Lawsuits Document Images Processing Classification

Chapter · September 2022
DOI: 10.1007/978-3-031-16474-3_4

11 authors, including:

Daniela Lopes Freire
University of São Paulo
26 PUBLICATIONS **66** CITATIONS

Leonardo Carneiro Feltran
University of São Paulo
2 PUBLICATIONS **0** CITATIONS

Kelly Ramos
University of São Paulo
2 PUBLICATIONS **0** CITATIONS

João Eduardo Ferreira
University of São Paulo
152 PUBLICATIONS **1,246** CITATIONS

Lawsuits Document Images Processing Classification

Daniela L. Freire[1(✉)], André Carlos Ponce de Leon Ferreira de Carvalho[1], Leonardo Carneiro Feltran[1], Lara Ayumi Nagamatsu[1], Kelly Cristina Ramos da Silva[1], Claudemir Firmino[1], João Eduardo Ferreira[1], Pedro Losco Takecian[2], Danilo Carlotti[1], Francisco Antonio Cavalcanti Lima[2], and Roberto Mendes Portela[2]

[1] University of Sao Paulo, Sao Paulo, Brazil
{danielalfreire,andre}@icmc.usp.br,
{leonardo.feltran,lara.nagamatsu,kelly.ramos.silva,cfirmino}@usp.br,
{jef,danilopcarlotti}@ime.usp.br

[2] TJSP - Justice Court of São Paulo State, Sao Paulo, Brazil
plt@ime.usp.br

Abstract. Natural Language Processing techniques usually fail to classify low quality lawsuit document images produced by a flatbed scanner or fax machine or even captured by mobile devices, such as smartphones or tablets. As the courts of justice have many lawsuits, the manual detection of classification errors is unfeasible, favouring fraud, such as using the same payment receipt for more than one fee. An alternative to classifying low-quality document images is visual-based methods, which extract features from the images. This article proposes classification models for lawsuit document image processing using transfer learning to train Convolutional Neural Networks most quickly and obtain good results even in smaller databases. We validated our proposal using a TJSP dataset composed of 2,136 unrecognized document images by Natural Language Processing techniques and reached an accuracy above 80% in the proposed models.

Keywords: Document image processing · Image classification · Deep learning techniques · Convolutional neural networks models · Feature extraction

1 Introduction

The São Paulo Court of Justice (in Portuguese, Tribunal de Justiça de São Paulo - TJSP) is the most significant in the volume of cases, with 25% of them in progress in Brazil. To ensure the excellent functioning and efficiency in the processing of these processes, the TJSP has 320 districts in the state and more than 40 thousand employees [26]. The parties must provide all documents necessary for the entry and processing of the lawsuit to the court, including proof of payment of court fees. The parties classify these and then pass through an

G. Marreiros et al. (Eds.): EPIA 2022, LNAI 13566, pp. 41–52, 2022.
https://doi.org/10.1007/978-3-031-16474-3_4

empirical conference by court officials. In many cases, classification errors favour fraud, such as using the same payment receipt for more than one fee. As the TJSP collection is physical and very voluminous, identifying these classification failures becomes unfeasible. Besides, the lawsuit documents are usually low-quality images captured in an unrestrained environment, so they typically have perspective distortions and uneven illumination.

The document image processing pipeline have used to automate the organization and indexing of document images. The first step of the pipeline is document image classification, which defines what class or classes a document belongs to. The survey [17] has been extended from the former search [4], presenting a current and broad literature analysis of document image classification, focusing on mostly-text document images, including non-mobile and mobile document images. The former regards document images produced by a flatbed scanner, fax machine, or converting an electronic document to the image format. In contrast, the latter refers to document images captured by mobile devices, such as smartphones or tablets.

