Deep Learning in Time-Frequency Domain for Document Layout Analysis

FELIPE GRIJALVA¹, (Senior, IEEE), ERICK SANTOS², BYRON ACUÑA.³, J.C. Rodríguez⁴, (Member, IEEE), and Julio Larco⁵, (Member, IEEE)

¹Faculty of Engineering and Applied Sciences (FICA), Telecommunications Engineering, Universidad de Las Américas (UDLA), Quito 170125, Ecuador (e-mail: felipe.grijalva@udla.edu.ec)
²Departamento de Electrónica, Telecomunicaciones y Redes de Información (DETRI), Escuela Politécnica Nacional, Ladrón de Guevara E11-235, Quito 170109, Ecuador (e-mail: erick.santos@epn.edu.ec)
³School of Electrical and Computer Engineering, University of Campinas (UNICAMP), CO 13083-852 BRA (e-mail: byronacunia@gmail.com)
⁴Green Energy Business Unit, Analog Devices International, Limerick, Ireland. V94 RT99 (e-mail: JuanCarlos.Rodriguez@analog.com)
⁵Departamento de Eléctrica, Electrónica y Telecomunicaciones. Universidad de las Fuerzas Armadas ESPE, Sangolquí - Ecuador (e-mail: jclarco@espe.edu.ec)

Corresponding author: Felipe Grijalva (e-mail: felipe.grijalva@udla.edu.ec).

ABSTRACT Document layout analysis plays an important role in the area of Document Understanding. It is responsible for identifying and classifying the different components of digital documents. Currently, there is no universal algorithm that fits all types of digital documents. This work presents a novel approach for identifying tables, figures, isolated equations, and text regions in scientific papers using deep learning and computer vision techniques. Our proposed approach is a three-stage system: (i) Obtaining the spectrograms of the horizontal and vertical intensity histograms of segmented regions of interest. (ii) Labeling segmented regions of interest into text, table, and figures using a deep convolutional neural network classifier. (iii) Identifying isolated equations in text regions using Bag of Visual Words (BOVW) with Zernike moments. We built a new dataset composed of 11007 papers to perform the experiments, using two common segmentation metrics to evaluate our model: (1) Adjusted Rand Index (ARI) and (2) Variation of Information (VI). The proposed document layout analysis system reached an overall accuracy of 96.2685%, outperforming prior art with a less computational cost.

INDEX TERMS Computer-Vision, Deep-Learning, document layout analysis, feature engineering, PDF.

I. INTRODUCTION

Document layout analysis (DLA) [1] is still one of the most challenging areas of information retrieval [2] due to the wide variety of documents that can be authored and the lack of structured information [3] in standardized formats for exchanging information such as Portable Document File (PDF). The main feature of PDFs created digitally is the preservation of the visual structure of the document in any electronic device, turning PDF files into the current standard format for electronic document exchange [4]. In this work, we explore the use of computer vision to segmenting tables, graphs, and equations because this is not feasible using text mining on digital documents.

Currently, there is no universal algorithm [5] that fully understands all regions of a digital document, i.e., identifying and segmenting all the individual elements such as tables, graphs, inline/isolated equations, paragraphs, etc. The problem of identifying and classifying the elements of a digital document based on its visual structure can be grouped into three categories [6]: (i) foreground regions, (ii) background regions, and (iii) both foreground and background regions. Foreground-based approaches perform page segmentation by analyzing the foreground pixels, which normally are the text characters. Background-based methodologies generally group the background pixels (“white space”) for segmentation. The approaches that analyze both foreground and background pixels try to ensemble the results of both individual approaches.

Convolutional Neural Networks (CNNs) have demonstrated enormous potential in DLA [7] due to the ability to find patterns in large amounts of data. In general, the incorporation of machine learning to solve the DLA problem is based on supervised learning approaches using CNN. Models are trained using pre-labeled blocks of text and non-text (e.g., tables, graphs, etc.). Despite significant advances achieved using Machine Learning (ML) approaches, layout segmentation is challenging due to the high intra-class, and low inter-class variance [5]. High intra-class variance is a...
result of the presence of digital documents with arbitrary layouts. Low inter-class variance is a result of the presence of tables, figures containing equations, ruling lines, or texts with a different orientation modifying its inter-class layouts.

In this paper, we present a new approach for DLA, which consists of:

1) Segmentation of the regions of interest;
2) Generation of spectrograms using horizontal and vertical pixels profile projections of the regions of interest;
3) Implementation of a deep CNN, trained for three classes: text, table, and figures; and
4) Use of the Bag of Visual Words (BOVW) technique to identify lines with isolated equations within the text regions.

