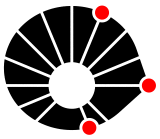
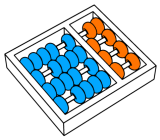




Scientific Integrity Analysis of Misconduct in Images of Scientific Papers

J. P. Cardenuto *A. Rocha*

Relatório Técnico - IC-PFG-19-53
Projeto Final de Graduação
2019 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Scientific Integrity
Analysis of Misconduct in Images of Scientific Papers

João Phillipe Cardenuto* Anderson Rocha†

Abstract

The pressure of "publish or perish" in the competitive research environment of science leads many scientists to misconduct. Aiming to foster scientific integrity, this work proposes a framework for detecting suspicious images of scientific articles. Its workflow begins with a PDF file of a scientific publication and ends with highlighting of suspected fraud regions. This workflow is divided into four operation steps: image extraction, image segmentation, clustering, and copy-move forgery detection. Each module of the framework was validated individually. As a result, the framework has outperformed existing methods for accomplishing each task. The image extraction achieves better results on efficiency and effectiveness than famous extraction images tools (e.g pdfimages); the segmentation achieves 98% accuracy on detecting relevant images regions and a proposed fusion of copy-move forgery detection achieves the best result of 19% average IoU on a dataset with 100 images proposed by this work. In addition, a real case of fraud was used to validate the framework as a whole. The images highlighted by the framework were the same as described in the case's retraction note.

1 Introduction

In an increasingly competitive environment of academic life, where the importance of a scientist is measured in numbers of published articles, researchers are under pressure to achieve a high number of publications per year. This phenomenon, coupled with the ease of using image-editing tools such as Adobe Photoshop or GIMP, leads many researchers to mishandle their results so that data and figures fit their hypothesis. Fanelli [1] collected data from surveys and meta-analysis to answer how many scientists falsify or fabricate their scientific results. With his work, he concluded that while some scientists (2%) admit an

*Institute of Computing (IC) - University of Campinas, 13084-971, Campinas, SP, Brazil. Research supported by FAPESP, process 2018/15864-4

†Institute of Computing (IC) - University of Campinas, 13084-971, Campinas, SP, Brazil.

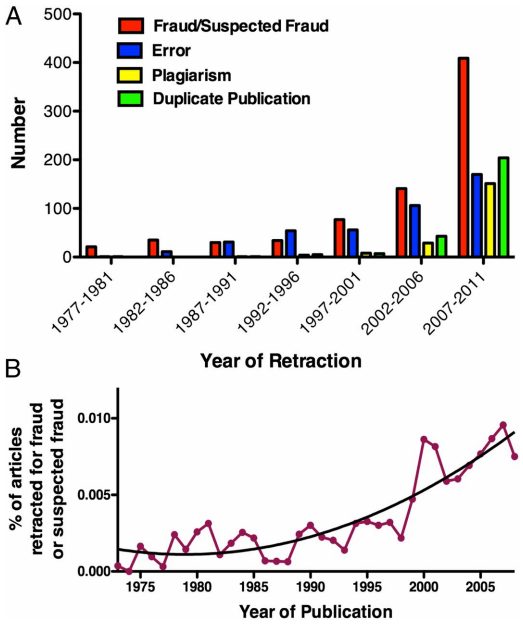


Figure 1: (A) Number of retracted articles for specific causes by year of retraction. (B) Percentage of published articles retracted for fraud or suspected fraud by year of publication. Image extracted from [3] figure 1

act of misconduct during their career, when asked about their colleague's behavior, the rate increases to 14% and for a questionable act of their colleague, 72%. To understand the correlation between journal impact factor and misconduct data duplication of their authors, Oksvold [2] selected at random 120 articles from three different journals related to cancer, with different impact factors. With his analysis, he found that 25% of journal articles of impact factor less than five or greater than twenty contains data duplication. For the journals with impact factor of 5-10, the index was 22%. In a similar work, Fang et al. [3] analyzed two thousand of retracted papers from the 1970s to 2012 on PubMed repository[1]. Figure 1 extracted from Fang et. al. work [3] describes reasons of retracted papers by year. Fraud/Suspect of Fraud are the most common retraction causes. With that data, Fang et al. concluded that 67.4% of retractions were attributable to misconduct and pointed to the raising of this percentage over time (Fig 1b).

[1]PubMed: An online repository of biomedicine papers. Website: https://www.ncbi.nlm.nih.gov/pubmed/ (Last Access 6 of September of 2019

Definition of Inappropriate Duplicated Image

Following the importance of any kind of research to prevent and detect misconduct, this work concerns image manipulation/duplication detection in scientific publications. We based our research on the work of Bik et al. [4], who manually screened 20,000 biomedical papers from 40 different journals from 1995 to 2014 and found that 3.8% contained problematic figures. In their study, they classify three different categories to inappropriate duplicated images.

1. **Simple Duplication:**
   "Figures containing two or more identical panels, either within the same figure or between different figures within the same paper, purporting to represent different experimental conditions" [4]. Examples of Simple Duplication category can be found in Figure 2a;
2. **Duplication with alteration:**
   "Images that were altered with complete or partial duplication of lanes, bands, or groups of cells, sometimes with rotation or reversal with respect to each other, within the same image panel or between panels or figure" [4]. Examples of Duplication with Alteration category can be found in Figure 2b;
3. **Duplication with repositioning:**
   "Images with a clear region of overlap, where one image had been shifted, rotated, or reversed with respect to the other" [4]. Examples of Duplication with Repositioning category can be found in Figure 2c;

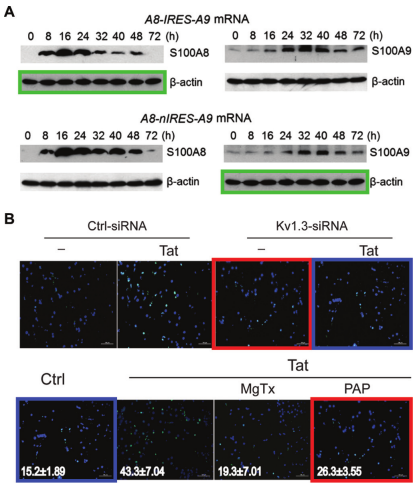
Definition of Research Misconduct

According to ORI [2] "research misconduct means fabrication, falsification, or plagiarism in proposing, performing, or reviewing research, or in reporting research results" [5].

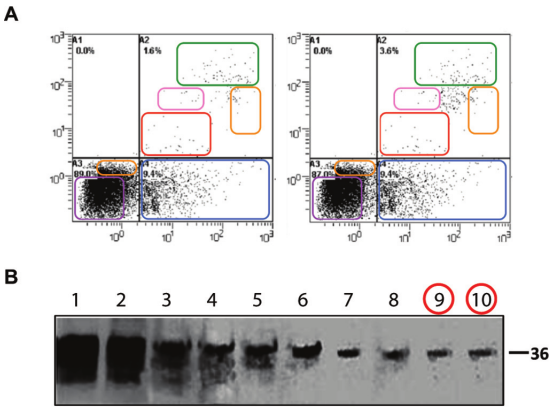
1. **Fabrication** is making up data or results and recording or reporting them.
2. **Falsification** is manipulating research materials, equipment, or processes, or changing or omitting data or results such that the research is not accurately represented in the research record.
3. **Plagiarism** is the appropriation of another person's ideas, processes, results, or words without giving appropriate credit.
4. Research misconduct does not include honest error or differences of opinion.

[2]ORI: *The Office Research Integrity* : https://ori.hhs.gov/ (Access at 30/09/2019)

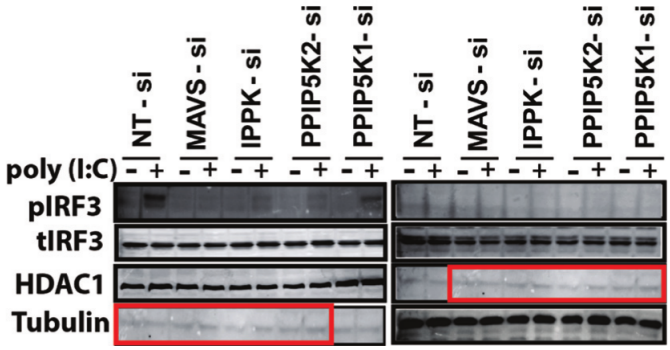
Linking this definition with Bik et al. [4] we can notice that in some circumstances we can find an image duplication or manipulation without a misconduct act. Concerning this, we do not intend to propose context or human revision free method. With this background, we developed a framework to highlight suspected duplicated images. The method begins with raw PDF document (standard for scientific articles) and ends highlighting the more suspicious regions of their images. To validate the proposed framework we performed experiments on each individual part of it and analyzed its behaviour for a real case of fraud, explained in section 5.



(a) Examples of simple duplication. Image extracted from [4] figure 2.



(b) Duplication with Alteration. Image extracted from [4] figure 4.



(c) Duplication with repositioning. Image extracted from [4] figure 3.

Figure 2: Types of duplication of scientific biomedical images. (a) Simple Duplication could be caused by a human not intentional error. (b) Duplication with Alteration: Probably intention misconduct. (c) Duplication with Repositioning: Probably consist of elaborated fraud. All figures were extracted and modified from [4]

This work is included in two projects concerned with data integrity. The first one is the DejáVù Thematic Project[3] at RECOD Laboratory (Reasoning for Complex Data) IC-UNICAMP (Institute of Computing - University of Campinas) supported by FAPESP 2018/15864-4. The second is the Media Forensics Integrity Analytics (MediFor[4]) supported by Advanced Research Projects Agency (DARPA-USA).