Research works have dealt with problems with document image classification in Law, such as the identification of the parties in legal proceedings [19], classification of documents from lawsuits into administrative labels [3,21,23], and prediction of the lawsuit domain lawsuit [25]. Usually, Natural Language Processing (NLP) tools solve these problems. Meantime the particularity and lack of knowledge of the language used in Law turn this field into a challenge for NLP applications, especially in Brazil, where there is a more extensive jurisdiction with different expressions in the legal language to designate the same reality depending on regions of the country [22]. Furthermore, many legal documents are illegible, so it negatively affects the classification performance carried out by NLP tools since they can extract text containing errors of images with low quality. A solution to the classification of low-quality document images is visual-based methods. They describe the appearance of the document images roughly and catch information that enables one to recognize the documents "at a glance." [7]. One of the visual-based methods is the 'deep features' that use Convolutional Neural Networks (CNNs) to extract features from the images [17] automatically. CNN is a type of artificial neural network where individual neurons are arranged side by side so that they respond to overlapping regions in the visual field [14]. Two components make up a typical convolutional neural network architecture. The first component is faintly connected and is responsible for extracting images' features. The other component is fully connected and is responsible for classifying images using the features that, for this reason, are called deep features. This article proposes two classifications for lawsuit document image processing. The first is a classification model for eliminating images without useful information, and the second is a classification model for splitting lawsuits document images into different kinds of documents. We validated our proposal using a TJSP dataset composed of 2,136 unrecognized document images. The proposed models reached an accuracy above 80% in the proposed models. The rest of this article is organised as follows: Sect. 2 discusses works related to document image classification and retrieving; Sect. 3 provides background information over image processing techniques; Sect. 4 exposes the proposal of a lawsuits document image processing pipeline; Sect. 5 reports experiments and statistics to validate the proposal; and, Sect. 6 presents our conclusions.

2 Related Work

This section gathers works that proposed using artificial intelligence techniques to support the work in the court system. Most works concern the automatic classification of court decisions by distinct techniques. Others predict the legal area from which a case belongs to and make the prediction of the court's decision based only on the textual content of the documents. Besides, we selected works that used Convolutional Neural Networks for classifying documents images or for content-based image recognition.

[25] developed a mean probability ensemble system combining the output of multiple Support Vector Machine (SVM) classifiers that reached 98% average F1 score in predicting a court decision, 96% F1 score for predicting the class of lawsuits, and 87.07% F1 score on estimating the date of a court decision. [3] proposed a classifier of 6,814 lawsuits [23] of the Supreme Court of Brazil into six classes. The authors used a Bidirectional Long Short-Term Memory model that reached an F1 score of 84%, dismissing the need to run an OCR on the remaining pages of the document. [27] mined the comprehensive information of enterprises and extracted features, combined with two models of machine learning. The first model is the combinatorial prediction model using the Light Gradient Boosting Machine (LightGBM) model, and its Top 1 accuracy was 40.868%, while its Top 2 accuracy was 21.826%. The second model is the Artificial Neural Network (ANN) model to classify and predict categories of lawsuits, its Top 1 accuracy is 40.803% and Top 2 accuracy is 21.243%. The dataset used is from the IEEE ISI World Cup 2019, which seventeen classes of lawsuits from 3,500 listed enterprises. [8] performed training of region-based classifiers and ensembling for document image classification. First, the authors used a primary level of 'inter-domain' transfer learning by exporting weights from a pre-trained VGG16 architecture on the ImageNet dataset to train a document classifier on whole document images. After, they studied the nature of region-based influence modelling and trained deep learning models for image segments in a secondary level of 'intra-domain' transfer learning. Finally, they integrated the predictions of the base deep neural network models with a stacked generalization based ensembling. The dataset was composed of a subset of the IIT-CDIP Test Collection known as the RVL-CDIP dataset [13], which consists of scanned grey-scale images of documents from lawsuits against American Tobacco companies, which is segregated into 16 categories or classes. The proposed method achieves state-of-the-art accuracy of 92.21%. [1] used an approach for learning visual features for document analysis in an unsupervised way, increasing the amount of data through a data augmentation technique. First, the authors used the feature extractor to represent document images s for an unsupervised classification task. Then, they used a

small amount of annotated data in the parameters initialization of the model used in a supervised classification task. In their experiments, the authors used the Tobacco-3482 [16], which contains 3,482 document images and 10 document classes and a small subset of the RVL-CDIP dataset [13], which contains 5,000 document images and 16 document classes. The median accuracy reached by the model in the classification task was 68.86%.

