2022

Vietnamese Document Analysis: Dataset, Method and Benchmark Suite

Khang Nguyen (0000-0002-6571-7075)

An Nguyen

Nguyen D. Vo

Tam Nguyen

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 August 2022, accepted 20 September 2022, date of publication 30 September 2022, date of current version 14 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3211069

RESEARCH ARTICLE

Vietnamese Document Analysis: Dataset, Method and Benchmark Suite

KHANG NGUYEN[1,2], AN NGUYEN[1,2], NGUYEN D. VO[1,2], AND TAM V. NGUYEN[3], (Senior Member, IEEE)

[1]University of Information Technology, Ho Chi Minh City, Vietnam
[2]Vietnam National University, Ho Chi Minh City, Vietnam
[3]University of Dayton, Ohio, OH, USA

Corresponding author: Khang Nguyen (khangnttm@uit.edu.vn)

This research was supported by The VNUHCM-University of Information Technology's Scientific Research Support Fund.

ABSTRACT Document image understanding is increasingly useful since the number of digital documents is increasing day-by-day and the need for automation is increasing. Object detection plays a significant role in detecting vital objects and layouts in document images and contributes to providing a clearer understanding of the documents. Nonetheless, previous research mainly focuses on English document images, and studies on Vietnamese document images are limited. In this study, we extensively benchmark state-of-the-art object detectors and analyze the performance of each method on Vietnamese document images. Moreover, we also investigate the effectiveness of four different loss functions on the experimental object detection methods. Extensive experiments on the UIT-DODV dataset are conducted to provide insightful discussions.

INDEX TERMS Convolutional neural network, deep learning, page object detection, Vietnamese document image, image processing.

I. INTRODUCTION

Understanding the content of documents is the key task in The Fourth Industrial Revolution (4IR) [1]. Document Image Understanding (DIU) is an automatic process that extracts useful information from the image of a documentation page. DIU combines image analysis techniques and pattern recognition to process and extract information from image documents; it refers to the logical and semantic analysis of image documents to extract information that humans can understand and encode into a machine-readable form. The DIU problem can be divided into several subproblems (e.g., Table Detection [2], Document Image Classification [3], etc.) with each problem rising from the result of the prior problem. Two common and significant stages of these subproblems are segmentation-defining feature regions (also known as page physical structure analysis) and labeling-assigning labels to defined regions (also known as page logical structure analysis) [4]. Once solved, these two stages are extremely meaningful and are baselines for other complex problems, such as document forgery detection [5], document image retrieval [6], and visual document question answering [7]. However, the Document Image Understanding field has many major challenging issues, receiving attention from document

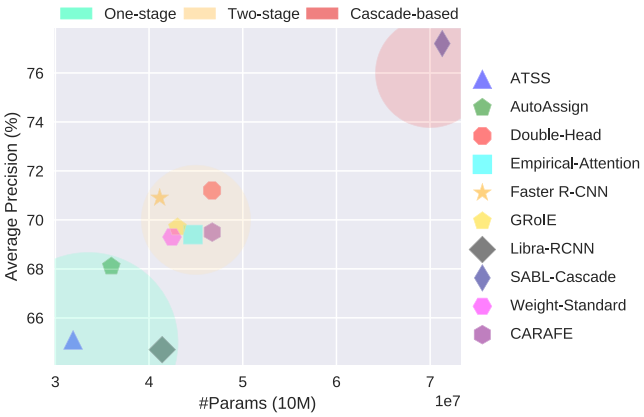


FIGURE 1. Comparison performance in terms of Average Precision and the number of parameters between advanced object detection models on the UIT-DODV dataset.

The associate editor coordinating the review of this manuscript and approving it for publication was Khoa Luu.

recognition, analysis and information & database communities.

In this study, we focus on object detection in the document image data problem using the Vietnamese UIT-DODV dataset [8]. The dataset includes 2,394 scanned images of Vietnamese documents, with four object classes: Table, Figure, Formula, and Caption. The input to the problem is a document page image; the output from the problem is the objects' locations (if any) in the image (Figure 2).

By observation and analysis, we recognize some challenges of the problem of each object type in the document not only from external factors but also from internal factors of the documents.

- External factors resulting from the quality of images such as blurred images, blurred, obscured objects, low resolution, and distorted objects. Moreover, the difference between the quality of scanned images and PDF images is very large.
- In addition to external factors, the problem will face challenges from within, such as page layout variation, uneven object distribution, elongation of space between objects (spacing), and diversity in the morphology of objects, such as border and nonborder categories. Moreover, unlike English documents, the typical extract of Vietnamese document images faces significant difficulties due to their own expressions in the text language. The most obvious is that the feature classes are expressed in terms of terms meaning Caption. Separately, the Formula object class (Formula), in addition to the usual mathematical formulas containing equations and math symbols learning, is also represented as text (not belonging to the math area), which is also a major challenge for the problem.

In this study, we focus on Vietnamese image documents. There are some different characteristics between Vietnamese documents and English documents. First, although English and Vietnamese both use Roman characters, the Vietnamese language further uses diacritics, and UTF-8 characters display them. This observation means that Vietnamese documents use many more characters than English documents, and this difference should make the models trained on English documents work poorly in Vietnamese. The reason can be technically explained by the fact that the CNN-based backbones trained on English documents cannot produce the feature maps that describe the diacritics' information in images, leading to poor performance in caption detection. Moreover, caption objects are often placed near tables or figures; this may affect the pattern recognition characteristic of deep CNN models, and the detection performance on tables and figures may also be poor. We also do experimentation to confirm this hypothesis. Second, the Vietnamese document layout is also different. While English documents often use a no-border style with large tables (Figure 3a), Vietnamese documents have bordered smaller tables (Figure 3b). In addition, English documents place the table either on top or at the bottom of the document. Meanwhile, small tables can be arbitrarily found in a Vietnamese document. Also, the positions of captions are worth discussing; they sometimes are placed next to figures or tables (Figure 4a) instead of above or below them (Figure 4b). Therefore, there is a legitimate need to explore and develop a specific deep learning-based object detection model for page object detection in Vietnamese document images.

Our prior work on object detection in Vietnamese document images was published in CAIP 2021 [8]. In this journal version, we further extend the conference version. We would like to highlight the novelty and contribution of our paper. We expand the experiments on nine object detectors that were published in three recent years. We review all available loss functions in the MMDetection toolbox on these object detectors and propose the combined loss function for the improvement. SABL-Cascade achieves the highest results by experiments; therefore, we extend investigations by replacing default RoIAlign with PrRoI in the RoI Pooling module on SABL-Faster. The state-of-the-art performance demonstrates the efficiency of this change. Our contribution is summarized as follows.

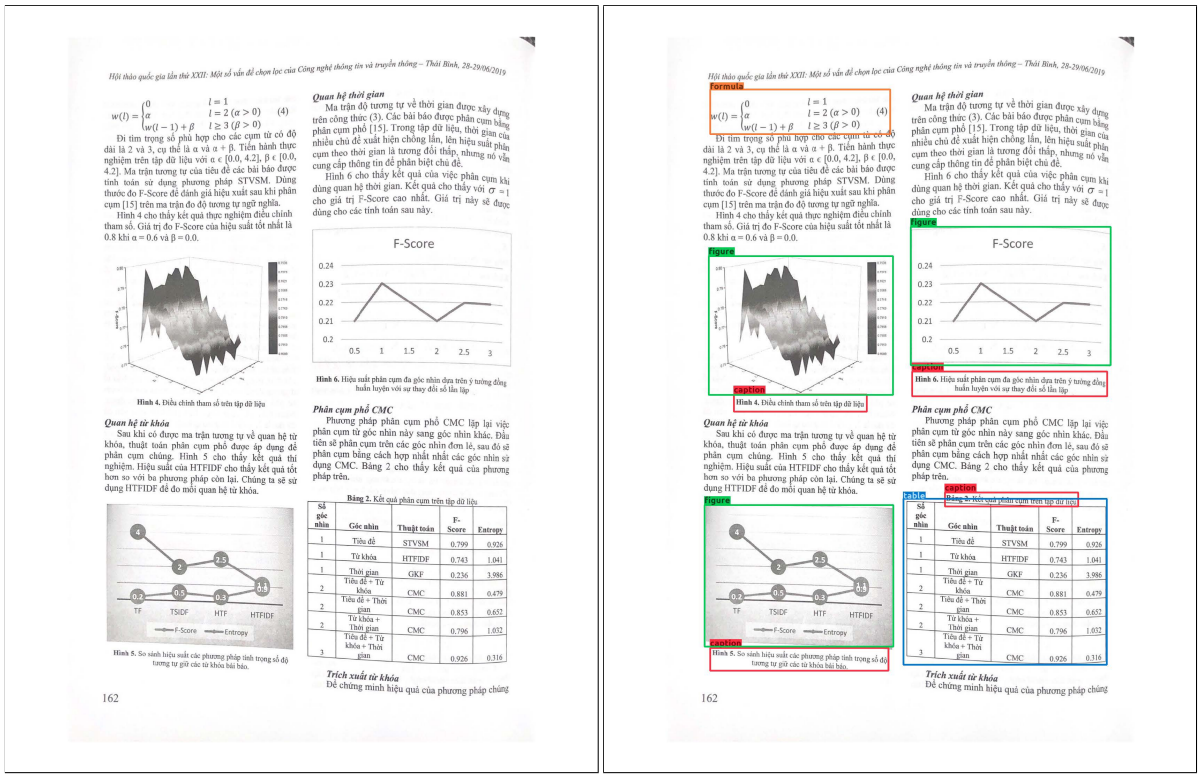
- To the best of our knowledge, **we are among the first to conduct research on Vietnamese document image understanding.**
- **We conduct the extensive benchmark on the UIT-DODV dataset, which is the first Vietnamese dataset.** Specifically, recent advanced models such as AutoAssign [9], ATSS [10], Double Head [11], GRoIE [12], SABL-Cascade [13], Faster RCNN [14], Generalized Attention [15], Libra R-CNN [16], Weight Standard [17], and CARAFE [18] methods are investigated in this article. In Figure 1, we briefly compare object detection methods on the UIT-DODV dataset regarding AP score and the number of parameters after experiments. Finally, **we conduct experiments using four different loss functions for the classification task.** The classification losses used to run experiments are cross-entropy loss, focal loss [19], fused loss [8] and GHM loss [20].