2 Objectives

The goal of this work is to research and develop an automatic open-source method that provides a forensic analysis of manipulated images in scientific publications. This method does not have the intention to provide a definitive result, but it highlights the most suspicious image regions to a human revision.

3 Related Works

To the best of our knowledge, there are just two works concerning automatic methodology to detect image manipulation from a scientific paper in the literature: Acuna et al. [6] and Bucci [7].

Work 1 - Acuna et al.

Acuna et al. [6] proposed an automatic workflow that starts with a collection of scientific images and ends with the most suspicious cases of misconduct.

During the first step of their approach, each image in the collection is forward to a copy-move detector. In the next step, they classify all the matches from the copy-move detectors as biological or not, aiming to highlight the biological matches. In the end, to confirm if any misconduct occurred, a human checks the results and judges each case.

The main contribution of their work is an automatic method used on several images, filtering the suspicious ones to a human revision. This is important since a human could not compare hundreds of papers as fast as a machine. On the other hand, a machine can not easily distinguish reused figures as legit or fraud.

As mentioned by the authors, the implementation of the framework is limited by the time complexity of copy-move detector. For this task, they extracted and matched features with SIFT from each image of the collection, which seems to be an interesting area of investigation.

[3]DejáVù: `http://www.ic.unicamp.br/~dejavu` (Accessed 25/10/2019)
[4]MediFor: `https://engineering.purdue.edu/MEDIFOR` (Accessed 25/10/2019)

Work 2 - Bucci

Bucci [7] investigated an automated workflow to analyze image manipulation specialized in western blot image.[5] His main contribution is a pipeline that starts with an article in PDF format and ends with suspect image regions highlighted.

In his method, each page from a PDF input is converted to a JPG image. Then, a segmentation step extracts all figure panels, using morphological operations. After this, a handcrafted feature-based algorithm checks if two aligned blots from the same page were cloned. In addition to the blot duplicate detection, Bucci used a commercial software "Visual Similarity Duplicate Image Finder Pro" (MindGems Inc) to analyze image duplicate from different documents.

4 Methodology

The methodology is divided into four sub-sections and summarized in figure 3:

1. Image Extraction
2. Segmentation
3. Detection
4. Clustering

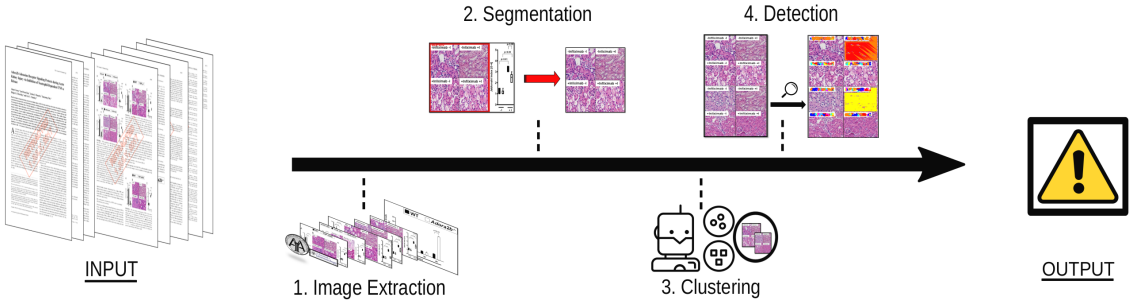


Figure 3: Proposed Method to highlight candidates image regions of misconduct. 1. Image Extraction- Extract all images embedded from a PDF document. 2. Segmentation- select regions of an image semantically interest to investigate. 3. Clustering- Group similar images based on their content. 4. Detection- Highlight suspect regions.

[5]Western Blot Image: An image resulting from a technique to detect the presence of a specific protein extracted from a biological substance. "The vast majority of the identified papers contain manipulations on western blots". [7]

4.1 Image Extraction

"A typical PDF file contains many thousands of objects, multiple compression mechanisms, different font formats, and a mixture of vector and raster graphics together with a wide variety of metadata and ancillary content"[8]. All image from a PDF are stored in a stream dictionary embedded in the PDF document. This dictionary includes metadata and a reference to a table where the image is located (*xref* table). This metadata contains the image position, compression type, and others information used by a PDF reader software. To extract an image we use a free-software library *PyMUPDF*[9] to parse and locate the dictionaries containing each image of each page. During the image extraction process we notice two main issues that were corrupting the stream dictionary and causing miss interpretation:

1. Stencil Mask issue
2. One Image - Multiple Objects issue

4.1.1 Stencil Mask

The Stencil Mask issue was noticed during the extraction of images with (alpha) transparency layer. Investigating this issue, we discovered that a lot of software (e.g. pdfimages[6]) that claimed to extract images from PDF, also extract those images in a corrupted way. This issue was caused by a stencil mask that represents the transparency of the image in another band, showed in Figure 4. This additional bits layer was not saved along with the other images bands, and it was confusing all of the image extractors that we tested.

In order to address this issue, we follow the documentation of *PyMuPDF* [9]. But, instead of adding the stencil mask on the image, we discard its transparency. Hence, the extraction result is an RGB image, which does not affect the next steps of the proposed pipeline. Figure 5 shows a comparison of the standard extraction and our solution for the image extraction solving the stencil-mask issue.

4.1.2 One Image - Multiple Objects

During an experiment with a set of files in PDF format, we noticed that some of the standard software extract were cropping the extracted images into several slices, as pointed by Figure 6. Inspecting the PDF files, we noticed that the stream object of the corrupted image was divided into multiples ones. Therefore, in order to increase corruption robustness to the extraction, we developed a method that considers the object neighborhood. Hence, during the image extraction process, we consider the location of each image on a page. If the

[6]Linux Portable Document Format (PDF) image extractor

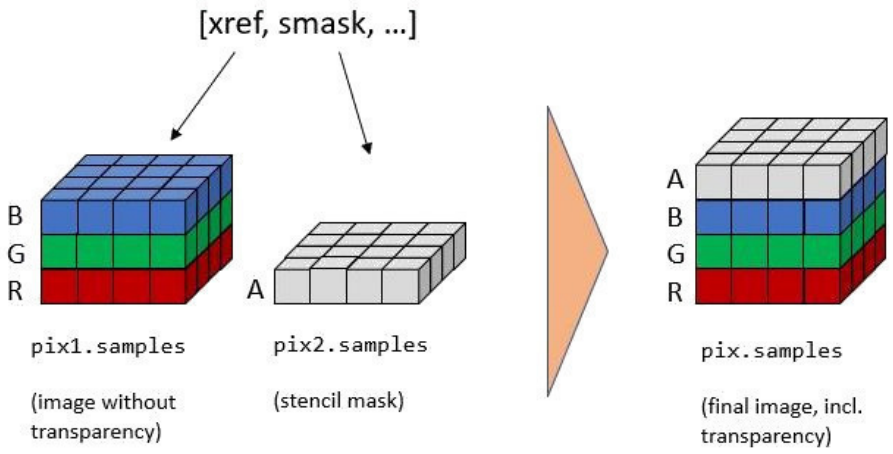


Figure 4: Stencil Mask image showing the transparency bits layer of an RGB image (PyMuPDF Documentation 2019)[10]

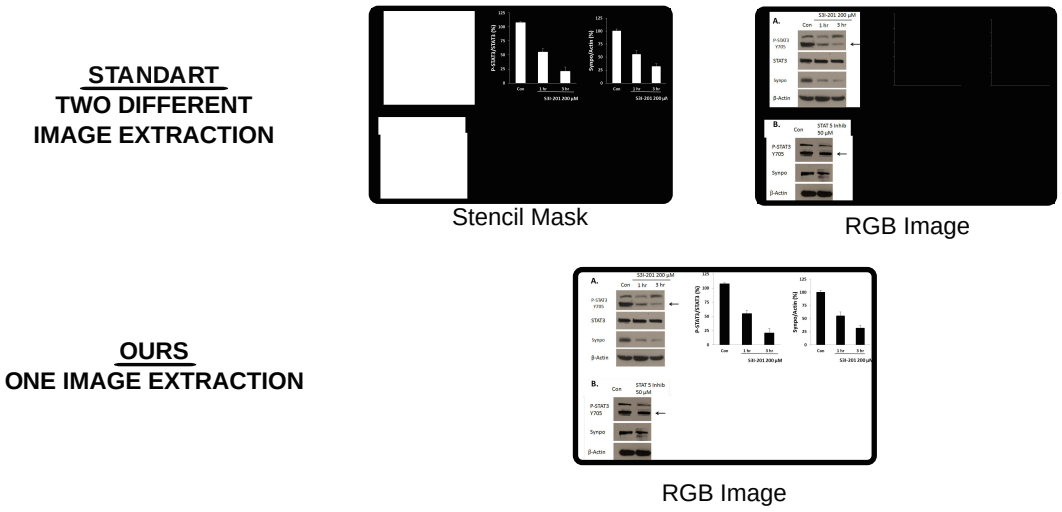
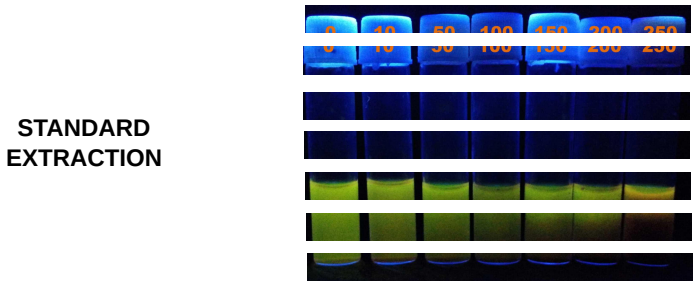
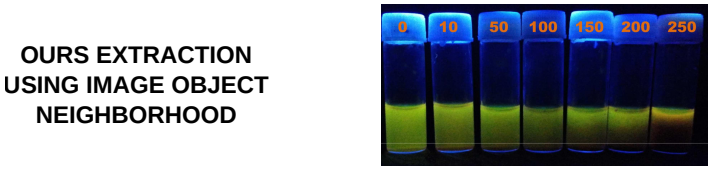


Figure 5: Comparison of standard and ours method extracting images with alpha layers

distance between two images are less than a threshold ϵ we merge them into a single one. We apply this process in a recursive fashion, considering the merged image position to the next iterations. The full method is presented in the pseudo-algorithm 1.