3 Background

Deep learning techniques are being used as feature extractors because these techniques are very robust when extracting complex features that express the image in much more detail, very efficient in learning the task-specific features and are very efficient [10,11,18,28]. The advantage of deep learning is that instead of providing the filters for performing convolution over an image to extract various features of an image (such as vertical edges, horizontal edges, noise distribution, son on), the model itself learn the filters and weights during each epoch. For this problem, we have used transfer learning for feature extraction. Transfer learning is a machine learning method that uses the weights of pre-trained models for a given task in models focusing on solving other problems. Due to this previous learning, training these networks takes less time and requires a smaller database to obtain good training.

There are several constructed convolutional neural networks models that use pre-trained models as starting points in computer vision problems, such as VGGNet [24]. VGGNet constructed CNNs with a depth of 16–19 layers. The VGG16 model comprises 13 convolutional layers, 3 fully connected layers, and five pooling layers. The 13 convolutional layers and the three fully connected layers have weight coefficients, composing a structure of 16 weighting layers, so it is called VGG16. The pooling layers do not count because they do not have weights. The model is constructed by piling several convolutional layers and pooling layers. The same convolution kernel parameters are used in every convolutional layer, and the same pooling kernel parameters are used in every pooling layer. The VGG16 structure is divided into blocks, and each block contains several convolutional layers and a pooling layer. In the same block, the number of channels of the convolutional layer is the same. Despite a simple structure, VGG is pretty adaptable because it contains many parameters, reaching 139,357,544 weights to convolution kernel and to fully connected layer. VGG19 is a CNN that comprises 19 layers of 16 convolution layers and 3 layers fully connected to classify the images, with 5 pooling layers. It uses multiple 3×3 filters in each convolutional layer because VGG19 is a popular method for image classification. These models are trained on the ImageNet dataset that contains a million images of 1000 categories.

In this article, we used Keras [15], an open-source software library that provides a Python interface for artificial neural networks and performs as an interface for the TensorFlow library. Keras Applications are deep learning models that are made available alongside pre-trained weights.

4 Proposal

This section describes our proposal of classification for unrecognized lawsuit document images. These documents make up a heterogeneous dataset containing document images with and without helpful information. We consider images without helpful information when they are entirely blurred, stains or small scribbles, while images with helpful information are those in which there is any content like text, figures, tables, and so on. The proposal has two main steps: preprocessing and classification.

4.1 Preprocessing

Preprocessing step has three activities: Format Conversion, Data Augmentation, and Histogram Equalization, as shown in Fig. 1. In Format Conversion, the unrecognized documents images are read from a repository and converted from `.pdf` to `.tiff` format. In contrast to other graphic formats such as JPEG, TIFF has an alpha channel that can store the transparency of individual pixels and colour information. The advantage of this method is the simple and, therefore, fast compression and decompression of such files with lossless quality. In Data Augmentation activity, since the unrecognized document images dataset is usually small (less than 1000 images), the data augmentation increases the dataset by applying several transformations such as rotation, width and height shift, shear, and zoom whitening, brightness, so on.

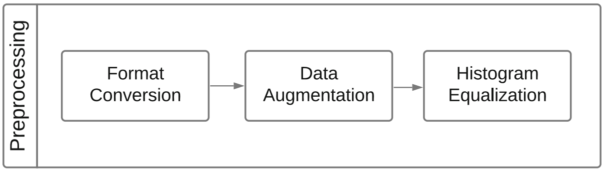

Fig. 1. Preprocessing step.