II. RELATED LITERATURE

A. EXISTING DATASETS

Detecting objects in image documents is one of the problems that has received the research community's interest in document layout analysis and document image understanding. There are many related studies as well as benchmarks for this problem that have been published worldwide. Details of the datasets mentioned above are described in Table 1.

The **Marmot [21]** dataset of 2,000 pages in PDF format was used for the article table detection algorithm; other datasets for the task of formula detection are taken from 400 pages of research papers with 1,575 isolated formulas and 7,907 embedded formulas from 194 digitally sourced PDF documents.



a. Input Document Image **b. Output Detection Results**

FIGURE 2. Object detection problem in Vietnamese document images. a. Input is a document page image; b. Output is the position of formulas (orange), caption (red), table (blue), figure (green). Please zoom in for viewing ease.

The **POD** contest dataset [22] includes 2,000 images of language document pages that were selected from 1,500 scientific articles by CiteSeer. The dataset represents the diverse formats for both page layout and object types, including single-column pages, two-column pages, multi-column pages and different types of formulas, tables, graphics, and figures.

The **TableBank** dataset [23] includes more than 278,000 images with more than 47,000 table objects. A total of 200,000 images edited in Latex are scientific articles collected from the ArXiv.org site.

PubLayNet [24] is the largest document image dataset ever, including 358,353 photos from research documents and scientific articles in medical fields with five object classes. The main object includes important elements related to document layout: title, text, figure, table and list. PubLayNet is used in the ICDAR 2021 competition in document layout recognition and detection board tasks.

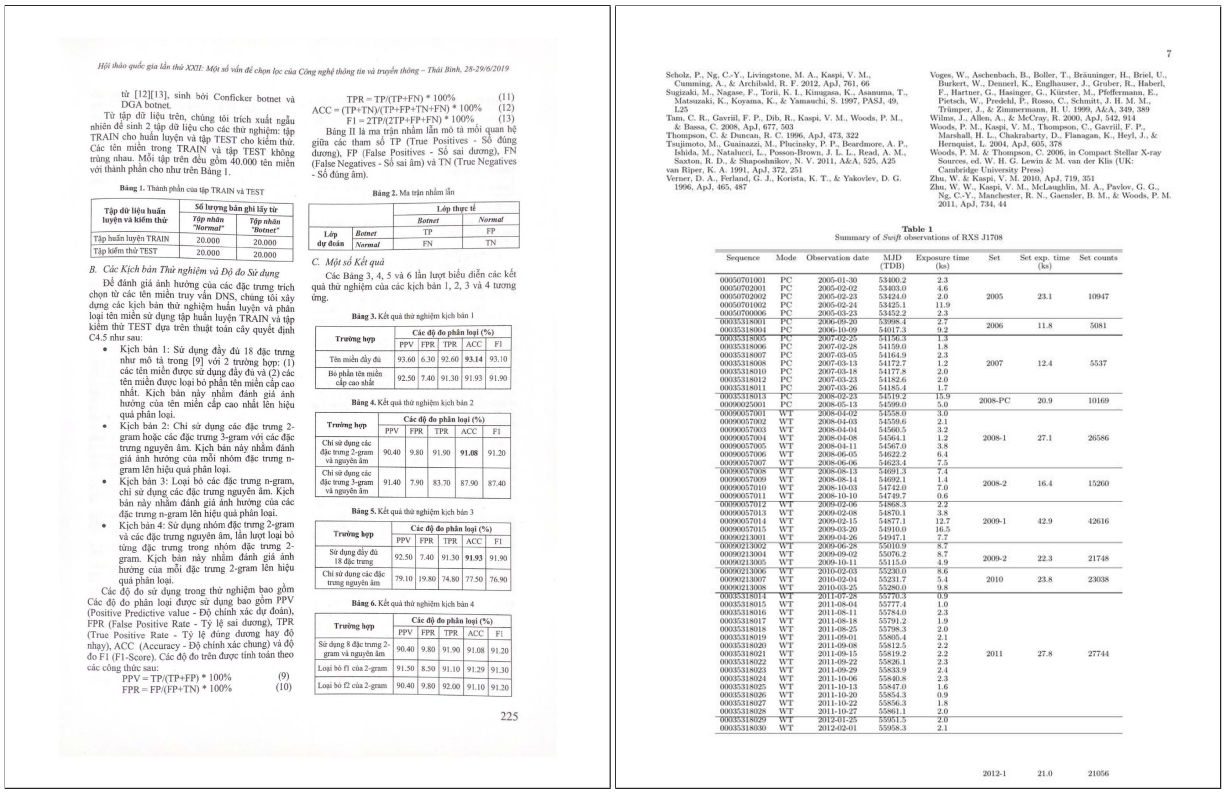
In the ICDAR 2019 competition, **cTDaR 2019** [25] is the dataset used with two New editions, including modern printed materials and archives. This is the first dataset that contains historical documents with handwritten and printed tables. The number of images in the cTDaR dataset is dependent on the tracks of the competition; however, the maximum is 799 and 840 images for historical and modern datasets, respectively.

DocBank [26] is an extended version of the TableBank dataset, which contains linguistic units. Other meanings are also included for document layout analysis. In this dataset, the following semantic structures are annotated in DocBank: Abstract, Author, Caption, Equation, Figure, Footer, List, Paragraph, Reference, Section, Table and Title.

UIT-DODV [8] is the first Vietnamese document dataset with 2,394 document images with 4 object classes, including Table, Figure, Caption, and Formula.

B. RELEVANT WORKS

In 2018, Kerwat *et al.* [28] used SSD [29] for object detection tasks in image documents on the ICDAR 2013 dataset. YOLOv3 [30] is a well-known algorithm for real-time object detection, which Huang *et al.* [31] used for a table detection task in 2019. Later, Ren *et al.* [32], for document layout detection combined context information to improve region detection performance. The experimental results have shown that the proposed method has 23.9% better mAP and 14 times faster processing speed than the text-based technique. Sun *et al.* [33] proposed the combination of the Faster R-CNN method and corner locating for table detection. The proposed method includes two stages: 1) the table detection results and the angular coordinate of the original object will be predicted by Faster R-CNN; and 2) the coordinate matching algorithm will group the angular coordinates that belong to the same table object. The result achieves F1 94.9% accuracy on the ICDAR2017 dataset, 2.8% higher than the traditional Faster R-CNN. Siddiqui et al. [34] introduced



a. An example from the UIT-DODV dataset **b. An example from the DocBank dataset**

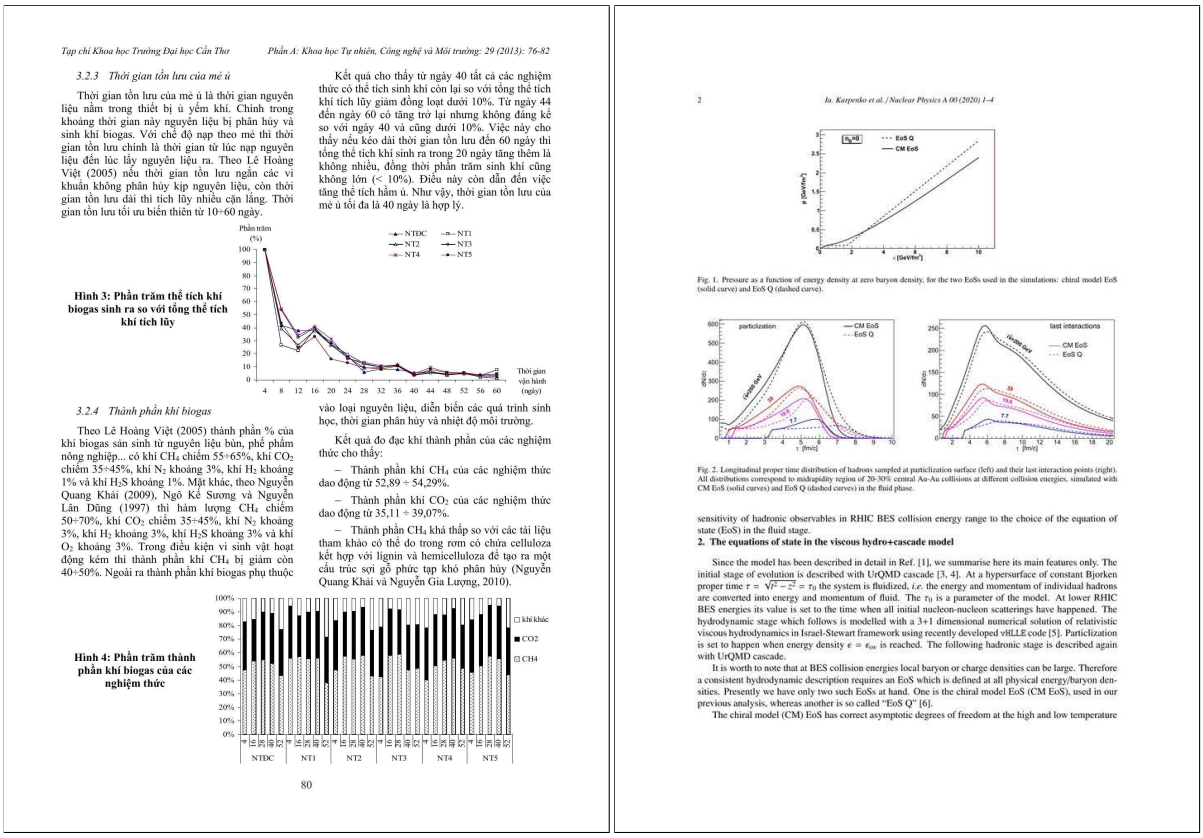
FIGURE 3. **Illustration of the difference in table style in Vietnamese and English document images.**