(a) Seven different object to represent the same image



(b) One merged object to represent the image

Figure 6: Comparison of standard methods and the neighborhood based proposed.

Algorithm 1 Image Extraction

```
1: pdf : Input PDF file
2: imageExtractionSet : Output Images
3: procedure ImageExtraction(pdf)
4:     imageExtractionSet ← {}
5:     for page in pdf.pages do
6:         imagePageSet ← {}
7:         for imgA in page.ImgObjects do
8:             if imgA ∈ ARGB then
9:                 Discart band A                          ▷ Image has stencil mask
10:            end if
11:            for imgB in imagePageSet do
12:                if |imgA.position − imgB.position| < ϵ then
13:                    imagePageSet ← imagePageSet \ {imgB}
14:                    imgA ← MergeObjects(imgA, imgB)      ▷ Concatenate images
   horizontally or vertically
15:                end if
16:            end for
17:            imagePageSet ← imagePageSet ∪ {imgA}
18:        end for
19:        imageExtractionSet ← imageExtractionSet ∪ imagePageSet
20:    end for
21:    return imageExtractionSet
22: end procedure
```

4.2 Segmentation

The goal of this step is selecting regions of an image that is semantically interesting to investigate. In other words, we are removing text regions, journal logos, bar error graphs and any other content that could not represent any sort of misconduct from our pipeline. As a consequence of this filtering, we expect to increase efficiency and effectiveness to the framework, since it will discard images that could confuse the copy-move detectors. The image segmentation process is divided into two steps: **Preprocessing**(4.2.1) and **Classification**(4.2.2) .

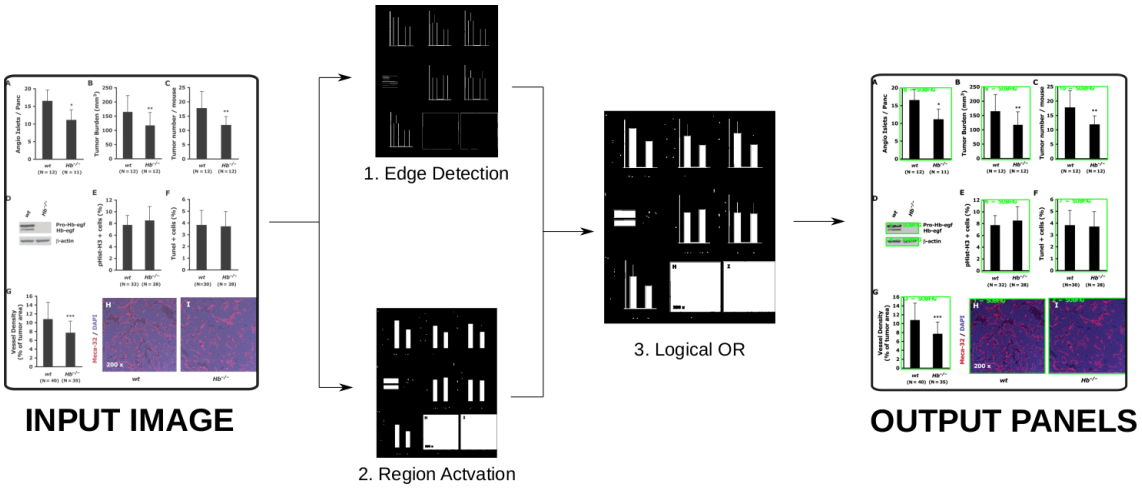


Figure 7: Segmentation Pre-processing

4.2.1 Pre-processing

During the segmentation pre-processing we aim to separate the panels from the background, all steps are summarized in the Figure 7.

Firstly, we check the histogram of an image to discover if the input image has or does not have a background. Our heuristic is:

- ☐ If the highest intensity of the histogram is greater than α (in our case $\alpha = 230$) and its frequency in the histogram is greater then f (in our case $f = 5\%$), the input image has a background.

If the image does not have a background, then we assume that the whole image is a panel and it goes directly to classification step.

If the image has background, we locate its the inside panels. For this, we activate all edges (Fig 7.1) and object regions using morphological operations (Fig 7.2). After this, we build an activation map doing a *logical OR* with the edges and the activated regions (Fig 7.3). At the end of the pre-processing, we find all contours of a connected component from the activation map, resulting in each subfigure panel of the image.

4.2.2 Classification

The classification process uses three trained *CNN* (Convolution Neural Networks) and a *SVM* (Support Vector Machine) model to classify each subfigure as interesting or not. In this subsection, we will explain the training process of each CNN and the SVM model.

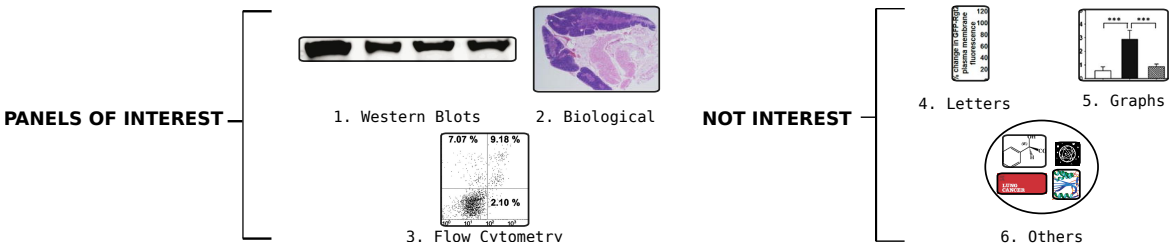


Figure 8: Panels Classes

Panel Classes

To address the problem of finding interesting panels, we define six classes that aim to represent the diversity of a biomedical paper panel, as shown in Figure 8.

1. Western Blots
2. Biological – any part of a biological system
3. Flow cytometry images (FACS analysis)
4. Letters
5. Graphs
6. Others – includes drawings, journal logos etc

CNN Training

Our proposed CNN is a fine-tuned version of VGG-16 [11] CNN pre-trained on the dataset ImageNet [12]. To address the fine-tuned process we remove the last layer (classification) of the network and insert a layer of size six - number of classes of the problem - using softmax as the activation function.

In contrast to the CNN input (224×224), the size and aspect ratio of the panels varies on a range from 20 to 10000 on each dimension. Due to this variation, we notice that the process of resizing was adding a lot of artifacts and confusing the CNN. To solve this issue, we used three different models that were trained with different input resize approaches. Figure 9 summarizes the CNN training process.

1. **Stretched**: Resize the input image to 224×224 using bilinear interpolation, not preserving the original aspect ratio.

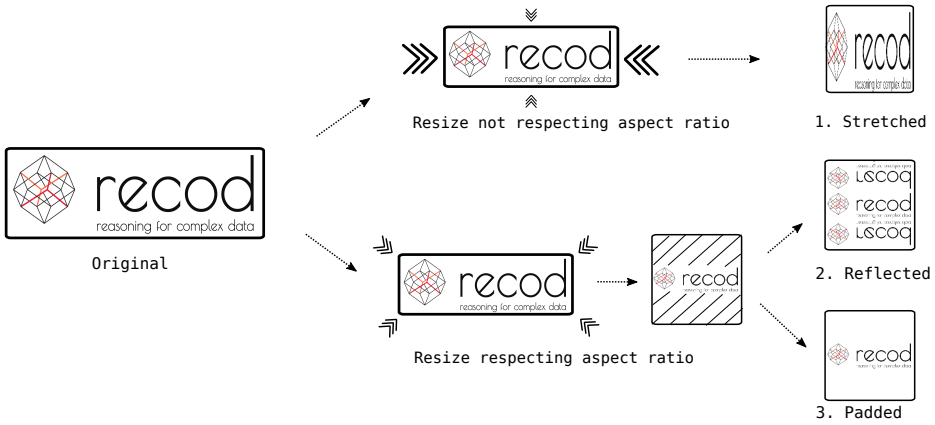


Figure 9: Resizes approach to each CNN input

2. **Reflected**: The first step to this process is resizing the image proportionally so that the large dimension of the image turns 224. After this, we centralize the image on a sketch image that is 224×224, and reflect the smaller dimension in order to fill the rest of the sketch.

3. **Padded**: This process is similar to the reflected one, but instead of filling the sketch reflecting the image, we fill the sketch with a white background.

SVM Training

After training the three CNNs models, we trained an SVM model with kernel rbf on top of them, which received as input the feature vector from the decision layer of each CNN. The SVM model predict two classes (Fig 10):

1. **Panel of Interest** composed by Biological images, Western Blots and Flow Cytometry

2. **Not Interest** composed by Letters, Graphs and Others.

4.3 Clustering

Due to the time complexity of the detection step, clustering is crucial for scalability of our proposed pipeline (Fig 3), since it limits the number of analyzed images to the most similar ones.