Thus, in scientific papers of standard format, i.e., journal-style scientific papers of one or two columns layout subdivided into the following sections: Title, Authors and Affiliation, Abstract, Introduction, Methods, Results, Discussion, Acknowledgments, and Literature Cited. Our proposed approach is able to segment and identify: (i) Figure regions, (ii) Table regions, (iii) Text regions, and (iv) Isolated equations.

II. RELATED WORK

While many existing methods focus on deep learning with convolutional networks directly, we argue the importance of explore time-frequency domain structures to improve the performance of the models. To the best of our knowledge, a deep learning in time-frequency domain method for document layout analysis has yet to be proposed for recognizing scholarly publications, which is a crucial stage in document understanding [8]–[10]. This task has become of great relevance because PDF documents dominate scholarly publications and this format does not allow extract the information in a structured way, which causes complications in the implementation of artificial intelligence algorithms to retrieve knowledge from scholarly publications. Almost all existing solutions attempt to produce high performance filters for determined images, typically by the use of image processing techniques and/or deep learning techniques. Our solution instead uses frequency domain information for segmentation and classification. Deep learning algorithms have been successfully applied to DLA in the contemporary literature [1], and currently is considered as an especially semantic segmentation task [11], whose goal is performing a pixel-level understanding of the segmentation object. For instance in [12] document layout analysis and recognition was treated as a pixel-by-pixel classification task using a multimodal fully convolutional neural networks, however in this work neural network is used to assist heuristic algorithms to classify candidate bounding boxes. In [13] a feed-forward neural network was trained with textual and statistical features extracted by processing a mask function across the image pages for the text vs. non-text classification. The model learns texture features by using dilated convolutional layers to segment text, image, table, mathematical expression (displayed equations), and flow charts. Authors reported improvements in the classification stage, but the performance about the bounding boxes regression was not reported, which is also explored in this work. In [14] a fast one-dimensional CNN-based approach for document image layout analysis was proposed. The methodology first segments blocks of content in document images and then divides them into several tiles to compute their horizontal and vertical projections. There were jointly used by a one-dimensional CNN model that classifies them into text, tables, and images. Finally, the tiles classification output is combined with a simple voting scheme that defines the final class for each block of content in the document image. The aforementioned voting scheme increases the number of calculations to obtain a prediction for each region. On [12] document layout analysis was performed using an encoder-decoder using both appearance (to distinguish text from figures, tables, and line segments) and semantics (for paragraphs and captions). Compared with [12] our approach does not require another neural network for feature engineering.

Previous studies on equations detection in documents propose different approaches. In [15], [16] and [17] the classification of regions with equations is carried out based on features that contain statistical information of the pixels present in the regions. In [18] the classification is done with the help of the properties and information embedded in the PDF file structure.

The accuracy and precision of identifying and segment tables, figures, and equation regions reported in previous works are still far from reliable. Due to this, human verification is required after using these algorithms in new documents. In this work, we are exploring the opportunities of generate more robust features using frequency domain. Additionally, we present a comparison of our approach to the current state of the art [14]. Upon publication, we will make available our new dataset built with 11007 PDF documents of scholarly publications.

We propose a new method that converts a digital document into a semi-structured document using an image-based deep learning approach that extracts features from blocks to provide predefined labels such as: (i) Figure Region, (ii) Table region, (iii) Text region and (iv) Isolated equation. The contributions of this paper are as follows:

- The introduction of a novel DLA, leveraging the potential of deep learning. Namely, we present the design of Spectrogram-CNN for region classification (text, figure, table, and isolated equations), which is based in a time-frequency representation that captures relevant features of the document’s regions.
- The proposed methodology is able to identify and segment mathematical expressions (isolated-equations) in blocks of text (paragraphs) from carefully chosen features such as the Zernick moments and other statistical features.
- The proposed approach outperforms the region classification with fast one-dimensional CNN proposed by...
[14], by considering a new dataset of 121 703 pages for evaluation.

III. MATERIALS AND METHODS

This section explains our proposed DLA system and its evaluation. Specifically, in Section III-A we describe the dataset employed. Then, the following sections present our DLA system and finally in Section III-E we describe the experimental setup employed to evaluate our system and its stages.

Fig 1 summarizes our approach that consists of the following stages:

1) Input preprocessing (Section III-B),
2) Region classification (Section III-C),
3) Isolated equations classification (Section III-D).

In summary, our DLA method receives an image page from a document and then it creates an intermediate representation by calculating the spectrograms of the vertical and horizontal projection profiles of segmented regions (i.e., the input preprocessing stage). Next, a CNN classifies these spectrograms into text, table and figure regions. Finally, we extract several features from the regions classified as text by the CNN based on statistical properties and the Zernick moments within a Bag of Visual Words approach. These features feed an SVM classifier to discriminate text from isolated equations.