TABLE 1. ***The statistics of publicly available datasets for the task of document image analysis. Note that UIT-DODV*** **[8] collected by us is the first dataset about Vietnamese document images.**

Dataset	Images	Categories	Coverage	Source	Year
Marmot[21]	2,000	Table	English, Chinese	Founder Apabi Library and Citeseer website	2012
	400	Formula			
POD [22]	2,000	Table, Figure, Formula	scientific papers English	CiteSeer	2017
TableBank [27]	417,234	Table	English, Chinese, Japanese, Arabic	Miscellaneous	2019
PublayNet [24]	358,353	Title, Text, List, Table, Figure	Medical	PubMed Central	2019
CTDaR2019	840	Table	Printed documents	Miscellaneous	2019
Modern Dataset [25]	799		Historical documents		
DocBank [26]	500,000	Title, Abstract, Author, Table, Section, Footer, Equation, Figure, Caption, Paragraph	Physics, Mathematics, Computer Science	arXiv.com	2020
UIT-DODV [8]	2,394	Table, Figure, Formula, Caption	Vietnamese research papers	VNICT (@),CTU	2021

the combination of Faster R-CNN and deformable convolutional neural network to analyze the tabular structure of image documents. The two datasets used are ICDAR2013 and TabStructDB. The method achieves the highest efficiency on ICDAR2013 at the publication time (F1 92.98%). On the TabStructD dataset, the method achieved accuracy (F1 93.72%).

Recently, Zhong *et al.* [35] collected a large dataset of scientific papers named PubTabNet and proposed an EDD



a. An example from the UIT-DODV dataset **b. An example from the DocBank dataset**

FIGURE 4. **Illustration of the difference in the position of the caption in Vietnamese and English document images.**

method based on the encoder-decoder structure to convert PDF documents to HTML structures at the same time. TED measurements are also suggested to evaluate effectiveness. Zheng *et al.* [36] introduced an end-to-end framework that not only detected but also recognized table structures in document images. Module Global Table Extractor (GTE), which is proposed, can be placed at the top of object detection methods. The study also introduces the FinTabNet dataset in the financial field. Agarwal *et al.* [37] presented the composite deformable cascade network (CDeC-Net) method to solve detecting tables in document images. This study is based on a novel cascade Mask R-CNN [38] and a dual backbone architecture [39].

Many studies have applied and improved common object detection methods such as SSD, YOLOv3, Faster R-CNN, and Mask R-CNN in the Document Image field. The evaluation and analysis of new methods in recent years on Vietnamese document image - UIT-DODV promises to provide a lot of helpful information as a fundamental for future extensive studies.

III. OBJECT DETECTION MODELS

Page object detection can be regarded as an object detection task in document images based on popular object detection algorithms. Therefore, we review the state-of-the-art object detection methods for page object detection in this section, which are leveraged for page object detection in our research.

To the best of our knowledge, object detection algorithms can be divided into one-stage object detection and two-stage object detection.

A. ONE-STAGE METHOD

In this subsection, we review AutoAssign [9] and ATSS [10] object detection methods. AutoAssign is a dense object detector considered a one-stage anchor-free detector. ATSS is not a complete object detection method, just a module which can be integrated into any one-stage anchor-free (such as FCOS) or one-stage anchor-based method (such as RetinaNet). However, for simplicity, we list them as ''one-stage'' methods.

1) AutoAssign

AutoAssign [9] is a single-stage object detection method. It requires very little prior knowledge (thresholds for selecting positive and negative samples) and is highly efficient through a weight distinction mechanism.

As shown in Figure 5, the grey framework illustrates the network architecture. The first followed an anchor box-free method such as FCOS (fully convolutional one-stage) [40]

to remove predefined anchors and directly predict objects at each feature position. The network architecture has three outputs: the classification score, the implicit objectness score, and the feature coordinates. During training (blue framework below), all the predictions of the above architecture are converted into a common confidence index first. Above all, a weighting mechanism has been proposed, which consists of a module of center weighting and confidence weighting. The center weights module is designed to respond to the pre-centrality property inherent in the data and adapt to each class's specific patterns. It starts from the standard central attribute and then learns the distribution of each class in the data. The confidence weights module is used to assign the most appropriate positions of each sample based on its occurrence and size accordingly. Both modules combine to generate positive and negative weights for each position i in the ground-truth bounding box. Finally, the positive and negative loss functions will be calculated, and the positive-negative sample labelling will be optimized along with the network architecture.

From a positive-negative labelling point of view, for an object, AutoAssign can automatically find its appropriate scale on FPN (feature pyramid network) levels and spatial locations based on the network's output. As a result, labeling is appropriately resolved in a uniform, recognizable and distinguishable manner.

2) ADAPTIVE TRAINING SAMPLE SELECTION (ATSS)

ATSS [10] is a method that automatically selects positive and negative samples based on the statistical characteristics of proposal regions.

For each ground-truth envelope g on the image, Zhang et al. first looks for positive samples. At each feature level, they choose k anchor boxes whose center coordinates are closest to the g-box center coordinates based on the L2 distance. Assuming there are L feature classes, the label box g has $k \times L$ positive recommendation regions. Then, calculate the IoU between these proposal boxes and the ground truth set. With these statistics, the IoU (intersection over union) threshold for the ground truth box g is adjusted using the formula $t_g = m_g + v_g$. Finally, they select proposal boxes with $IoU \geq t_g$ as the last positive proposal boxes.

Note that ATSS also restricts positive samples neighboring ground truth g. In addition, if an anchor box is assigned to more than one ground-truth box, the box with the highest IoU is selected, and the rest are considered negative samples.

B. TWO-STAGE METHODS

In this subsection, we review the state-of-the-art two-stage object detection methods. Note that all methods in this section are almost improved versions of Faster R-CNN [32], which is mainly focused on Balance Sampling [16], Deep Convolution Network for Feature Extraction [17], [15], Feature Pyramid Network (FPN) [16], [12], [18], and Regression and Classification tasks [11], [13].

1) FASTER R-CNN

Faster R-CNN is the improved version of Fast R-CNN. Ren *et al.* [32] proposed a region proposal network (RPN), replacing selective search to generate better proposal regions; this architecture will then be trained with Fast R-CNN. These improvements have reduced the number of proposal regions and increased the operating speed during model testing to near real-time with the best performance, approximately 5 fps on a single GPU. Faster R-CNN is the premise method for many later object detection methods. Within Faster R-CNN, an input image will pass through a CNN architecture and output a feature map. This feature then goes through the RPN to generate suggested regions with or without objects. These regions will pass through the RoI pooling layer to be resized to the same size and then classified and location refined by Fast R-CNN.

2) WEIGHT STANDARD

Batch normalization is a data normalization technique that gives outstanding results. However, Qiao *et al.* [17] argue that with the microbatch training schedule, this method has limitations. The reason is that when training a microbatch on multiple GPUs, each GPU only receives 1-2 images, thus causing batch normalization to reduce performance significantly. Indeed, one GPU receiving too few images is a common problem due to insufficient resources in the computer vision field. Therefore, the Weight Standard was proposed to overcome this issue.

The main idea of the Weight Standard is to normalize the weights on the kernel. Typically, a convolution is defined as follows:

$$\mathbf{Y} = \mathbf{W}.\mathbf{X} \tag{1}$$

where $W \in \mathbb{R}^{O \times I}$ is the weight in the kernel and $\mathbf{X}, \mathbf{Y}$ are the output and the input of the convolution. O is the number of output channels, and I is the number of input channels in the kernel of each output channel. The Weight Standard Weight Standard will normalize the weight $\mathbf{W}$ according to the following formula:

$$\widehat{\mathbf{W}} = \left[\widehat{\mathbf{W}}_{i,j} \mid \widehat{\mathbf{W}}_{i,j} = \frac{\mathbf{W}_{i,j} - \mu \mathbf{w}_i, .}{\sigma \mathbf{w}_i, .} \right] \tag{2}$$

where:

$$\mu \mathbf{w}_{i,.} = \frac{\mathbf{1}}{\mathbf{I}} \sum_{\mathbf{j=1}}^{I} \mathbf{W}_{i,j} \tag{3}$$

$$\sigma \mathbf{w}_{i,.} = \sqrt{\frac{\mathbf{1}}{I} \sum_{j=\mathbf{1}}^{\mathbf{I}} \mathbf{W}_{i,j}^2 - \mu \mathbf{w}_{i,}^{\mathbf{2}} \ . + \epsilon} \tag{4}$$

Therefore, the output of the convolution is now calculated as follows:

$$\mathbf{Y} = \widehat{\mathbf{W}}.\mathbf{X} \tag{5}$$

3) GENERALIZED ATTENTION

Generalized Attention [15] is a synthesized study of different spatial attention factors in a general attention formula, including the attention mechanism in transformer architecture.