To perform clustering we use a feature extracted from the last *Fully Connected* layer from the Reflected CNN (classification step 4.2.2). With these features, we measure pairwise

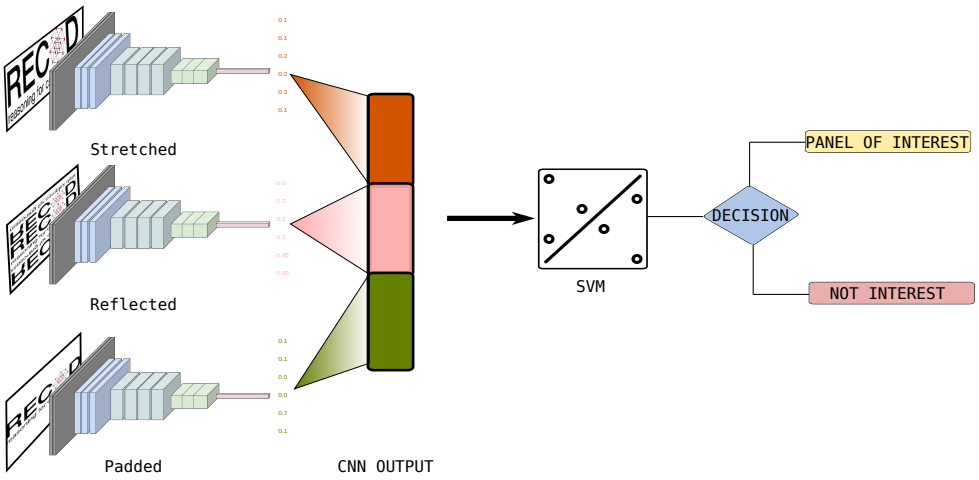


Figure 10: Proposed model for classification of scientific article panel

similarity using the cosine similarity. With this, we build a rank and we assume that all top-k (in our case $k = 50$) retrieval results are in the same cluster. The cosine similarity between two vectors $\boldsymbol{A}$ and $\boldsymbol{B}$ is defined as:

$$similarity = cos(\theta) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

4.4 Detection

After reviewing the literature of copy-move forgery detection (*CMFD*), we find the work of Christlein et al. [13] with an evaluation of *CMFD* state-of-art approaches. The literature describes a generic pipeline of a *CMFD* detector as presented in Figure 11, which outputs a binary image with the suspected *CMFD* region. The pre-processing (Fig 11.1) usually consists of denoising and converting the image to grayscale. Two different approaches to select regions can be done, after the pre-processing step. The first one detects the keypoints of an input (Fig 11.2.1), using the regions of high entropy, and selects the adjacent region of each keypoint. The second approach divides the image into several rectangular regions as if they were tiling on a wall (Fig 11.2.2). For each region a feature vector is extracted using one of the methods of the list:

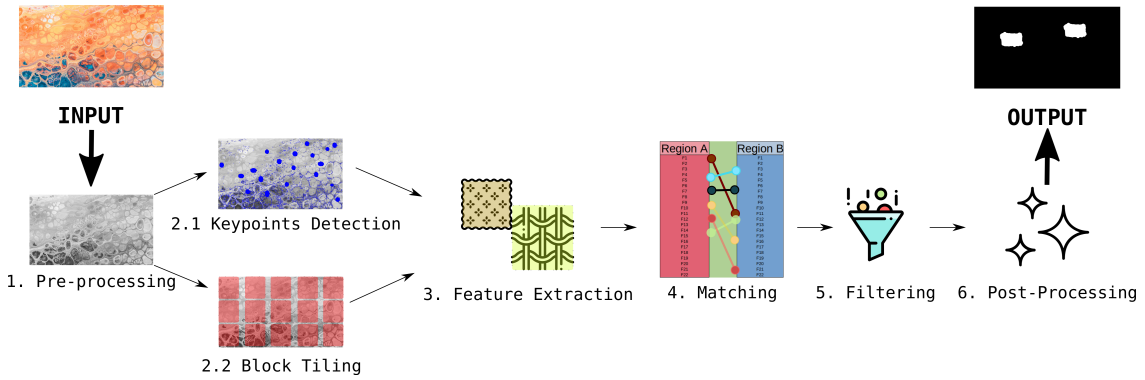


Figure 11: Typical processing pipeline of copy-move forgery detection. The output is a binary mask where suspected regions is highlighted with white.

1. Keypoints-based
   (a) Speed Up Robust Features - SURF [14]
   (b) Multi-scale analysis and voting processes of a digital image with SURF features - SILVA [15]
   (c) Scale Invariant Feature Transform - SIFT [16]
2. Block-based
   (a) Blur moments invariant - *BLUR* [17]
   (b) Circle Block - *CIRCLE* [18]
   (c) Discrete cosine transform - *DCT* [19]
   (d) Discrete wavelet transform - *DWT* [20]
   (e) Fourier-Mellin Transform - *FMT* [21]
   (f) Average grayscale intensities of a block and its sub-blocks - *LIN* [22]
   (g) Zernike moments - *ZERINKE* [23]

The matching step uses the similarity of features extracted by the same method on different regions to indicate a suspect area (Fig 11.4). To remove some false positives, a filtering process is applied to the matched features, discarding spatially close regions that have similar content (Fig 11.5). At the end of the pipeline, post-processing is done to enhance the suspect regions with common behavior and delete outliers, generating the binary output (Fig 11.6).

CMFD Fusion Voting

We proposed a fusion method with the *CMFD* state-of-art approaches to maximize their performance. Our proposed Fusion Voting approach (Figure 12) interprets the binary output as a voting map, where suspect regions have voting 1 and non-suspect 0. Aiming to add scale robustness we perform a detection on four scaled versions of the input image. Each

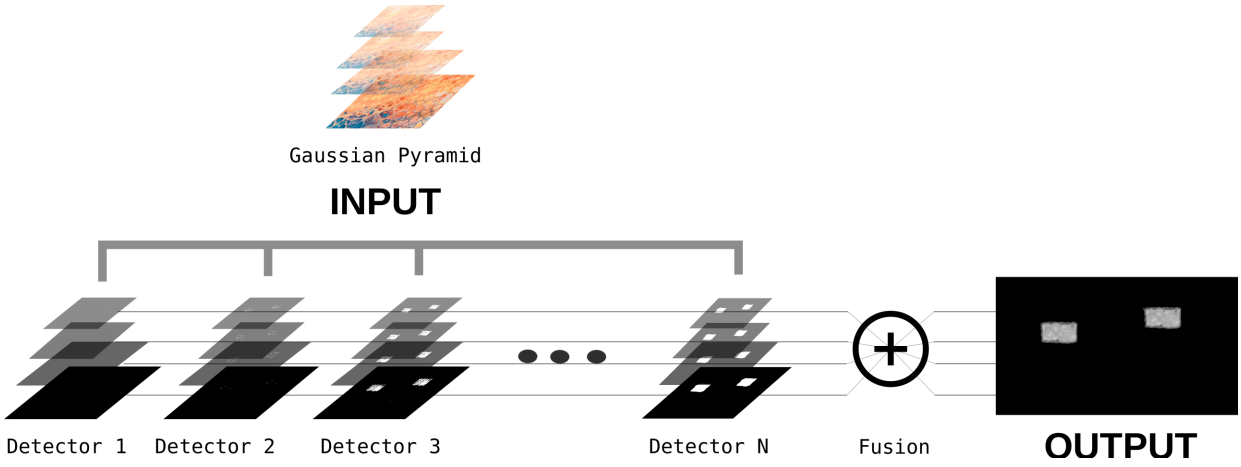


Figure 12: Fusion proposed method using a Gaussian Pyramid from the input and N detectors.

one is scaled using a Gaussian pyramid with scales $0.6, 0.8, 1, 1.2$. Therefore, for each detector, we have four different voting maps for each pixel. The outputs of each different scale are rescaled to the original one. After the fusion process of all binary maps, we generate a grayscale image where each pixel represents the number of votes received. Pixels with less than 20% of votes are discarded.

5 Experiments

In order to evaluate the proposed pipeline (Fig 3), we made a set of experiments for each module. This section is divided into five subsections, the first four address the validation of each module and the last one shows a qualitative example of the whole framework.

5.1 Image Extraction

We collect 535 biomedical articles in PDF format. With this collection, we perform image extraction using four different approaches:

1. **Safe**: This method addresses the problem of multiple image referencing and incorrect declared. For this, it extracts images declared on the reference table, just once. Also, this solution solves the stencil-mask issue without adding the neighborhood-based extraction solution explained on section 4.1.2.

2. **Neighborhood-based**: This method address both the stencil-mask and one-image multiple-objects issues. Since this method aims to fix damaged images, it gets all

image objects even it was not declared or it was declared more than once in the reference table.

3. **Unsafe**: This method extracts all image objects embedded on a PDF document, even it was not declared on the reference table or was referenced more than once. This method does not apply any solution to solve the stencil-mask neither the one-image multiple-objects issue.

4. *PDFImages*: This approach uses the PDFImages tool provided from poppler-utils "a rendering library based on the xpdf-3.0 code base" [24]. Pdfimages is a popular and free software to extract images from pdfs. The version used in the experiments was 0.62.0.

Effectiveness

Aiming to compare the effectiveness of the proposed approaches and the Pdfimages, we randomly choose 178 PDF documents from the collection of biomedical articles. In order to decide if the image extraction was successful or not, we were guided by the following protocol:

- If all extracted images were similar to the figures available online at the official repository of the article, then we considerate it as **successful** extraction.
- If all relevant panels of the paper were extracted even that the figure was divided into several images panes, then we considerate it as **successful** extraction.
- If besides the correct figures more images not mentioned on the official repository of the article were extracted, then we considerate it as **successful** extraction.
- If there was a cut panel or a stencil-mask issue, then we considerate it as **failed** extraction.
- Any other case, we considered it as **failed** extraction.

