative has higher computational cost, therefore the first order optimizers are often used. For this work, a Stochastic Gradient Descent (SGD) algorithm is used [29].

B. Integration on Arquivo.pt

The developed solution to classify image contents as NSFW (or its opposite) is integrated on the Arquivo.pt ISS indexing workflow. This workflow extracts images and related metadata, from the ARC file.

The integration that is proposed and implemented is modular, and can be extended by changing the underlying model. It also supports a real time classification, exposed as a Web Service. The workflow of the solution is depicted in Figure 5. The ISS indexing workflow is composed by one Hadoop job that processes each WARC file and splits it into smaller units named WARC Records. Each WARC Record contains a Web resource representation which is processed by a Hadoop map. The Hadoop map extracts text and images included in the WARC Records and builds an index record with the image and its derived metadata. The NSFW classification score is added as a metadata field.

The images contained in each WARC file are extracted and analyzed in order to get a feature vector. This vector is used by a classifier to discriminate if a image is NSFW or not, and finally the results are stored. The Hadoop cluster processes the WARC data as follows:

(1) The Hadoop task processes the WARC and the extracted images are queued.
(2) The Workers then fetch the images from the queue and perform the classification.
(3) The classification result is reported back to the message queue.
(4) Finally, the Hadoop task fetches the classification result and proceeds with the indexing workflow, generating the Solr[7] indexes.

[7] http://lucene.apache.org/solr/

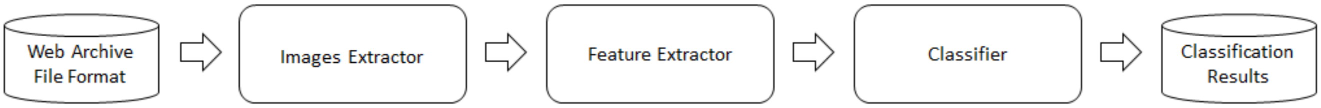


Fig. 5. Workflow of the proposed classification system.

The Web API also handles the classification requests, posting the images to be classified to the queue.

V. EXPERIMENTAL EVALUATION

This Section reports the experimental evaluation taken on the proposed solution, and it is organized as follows. Section V-A describes how a ground-truth dataset was build, to perform this evaluation. Section V-B presents the software and hardware platforms used to perform these evaluations as well as the standard evaluation metrics. The image preprocessing steps are detailed in Section V-C and some experimental results are reported in Section V-D. The solution integration is reported in Section V-E and the last Section V-F addresses the experimental results of the improvement on the models.

A. Building the Initial Evaluation Dataset

There are few datasets available for NSFW classification. Most of them[8] are raw datasets containing only the location of the content. However, those locations are often unavailable. Since Arquivo.pt has a considerable amount of NSFW images, collected over the years, and since our goal is to put the model in production on that platform, we decided to build our own NSFW dataset. It is composed by 17 655 images manually labeled from Arquivo.pt with 8 273 labeled as NSFW and 9 382 as SFW (see Table III). The remaining 18 626 images are not (yet) labeled.

On Arquivo.pt, the total of images that belong to the NSFW class is much less than the images from the SFW class. The main difficulty at this task is to find enough images from the NSFW class in order to build a dataset with a significant number of images, guaranteeing that both classes are balanced.

These images were acquired from Arquivo.pt using two methods. The first method to acquire the images was through the existent Beta Images Indexes. These are Lucene indexes provided by the Solr platform. Each index document is an image resource with its corresponding metadata, for instance the image width and height, Uniform Resource Locator (URL) of the origin site and text extracted from the image tags properties. The Solr platform exposes a REST API that was used to process queries to return images to be labeled. The second method uses Arquivo.pt Text Search API [30], querying the API to retrieve Web pages. From those Web pages, the images were extracted to be labeled. Arquivo.pt Text Search API was queried to retrieve Web pages that contain the terms *porn*, *blowjob*, or *fuck*. Those terms are usually associated with NSFW contents [31]. Web pages that contain these three keywords are more likely to have pornographic content, increasing the amount of retrieved relevant images. With the list of the archived Web pages URL returned, each archived Web page was scrapped by their Hyper-Text Markup Language (HTML) img tags and the available archived images downloaded. A total of 18 000 images were retrieved this way.

TABLE III

SUMMARY OF THE COLLECTED DATASET.