Given a query element and a set of element keys, an attention function will aggregate the content keys based on attention weights that measure the correspondence of the query-key pair. For the model to participate in content keys from different representation subspaces and locations, the outputs of many attention functions are linearly summed with learnable weights. Let q index a query element with content z_q, and let k index a key element with content x_k. Then, the multihead attention y_q feature will be computed by the formula:

$$y_q = \sum_{m=1}^{M} W_m \left[\sum_{k \in \Omega_q} A_m(q, k, z_q, x_k) \odot W'_m x_k \right] \tag{6}$$

where m indicates the attention head, Ω_q identifies the key regions that support the query, $A_m(q, k, z_q, x_k)$ denotes the attention weights in the m^{th} attention head, and W_m and W'_m are the weights that can be learned. Usually, attention weights are normalized inside Ω_q, such as $\sum_{k \in \Omega_q} A_m(q, k, z_q, x_k) = 1$.

In recent related studies on transformer attention, the attention weight of each query-key pair is calculated as the sum of four parts $\{\varepsilon_j\}_{j=1}^{4}$ based on different attention factors, such as:

$$A_m^{Trans}(q, k, z_q, x_k) \propto \exp\left(\sum_{j=1}^{4} \varepsilon_j \right) \tag{7}$$

normalized by $\sum_{k \in \Omega_q} A_m(q, k, z_q, x_k) = 1$ when key regions supporting Ω_q span element keys (e.g., the whole input sequence). By default, 8 attention heads will be used in the research.

Zhu *et al.* [15] incorporate various attention mechanisms into deep networks to explore their effects. For object detection tasks, ResNet50 is chosen as the backbone for feature extraction, and only the self-attention mechanism is involved. In detail, the self-attention mechanism is incorporated into the residual block, called the "attended residual block," and only applied in the last two stages (conv4 and conv5 stages). Faster R-CNN with a feature pyramid network is chosen as the baseline detector for experiments.

4) LIBRA R-CNN

Libra R-CNN [16] is an innovative object detection method that addresses imbalances in the training process. Pang et al. suggested that this imbalance lies at three levels: the sample level, feature level, and object level. To solve this, balanced sampling based on IoU (IoU-balanced sampling), balanced feature pyramid (balanced feature pyramid) and Balanced L1 loss function (L1-balanced loss), which respond to the above three imbalance problems, have been proposed.

a: BALANCE IoU SAMPLING

Based on the conclusion that imbalance will cause difficult samples to be masked by thousands of easy samples, the IoU sampling balancing method is proposed to find more difficult samples at no additional cost.

Suppose we need to select N difficult samples in M proposed regions. The probability of being selected for each sample in random sampling is:

$$p = \frac{N}{M} \tag{8}$$

To increase the selectivity of hard negatives, the sampling interval is divided into K equal parts based on IoU. N is the number of hard-to-negative samples in each of these equal parts. Then, samples were uniformly selected from these sections. Therefore, the possibility of taking these difficult samples is redefined as follows:

$$p_k = \frac{N}{K} \times \frac{1}{M_k}, \quad k \in [0, K) \tag{9}$$

where $\mathbf{M_k}$ is the number of proposals in the $\mathbf{K_{th}}$ part. The original paper is divided into three parts ($K = 3$).

b: EQUALIZING PYRAMID FEATURES

Unlike previous studies on FPN, PAANet combines multilevel features using the two-way connection; the idea is to reinforce multilevel features by using balanced semantic features. Full integration includes four steps: resizing, integrating, refining and strengthening.

c: OBTAIN BALANCED SEMANTIC FEATURES

the feature at $\mathbf{l}$ - level resolution is denoted as $\mathbf{C_l}$. The number of multilevel features is denoted as $\mathbf{L}$. The highest and lowest levels are denoted by $\mathbf{l_{min}}$ and $\mathbf{l_{max}}$, respectively. In the image above, $\mathbf{C_2}$ is the top-level resolution. To integrate multilevel features and keep semantic hierarchies at the same time, the multilevel features are resized $\mathbf{C_2}, \mathbf{C_3}, \mathbf{C_4}, \mathbf{C_5}$ to an intermediate size. Once the features are rescaled, the balanced semantic feature is obtained by simple averaging:

$$\mathbf{C} = \frac{1}{L} \sum_{l=l_{min}}^{l=l_{max}} C_l \tag{10}$$

The obtained features are then scaled using the same but inverse process to enhance the original features.

d: REINFORCE SEMANTIC FEATURES

equilibrium semantic features will be consolidated later. The consolidation step will help enhance specific features to improve results. With this method, low-level to high-level features will be aggregated at the same time. Output P_2, P_3, P_4, P_5 is used for FPN-like object detection.

e: EQUILIBRIUM LOSS FUNCTION L1

In this paper, a balanced loss function L1 is proposed. Balanced L1 is used to increase the contribution of current

observations (inliers); this cost is defined as follows: (11), as shown at the bottom of the page, where b ensures $L1_{balanced}\left(\hat{x}=1\right)$ is a continuous function, Z is a constant and the relationship between the coefficients α, b, γ is determined as follows:

$$\alpha \ln\left(b+1\right) = \gamma \tag{12}$$

As recommended in the original article, α and γ are set as 0.5 and 1.5, respectively.

5) CONTENT-AWARE REASSEMBLY OF FEATURES (CARAFE)

Feature upsampling is a common practice in thick detection problems such as object detection or object segmentation. It is an integral part of high-to-low or low-to-high feature hybrid architectures such as FPN, U-Net, and Stacked Hourglas. Content-Aware ReAssembly of Features (CARAFE) [18] is a universal, simple and highly efficient operator for this purpose.

CARAFE acts as a clustering operator with content-aware kernels consisting of two steps. The first step is to predict a clustered kernel for each target location based on its content, and the second step is to cluster the features with the predicted kernel. Given a feature X of size $C \times H \times W$ and size increase ratio σ. CARAFE will produce a new feature X' of size $C \times \sigma H \times \sigma W$. Any target location $l' = (i', j')$ of X' will have a target location $l = (i, j)$ at input characteristic X, where $i = \lfloor j'/\sigma \rfloor$ and $j = \lfloor j'/\sigma \rfloor$. Here, $N\left(X_l, k\right)$ is denoted as a subregion of size $k \times k$ of the input feature X located between position l, so-called neighborhood X_l.

6) DOUBLE HEAD

The two-layer structure (one fully connected layer and one convolutional layer) is used extensively in R-CNN-based object detection methods for two jobs: recommendation box classification and coordinate regression. However, in their study, Wu *et al.* [11] suggest that there is a certain lack of understanding of how these two classes can work for both. The results show that the fully connected layer is more suitable for the proposed box classification, and the convolution layer is more suitable for coordinate regression. Here, the output of the fully connected layer is more spatially sensitive than that of the convolutional layer. Therefore, the double-head method was proposed, i.e., using a fully connected layer for classification and a convolution layer for box regression.

7) GENERIC RoI EXTRACTOR (GRoIE)

In two-step object detection methods such as Faster R-CNN, the region of interest layer plays an important role. Specifically, it is used to extract a consistent subset of features from an FPN network layer placed at the top of the architecture. Realizing that previous RoI classes only selected the best layer from the FPN as a limitation, Rossi *et al.* [12] proposed the Generic RoI Extractor (GRoIE), which introduces nonlocal building blocks and an attention mechanism to increase performance.

Specifically, GRoIE is mentioned to include the following 4 modules:

a: RoI POOLER MODULE

Here is a module that uses RoI Align on heterogeneous proposed regions to obtain fixed size representations.

b: PREPROCESSING MODULE

The goal of this module is to apply preliminary pooling to pooled regions. This module is used to preprocess the features and is usually applied by a convolution layer associated with each aspect ratio.

c: AGGREGATION MODULE

This module defines how the single RoIs coming from each branch can be aggregated. The most commonly used operators are concatenation and summation.

d: POSTPROCESSING MODULE

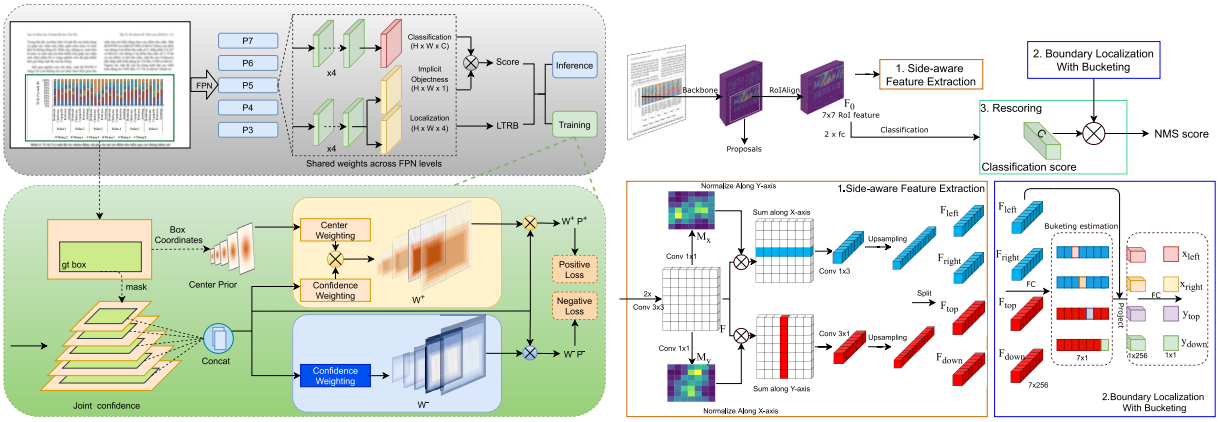
This is an additional postprocessing step that applies to features that have been merged before returning. It allows the network to learn global features considering all dimensions.