With this protocol, the **Safe** technique had the best result (Fig 13 just missing images that had their panels cut because of corruption data.

Although the high score of the **Neighborhood-based** method, we noticed that an error was generating duplicated panels. This error was caused by images that were not included in the reference table of the PDF. We noticed that due to corruption data some object images were placed twice in the same position, causing the false duplication. The **Unsafe** technique, as expected was the worst of all. All errors that appeared in the other methods

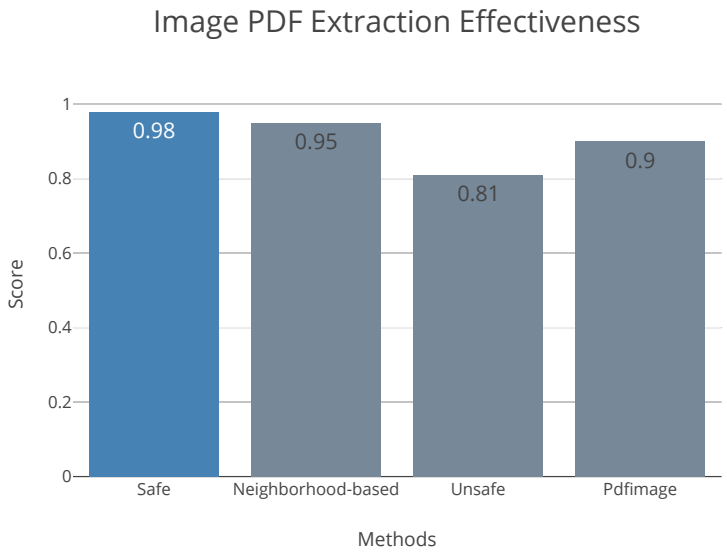


Figure 13: Comparison of image extraction effectiveness for each method.

also appear in this one. The most common error in this approach was the stencil mask issue. In spite of the Pdfimages tool had a high score, it did not reach the Neighbor-based and Safe methods. We notice errors of false duplicate panels and stencil-mask issue on its results.

Efficiency

We performed an efficiency comparison of the Safe, the Neighbor-based, and the Pdfimages approaches. All experiments were executed five times in the same machine with twenty-four CPUs Intel(R) Xeon(R) Gold 6126 CPU @ 2.60GHz and memory of $377GB$. Figure 14 shows the average extraction runtime of each procedure applied on the dataset of 535 PDF documents. To visualize the result of the experiment, we sort the dataset in increase runtime order by the Safe approach. Hence, the index 1 of the dataset represents the PDF with the lowest average extraction runtime from the Safe method and index 535, the highest. The dataset samples are represented by their index from [1 - 535] on the x-axis of Figure 14. The points in Figure 14 represent the difference from the Neighbor-based and the Pdfimages average extraction runtime to the Safe technique. The red points are the cases in which the Safe method is worse than the other approaches, and the blue points, in which it is better. We noticed that the Safe procedure had the best average extraction runtime in the massive majority of the collection.

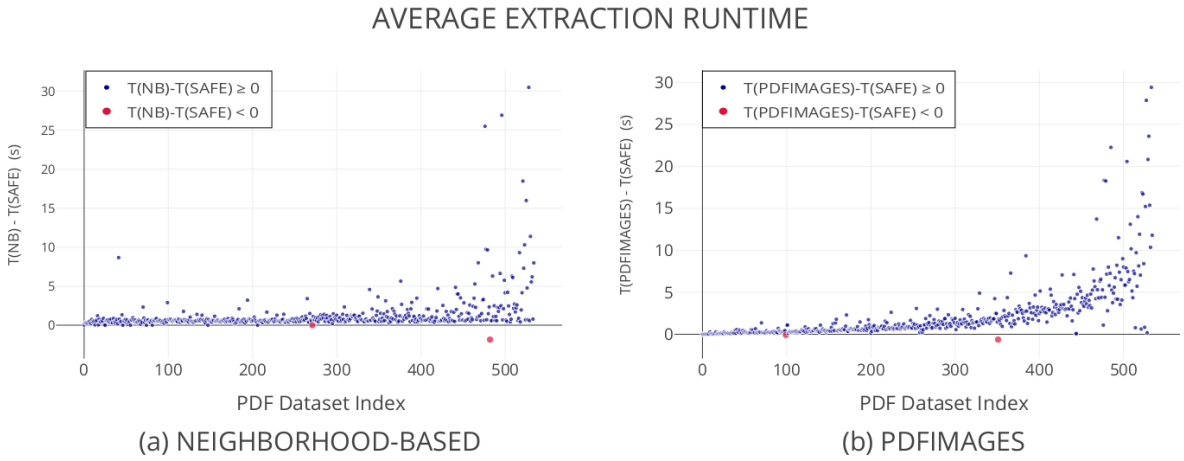


Figure 14: Average extraction runtime comparison between (a) Neighborhood-based and Safe ;(b) Pdfimages and Safe. For each point in the graphs (a) and (b) the average runtime was subtracted from the average of Safe procedure in the same document. T(NB), T(SAFE), and T(PDFIMAGES) represent the average extraction runtime in seconds of Neighborhood-based, Safe, and Pdfimage, respectively.

5.2 Segmentation

To validate the segmentation step we perform a panel classification experiment, with panels pre-processed using the method described in section 4.2.1. This experiment uses individually the CNNs Stretched, Reflected, Padded explained in section 4.2.2 and the fusion of all CNNs using an SVM model (Fig 10). During the experiments, a panel was predicted as Panel of Interest (Biological/Western blot/Flow cytometry) or Not Interest (Letters/Graphs/Others) as described in Figure 8.

Classification Metric

The metric used in this experiment was the accuracy, which is the sum of True Positives with True Negatives divided by the total of samples.

$$ACC = \frac{TruePositive(TP) + TrueNegative(TN)}{TOTAL}$$

Classification Dataset

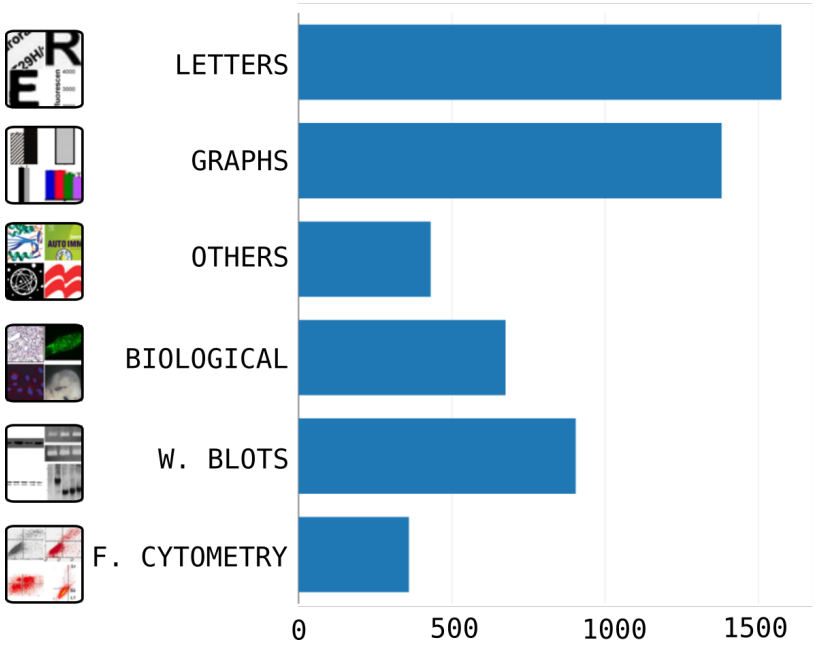


Figure 15: Distribution of samples on test set for the Classification step

The dataset is composed by $30,000$ images for training and $5,325$ for testing. All images were extracted with the pre-processing step (subsection 4.2.1, and manually labeled. The images from the training are equally divided per class. We performance data augmentation on train data using standard normalization, horizontal flip and height, and width shift. The test set is not equally divided, and its distribution is described in figure 15.

Classification Results

In Table 1, **I** indicates Panel of Interest and **NI** Not Interest. We noticed with the presented table that the model with SVM was the best on the test set (0.98). In addition to that, the SVM model had the lowest rate of False Negative.

Table 1: Classification results on test set

CNN Reflected			
		Predicted	
Ground-truth		I	NI
	I	1881	78
	NI	82	3304
ACC		**0.97**	

CNN Stretched			
		Predicted	
Ground-truth		I	NI
	I	1886	53
	NI	350	3036
ACC		**0.9243**	

CNN Padded			
		Predicted	
Ground-truth		I	NI
	I	1889	50
	NI	69	3317
ACC		**0.977**	

SVM ON TOP			
		Predicted	
Ground-truth		I	NI
	I	1908	31
	NI	67	3319
ACC		**0.9816**	

5.3 Clustering

To validate our clustering method, we used t-SNE (t-Distributed Stochastic Neighbor Embedding) [25] and PCA (Principal Component Analysis) to visualize the ranked features. In addition, we measure the similarity between the features extracted from the Reflected CNN for each image. For this, we use the cosine similarity distance explained in section 4.3.