	SFW	NSFW	Total
Labeled Images	9 382	8 273	17 655
Non-Labeled Images	-	-	18 626

B. Evaluation Setup and Metrics

The hardware used to evaluate these models is a common laptop with 8 GB RAM, a GeForce GTX 860M as GPU and a Intel(R) Core(TM) i7-4710HQ CPU @ 2.50GHz. The models were also tested using server class hardware available at Arquivo.pt infrastructure. The server is a Dell PowerEdge R730xd model with 256 GB RAM and an Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.4 GHz.

One of the evaluation metrics used to compare the models was the Area Under the Curve (AUC), computed over the Receiver Operating Characteristic (ROC) curves. This is a common evaluation metric used on binary classifiers, and it evaluates the classifier varying a cut-off threshold [32]. It is useful to evaluate the models before a threshold value is chosen as cut-off. Other evaluation metrics were calculated such as *accuracy*, *precision*, *recall*, and *f-score*, defined as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{and} \tag{4}$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5}$$

True Positives (TP) is the number of NSFW images correctly classified as NSFW and True Negatives (TN) the number of SFW images correctly classified as SFW. False Positives (FP) is the number of SFW images that were classified as NSFW and False Negatives (FN) is the number of NSFW images classified as SFW.

Another metric used as evaluation is the amount of images a model can classify per second with a specific hardware. Using limited computational resources, a model with a higher

[8] E.g. https://github.com/EBazarov/nsfw_data_source_urls

classifying speed can be used as a trade-off for choosing a model with less accuracy, for instance when the model needs to be used online.

C. Image Preprocessing

The models were evaluated using the dataset described in Section V-A. For each model, all the images were preprocessed and passed through the network. Before each image is passed through the CNN, a preprocessing phase was performed. This preprocessing depends on the library used to read the images, on the CNN model, and how it was trained. The models are using as input a $b \times 3 \times 256 \times 256$ vector, where $b >= 1$ is the batch size (the number of images that will be classified in a batch), and the other indexes correspond to the number of image channels, image width and image height (196 608 dimensions per image, with $b = 1$). Depending on the library that reads the images, the input vector must be arranged to match the neural network models input. On this evaluation test, the Python Pillow library [33] and the Caffe framework [34] were used for image processing. Caffe natively expects the color channels ordered as BGR instead of RGB, so a transposition is needed to switch the channels. The mean is subtracted from the data and the values are scaled from the [0,255] range to the [0,1] range. Finally, a bilinear image resizing to the 256×256 spatial resolution is performed. The image preprocessing is the same that it was used on the training phase of the pre-trained models networks, the best results.

D. Performance Evaluation Results

Figures 6 and 7 show the ROC curve for the models obtained from OpenNSFW and NSFWSqueezeNet, respectively. OpenNSFW has a better AUC with 0.98 comparing to the 0.85 of the NSFWSqueezeNet.

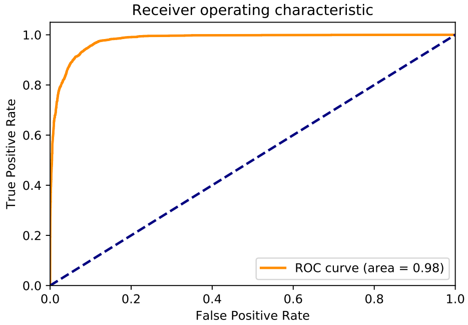


Fig. 6. ROC Curve evaluation of the OpenNSFW model.

Table IV shows the confusion matrix for the OpenNSFW model, and Table V reports the evaluation metrics. The confusion matrix indicates that the classification error is balanced between both classes (SFW and NSFW), with more FP (652) than FN (567). It happens to be a good result in this domain, as it is preferable to be more conservative and classify images as NSFW even if they are not, than the opposite.

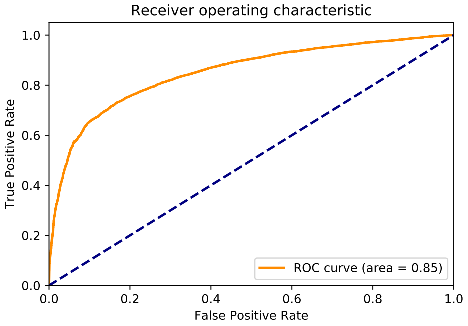


Fig. 7. ROC Curve evaluation of the NSFWSqueezeNet model.

TABLE IV

CONFUSION MATRIX OF THE OPENNSFW MODEL.