Then, in Histogram Equalization, we applied techniques to enhance the images' quality. First, we transformed the images into grey-scale and applied a Gaussian filter for smoothing out noises and for edge detection of the images. Simultaneously suppressing noise, the Gaussian filter can cause distortion, edge position shift, edge vanishing and ghost edges [9]. Furthermore, some images produced by data augmentation were distorted. Then, we use Contrast Limited Adaptive Histogram Equalization (CLAHE) technique for the histogram equalization. This technique carries out histogram equalization in small stains with high accuracy and contrast limiting. We used Otsu algorithm [20] to find the optimal value for a binary threshold. Otsu is the most popular global threshold iterative algorithm where the intensity levels are split into background and foreground for all possible intensity values in the image [2]. Then, we dismiss the distorted images by calculating the percentage of black pixels of images binarized (*perc*), i.e., one minus the mean of pixel values divided by 255, because by convention, 0 is usually black, and 255 is white. Since we do not want to lose any image document with any helpful information, we assume a low value for the minimal percentage of black. Therefore, we considered an image with helpful information if it has at least 2% of black pixels, i.e., if an image has $perc < 2$, it is discarded. We use the resultant dataset to create a model for classifying new images into the classes: with and without helpful information.

4.2 Classification

The classification step is performed in two stages: Classification I and II. In the first stage (Classification I), we used the Haralick algorithm [12] to extract textural features from images. The Grey Level Co-occurrence Matrix (GLCM) is the main idea of Haralick Texture features. GLCM looks for pairs of adjacent pixel values that occur in an image and records them over the entire image. There are four types of adjacency: Left-to-Right, Top-to-Bottom, Top Left-to-Bottom Right, and Top Right-to-Bottom Left. First, the Haralick features are extracted for all four types of adjacency after obtaining the mean of all four types of GLCM. Then, the feature vector for the image which describes the texture is returned. The features are then stored as feature vectors in `.npy` format files to be used in the training of machine learning classification models. We created five machine learning classifiers to decide if an image has helpful information or not, namely Linear Support Vector Machine, Random Forest, Decision Tree, Gaussian Naive Bayes and Neural Networks. All models are trained with the Haralick texture vectors. A report with the list of images and their respective class is saved into a file `.csv` format.

The second stage (Classification II) splits images with helpful information into separate document classes because there may be interest in a specific class, such as payment receipt, judicial deposits, others, and trash. We used VGG19 because it is an advanced CNN with pre-trained layers and a great understanding of what defines an image in terms of shape, colour, and structure. We froze its layers and appended a shallow six-layer network on top of it to perform the classification task of identifying images with and without trees. A report with the list of images and their respective class is saved into a file `.csv` format. Figure 2 depicts the classification step.

5 Experiment

This section describes experiments to validation of the proposal of lawsuit document images classification in São Paulo Court of Justice. First, we contextualize how our research integrates the document image processing pipeline of the TJSP. Afterwards, we describe the database and the classification's models, and finally, we present the results obtained.

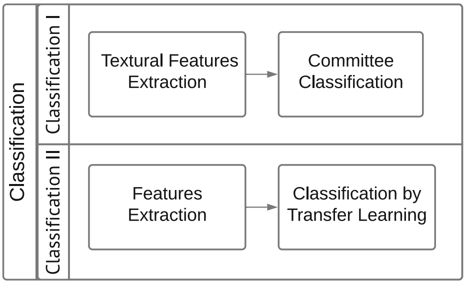

Fig. 2. Classification step.

This experiment is part of the project to identify failures of classification, mainly the reuse of the same payment receipt issued by the TJSP, which gives evidence of fraud in the judicial system. The project's first stage seeks to identify document duplication through information obtained from texts extracted from images of scanned documents, using natural language processing techniques. The proposed pipeline deals with images of payment slip documents from which no information can be extracted and converted into texts. Usually, they are images whose text is unreadable or of low quality, containing much noise, low resolution, or even crumpled and poorly positioned document images.

The dataset used in the research project is composed of 139,603 processes that have 416,316 images of documents belonging to the 4th Civil Court of the Regional Forum XII - Nossa Senhora do Ó, in the city of São Paulo. Only 0.5% of the total number of documents were not recognized by the previous text extraction steps, leaving for our experiment 2,136 unrecognized document images, including lawsuit payment receipt images and other kinds of document images such as personal documents, reports, lawsuit decisions.

5.1 Preprocessing

We used Python Imaging Library (PIL) [5] to convert `.pdf` files to `.tiff` format. PIL adds image processing capabilities to the Python interpreter and provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.