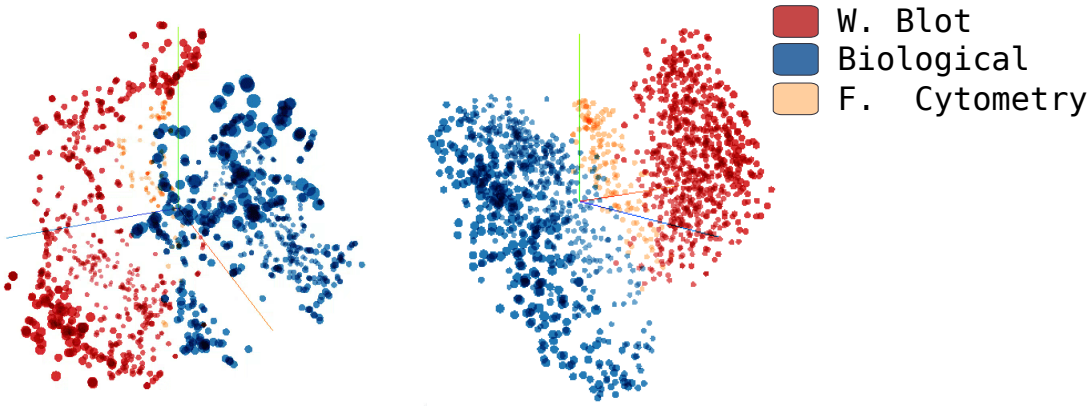
Dataset

We collected a total of 1528 images from the Panel of Interest class. From this total, 694 images were from Western Blot class; 141 from Flow Cytometry and 693 from Biological.

Results Clustering

Figure 16 shows a 3D visualization of the Clustering dataset. We noticed that the data was well divided into clusters from the same class in both methods applied.

Figure 17 shows a pairwise cosine distance from the Clustering dataset. We structured the dataset so that features with index $0 - 693$ are Western Blots; $694 - 835$ are Flow Cytometry and $836 - 1528$ are Biological. With this structured data, we expected to highlight three different areas on the pairwise distance matrix. This expected area is noticed



(a) t-SNE (b) PCA

Figure 16: Features projection of panel of interest using (a) t-SNE and (b) PCA.

in Figure 17. This result gives another clue for the spatial arrangement of features in the space.

Based on the experiments of clustering we concluded that the features can be cluster by class similarity, as we were intended to do.

5.4 CMFD Detection

To analyze the copy-move forgery detectors described in section 4.4 we compared the output of each scaled and fused output described in section 4.4 with its ground-truth. The metric used in this experiment was intersection over union **IoU** (Fig 18). The **IoU** measures if the output mask is at the same location of the ground-truth (using intersection) and penalizes each occurrence of false-positive (using union).

CMFD Detectors

The experiments with the CMFD detectors used an existent implemented version of all detectors. The implementation of [SURF, BLUR, CIRCLE, DCT, DWT, FMT, LIN] was provided from the Christlein et al. [13] CMFD library [27]. The implementation of [SILVA] was produced by Silva et al. [15]. For the [SIFT, ZERNIKE], we used the implementation of Ehret [28], which was based on the work of Cozzolino et al. [29] on rotation-invariant features computed densely on images.

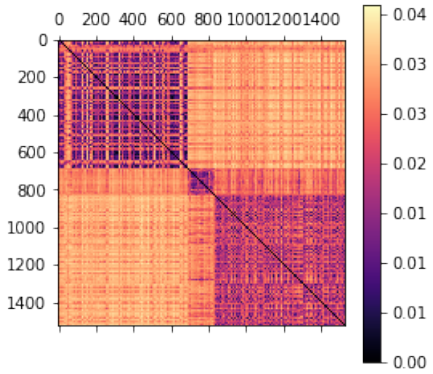
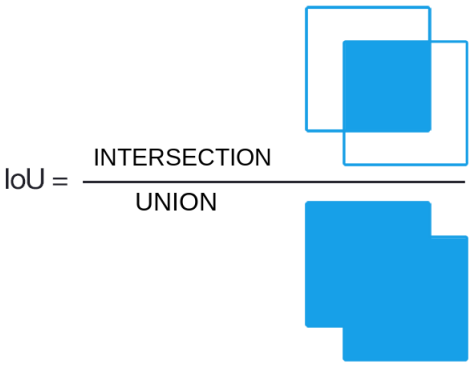
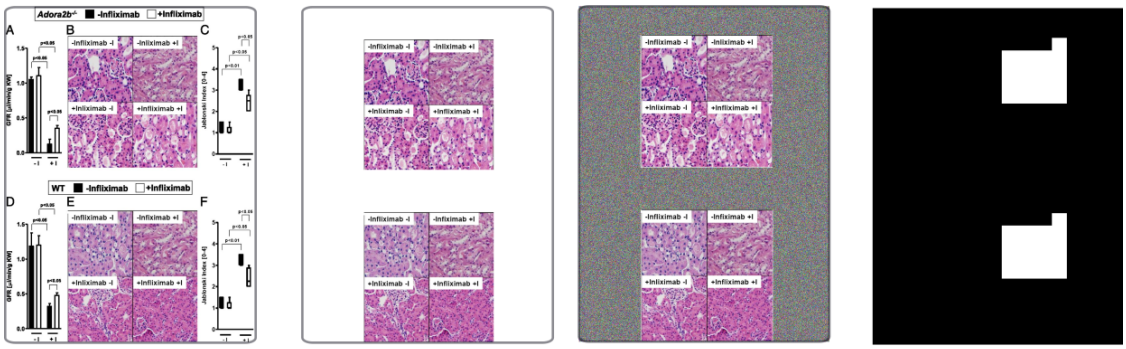


Figure 17: Pairwise cosine distance matrix. The color intensity of each cell is brighter as the distances between two features i, j are higher, where i is the index feature of the row and j index of the column. Index $0 - 693$ are Western Blots features, $694 - 835$ Flow Cytometry and $836 - 1528$ Biological.



Found on https://www.pyimagesearch.com [26]

Figure 18: IoU Metric.



(a) Original (b) White Background (c) Random Background (d) Ground-truth

Figure 19: Sample of CMFD dataset. (a) Original version without segmentation. (b) Segmented version with non-relevant area colored with white. (c) Segmented version with non-relevant area colored with random noise. (d) CMFD Ground-truth.

Dataset

In partnership with the University of Naples Federico II (*Unina*), we created a dataset for CMFD. This dataset has a hundred images with binary masks labeled pixel-by-pixel. White pixels were labeled as suspected and black as non-suspected (Fig 19). To measure the impact of the proposed segmentation step 4.2.2 on this dataset, we created a segmented version of this dataset, using our segmentation solution. To be comparable with the ground-truth, we painted the not relevant area with two types of background.

- White Background
  Pixel with intensity 255 was placed on the background for each band of the image (Fig 19-b).

- Random Noise Background
  Random noise on a range [0-255] was placed on the background for each band of the image (Fig 19-c)

CMFD Result

Figure 20 shows the results of all CMFD detectors methods (scaled and fused) for the original dataset without segmentation. Figure 21 shows the results for the dataset segmented with a white background, and Figure 22 with a random noise background.

In both case white background (Fig 21) and random background (Fig 21) the segmentation improves the CMFD detection for the majority of the approaches. This occurred,

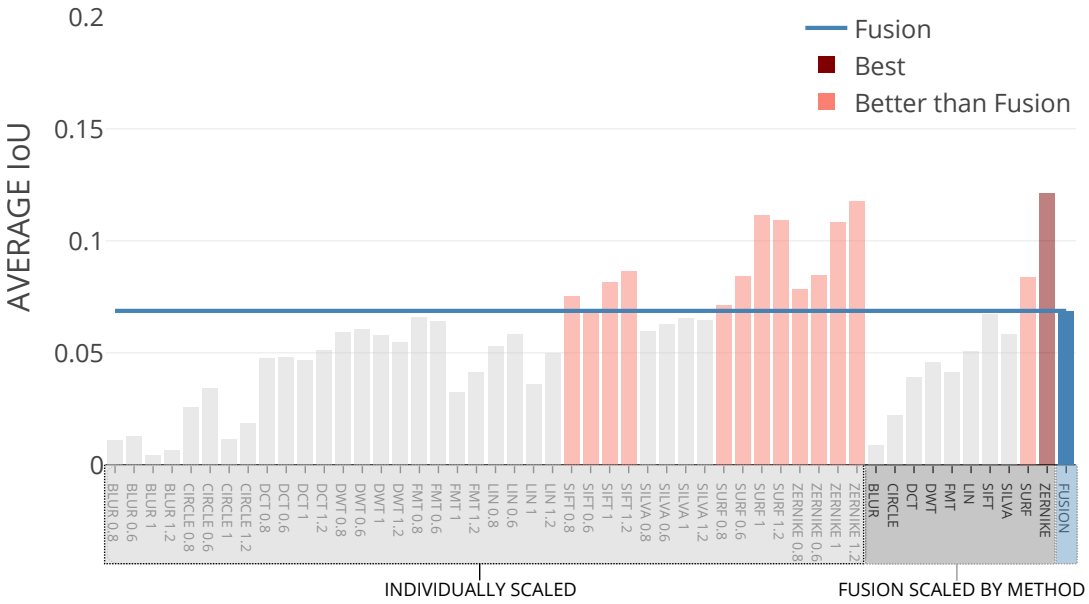


Figure 20: Average IoU for the dataset Without Segmentation. All of the red-colored bars represent techniques that perform better than the Fusion approach.