A. DATASET

We built a dataset of labeled regions from text, figures, tables, and mathematical equations found in images of 121 703 pages from standard-format scientific papers in PNG format with a resolution of 500 dpi. Fig. 2 shows an example page of the dataset and their corresponding labels.

These images have been collected systematically from PDF papers with their respective \texttt{BibTeX} files downloaded from the ARXIV platform, which are of public-domain status. Each image page was labeled with the bounding boxes coordinates of the regions of interest, i.e., regions of text, tables, figures, and isolated equations.

The labeling process was performed using the approach explained in [19]. This process consists of modifying the semantic structure of each \texttt{BibTeX} file automatically by inserting appropriate \texttt{BibTeX} tags to draw color bounding boxes in the regions of interest. The coordinates of each bounding box and its corresponding region label (i.e. the ground truth annotations) are obtained considering a coordinate system located in the upper-left part of each page of the document. Due to the modification process in the \texttt{BibTeX} files for performing the automatic labeling process, compilation errors sometimes occurred. Hence, it was necessary to carry out a manual verification with human curators to ensure a high quality dataset.

B. INPUT PREPROCESSING

Since the region classification stage is a Convolutional Neural Network (CNN) that accepts fixed size inputs of the regions of interest, we need to normalize the document page’s regions. Besides, it is critical to feed the CNN with a meaningful input representation. Thus, the input preprocessing is composed of two stages:

1) Region segmentation, which employs image processing techniques to obtain plausible regions of interest.
2) Intermediate representation, which converts the segmented regions of variable size into a fixed size 2D spectrogram representation compatible with the expected fixed size CNN input.

1) Region Segmentation

The region segmentation stage receives an RGB image page in PNG format as input. First, we binarize the input image. Then, we identify the presence of a one-column or a two-column layout by checking the binary vertical projection profile of the image page. If any value of this projection is below the threshold of $50\%$ of the root mean square value of the vertical projection profile, a two-column layout is assumed (see Fig. 3a).

Finally, we performed a block segmentation using Run Length Smoothing Algorithm (RLSA) [20], with kernel size of $20 \times 20$ pixels for dilatation. We also set a horizontal RLSA threshold $= 70\%$ of the space between columns measured in pixels for two-column layouts, and 120 pixels for one-column layouts. The output of the Region Segmentation stage is shown in Figure 3b. All the region segmentation parameters where determined by visually inspecting several random pages.

2) Intermediate Representation:

Using the bounding boxes obtained in the region segmentation stage, we extract the binary vertical and horizontal projection profile of each region of interest in the image page (see Fig. 4). We propose to use an intermediate representation by calculating the spectrograms of the vertical and horizontal projection profiles so that these spectrograms have a fixed size in order to feed the Convolutional Neural Network that expects a fixed size input. The 2D spectrogram representation captures the time and frequency variability of the projection profiles that are very distinctive among the regions of interest (text, table, etc.) as shown in Fig 4.

However, by directly calculating the spectrogram of the projection profiles does not guarantee ending up with spectrograms with the same size since the regions of interest are not of fixed size (e.g., a text region is normally bigger than an isolated equation). This was solved by upsampling or downsampling the projection profiles to a fixed value of $N = 2016$ samples when the projection profile length is smaller or larger than $N$, respectively.

Once each projection profile is normalized to a fixed size, we calculate the spectrograms for the vertical and horizontal projection profiles with a Blackman-Harris window of length 126 and overlapping of 96 samples to obtain an spectrogram of size $64 \times 64$. Figures 5, 6 and 7 show some examples of the spectrograms obtained using the aforementioned process.
We trained a Deep Convolutional Neural Network (CNN) with two branches to classify three classes: (i) Figure, (ii) text, and (iii) Table. The CNN architecture is shown in Fig. 8. The two CNN branches receive as inputs the horizontal and verticals spectrograms, respectively and then both branches are merged to produce a single classification result. Next, we describe the hyperparameters used for each layer:

- **Convolutional Layer**: All convolutional layers uses a filter (kernel) size = 3 × 3, with stride = 1 × 1. We employed 32 filters in the first two convolutional layers and 64 filters in the last two layers.

- **Batch Normalization (BN) Layer**: BN speeds up the training process and acts as a regularizer to prevent over-fitting [21]. Empirically, we found that the best momentum (hyper-parameter) used in BN layers for our architecture was 0.99 (range of Momentum: 0-1) because during training we used small batches compared to the size of the dataset.