8) SIDE-AWARE BOUNDARY LOCALIZATION (SABL)

Existing object detection methods depend on bounding box regression for object locating. Although there have been attempts to improve processes in recent years, the accuracy of envelope regression has not been satisfied, leading to this being a limitation of object detection. Wang *et al.* [13] found that previous approaches focused only on predicting center coordinates and dimensions (x, y, w, h), which is not an efficient way to perform regression bounding boxes, especially when large displacements and variances exist between the anchor boxes and the ground truth. Therefore, the Side-Aware Boundary Localization (SABL) method is proposed, where each side of the envelope would be located in turn with a dedicated network branch.

SABL will first extract the horizontal and vertical features ($\mathcal{F}_x$ *and* $\mathcal{F}_y$) by combining the RoI $\mathcal{F}$ features along the X and Y axes, respectively. $\mathcal{F}_x, \mathcal{F}_y$ will be divided into side-aware features $\mathcal{F}_{left}, \mathcal{F}_{right}, \mathcal{F}_{top}, \mathcal{F}_{down}$. Then, on each side of the bounding box, the SABL first divides the target spaces into groups and searches for the boundary container by taking advantage of side-aware features. It refines the

$$L1_{balanced}(\hat{x}) = \begin{cases} \frac{\alpha}{b}\left(b \mid \hat{x} \mid +1\right) \ln\left(b \mid \hat{x} \mid +1\right) - \alpha \mid \hat{x} \mid, & \text{if } \mid \hat{x} \mid < 1 \\ \gamma \mid \hat{x} \mid + Z, & \text{if } \mid \hat{x} \mid \geq 1 \end{cases} \tag{11}$$



AutoAssign [9] Side-Aware Boundary Localization [13]

FIGURE 5. **Two notable object detection methods, namely, AutoAssign (one-stage) and Side-Aware Boundary Localization (two-stage).**

boundary coordinates $x_{left}, x_{right}, y_{top}, y_{right}$ by further predicting their offset. Such a two-step clustering framework reduces regression variance and eases prediction difficulties. Furthermore, the reliability of the estimated groups can also help to correct the classification score and further improve the performance. Not only is it applicable to two-stage methods, it is also applicable to one-stage detection methods.

In Figure 5, we depict the input processing as document images on two methods, AutoAssign and SABL, representing the one-stage and two-stage detectors in our experiments. AutoAssign improves the label assignment task in anchor-free detectors by proposing two modules, center weighting and confidence weighting, to calculate positive and negative weights for adjusting the category-specific prior distribution and the instance-specific sampling strategy in both spatial and scale dimensions. Meanwhile, SABL focuses on improving the localization task, where each side of the bounding box is respectively localized with a dedicated network branch. Figure III-B6 and Figure III-B6 show the AutoAssign and SABL method processing on the input Vietnamese document image.

C. LOSS FUNCTIONS

Loss functions are an essential factor affecting the detection performance in object detection tasks. The loss functions of object detection are categorized into classification loss and localization loss. In our research, we focus on exploring the effect of the classification loss function on object detectors. This improves the precision of classifying semantic classes, which is a challenging problem in analyzing document images.

1) CROSS ENTROPY LOSS (CE)

Let $\mathbf{p}$ be the label probability, $\mathbf{q}$ be the prediction probability, and $\mathbf{C}$ be the number of classes. Cross-entropy loss is defined as follows:

$$\mathcal{L}_{\mathbf{CE}} = -\sum_{\mathbf{i=1}}^{\mathbf{C}} \mathbf{p_i log(q_i)} \tag{13}$$

The CE loss is used for the problem of classifying positive or negative suggested boxes, which means the number of classes is 2 (C = 2). For CE, the class distribution is assumed to be balanced; however, we would like to consider the unbalanced scenario where we need a different loss function that handles the minority classes to be classified more accurately. This case is problematic to the object detector since the positive recommended regions are few, whereas the negative proposal regions dominate.

2) FOCAL LOSS (FL)

Originally introduced by Lin *et al.* [19] in an attempt to improve the single-stage method, this loss function is applied to the method and named RetinaNet. Focal loss is defined as follows:

$$\mathcal{L}_{\mathbf{FL}} = -\sum_{\mathbf{i=1}}^{\mathbf{C}} (1 - q_i)^{\gamma}\, \mathbf{p_i log(q_i)} \tag{14}$$

As shown, focal loss adds the factor $(1 - q_i)^{\gamma}$ to the CE function. This multiplier is very effective in adjusting the effect of labels on the loss function and gradient descent simultaneously. For classes with majority samples, the probability of guessing these samples is usually correct and larger, $(1 - q_i)^{\gamma}$ will tend to be smaller and have almost no impact on the loss function. For classes with minority samples, the probability of predicting these samples is small, making $(1 - q_i)^{\gamma}$ closer to 1, and the impact will be larger.

3) FUSED LOSS

To take advantage of both CE and FL loss functions, we combine both loss functions with a trade-off parameter:

$$\mathcal{L}_{\mathbf{Fused}} = \alpha \mathcal{L}_{\mathbf{FL}} + (1 - \alpha)\, \mathcal{L}_{CE} \tag{15}$$

TABLE 2. Experimental results of different object detection methods on dataset UIT-DODV. The highest results of each method are highlighted in **bold** style. The AP scores of the most outperformed method are highlighted in the grey cell.

Method	Classfication loss	Regression loss	$AP_{caption}$	AP_{figure}	AP_{table}	$AP_{formula}$	AP	AP@50	AP@75
YOLOv4 [8]		CIoU	60.8	78.0	84.2	40.2	65.8	90.2	75.2
YOLOv4 mish [8]			61.3	75.7	82.0	45.2	66.1	90.7	77.7
AutoAssign		GIoU	68.1	77.0	90.3	35.9	67.8	87.6	73.8
		DIoU	69.3	78.2	88.0	37.0	**68.1**	88.9	73.7
		CIoU	67.5	77.9	89.3	37.2	68.0	88.3	74.1
ATSS	Focal Loss	L1	**62.1**	**77.6**	**88.5**	**33.1**	**65.1**	**85.7**	**70.5**
	Fused Loss		31.4	54.3	65.8	7.9	39.9	65.5	41.6
	GHM Loss		51.1	65.6	76.8	14.3	52.0	75.2	56.6
Double Head	Cross Entropy	L1	**66.9**	79.8	**91.4**	**46.9**	**71.2**	**88.6**	**79.9**
	Focal Loss		65.4	78.8	90.4	43.5	69.5	88.4	76.5
	Fused Loss		66.7	**79.9**	91.0	46.0	70.9	88.1	78.6
GRoIE	Cross Entropy	L1	**64.2**	**79.4**	**91.0**	**44.1**	**69.7**	**88.5**	**77.1**
	Focal Loss		61.2	76.1	89.3	38.2	66.2	86.1	72.3
	Fused Loss		62.6	78.5	90.7	43.7	68.9	87.6	76.4
SABL-Cascade	Cross Entropy	SmoothL1	**76.2**	**86.6**	**95.9**	50.1	**77.2**	**90.6**	**83.5**
	GHM Loss		73.9	85.6	94.7	**50.6**	76.2	89.9	82.9
Faster R-CNN	Cross Entropy	L1	**65.3**	79.4	90.0	46.4	70.2	**89.3**	**78.6**
	Fused Loss		61.5	**81.3**	**92.8**	**47.8**	**70.9**	87.1	77.7
	GHM Loss		61.2	74.0	87.8	36.1	64.8	87.0	71.6
Generalized Attention	Cross Entropy	L1	65.1	78.4	**90.5**	42.9	69.2	**88.8**	75.6
	Fused Loss		**65.2**	**78.7**	89.7	**43.9**	**69.4**	88.3	**76.5**
	GHM Loss		52.8	65.7	82.7	33.8	58.7	84.6	64.4
Libra R-CNN	Cross Entropy	BalancedL1	54.6	75.0	86.3	39.1	63.8	85.0	71.0
	Fused Loss		55.3	**75.9**	86.8	**40.2**	64.6	84.9	**72.0**
	GHM Loss		**58.4**	75.3	**87.0**	38.0	**64.7**	**86.7**	71.4
Weight Standard	Cross Entropy	L1	**63.1**	**78.4**	90.7	**44.9**	**69.3**	**88.8**	**76.3**
	Fused Loss		59.5	**78.4**	**91.4**	42.2	67.9	85.9	75.3
	GHM Loss		61.5	76.9	88.9	41.3	67.2	**88.8**	74.3
CARAFE	Cross Entropy	L1	**66.2**	78.0	89.1	41.7	68.7	**89.5**	**75.7**
	Fused Loss		64.7	**79.5**	**91.3**	**42.4**	**69.5**	88.5	75.5
	GHM Loss		63.0	76.9	89.3	42.0	67.8	89.0	74.1