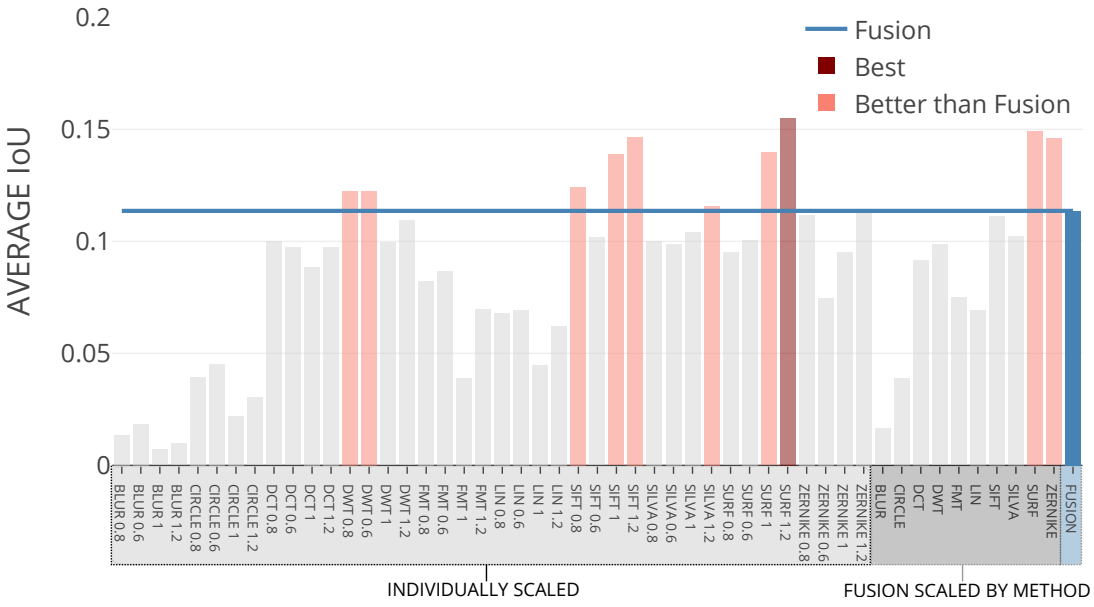


Figure 21: Average IoU for the dataset White Background. All of the red-colored bars represent techniques that perform better than the Fusion approach.

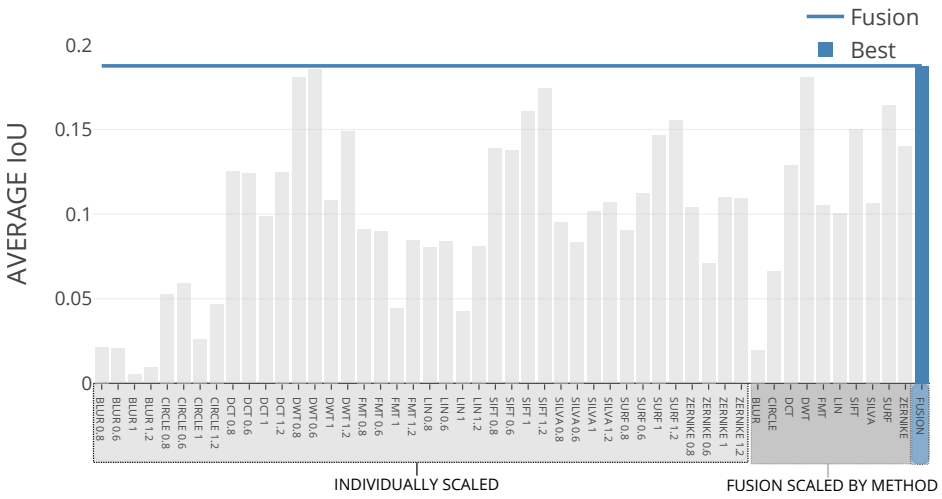


Figure 22: Average IoU for the dataset Random Noise Background.

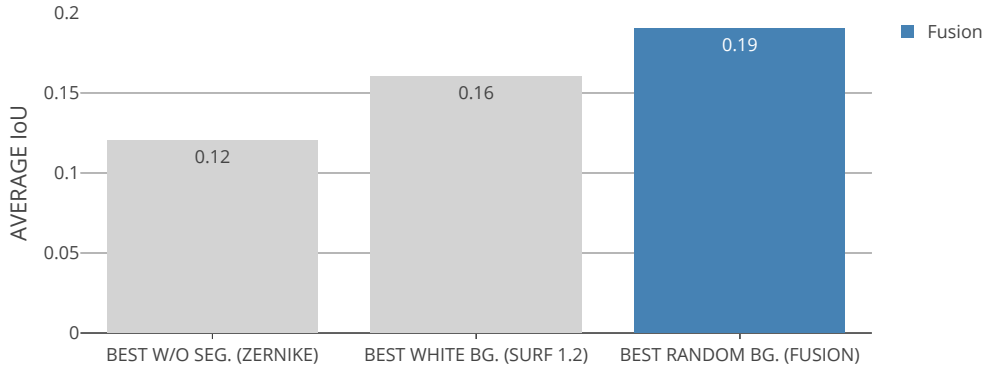


Figure 23: Best average IoU results for the CMFD experiments. The left bar shows the best score for the dataset without segmentation; the middle bar, for the dataset White Background; and the right bar, for the Random Noise Background.

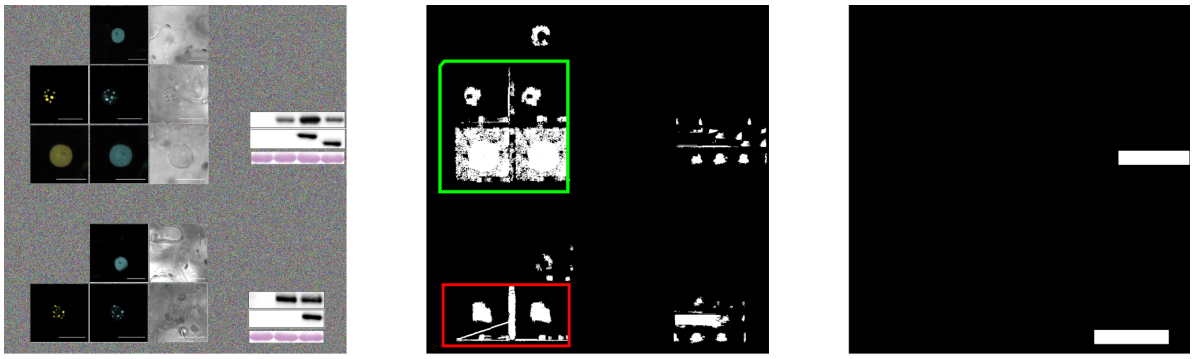


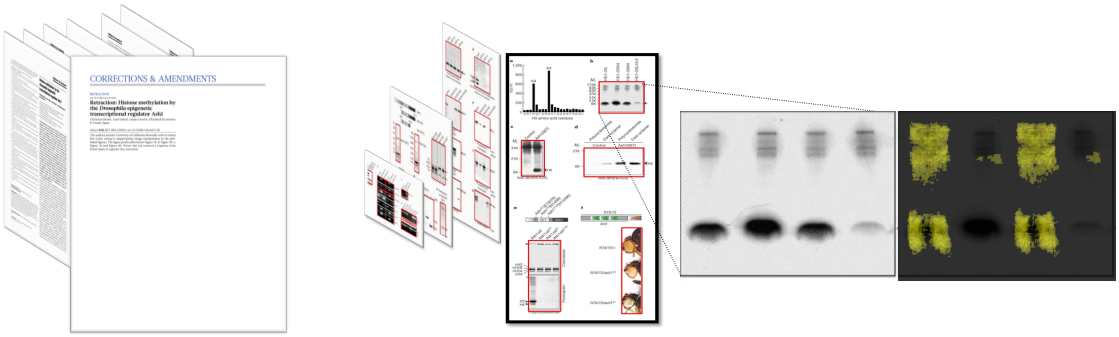
Figure 24: Example of correct detection of duplicate region but wrong semantic assigned. The colored rectangles highlight the regions that were detected, but do not configure misconduct.

because of the decrease of false positive when applied segmentation. This result shows the importance of the segmentation step for the pipeline.

The best average IoU result was achieved with the fusion of all detectors on a segmented version of the dataset with a random noise background (Fig 23). Based on this results we concluded that detection techniques can be fused to increase their robustness. As we noticed in Figure 23, the best average IoU was 0.19. Since IoU is a metric that hard penalizes any type of mismatch between the output and the ground-truth, we expected for CMFD problem a low score, due to false-positives. In favor of this argument, we check the output of some images and noticed that some outputs were correctly detecting legit duplicated regions, which are not considered as misconduct. This type of detection decreases the average IoU, due to a semantic meaning of the region that the detectors are not able to predict. Figure 24 shows an example of this false-positive occurrence. The red and green rectangles of the image show regions that were highlighted with different fluorescent reagents but have the exactly same shape. These regions are a different visualization of the biological experiment, and do not configure misconduct.

5.5 Entire Framework Experiment

To validate the entire framework, we input a paper that was charged with fraud by the Office of Research Integrity [30]. The chosen article was a Nature paper [31] with some problematic images. A PDF version of [31] was input on the framework (Fig 25a). The method correctly extracts its images. For each image the segmentation process selected the relevant regions highlighted by the red rectangle in Figure 25b. Each highlighted image was



(a) INPUT PDF (b) EXTRACTED AND SEGMENTED IMAGES (c) OUTPUT

Figure 25: Full framework working on a real case. (a) The input is the article [31]. (b) The results of the segmentation are represented with the red rectangles. (c) The Fusion detection method outputs the yellow area highlighted. This area is the same as described in the retraction notice.

directly used on the detection step. The main output that calls our attention is the image showed in Figure 25c. This figure is the same one described on the retraction notice [31] and one of the charging reasons [30].