After, we manually labelled the 2,136 unrecognized document images into two classes: with and without helpful information. There were 230 images without helpful information and 1906 with helpful information. We separated 30 images of each class from testing, from which remained 200 images without helpful information and 1,876 with helpful information to train. Therefore, we needed to increase the number of images without helpful information to balance the training dataset.

In the Data Augmentation activity, we automatically used the Keras deep learning library for data augmentation by the ImageDataGenerator class. We used shift, brightness, zoom, and shear techniques to produce new images for each of the 200 images without helpful information. 1,636 new images were produced in this activity, totalling 1,836 images without helpful information.

In the Histogram Equalization activity, we applied the CLAHE technique, c.f., Fig. 3 that shows the same image in three different situations: in greyscale, with Gaussian filter, and with Otsu's Thresholding optimization. It resulted in 511 distorted images that were discarded, with 1,325 images without useful information left.

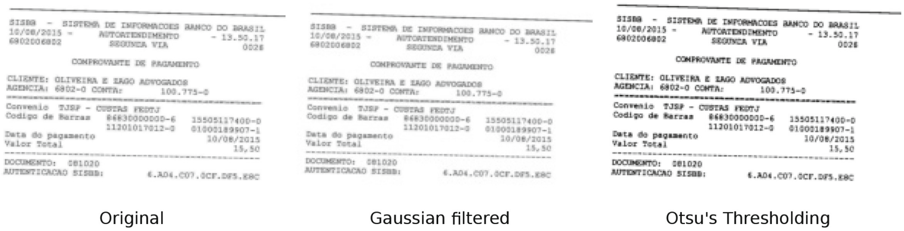

Fig. 3. Example of CLAHE technique application.

5.2 Classification

In Classification I, we used Mahotas [6], a computer vision and image processing library for Python, to apply the Haralick Texture algorithm to extract features. We defined a function that computes features and takes the mean of for directions of the features, returned by `mahotas.features.haralick`. After feature extraction, we stored filed in `.npy` format because this way, we do not need to repeat the extraction for future training. Then, we created and trained five classifiers: Support Vector Machine (SVM), Random Forest (RF), Decision Tree (DT), Gaussian Naive Bayes (NB) and Neural Networks (NN). There were 1,325 images without useful information remaining, and we decided to use the same number of images with useful information to balance the training dataset. The five classifiers classified the 60 document images of the test dataset into with and without helpful information classes. Figure 4 shows the confusion matrix for this classification. All Classifiers correctly classified the thirty images with helpful information class, except the Linear SVM classifier, which misclassified the classes. Of the thirty images in the without helpful information class, SVM and NN correctly classified 21 images, RF correctly classified 16 images, DT correctly classified 14 images, and NN correctly classified 28 images. Table 1 lists the performance metrics of each classifier.

In the second stage (Classification II), we used a dataset with previously labelled document images into four classes: 175 payment receipts (class 1), 175 judicial deposits (class 2), 136 others (class 3), and 135 trash (class 4). As the number of the samples is small, We used the Keras ImageDataGenerator class that provides real-time data augmentation and ensures that the model receives new variations of the images at each epoch (or iteration) without adding them

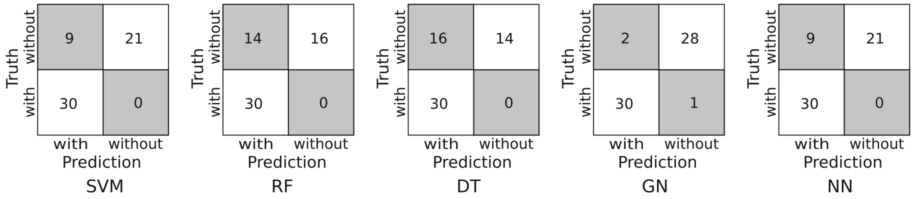

Fig. 4. Confusion matrix of classification I.