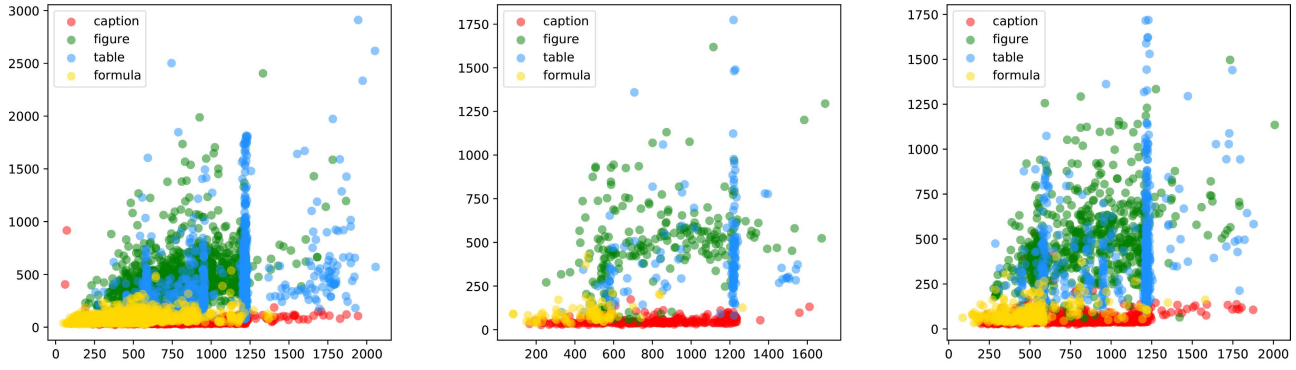


FIGURE 6. Distribution of various sizes of objects in three sets: training, validation and testing sets of the UIT-DODV dataset. Please zoom in for viewing ease.

This combined loss function will default to consider the contributions of the classes as fair, in addition to the contributions of the minority classes.

4) GHM LOSS

In an attempt to solve the class imbalance of proposal boxes. In the study [20], the loss function gradient harmonized mechanism-classification (GHM-C) is proposed:

$$L_{\text{GHM-C}} = \frac{1}{N}\sum_{i=1}^{N}\beta_i L_{CE}\left(p_i, p_i^*\right) = \sum_{i=1}^{N}\frac{L_{CE}\left(p_i, p_i^*\right)}{GD(g_i)} \quad (16)$$

where p is the probability of the suggested boxes; p^* is ground truth, β_i is the harmonic gradient density parameter of the i-th sample, and $GD(g_i)$ is the gradient density of sample I, determined by: $\beta_i = \frac{N}{GD(g_i)}$.

IV. BENCHMARK SUITE

In this section, we present the benchmark for Vietnamese document analysis. We mainly discuss the benchmark dataset for Vietnamese document images and the experimental process based on the aforementioned methods in Section III.

A. DATASET

The UIT-DODV dataset [8] is used for our extended experiments. There are four classes in this dataset: formula, figure, table, and caption, and the dataset is split into three sets: training (1,440 images), validation (234 images), and testing (720 images).

The UIT-DODV dataset is collected from various domains, *i.e.*, PDF (1,696 images), scanned by smartphone (451 images) and scanned by the physical scanner (247 images). Due to the variety of images, UIT-DODV poses many challenges for object detectors to work well in multiple domains. In addition, there are many research articles from many sources. Therefore, there are significant differences in layout. For example, it can organize the page into one column or double columns, depending on the template of the conference or journal. As a result, the location of objects (e.g., table, figure) is not fixed on different pages. Moreover, the primary language on UIT-DODV is Vietnamese, which has high differences in character with some accent symbols *(', ',?,.,∼)* and derivative characters *(ô, ò, ê, â, ă, ù)*. This contributes to the significant challenge of detecting semantic classes (e.g., formula, caption).

Moreover, we also visualize the distribution of object sizes in the training, validation and testing sets of the UIT-DODV dataset. The distributions seem quite similar between the three sets. Almost width values are between 0 to approximately 1250px, and the height value tends to be stable. The formula objects seem to be similar to caption objects. However, the range of width values is shorter. Width values of table objects seem to cluster at approximately 1200px and 500px whereas the height values are various, which are almost from approximately under 250px to 1000px. The figure objects have the most beautiful distribution between other types of objects, it seems to be quite linear along with width-axis and height-axis. Table objects tend to have higher height values (almost up to 2,000 px, and the maximum is approximately 3,000 px). This analysis shows that UIT-DODV reflects the natural distribution of Vietnamese documents in reality quite well. The number of objects is adequate to evaluate the performance of object detectors, and the experimental results on this dataset are worth discussing.

B. EXPERIMENTAL SETTINGS

In this study, we run all experiments by using the MMDetection toolbox [41], which implements recent new methods. The configuration we use: Intel(R) Xeon(R) CPU @ 2.30 GHz 2 cores; 25 GB Ram; and 01 Nvidia Tesla P100-PCIE 16 GB GPU. For a fair comparison, we use ResNet-50 as the backbone for feature extraction in all ten object detectors. All models mentioned in Section III are trained in 24 epochs, and the best of each is recorded. Training within 24 epochs is proven to help the models converge when trained on the MS-COCO dataset in the MMDetection toolbox [41]. Each detector is tested using the four different classification loss functions mentioned in Section III-C. However, several detectors are not adapted well with some loss functions. In particular, the loss value is NaN in the very first epochs. Therefore, not every method is tested with all four loss functions.

C. EVALUATION METRICS

We calculate the average precision metric using the COCO API.[1] We calculated the AP scores of all classes and took their average as the mean AP score (mAP). This process is formally defined as follows:

$$\text{AP}_\text{c} = \frac{1}{\#\text{T}}\sum_{IoU\in\text{T}} AP[c, IoU] \quad (17)$$

$$\text{mAP} = \frac{1}{\#\text{C}}\sum_{c\in\text{C}}\text{AP}_\text{c} \quad (18)$$

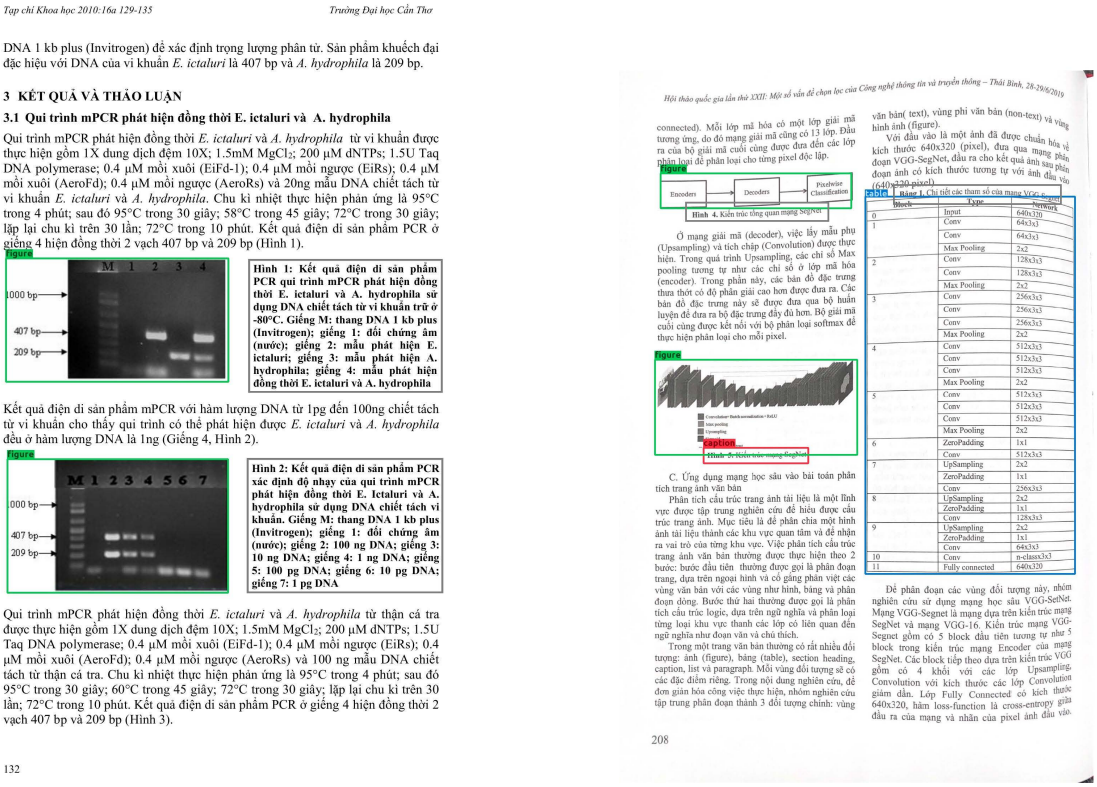
where AP_c is the average precision of the c-th class; C is the set of all classes in the dataset; and T is the set of IoU thresholds $\text{T} = 0.50 : 0.05 : 0.95$. In addition, we also calculate the mAP scores at $\text{IoU} = 0.5$ and $\text{IoU} = 0.75$, which are called AP@50 and AP@75, respectively.

D. EXPERIMENTAL RESULTS

All the aforementioned methods are available in the MMDetection toolbox. Furthermore, we perform experiments with different loss functions of classification and regression tasks in the second stage of detectors. Note that there are some modifications as follows.

- AutoAssign: This detector does not provide classification loss in its configuration; thus, we only perform experiments on regression loss.