6 Conclusion and Future Works

This work presents a framework for analyzing images from scientific publications. The framework starts with a raw PDF document of a scientific paper and ends highlighting the most suspicious regions of its image. During its pipeline, it performs an image extraction, followed by a segmentation step that filters regions of interest and discards non-related regions to our problem (e.g equations, molecule structure, journal logos). After this, the pipeline performs clustering to group the most similar images, aiming to decrease the run-time of the detection step. After clustering, a copy-paste forgery detection is done in all the similar images, highlighting the suspect regions.

Each step of the framework was individually validated. The proposed image extraction has shown better results in effectiveness and efficiency than the software pdfimages, established as a popular tool to extract images from a PDF.

We validate the Segmentation step using images extracted from biomedical papers. After cropping the panels of each image, a classification of relevance was performed achieving 98% accuracy.

To validate the Clustering step, we visualized the projection of the features used to classify each panel. Visually, we noticed that the samples of the same classes were well-

grouped in the same cluster. Also, using the pairwise cosine distance matrix, we confirm this result.

In this work, we also proposed a fusion of state-of-art CMFD to improve their isolated results. The fusion approach performed better than all the other individual detectors. Using a dataset created in partnership with UNINA, we demonstrate the importance of the Segmentation step for the pipeline, since the best result of the CMFD detectors in average IoU on the dataset without segmentation was 12% and with segmentation was 19%. Along with all these steps validated individually, we setup an experiment using a real case of image misconduct [31]. We were able to highlight the same issue described in its retraction notice.

As future work, an interesting area of research would be copy-move detectors specialized in scientific images. Since we used an implemented version of CMFD detectors for natural images, we believe that a specialized version will perform better.

Since this work deals with a quite new area in the literature, we could not find any mention to an establish dataset or metric to evaluate the proposed framework as a whole. Hence, as future work, we indicate the construction of a big dataset and metric to measure the proposed framework. We believe that a well-structured dataset will motivate the designing of new CMFD detectors and fostering innovative works in the area of scientific integrity.

7 Acknowledgments

We would like to thank *Fundação de Amparo à Pesquisa do Estado de São Paulo* (FAPESP) process 2018/15864-4 and Advanced Research Projects Agency (DARPA) for supporting this research. In addition, we would like to thank the entire team of RECOD and the Scientific Integrity Project of MediFor.

References

[1] D. Fanelli, "How many scientists fabricate and falsify research? a systematic review and meta-analysis of survey data", *PLoS ONE*, vol. 4, no. 5, T. Tregenza, Ed., e5738, 2009. DOI: 10.1371/journal.pone.0005738. [Online]. Available: https://doi.org/10.1371/journal.pone.0005738.

[2] M. P. Oksvold, "Incidence of data duplications in a randomly selected pool of life science publications", *Science and Engineering Ethics*, vol. 22, no. 2, pp. 487–496, 2015. DOI: 10.1007/s11948-015-9668-7. [Online]. Available: https://doi.org/10.1007/s11948-015-9668-7.

[3] F. C. Fang, R. G. Steen, and A. Casadevall, "Misconduct accounts for the majority of retracted scientific publications", *Proceedings of the National Academy of Sciences*, vol. 109, no. 42, pp. 17 028–17 033, 2012. DOI: 10.1073/pnas.1212247109. [Online]. Available: https://doi.org/10.1073/pnas.1212247109.

[4] E. M. Bik, A. Casadevall, and F. C. Fang, "The prevalence of inappropriate image duplication in biomedical research publications", *mBio*, vol. 7, no. 3, 2016. DOI: 10.1128/mbio.00809-16. [Online]. Available: https://doi.org/10.1128/mbio.00809-16.

[5] T. office of Research Integrity. (2019). Definition of research misconduct, [Online]. Available: https://ori.hhs.gov/definition-misconduct (visited on 09/30/2019).

[6] D. E. Acuna, P. S. Brookes, and K. P. Kording, "Bioscience-scale automated detection of figure element reuse", 2018. DOI: 10.1101/269415. [Online]. Available: https://doi.org/10.1101/269415.

[7] E. M. Bucci, "Automatic detection of image manipulations in the biomedical literature", *Cell Death & Disease*, vol. 9, no. 3, 2018. DOI: 10.1038/s41419-018-0430-3. [Online]. Available: https://doi.org/10.1038/s41419-018-0430-3.

[8] J. Whitington, *PDF explained*. OReilly Media, 2012.

[9] J. X. McKie. (2019). Pymupdf documentation, [Online]. Available: https://pymupdf.readthedocs.io/en/latest (visited on 09/30/2019).

[10] J. X. McKie. (2019). How to handle stencil masks, [Online]. Available: https://pymupdf.readthedocs.io/en/latest/faq/#how-to-handle-stencil-masks (visited on 09/30/2019).

[11] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2014. arXiv: 1409.1556 [cs.CV].

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database", in *CVPR09*, 2009.

[13] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou, "An evaluation of popular copy-move forgery detection approaches", *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 6, pp. 1841–1854, 2012. DOI: 10.1109/tifs.2012.2218597. [Online]. Available: https://doi.org/10.1109/tifs.2012.2218597.

[14] B. L. Shivakumar and L. D.S. S. Baboo, "Detection of region duplication forgery in digital images using surf", 2011.

[15] E. Silva, T. Carvalho, A. Ferreira, and A. Rocha, "Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes", *Journal of Visual Communication and Image Representation*, vol. 29, pp. 16–32, 2015. DOI: 10.1016/j.jvcir.2015.01.016. [Online]. Available: https://doi.org/10.1016/j.jvcir.2015.01.016.

[16] H. Huang, W. Guo, and Y. Zhang, "Detection of copy-move forgery in digital images using SIFT algorithm", in *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, 2008. DOI: 10.1109/paciia.2008.240. [Online]. Available: https://doi.org/10.1109/paciia.2008.240.

[17] B. Mahdian and S. Saic, "Detection of copy–move forgery using a method based on blur moment invariants", *Forensic Science International*, vol. 171, no. 2-3, pp. 180–189, 2007. DOI: 10.1016/j.forsciint.2006.11.002. [Online]. Available: https://doi.org/10.1016/j.forsciint.2006.11.002.

[18] J. Wang, G. Liu, H. Li, Y. Dai, and Z. Wang, "Detection of image region duplication forgery using model with circle block", in *IEEE International Conference on Multimedia Information Networking and Security*, IEEE, 2009. DOI: 10.1109/mines.2009.142. [Online]. Available: https://doi.org/10.1109/mines.2009.142.

[19] A. J. Fridrich, B. D. Soukal, and A. J. Lukáš, "Detection of copy-move forgery in digital images", in *in Proceedings of Digital Forensic Research Workshop*, 2003.

[20] M. Bashar, K. Noda, N. Ohnishi, and K. Mori, "Exploring duplicated regions in natural images", *IEEE Transactions on Image Processing*, 2010. DOI: 10.1109/tip.2010.2046599. [Online]. Available: https://doi.org/10.1109/tip.2010.2046599.

[21] S. Bayram, H. T. Sencar, and N. Memon, "An efficient and robust method for detecting copy-move forgery", in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2009. DOI: 10.1109/icassp.2009.4959768. [Online]. Available: https://doi.org/10.1109/icassp.2009.4959768.

[22] H.-J. Lin, C.-W. Wang, and Y.-T. Kao, "Fast copy-move forgery detection", *WSEAS Trans. Sig. Proc.*, vol. 5, no. 5, pp. 188–197, 2009, ISSN: 1790-5052. [Online]. Available: http://dl.acm.org/citation.cfm?id=1639329.1639332.

[23] S.-J. Ryu, M.-J. Lee, and H.-K. Lee, "Detection of copy-rotate-move forgery using zernike moments", in *Information Hiding*, R. Böhme, P. W. L. Fong, and R. Safavi-Naini, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 51–65.

[24] Poppler. (2018). Poppler Homepage, [Online]. Available: https://poppler.freedesktop.org (visited on 09/30/2019).

[25] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE", *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

[26] A. Rosebrock, *Intersection over union (iou) for object detection.* Available: ¡https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection¿, visited on 24/10/2019, 7 Nov. 2016.

[27] V. C. Christian Riess. (2012). Copy-move forgery detectors and ground truth generator, [Online]. Available: https://www5.cs.fau.de/research/groups/computer-vision/image-forensics/evaluation-of-copy-move-forgery-detection/ (visited on 09/30/2019).

[28] T. Ehret, "Automatic Detection of Internal Copy-Move Forgeries in Images Zernike Moments", vol. 8, pp. 167–191, 2018.

[29] D. Cozzolino, G. Poggi, and L. Verdoliva, "Efficient Dense-Field Copy – Move Forgery Detection", *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 11, pp. 2284–2297, 2015. DOI: 10.1109/TIFS.2015.2455334.

[30] T. office of Research Integrity. (2017). Case summary: Sauer, frank, [Online]. Available: https://ori.hhs.gov/case-summary-sauer-frank (visited on 09/30/2019).

[31] C. Beisel, A. Imhof, J. Greene, E. Kremmer, and F. Sauer, "Histone methylation by the drosophila epigenetic transcriptional regulator ash1", *Nature*, vol. 419, no. 6909, pp. 857–862, Oct. 2002. DOI: 10.1038/nature01126. [Online]. Available: https://doi.org/10.1038/nature01126.