	NSFW’	SFW’
NSFW	7706	567
SFW	652	8730

Tables VI and VII report the results for the NSFWSqueezeNet. In this model, the most frequent type of errors are FN, which is not suitable for our system. The OpenNSFW model reports the best score with 0.93 accuracy in comparison to 0.72 from NSFWSqueezeNet.

A further analysis on misclassified images lead to the following observations. As false positives, the most common sources of error are images with women with a significant amount of nudity, but not explicit. At false negatives samples, common occurrences are sexual acts with animals, and sexual explicit animations. This is expected, since the model was trained to filter pornography.

Table VIII reports the classification speed obtained with different setups. The NSFWSqueezeNet model has the best classification speed, being able to classify 77 images per second in comparison with the OpenNSFW model, with 40 images per second, both using the GPU GTX860M setup.

These results show the adequacy of the proposed classification approach, since it is supposed to be used on a offline basis.

E. Solution Integration

The image classification task is performed offline, thus the extra workload that the system needs to perform is very low. For each indexed document representing an image, a field named *safeimage* was added, with values ranging between 0 and 1. When performing a query, Solr filters out NSFW images based on the value of this field (above 0.5). For 6 468 single term queries requested to Solr with the filtering of

TABLE V

EVALUATION METRICS FOR THE OPENNSFW MODEL.

	Precision	Recall	F1-Score	Accuracy
NSFW	0.92	0.93	0.93	-
SFW	0.94	0.93	0.93	-
Average / Total	0.93	0.93	0.93	0.93

TABLE VI
CONFUSION MATRIX OF THE NSFWSQUEEZENET MODEL.

	NSFW'	SFW'
NSFW	3697	4576
SFW	333	9049

TABLE VII
EVALUATION METRICS FOR THE NSFWSQUEEZENET MODEL.

	Precision	Recall	F1-Score	Accuracy
NSFW	0.92	0.45	0.60	-
SFW	0.66	0.96	0.79	-
Average / Total	0.78	0.72	0.70	0.72

TABLE VIII
CLASSIFICATION SPEED PERFORMANCE (IMG/SEC) WITH DIFFERENT HARDWARE AND BACKEND SETUPS.

	OpenNSFW	NSFWSqueezeNet
BVLC Caffe i7 + OpenBLAS	7	17
BVLC Caffe GTX860M CUDA	40	77
Intel Caffe CPU Xeon + MKL	9	28

NSFW images, the average query response time was 9 ms. Without the filtering, it was 1 ms. Figure 8 depicts the image filtering integration on the user interface. The users can opt for excluding this kind of image contents or not. Figure 9 shows the results of the classifier exhibiting one example of a misclassified image (hidden by a gray rectangular box). Figure 10 shows the results retrieved by the same query as in Figure 9, without using the classifier. We can now observe many NSFW images hidden by a gray rectangular box.

F. Improving the model

In order to improve on the model, techniques of data augmentation can be applied to increase the available dataset and consequently improve the algorithms accuracy on those tasks. It is known that applying simple data augmentation techniques such as cropping, rotating and flipping input images reduce the overfitting and class imbalance problems typically found in small datasets [35], [36]. Augmentator [37], an image data augmentation library for machine learning, was used to generate new images by applying transformations such as mirroring, shearing, and flipping. These new images were added to the training set.

The learned models can be improved through fine-tuning of the values of some parameters. There are several methods to fine tune a DNN model, for instance, training the network with our labeled data and performing small adjustments to the output layer, or adjusting more layers, such as the last 3 layers. All the network layers can also be retrained, using the network weights from a pre-trained model. These weights usually are good initial weights to start training a network instead of using other weight initialization techniques [38].

Experiments to evaluate if a model can improve its accuracy have been made. Both models were retrained using a 4-fold methodology on the built dataset and the following heuristics were applied:

- Retraining each model for 1 epoch and augmenting the dataset.
- Retraining each model for 5 epochs and augmenting the dataset.

One epoch means one pass of the full training set. It contains several iterations where each iteration is a training image being passed forward through the network.

To improve the NSFWSqueezeNet model, all network model layers were updated using the SGD solver, with the following training parameters:

- Learning rate policy - poly;
- Power - 1.0;
- Momentum (α) - 0.9;
- Base learning rate (η) - 0.001;
- Weight decay - 0.0000001.