- **Leaky ReLU Layer**: This layer was stacked in order to avoid the vanishing gradient problem since this layer implements an activation function that does not cause saturation for positive and negative inputs. Compared with ReLU, Leaky ReLU compresses the negative part rather than mapping it to zero. This allows for a small, non-zero gradient when the unit is not active [22], [23].

- **Average Pooling Layer**: This layer provides an operation to reduce the dimensionality of the feature maps. We used an average-pooling of size = 2 × 2, stride = 2 × 2, and without zero-padding.

- **Dropout Layer**: This layer was located in both convolutional blocks and the fully connected layers with a drop rate of 0.1. Dropout is a regularization method to avoid overfitting [24].

Observe that the CNN does not discriminate isolated equations because we experimentally found that including isolation equations as a fourth class increases the error, since they are easily confused with text regions by the CNN or they are incorrectly segmented by the region segmentation stage described in III-B1. Instead, we decided to design an isolated equation classification stage that we describe in the next section.
FIGURE 3: Approach used to segment pages into blocks of regions of interest: (a) Input document page with their binary projection profiles (blue), thresholds (orange lines) and thresholded projection profiles (green). Vertical projection threshold is used to determine if the page corresponds to a one-column or two-column layout; and (b) Binary image resulting from the RLSA operations.

FIGURE 4: Vertical and horizontal projection profiles of (a) Figure region (b) Text region and (c) Table region.
we first segmented the text regions into lines. Then, we extracted a global feature vector for each segmented line, which is constructed by concatenating two vectors. The first vector stores statistical information extracted directly from the pixels of each segmented line. The second vector is a dictionary built from the information obtained with the Zernike moments of each symbol found in a line. Finally, the global feature vector is sent to a Support Vector Machine binary classifier to discriminate text lines from isolated Equations. Next, we describe in detail these stages.

D. ISOLATED EQUATIONS CLASSIFICATION

After the CNN region classification stage, some regions classified as text region contain equation regions, as shown in Fig. 9.

Thus, we first segmented the text regions into lines. Then, we extracted a global feature vector for each segmented line, which is constructed by concatenating two vectors. The first vector stores statistical information extracted directly from the pixels of each segmented line. The second vector is a dictionary built from the information obtained with the Zernike moments of each symbol found in a line. Finally, the global feature vector is sent to a Support Vector Machine binary classifier to discriminate text lines from isolated Equations. Next, we describe in detail these stages.

1) Line Segmentation Stage

Line segmentation was performed using a dilation with a kernel of vertical size = 4 pixels and a horizontal size equal to half the width in the text block’s pixels. We used this dilation structured element since the pixels of a line (text or isolated equation) follow a rectangle structure. Then, we obtained the bounding boxes of the lines using the algorithm reported in [25]. An example of the line segmentation stage is shown in Fig. 10.

2) Feature Extraction Stage

For each line detected in the line segmentation stage, we construct a global feature vector by concatenating statistical features obtained directly from the line pixels and features extracted from the Zernike moments of each symbol in a line.

With respect to the statistical features, they are the following:

- **Centroid fluctuation**: This feature evaluates the variation of the centroid of each symbol contained in a line and is mathematically defined in [15]. The intuition behind this feature is that lines with isolated equations produce a higher centroid than lines without isolated equations.
- **Variance of the height of symbols**: Prior works [17], and [15] reported that lines with isolated equations produced higher variance than lines without isolated equations.
- **Sparse Ratio**: In [17] and [15] it was found that lines with isolated equations produce a higher sparse ratio (density of foreground pixels with respect to the total pixels) than lines without isolated equations.

In addition to these three features, we propose a new set of features based on a Bag of Visual Words (BoVW) of each symbol contained in a line.

FIGURE 5: The $64 \times 64$ spectrograms of the projection profiles for the figure region illustrated in Fig 4a.

FIGURE 7: The $64 \times 64$ spectrograms of the projection profiles for the table region illustrated in Fig 4c.

FIGURE 6: The $64 \times 64$ spectrograms of the projection profiles for the text region illustrated in Fig 4b.
The methodology to obtain our proposed set of features based on BoVW follows:

1) Computing keypoint descriptors using Zernike moments [26],
2) Clustering keypoint descriptors using Gaussian Mixture Model (GMM) [27],
3) Feature encoding.

**Computing keypoint descriptors**: We obtain the Zernike moments of each symbol in the line segmented considering the invariance both in translation and in scale, with fifth-order Zernike polynomials. This produces 12 complex Zernike moments for each symbol. Then, we extract the magnitudes from these 12 complex moments to form a keypoint descriptor per symbol. Therefore, a line has the same number of keypoint descriptors as symbols.