to the original image dataset. Furthermore, as the images are loaded in batches, much memory is saved. To monitor and control the training, we used some functions (callbacks) provided by Keras, namely, ModelCheckpoint and ReduceLROnPlateau. ModelCheckpoint function was used to save the model when a validation accuracy surpassed one previous reached. ReduceLROnPlateau was used to reduce the learning rate by 0.1 every five epochs when the validation accuracy has stopped improving. We used the weights of the VGG19 model. However, we did not train VGG19 any further. We only froze its layers and appended a shallow seven-layer network on top of it to perform the classification task of identifying document images in four classes, c.f. shown in Fig. 5.

```
conv_base = VGG19(weights='imagenet', include_top=False, input_shape=input_shape)
model = models.Sequential()
model.add(conv_base)
model.add(layers.GlobalAveragePooling2D())
model.add(layers.BatchNormalization())
model.add(layers.Flatten())
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dropout(0.6))
model.add(layers.Dense(4, activation='softmax'))
```

Fig. 5. Model used in classification II.

Global Average Pooling, a pooling operation that replaces fully connected layers, generates one feature map for each corresponding class of the classification task in the last layer, taking the average of each feature map. The resulting vector is fed directly into the softmax layer. The batch normalization layer normalizes its inputs, applying a transformation that maintains the mean output close to 0 and the output standard deviation close to 1. The Flatten layer is used to flatten the Dense layer's input with the RELU activation function that activates only certain neurons. A Dropout layer randomly selects neurons that will be ignored during training, thus creating less network computation and preventing overfitting. The Dense layer uses the SOFTMAX activation function, reducing the output to four neurons. Figure 6 shows the confusion matrix for this classification. Table 2 lists the performance metrics of each classifier.

Table 1. Metrics classification I.

	Classifiers				
	SVM	RF	DT	NB	NN
Accuracy	0.85	0.76	0.73	0.95	0.85
Sensitivity	1	1	1	0.96	1
Specificity	0.70	0.53	0.47	0.93	0.70

Table 2. Metrics classification II.

Metric	Value
Accuracy	0.968
Sensitivity	1.000
Specificity(class 0)	1.000
Specificity(class 1)	1.000
Specificity(class 2)	0.8857
Specificity(class 3)	1.000

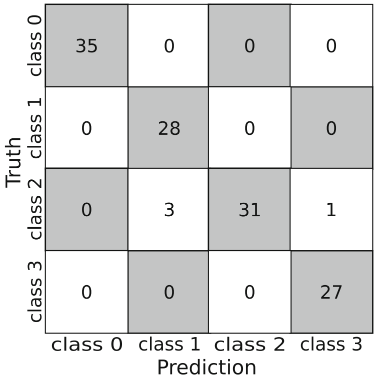

Fig. 6. Confusion matrix of classification II.

6 Conclusion

In this article, we present a proposal of classification for lawsuit document images. Our goal was to offer an alternative for document processing when extracting information from these document images is not feasible by NLP tools.

First, we carried out a preprocessing in images using Data Augmentation and Histogram Equalization techniques, then, we performed in two classification. In the first classification, we used the Haralick algorithm to extract textural features from images and created five machine learning classifiers to decide if an image has helpful information or not, namely Linear Support Vector Machine, Random Forest, Decision Tree, Gaussian Naive Bayes and Neural Networks. The Naive Bayes was the best model with 0.95 of accuracy, 0.96 of sensitivity, and 0.93 of specificity. In the second classification, we used VGG19, an advanced CNN with pre-trained layers, to extract features from . We froze its layers and appended a shallow seven-layer network on top of it to perform the classification task of identifying document images in four TJSP interest classes. This classifier obtained 0.96 of accuracy, 1 of sensitivity, and specificity above 0.88.