During the network model training, the learning rate parameter was updated following a polynomial decay:

$$\eta_{iter+1} \leftarrow \eta_{iter} * \left(1 - \frac{iter}{max\_iter}\right)^{power}. \qquad (6)$$

Table IX reports the experimental results of the improvement on the NSFWSqueezeNet model, while Table X does the same for the OpenNSFW model.

TABLE IX
NSFWSQUEEZENET FINE-TUNING ACCURACY AND LOSS.

Model	4-Fold Accuracy	4-Fold Loss
NSFWSqueezeNet 1 Ep. Aug. 10K	0.88 ± 0.002	0.28 ± 0.004
NSFWSqueezeNet 1 Ep. Aug. 10K	0.89 ± 0.002	0.27 ± 0.006

There was a significant accuracy improvement, from the initial model accuracy of 72% to 89% accuracy. The training network configuration, solver configuration and training logs are available online[9].

The OpenNSFW model is computationally more expensive to train. With the limited hardware available and time constraints, an attempt to improve it was made, freezing all the network layers and retraining only the last fully-connected and the softmax layers. The solver used was also the SGD with the same learning parameters as the model above.

TABLE X
OPENNSFW FINE-TUNING ACCURACY AND LOSS.

Model	4-Fold Accuracy	4-Fold Loss
OpenNSFW 1 Ep. Aug. 10K	0.92 ± 0.003	0.20 ± 0.006
OpenNSFW 5 Ep. Aug. 10K	0.94 ± 0.004	0.16 ± 0.007

There is a small accuracy improvement on Table X, not significant because the accuracy number is rounded to 2 decimal places.

VI. CONCLUSIONS

Arquivo.pt provides access to more than six million preserved resources. In order for users to find the information

[9] https://github.com/arquivo/SafeImage/tree/master/training/SqueezeNet

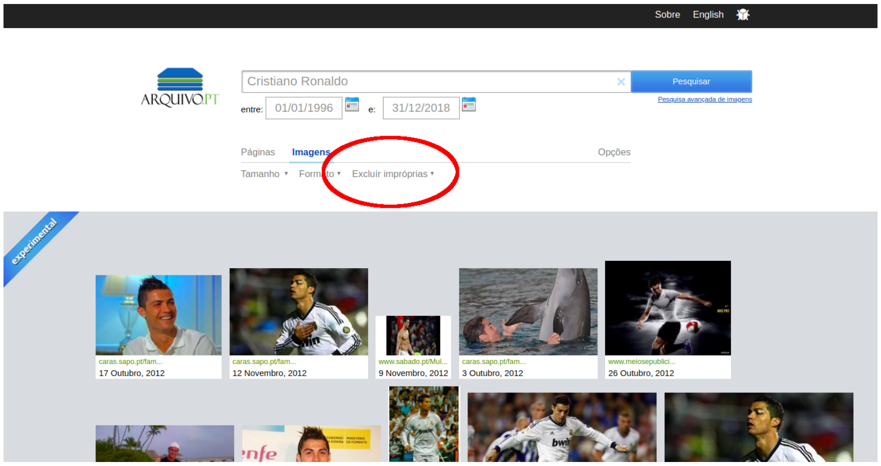


Fig. 8. Image classification interface integration, using as query term ‘Cristiano Ronaldo’. The red ellipse highlights the NSFW filter choice by the user.

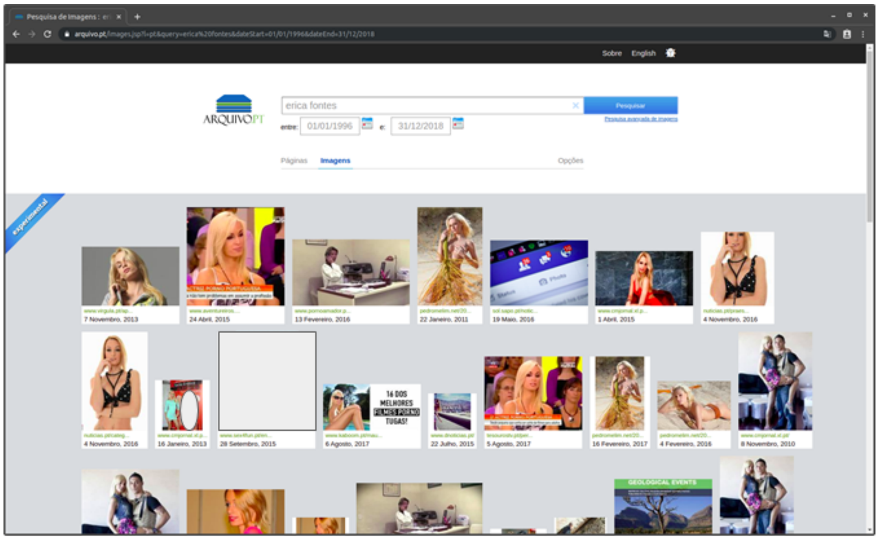


Fig. 9. Image filtering interface integration, using as query term ‘Erica Fontes’, with the NSFW classifier. The gray rectangular box (third image in the second row) highlights NSFW contents which were misclassified.