**Clustering keypoint descriptors**: Using GMM, each cluster generated is considered a visual word that represents a specific local pattern shared by the keypoint descriptors in that cluster. In this way, clustering using GMM creates a visual word vocabulary that summarizes different local patterns located on a line of text, and the number of clusters determines the size of the vocabulary.

Then, we used the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) [28] to determine the optimal number of components for the Gaussian Mixture Model. This is done by plotting the AIC and BIC scores as functions of the number of clusters, as is shown in Fig. 11. Experimentally, we found that 256 clusters produce the same results in terms of performance than a higher number of clusters (> 256). Hence, we fixed the number of clusters to 256.

**Feature encoding**: Feature encoding allows to build a global feature vector from a coding method of the keypoint descriptors. In this work, we consider the Vector Quantization (VQ) technique, which is a voting-based encoding method where each descriptor directly votes for the code word using a specific strategy. In this way, the vector quanti-
zation clusters the keypoint descriptors in their feature space into 256 clusters using the Gaussian Mixture Model, and encodes each keypoint by the index of the cluster it belongs to. Thus, with a vocabulary of 256 visual words it is possible to represent each line of a text block as a “bag of visual words”.

Finally, the three features aforementioned were concatenated together with the vector of length 256, generated by the Bag of visual words to obtain a final vector of length = 259 that represents the features of a line within a text block.

3) Line Classification Stage

This stage is a SVM classifier (binary classifier) with Radial Basis Function (RBF) kernel. It receives as input the feature vector of 259 obtained in the feature extraction stage and returns the classes: (i) Text Line, and (ii) Isolated Equation.

After the line classification stage, it was necessary to merge consecutive lines with the same label into a single region to obtain the results shown in Fig. 12.

E. EXPERIMENTAL SETUP

This section outlines the experimental design to assess the region classification and isolated equations classification stages. We also present the experimental setup used to evaluate our complete system.

1) Region Classification Experimental Setup

a: Data partition

For the region classification stage, we used five-Fold Cross Validation technique to estimate the CNN model performance. The training dataset was composed of horizontal and vertical spectrograms of: (i) 13 131 text regions, (ii) 13 131 table regions and (iii) 13 131 figure regions. We separated 20% of the samples from the training dataset to construct a validation set to monitor the accuracy across epochs during training.

b: CNN configuration

We used the Adam [29] optimizer with schedule learning rate decreasing the learning rate gradually, based on the epoch starting at a value of 0.001 until 0.0001. The early stopping technique with patience of 5 was used to avoid overfitting by monitoring the accuracy on the validation dataset.

c: Evaluation metrics

After training each fold, we evaluated the following metrics in the corresponding test set: precision, recall, F1 Score and AUC (Area Under the Curve). The final performance is obtained by averaging across folds. We compared our proposed Spectrogram-CNN to similar previous work [14] (hereinafter Borges-CNN), which also employs projection profiles but across several tiles of the page image. For a fair comparison, we trained Borges-CNN with the same dataset used for Spectrogram-CNN.

2) Experimental setup for Isolated Equations Classification

a: Data partition

To construct the GMM clustering model, we used a training set that contains 320 000 text symbols and 320 000 mathematical symbols with variable size and shape. These symbols were extracted from text regions that were not used to train the CNN to avoid data contamination. With these 640 000 symbols, we generated a GMM clustering model considering 256 clusters.

To train the SVM classifier, we considered 800 text regions and 2 800 equation regions, resulting in 4 948 lines with text symbols and 5 108 lines with math symbols so that the dataset is roughly balanced. Symbols of these lines come from different regions than the ones used for fitting the GMM model, to avoid data contamination. We split this dataset
composed of lines into a training set with 80% of random samples and a test set with the remaining ones.

b: SVM configuration
A global feature vector for each of the lines from the training set is calculated with BoVW using our pre-trained GMM clustering model. We obtained the SVM classifier hyperparameters (i.e., the soft margin constant $C$ and the $\gamma$ parameter for the RBF kernel) using a grid search approach under a five-fold cross-validation scheme.

c: Evaluation metrics
We computed the accuracy and AUC (Area Under the Curve) to evaluate the performance of the isolated equations classifier on the test set. So far, we have presented how to evaluate single stages of our approach. Next, we describe the complete system assessment.

3) Complete System Assessment
a: Dataset
We evaluated the complete system from Fig. 1 using 1500 pages with labeled regions (text, figure, table, and equation along with the coordinates of their bounding boxes). It is worth noting that the test dataset contains pages that have not been used in previous stages to avoid data contamination.

b: Evaluation metrics
Since the final goal of our approach is to determine the bounding box as well as the region label (i.e., text, equation, table or figure), our system can be evaluated as a typical image segmentation task. Hence, we used the following image segmentation metrics: (i) Adapted Rand Index (ARI), and (ii) Variation of Information (VI) [30] for global system evaluation.