[1] https://github.com/cocodataset/cocoapi



a) b)

FIGURE 7. Visualization of some common failure cases of caption objects. Gray bounding boxes mark missing objects.

- In the classification loss, we perform experiments on cross-entropy loss, focal loss, and fused loss by default. Here, we replace any incompatible loss in any detector with GHM loss. Except for the AutoAssign detector, all detectors experiment with different L1 losses in regression tasks. We experiment with different IoU losses in the AutoAssign detector because the L1 loss shows a very low performance on this detector.
- Regarding the fused loss function, we emphasize the focal loss function. As shown in Section III-C3, the focal loss has a better performance on the class imbalance; therefore, the classification performance is theoretically better. In particular, we apply $\alpha = 0.6$ to perform experiments in all methods where fused loss is applied. However, we also try different α values of 0.4, 0.5, and 0.6 in the double head method, which shows that $\alpha = 0.6$ is appropriate.

As a result, the highest result of each method ranges from 64.7% to 77.2% in terms of the mean average precision. The three methods that give the highest results are SABL (Cascade), Faster RCNN, and Double Head. Meanwhile, ATSS yields the lowest result 39.9%. We visualize the results of these 4 methods in Figure 12.

As shown in Figure 12, these four methods show good predictions for the Table class, and the predictions of the three other classes exhibit different mistakes depending on the detectors. SABL-Cascade detects enough and correct objects; the bounding boxes also perfectly surround objects belonging to four categories. The reason is that SABL Cascade is based on Cascade R-CNN, which contains multiple stages, and the next stage will improve the performance of the previous stage; therefore, the predictions can achieve the perfect performance. Moreover, SABL is the method that enhances the regression task by taking advantage of side-aware features; consequently, it is much more precise than other methods.

Compared to the SABL Cascade, the predicted boxes of the Double Head detector do not perfectly surround the objects. Double Head detects three Caption objects, while there are just two in the ground truth. Double-Head and Faster R-CNN show quite similar predictions because Double-Head is also based on Faster R-CNN. As shown in Figures IV-D and IV-D, redundant detections of Caption and Figure objects are observed in Double-Head and Faster R-CNN, respectively. This is a true reflection of the results in Table 2; the AP of Double-Head is higher than Faster-RCNN in the Figure class and vice versa.

The ATSS method only detects figures and tables, omits the two Caption objects, and gives the lowest results among the methods. The predicted bounding box of the figure objects also does not well surround the object. The reason is that ATSS is a single-stage detector; thus, its performance is

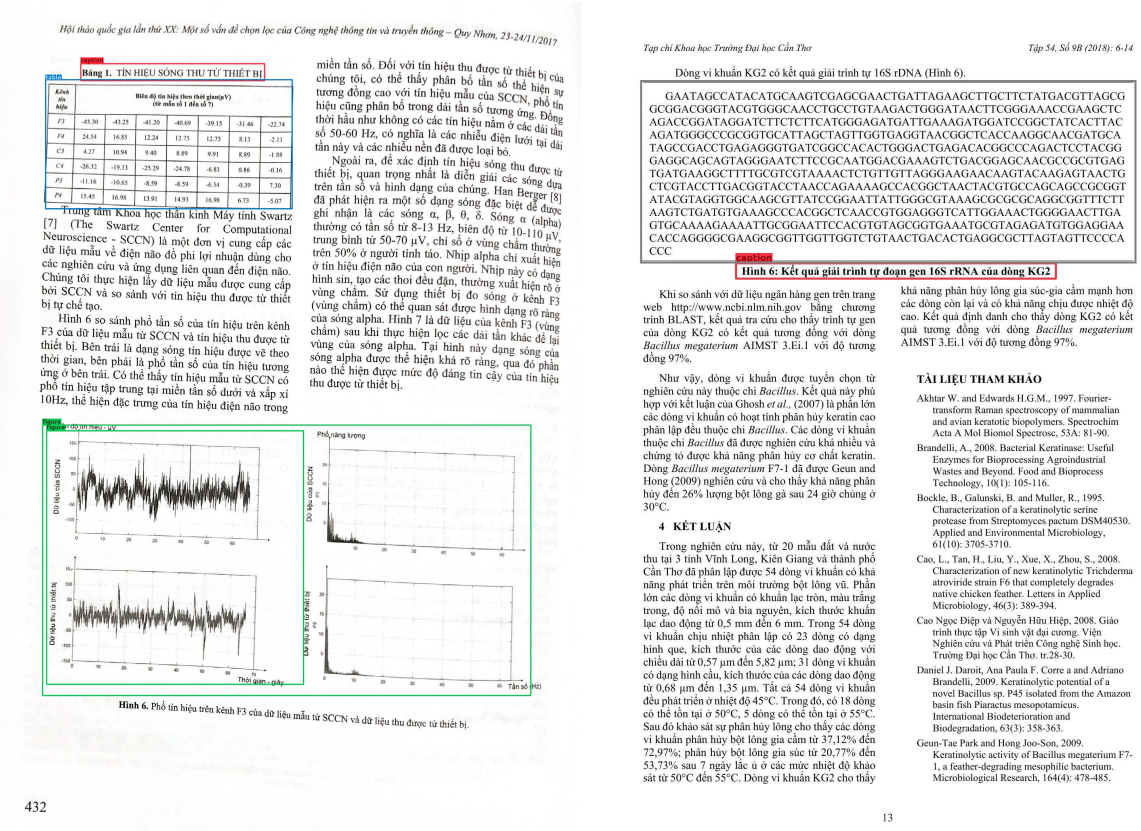


FIGURE 8. Visualization of some common failure cases of figure objects. Grey bounding boxes mark missing objects.

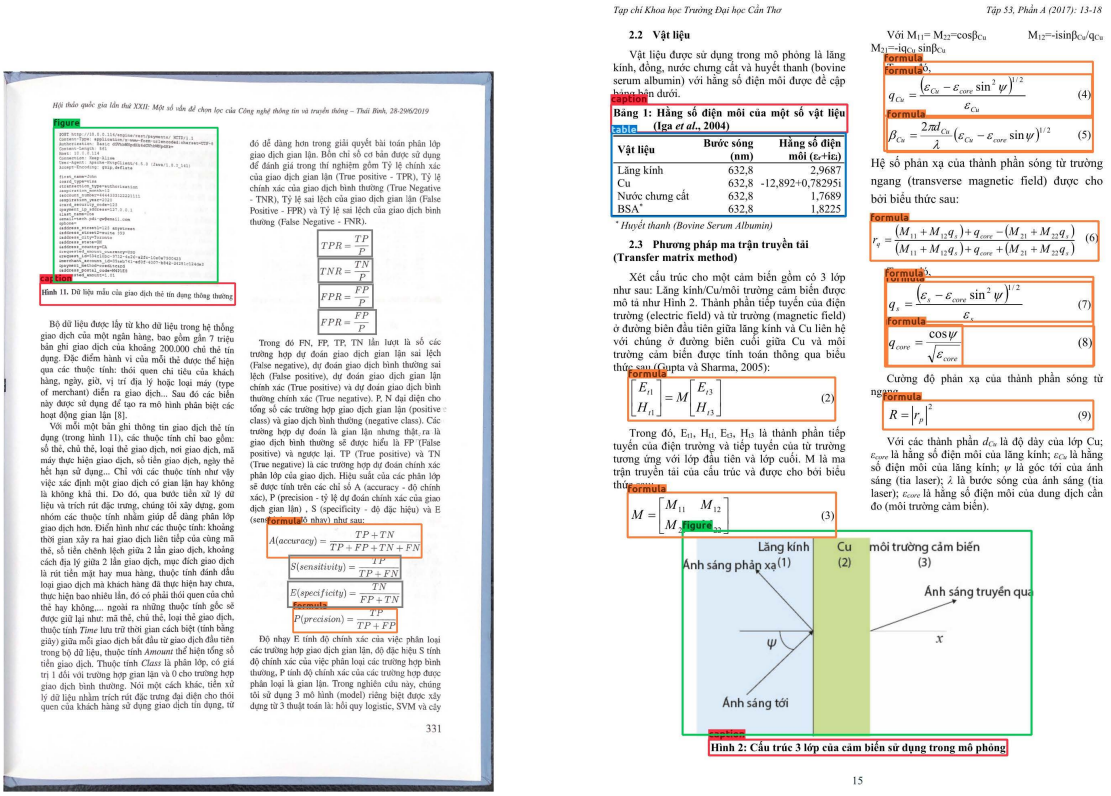
TABLE 3. Experimental results of different configurations of SABL (Faster RCNN).

RoI Pooling	$AP_{caption}$	AP_{figure}	AP_{table}	$AP_{formula}$	AP	AP@50	AP@75
RoI Align	66.1	**84.3**	**94.7**	49.0	73.5	88.7	79.6
PrRoI	**69.3**	83.9	94.4	**50.5**	**74.5**	**89.2**	**81.4**

obviously lower than the rest of the detectors. However, the prediction time in ATSS will be much faster, which is commonly seen in single-stage detectors.

The SABL (Cascade) method yields the highest AP results up to 77.2%. Among the four classes, the AP scores of Table and Figure are above 85%, Formula is 50.1%, and Caption is 76.2%. To improve the results on the dataset, it is necessary to focus on these two classes. The Formula class has a highly diverse expression, a big challenge for the UIT-DODV dataset. The SABL (Cascade) method achieves the highest mAP of 77.2% when using Cross-Entropy and 76.2% when using GHM Loss.