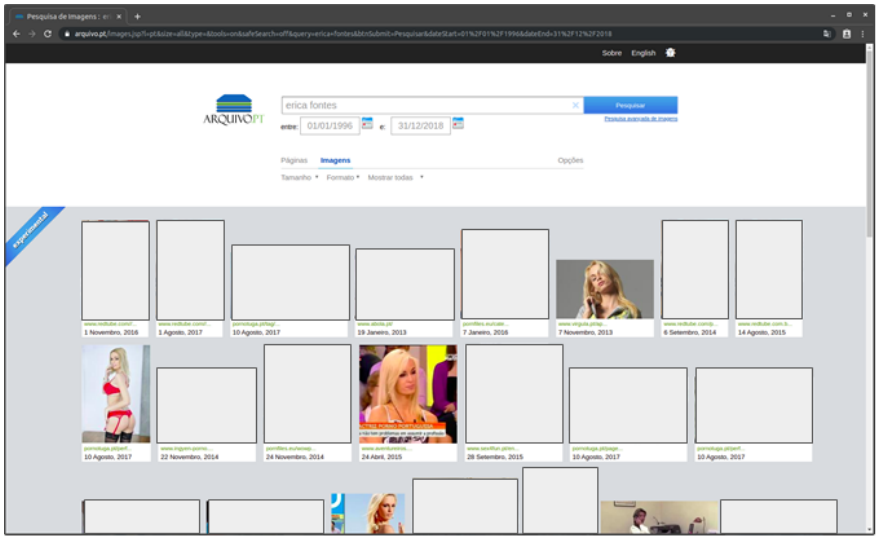


Fig. 10. Image filtering interface integration, using as query term ‘Erica Fontes’, without the NSFW classifier. The gray rectangular boxes highlight (and hide) NSFW contents.

they want, among all Arquivo.pt contents, information retrieval tools are provided to find pages with a specific text and an Image Search System (ISS) is under development to expand its searching capabilities. There are huge amounts of visual content on the Web, and thus, also in Arquivo.pt. The ISS provides an easy access to this visual content. However, some of this visual content can be offensive for users (for instance naked persons, violence, or pornography).

To mitigate this problem, a solution that automatically identifies images with this type of content was developed and integrated into Arquivo.pt ISS. The solution uses a Convolutional Neural Network to identify this content type and provides the classification result as an input to the ISS, which uses it to hide not suitable for work contents, from the results.

Two deep neural networks pre-trained to solve this kind of task were evaluated on a dataset built with Arquivo.pt images. Then, a fine tuning process was performed to improve the model's accuracy, identifying this type of content. The initial evaluation of the models reported 93% accuracy for the ResNet model and 72% for the SqueezeNet model. After fine tuning, the model's accuracy, was improved by 1% on the ResNet model (94% accuracy) and by 17% on the SqueezeNet model (89% accuracy). The best model was integrated in the ISS indexing workflow, using a message queue as a broker to be able to distribute and scale the classification work among several workers. The proposed solution is currently in production at Arquivo.pt (`https://arquivo.pt/image.jsp`).

A. *Future Work*

As future work, the models accuracy can be improved building a larger dataset and using more recent Deep Neural Network models, while simultaneously increasing the number of training epochs. Another improvement is that instead of using only 2 classes, NSFW or SFW, categorizing with more classes could help users to better define their needs. An example is having the choice of Strict, Moderate or Off, like the DuckDuckGo[10] search engine does. Moreover, they can be expanded to identify other types of NSFW content, for instance, violence, and offensive symbols, among others.

REFERENCES

[1] Gordon Mohr, Michael Stack, Igor Ranitovic, Dan Avery, and Michele Kimpton. Introduction to Heritrix, an archival quality web crawler. In *4th International Web Archiving Workshop (IWAW04)*, Bath, UK, 2004.