The ARI evaluates our system’s segmentation by comparing the existing differences in pixel pair connectivity between the ground truth and predicted segmentation. By treating our image segmentation as clusters employing ARI, it is possible to determine the precision, recall and F1-Score values of the entire system with the help of a penalty when an erroneous merger or split errors of the ground truth segments occur. Thus, we calculated an error score for ARI which is defined as $1 - \text{F1-Score}$.

The VI, in turn, calculates symmetric conditional entropies in order to identify whether the system is affected by problems of over-segmentation (splits score) or under-segmentation (merges score). Both scores are measured in bits and ideally they should be zero in the absence of splits or merges.

Finally, we also computed the False detection and Missing regions statistics. We define the false detection statistic as the percentage of the document pages in which our system detects regions that do not exist on the ground truth of each class. On the other hand, the Missing regions statistic determines the percentage of the document pages in which our system does not detect regions that should be present, when compared to the ground truth of each class.

FIGURE 13: (a) Confusion Matrix for Spectrogram-CNN (b) Confusion Matrix for Borges-CNN.

TABLE 1: Evaluation metrics obtained in Spectrogram-CNN and Borges-CNN. Bold type represents best scores.

| Region   | Architecture  | Precision | Recall | F1-Score |
|----------|---------------|-----------|--------|----------|
| Text     | Spectrogram-CNN | 0.98550  | 0.99218 | 0.98858  |
|          | Borges-CNN    | 0.96629  | 0.95536 | 0.96079  |
| Figure   | Spectrogram-CNN | 0.96689  | 0.96108 | 0.96497  |
|          | Borges-CNN    | 0.94502  | 0.86439 | 0.90291  |
| Table    | Spectrogram-CNN | 0.96735  | 0.96799 | 0.96767  |
|          | Borges-CNN    | 0.89251  | 0.97821 | 0.93340  |

IV. RESULTS AND DISCUSSION
In this section the results of the individual stages, i.e. region classification (Sec. IV-A) and isolated equations classification (Sec. IV-B), as well as the results of the complete system (Sec. IV-C) are discussed.

A. ANALYSIS OF REGION CLASSIFICATION RESULTS
In Fig. 13 we present the normalized Confusion Matrices from the prior approach [14] (Borges-CNN), and of our proposed approach (Spectrogram-CNN). The global metrics of precision, recall, and F1-score per class are detailed in Table 1. Our proposed approach’s overall accuracy reached 97.401%, overcoming the overall accuracy 93.296% of Borges-CNN.

Among the important aspects that influence the performance of this stage is the generation of spectograms, since it transforms the fundamental information contained in the waveforms of the projections in a 2D region.

On the other hand, one drawback in our approach can be inferred in these confusion matrices. Concretely, observe that our approach incorrectly predicts 2.9% as Tables instances when the ground truth is Figure. In Borges-CNN this error increases even more to 10.59%. We found that this is caused by the pixel projection profiles of tables and figures being very similar when figures presents diagrams composed mainly by lines, which are typically found in scientific papers.

In computational terms, although Borges-CNN has 35 803 trainable hyperparameters, while our proposed approach has 1 182 531, the latter uses a voting scheme that increases the number of calculations to obtain the prediction for a region. For instance, a region of size 700 × 700 pixels can obtain 529
tiles of size 100 × 100 pixels with overlapping displacement of 30 × 30 pixels, as specified in [14]. Thus, to obtain the prediction using the Borges-CNN approach one must predict the labels of the 529 tiles (i.e., a 529 fold increase of calculations), in contrast to a single prediction made by our proposed approach Spectrogram-CNN.

In Fig. 14a we present Receiver Operating Characteristic (ROC) [31] curves of Borges-CNN and Spectrogram-CNN, where it can be seen that our proposed approach overcomes the Borges-CNN methodology, both globally and per-class as shown by the 14b ROC micro-average.

B. ISOLATED EQUATIONS CLASSIFICATION RESULTS

In Fig. 15 we present the normalized Confusion Matrix of our SVM classifier, and Table 2 summarizes the global metrics of precision, recall, and F1-score per class: (i) Text, and (ii) Equation. We achieved a global accuracy of 98.249% using our proposed Bag of Visual Words approach with Zernike moments.

In Fig. 16, we can visualize our SVM classifier’s ROC curve. This ROC curve is very close to reaching the upper-left corner of the plot, which means that there is a noticeable separation between the two classification groups (text lines and equation lines). Hence, the SVM classifier has high specificity and sensitivity values, which explains the high global accuracy obtained by the model.