References

1. Abuelwafa, S., Pedersoli, M., Cheriet, M.: Unsupervised exemplar-based learning for improved document image classification. IEEE Access **7**, 133738–133748 (2019)
2. Ashir, A.M.: Multilevel thresholding for image segmentation using mean gradient. J. Electr. Comput. Eng. **2022** (2022)
3. Braz, F.A., et al.: Document classification using a bi-lstm to unclog brazil's supreme court. arXiv e-prints pp. arXiv-1811 (2018)
4. Chen, N., Blostein, D.: A survey of document image classification: problem statement, classifier architecture and performance evaluation. Int. J. Doc. Anal. Recogn. **10**(1), 1–16 (2007)
5. Clark, A.: Pillow (pil fork) documentation. Readthedocs (2015). https://Buildmedia.Readthedocs.Org/Media/Pdf/Pillow/Latest/Pillow.Pdf
6. Coelho, L.P.: Mahotas: Open source software for scriptable computer vision. arXiv preprint arXiv:1211.4907 (2012)
7. Csurka, G., Larlus, D., Gordo, A., Almazan, J.: What is the right way to represent document images? arXiv preprint arXiv:1603.01076 (2016)
8. Das, A., Roy, S., Bhattacharya, U., Parui, S.K.: Document image classification with intra-domain transfer learning and stacked generalization of deep convolutional neural networks. In: International Conference on Pattern Recognition, pp. 3180–3185 (2018)
9. Deng, G., Cahill, L.: An adaptive gaussian filter for noise reduction and edge detection. In: IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference, vol. 3, pp. 1615–1619 (1993)
10. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: self-supervised interest point detection and description. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 224–236 (2018)
11. Dusmanu, M., et al.: D2-net: a trainable CNN for joint description and detection of local features. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 8092–8101 (2019)
12. Haralick, R.M., Shanmugam, K., Dinstein, I.H.: Textural features for image classification. IEEE Trans. Syst. Man Cybern. **1**(6), 610–621 (1973)
13. Harley, A.W., Ufkes, A., Derpanis, K.G.: Evaluation of deep convolutional nets for document image classification and retrieval. In: International Conference on Document Analysis and Recognition (2015)
14. Jain, S., Dhar, J.: Image based search engine using deep learning. In: 2017 Tenth International Conference on Contemporary Computing, pp. 1–7. IEEE (2017)
15. Ketkar, N.: Introduction to Keras. In: Deep Learning with Python, pp. 95–109. Apress, Berkeley, CA (2017). https://doi.org/10.1007/978-1-4842-2766-4_7
16. Kumar, J., Doermann, D.: Unsupervised classification of structurally similar document images. In: International Conference on Document Analysis and Recognition, pp. 1225–1229 (2013)
17. Liu, L., Wang, Z., Qiu, T., Chen, Q., Lu, Y., Suen, C.Y.: Document image classification: progress over two decades. Neurocomputing **453**, 223–240 (2021)
18. Liu, Y., Xu, X., Li, F.: Image feature matching based on deep learning. In: IEEE 4th International Conference on Computer and Communication, pp. 1752–1756 (2018)
19. Nguyen, T.-S., Nguyen, L.-M., Tojo, S., Satoh, K., Shimazu, A.: Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts. Artif. Intell. Law **26**(2), 169–199 (2018). https://doi.org/10.1007/s10506-018-9225-1

20. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
21. Polo, F.M., Ciochetti, I., Bertolo, E.: Predicting legal proceedings status: approaches based on sequential text data. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law, pp. 264–265 (2021)
22. Polo, F.M., et al.: Legalnlp-natural language processing methods for the Brazilian legal language. arXiv preprint arXiv:2110.15709 (2021)
23. da Silva, N.C., et al.: Document type classification for Brazil's supreme court using a convolutional neural network. In: International Conference on Forensic Computer Science and Cyber Law (ICoFCS), Sao Paulo, Brazil, pp. 29–30 (2018)
24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations, pp. 1–14 (2015)
25. Sulea, O.M., Zampieri, M., Malmasi, S., Vela, M., Dinu, L.P., Van Genabith, J.: Exploring the use of text classification in the legal domain, pp. 1–5 (2017). arXiv preprint arXiv:1710.09306
26. de Tecnologia da Informação do Tribunal de Justiça de São Paulo, S.: Tribunal de justiça - estado de são paulo: a justiça próxima do cidadão (2022). https://www.tjsp.jus.br/QuemSomos
27. Xu, Y., Zhang, M., Wu, S., Hu, J.: Lawsuit category prediction based on machine learning. In: IEEE International Conference on Intelligence and Security Informatics, pp. 176–178. IEEE (2019)
28. Zhang, Z., Lee, W.S.: Deep graphical feature learning for the feature matching problem. In: IEEE/CVF International Conference on Computer Vision, pp. 5087–5096 (2019)