[2] Miguel Costa. *Information Search in Web Archives*. PhD thesis, Faculty of Sciences of the University of Lisbon, December 2014.

[3] T. Deselaers, L. Pimenidis, and H. Ney. Bag-of-visual-words models for adult image classification and filtering. In *2008 19th International Conference on Pattern Recognition*, pages 1–4, Dec 2008.

[4] D.A. Forsyth and M.M. Fleck. Automatic detection of human nudes. *International Journal of Computer Vision*, 32(1):63–77, Aug 1999.

[5] T. Lindeberg. Scale Invariant Feature Transform. *Scholarpedia*, 7(5):10491, 2012. revision #153939.

[6] Huicheng Zheng, Mohamed Daoudi, and Bruno Jedynak. Blocking Adult Images Based on Statistical Skin Detection. *Electronic Letters on Computer Vision and Image Analysis*, 4(2):1–14, 2004.

[7] Daoudi Mohamed. POESIA - Filtering Software @ONLINE, January 2018.

[8] C. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[9] Gu Jiuxiang et al. Recent advances in convolutional neural networks. *CoRR*, abs/1512.07108, 2015.

[10] Alex Krizhevsky, IIya Sulskever, and Geoffret E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information and Processing Systems (NIPS)*, pages 1–9, 2012.

[11] Y. Sun, B. Xue, M. Zhang, and G. G. Yen. Evolving deep convolutional neural networks for image classification. *IEEE Transactions on Evolutionary Computation*, 24(2):394–407, 2020.

[12] Dmirty Zhelonkin and Nikolay Karpov. Training effective model for real-time detection of nsfw photos and drawings. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 301–312. Springer, 2019.

[13] Ruolin Zhu, Xiaoyu Wu, Beibei Zhu, and Liuyihan Song. Application of pornographic images recognition based on depth learning. In *Proceedings of the 2018 International Conference on Information Science and System*, pages 152–155, 2018.

[14] Stanford University. ImageNet, http://www.image-net.org/, January 2018.

[15] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.

[16] Christian Szegedy et al. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[18] R. Duda, P. Hart, and D. Stork. *Pattern classification*. John Wiley & Sons, 2nd edition, 2001.

[19] F. Escolano, P. Suau, and B. Bonev. *Information theory in computer vision and pattern recognition*. Springer, 2009.

[20] Jay Mahadeokar. Open NSFW model code @ONLINE, January 2018.

[21] Noah Levitt. Brozzler - Distributed browser-based web crawler @ONLINE, January 2018.

[22] International Organization for Standardization. ISO 28500:2017 information and documentation – WARC file format, January 2018.

[23] Internet Archive. Internet Archive: CDX File Format Reference @ONLINE, January 2018.

[24] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. SqueezeNet. *arXiv, 1602.07360*, 2016.

[25] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*, 2010.

[26] He Kaiming et al. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. *CoRR*, abs/1502.01852, 2015.

[27] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[28] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. *CoRR*, abs/1702.05659, 2017.

[29] Sebastian Ruder. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747, 2016.

[30] Arquivo.pt. Arquivo.pt API v.0.2 (beta version), https://github.com/arquivo/pwa-technologies/wiki/Arquivo.pt-API-v.0.2-(beta-version), March 2018.

[31] slate.com. Words banned from Bing and Google's autocomplete algorithms., March 2018.

[32] Charles X. Ling, Jin Huang, and Harry Zhang. AUC: A better measure than accuracy in comparing learning algorithms. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 2671, pages 329–341, 2003.

[33] Alex Clark et al. Pillow: 3.1.0, January 2016.

[34] Yangqing Jia et al. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[35] Patrice Simard, Dave Steinkraus, and John C Platt. Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis.

[10] `https://duckduckgo.com/`

Proceedings of the 7th International Conference on Document Analysis and Recognition, pages 958–963, 2003.

[36] Sebastien Wong, Adam Gatt, Victor Stamatescu, and Mark McDonnell. Understanding Data Augmentation for Classification: When to Warp? In *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2016.

[37] Marcus D. Bloice, Christof Stocker, and Andreas Holzinger. Augmentor: An image augmentation library for machine learning. *CoRR*, abs/1708.04680, 2017.

[38] Sarfaraz Masood, M. N. Doja, and Pravin Chandra. Analysis of weight initialization techniques for Gradient Descent algorithm. In *12th IEEE International Conference Electronics, Energy, Environment, Communication, Computer, Control: (E3-C3), INDICON*, 2015.