C. ANALYSIS OF RESULTS OF THE COMPLETE SYSTEM

We computed the ARI metric for each class separately (i.e., text, figure, table, and equation), which was done with the pixels corresponding to the region that comprise the bounding boxes generated by our DLA system. Table 3 shows the Error score calculated from ARI, both per-class and globally. Hence, the error score of the entire system is 2.2526%.

Recalling that The splits score quantifies the amount of over-segmentation measured in bits while the merges score quantifies the amount of under-segmentation. The lower these values, the more precise the segmentation. A high

| Region  | Precision | Recall | F1-Score |
|---------|-----------|--------|----------|
| Text    | 0.98171   | 0.98217| 0.98194  |
| Equation| 0.98321   | 0.98278| 0.98300  |

FIGURE 14: Performance comparison between Spectrogram-CNN and Borges CNN

FIGURE 15: Normalized confusion matrix in isolated equations classification.

FIGURE 16: ROC curve for isolated equations classification.
merge score was obtained for the text and figure category, this means that our system tends to merge (under-segmentation) more text and figure than equation and table regions. On the other hand, the splits score varies slightly for the four classes.

Table 4 presents statistics of False detections and Missing regions. From these results, we found that the False Detection statistic for equations is higher than the other classes. These results suggest that the equation detection is affected by the propagation of errors from text segmentation using our proposed Spectrogram-CNN.

V. CONCLUSION

We developed a new Document Layout Analysis approach for identifying tables, figures, isolated equations, and text regions of standard format scientific papers in PDF format, using computer vision techniques with a deep CNN. The results are very encouraging in terms of error, split and merge scores calculated from ARI and VI metrics.

One of our algorithm’s weaknesses is the dependence that exists between the Spectrogram-CNN stage and the region segmenter. This means that in order to maintain the performance of the region classification (text, figure, and table), the segmenter must correctly extract the regions of interest. Therefore, to increase the entire system’s performance, it is recommended to use a more precise and exact segmenter.

Still, our proposed approach overcomes the previous work of Borgues-CNN [14] in terms of performance, with a less computational cost. The advantage that our architecture has when using the whole pixel spectrograms is the reduction of the computational cost to obtain the prediction unlike Borgues-CNN which is based on a voting scheme. This makes our system feasible for applications that require consuming the least amount of computational resources.

We presented also a new set of features based on a Bag of Visual Words (BoVW) of each symbol contained in a line, using Zernike moments to discriminate isolated equations from text. It should be noted that the construction of the visual vocabulary was trained with symbols of the English alphabet and with mathematical symbols, which means that this algorithm can be used for documents written in Latin alphabet. Hence, to adapt to another language such as Chinese, the algorithm should be trained again with documents containing text from the new alphabet.

Moreover, although there is a high performance in the classification of lines of equations and lines of text, information is being lost in the Zernike moments by only taking into account the magnitude of the moments (i.e., we did not consider phase). By taking advantage of this aspect in future studies, it should be possible to propose new approaches on this regard.

One concern that might arise is how future works can deal with inline equations (i.e., equations within lines of text). Since our method does not deal with inline equations, this would be an interesting topic for future work using Zernike moments. For instance, a good starting point might be to segment words and then to extract from them both phase and magnitude Zernike moments to create similar features as we do to train the SVM classifier.

REFERENCES

[1] S. Bhownik, R. Sarkar, M. Nasipuri, and D. Doermann, “Text and non-text separation in offline document images: a survey,” International Journal on Document Analysis and Recognition (IJDAR), vol. 21, no. 1-2, pp. 1–20, 2018.

[2] J. Liu, X. Kong, X. Zhou, L. Wang, D. Zhang, I. Lee, B. Xu, and F. Xia, “Data mining and information retrieval in the 21st century: A bibliographic review,” Computer Science Review, vol. 34, p. 100193, 2019.

[3] K. Jayaram and K. Sangeeta, “A review: Information extraction techniques from research papers,” in 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), IEEE, 2017, pp. 56–59.

[4] P. Association et al., “Iso 32000-2 (pdf 2.0),” Technical report, International Organization for Standardization, Tech. Rep., 2017.

[5] G. M. Binnakshashen and S. A. Mahmoud, “Document layout analysis: A comprehensive survey,” ACM Computing Surveys (CSUR), vol. 52, no. 6, pp. 1–36, 2019.

[6] K. Chen, F. Yin, and C.-L. Liu, “Hybrid page segmentation with efficient whitespace rectangles extraction and grouping,” in 2013 12th International Conference on Document Analysis and Recognition. IEEE, 2013, pp. 958–962.

[7] A. W. Harley, A. Utkes, and K. G. Derpanis, “Evaluation of deep convolutional nets for document image classification and retrieval,” in 2015 13th International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2015, pp. 991–995.

[8] C. Caragea, J. Wu, A. Ciobanu, K. Williams, J. Fernández-Ramírez, H.-H. Chen, Z. Wu, and L. Giles, “Citeseer x: A scholarly big dataset,” in European Conference on Information Retrieval. Springer, 2014, pp. 311–322.

[9] D. Kenny, S. Srirangaraj, D. David, K. B. Urala, and G. Venu, “Chart mining: A survey of methods for automated chart analysis,” IEEE transactions on pattern analysis and machine intelligence.

[10] K. Davila, S. Setlur, D. Doermann, U. K. Bhargava, and V. Govindaraju, “Chart mining: a survey of methods for automated chart analysis,” IEEE transactions on pattern analysis and machine intelligence, 2020.

[11] D. Ravi, M. Bober, G. M. Farinella, M. Guarnera, and S. Battiti, “Semantic segmentation of images exploiting dct based features and random forest,” Pattern Recognition, vol. 52, pp. 260–273, 2016.

[12] X. Yang, E. Yumer, P. Amente, M. Kraley, D. Kifer, and C. Lee Giles, “Learning to extract semantic structure from documents using multimodal fully convolutional neural networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5315–5324.

[13] S. C. Kovaraju, M. Masum, N. Z. Tsaku, P. Patel, T. Bayramoglu, G. Modgil, and M. Kang, “Dot-net: Document layout classification using texture-based cnn,” in 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019, pp. 1029–1034.
[14] D. Augusto Borges Oliveira and M. Palhares Viana, “Fast cnn-based document layout analysis,” in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 1173–1180.

[15] W.-T. Chu and F. Liu, “Mathematical formula detection in heterogeneous document images,” in 2013 conference on technologies and applications of artificial intelligence. IEEE, 2013, pp. 140–145.

[16] U. Garain, “Identification of mathematical expressions in document images,” in 2009 10th international conference on document analysis and recognition. IEEE, 2009, pp. 1340–1344.

[17] X. Lin, L. Gao, Z. Tang, X. Lin, and X. Hu, “Mathematical formula identification in pdf documents,” in 2011 international conference on document analysis and recognition. IEEE, 2011, pp. 1419–1423.

[18] J. B. Baker, A. P. Sexton, and V. Sorge, “Faithful mathematical formula recognition from pdf documents,” in Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, 2010, pp. 485–492.

[19] M. Li, Y. Xu, L. Cui, S. Huang, F. Wei, Z. Li, and M. Zhou, “Docbank: A benchmark dataset for document layout analysis,” 2020.

[20] K. Y. Wong, R. G. Casey, and F. M. Wahl, “Document analysis system,” IBM journal of research and development, vol. 26, no. 6, pp. 647–656, 1982.

[21] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” arXiv preprint arXiv:1502.03167, 2015. [Online]. Available: https://arxiv.org/abs/1502.03167

[22] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in Proc. icml, vol. 30, no. 1, 2013, p. 3.

[23] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shakhroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai et al., “Recent advances in convolutional neural networks,” Pattern Recognition, vol. 77, pp. 354–377, 2018. [Online]. Available: https://arxiv.org/abs/1512.07108

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” The journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.

[25] S. Suzuki et al., “Topological structural analysis of digitized binary images by border following,” Computer vision, graphics, and image processing, vol. 30, no. 1, pp. 32–46, 1985.

[26] W.-Y. Kim and Y.-S. Kim, “A region-based shape descriptor using zernike moments,” Signal Processing: Image Communication, vol. 16, no. 1, pp. 95 – 102, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0923596500000199

[27] X. Peng, L. Wang, X. Wang, and Y. Qiao, “Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice,” Computer Vision and Image Understanding, vol. 150, pp. 109–125, 2016.

[28] A. Chakrabarti and J. K. Ghosh, “Aic, bic and recent advances in model selection,” in Philosophy of Statistics. Elsevier, 2011, pp. 583–605.

[29] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[30] I. Arganda-Carreras, S. C. Turaga, D. R. Berger, D. Cireşan, A. Giusti, L. M. Gambardella, J. Schmidhuber, D. Laptev, S. Dwivedi, J. M. Buhmann et al., “Crowdsourcing the creation of image segmentation algorithms for connectomics,” Frontiers in neuroanatomy, vol. 9, p. 142, 2015.

[31] J. A. Hanley and B. J. McNeil, “The meaning and use of the area under a receiver operating characteristic (roc) curve.” Radiology, vol. 143, no. 1, pp. 29–36, 1982.

***