We continue to discuss the loss function in our experiment. In general, detectors maintain good performances in predicting tables and figures. However, as mentioned in IV-A, detectors have to struggle with the challenges of the UIT-DODV dataset, which leads to poor performance on caption and formula. Specifically, formula and caption are semantic classes that are a part of the text in the document. Therefore, the problem that must be overcome is the noise proposed region (i.e., a part of the text can be proposed to captions or formulas by the region proposal network). These problems relate to the classified background and foreground. We leverage focal loss to handle these problems; however, it is not as robust as expected to solve them. Fused loss is another solution that leverages both cross-entropy and focal loss to enhance classification precision. According to the experimental results, the fused loss gives better results than the cross-entropy loss in almost all methods (except the GRoIE and Double Head methods). This shows that applying fused loss is more effective than applying traditional cross-entropy loss. In addition, with different α values examined on the Double Head method, $\alpha = 0.6$ gives the highest result. Fused loss contributes to improving the AP on the caption


Hội thảo quốc gia lần thứ XXII: Một số vấn đề chọn lọc của Công nghệ thông tin và truyền thông – Thái Bình, 28-29/6/2019
figure
caption
Hình 11. Dữ liệu mẫu của giao dịch thẻ tín dụng thông thường
Bộ dữ liệu được lấy từ kho dữ liệu trong hệ thống giao dịch của một ngân hàng, bao gồm gần 7 triệu bản ghi giao dịch của khoảng 200.000 chủ thẻ tín dụng. Đặc điểm hành vi của mỗi thẻ được thể hiện qua các thuộc tính: thời quen chi tiêu của khách hàng, ngày, giờ, vị trí địa lý hoặc loại máy (type of merchant) diễn ra giao dịch... Sau đó các biến này được sử dụng để tạo ra mô hình phân biệt các hoạt động gian lận [8].
Với mỗi một bản ghi thông tin giao dịch thẻ tín dụng (trong hình 11), các thuộc tính chỉ bao gồm: số thẻ, chủ thẻ, loại thẻ giao dịch, nơi giao dịch, mã máy thực hiện giao dịch, số tiền giao dịch, ngày thẻ hết hạn sử dụng... Chỉ với các thuộc tính như vậy việc xác định một giao dịch có gian lận hay không là không khả thi. Do đó, qua bước tiền xử lý dữ liệu và trích rút đặc trưng, chúng tôi xây dựng, gom nhóm các thuộc tính nhằm giúp dễ dàng phân lớp giao dịch hơn. Điển hình như các thuộc tính: khoảng thời gian xảy ra hai giao dịch liên tiếp của cùng mã thẻ, số tiền chênh lệch giữa 2 lần giao dịch, khoảng cách địa lý giữa 2 lần giao dịch, mục đích giao dịch là rút tiền mặt hay mua hàng, thuộc tính đánh dấu loại giao dịch mà khách hàng đã thực hiện hay chưa, thực hiện bao nhiêu lần, đó có phải thói quen của chủ thẻ hay không,... ngoài ra những thuộc tính gốc sẽ được giữ lại như: mã thẻ, chủ thẻ, loại thẻ giao dịch, thuộc tính *Time* lưu trữ thời gian cách biệt (tính bằng giây) giữa mỗi giao dịch bắt đầu từ giao dịch đầu tiên trong bộ dữ liệu, thuộc tính *Amount* thể hiện tổng số tiền giao dịch. Thuộc tính *Class* là phân lớp, có giá trị 1 đối với trường hợp gian lận và 0 cho trường hợp giao dịch bình thường. Nói một cách khác, tiền xử lý dữ liệu nhằm trích rút đặc trưng đại diện cho thói quen của khách hàng sử dụng giao dịch tín dụng, từ đó dễ dàng hơn trong giải quyết bài toán phân lớp giao dịch gian lận. Bốn chỉ số cơ bản được sử dụng để đánh giá trong thí nghiệm gồm Tỷ lệ chính xác của giao dịch gian lận (True positive - TPR), Tỷ lệ chính xác của giao dịch bình thường (True Negative - TNR), Tỷ lệ sai lệch của giao dịch gian lận (False Positive - FPR) và Tỷ lệ sai lệch của giao dịch bình thường (False Negative - FNR).
formula
$TPR = \frac{TP}{P}$
formula
$TNR = \frac{TN}{P}$
formula
$FPR = \frac{FP}{P}$
formula
$FPR = \frac{FP}{P}$
Trong đó FN, FP, TP, TN lần lượt là số các trường hợp dự đoán giao dịch gian lận sai lệch (False negative), dự đoán giao dịch bình thường sai lệch (False positive), dự đoán giao dịch gian lận chính xác (True positive) và dự đoán giao dịch bình thường chính xác (True negative). P, N đại diện cho tổng số các trường hợp giao dịch gian lận (positive class) và giao dịch bình thường (negative class). Các trường hợp dự đoán là gian lận nhưng thật, ra là giao dịch bình thường sẽ được hiểu là FP (False positive) và ngược lại, TP (True positive) và TN (True negative) là các trường hợp dự đoán chính xác phân lớp của giao dịch. Hiệu suất của các phân lớp sẽ được tính trên các chỉ số A (accuracy - độ chính xác), P (precision - tỷ lệ dự đoán chính xác của giao dịch gian lận) , S (specificity - độ đặc hiệu) và E (sensitivity - độ nhạy) như sau:
formula
$A(accuracy) = \frac{TP+TN}{TP+FP+TN+FN}$
formula
$S(sensitivity) = \frac{TP}{TP+FN}$
formula
$E(specificity) = \frac{TN}{FP+TN}$
formula
$P(precision) = \frac{TP}{TP+FP}$
Độ nhạy E tính độ chính xác của việc phân loại các trường hợp giao dịch gian lận, độ đặc hiệu S tính độ chính xác của việc phân loại các trường hợp bình thường, P tính độ chính xác của các trường hợp được phân loại là gian lận. Trong nghiên cứu này, chúng tôi sử dụng 3 mô hình (model) riêng biệt được xây dựng từ 3 thuật toán là: hồi quy logistic, SVM và cây
331
Tạp chí Khoa học Trường Đại học Cần Thơ
Tập 53, Phần A (2017): 13-18
2.2 Vật liệu
Vật liệu được sử dụng trong mô phỏng là lăng kính, đồng, nước chưng cất và huyết thanh (bovine serum albumin) với hằng số điện môi được đề cập bảng bản dưới.
caption
Bảng 1: Hằng số điện môi của một số vật liệu (Iga *et al.*, 2004)
table
Vật liệu | Bước sóng (nm) | Hằng số điện môi (ε₁+iε₂)
Lăng kính | 632,8 | 2,9687
Cu | 632,8 | -12,892+0,78295i
Nước chưng cất | 632,8 | 1,7689
BSA* | 632,8 | 1,8225
* Huyết thanh (Bovine Serum Albumin)
2.3 Phương pháp ma trận truyền tải (Transfer matrix method)
Xét cấu trúc cho một cảm biến gồm có 3 lớp như sau: Lăng kính/Cu/môi trường cảm biến được mô tả như Hình 2. Thành phần tiếp tuyến của điện trường (electric field) và từ trường (magnetic field) ở đường biên đầu tiên giữa lăng kính và Cu liên hệ với chúng ở đường biên cuối giữa Cu và môi trường cảm biến được tính toán thông qua biểu thức sau (Gupta và Sharma, 2005):
formula
$\begin{bmatrix} E_{t1} \\ H_{t1} \end{bmatrix} = M \begin{bmatrix} E_{t3} \\ H_{t3} \end{bmatrix}$ (2)
Trong đó, E_{t1}, H_{t1}, E_{t3}, H_{t3} là thành phần tiếp tuyến của điện trường và tiếp tuyến của từ trường tương ứng với lớp đầu tiên và lớp cuối. M là ma trận truyền tải của cấu trúc và được cho bởi biểu thức
formula
$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$ (3)
figure
Lăng kính
Cu
môi trường cảm biến
(1)
(2)
(3)
Ánh sáng phản xạ
Ánh sáng truyền qua
ψ
x
Ánh sáng tới
caption
Hình 2: Cấu trúc 3 lớp của cảm biến sử dụng trong mô phỏng
15
Với $M_{11} = M_{22} = \cos\beta_{Cu}$ $M_{12} = -i\sin\beta_{Cu}/q_{Cu}$
$M_{21} = -iq_{Cu}\sin\beta_{Cu}$
formula
$q_{Cu} = \frac{\left(\varepsilon_{Cu} - \varepsilon_{core}\sin^2\psi\right)^{1/2}}{\varepsilon_{Cu}}$ (4)
formula
$\beta_{Cu} = \frac{2\pi d_{Cu}}{\lambda}\left(\varepsilon_{Cu} - \varepsilon_{core}\sin\psi\right)^{1/2}$ (5)
Hệ số phản xạ của thành phần sóng từ trường ngang (transverse magnetic field) được cho bởi biểu thức sau:
formula
$r_p = \frac{(M_{11}+M_{12}q_s)+q_{core}-(M_{21}+M_{22}q_s)}{(M_{11}+M_{12}q_s)+q_{core}+(M_{21}+M_{22}q_s)}$ (6)
formula
$q_s = \frac{\left(\varepsilon_s - \varepsilon_{core}\sin^2\psi\right)^{1/2}}{\varepsilon_s}$ (7)
formula
$q_{core} = \frac{\cos\psi}{\sqrt{\varepsilon_{core}}}$ (8)
Cường độ phản xạ của thành phần sóng từ ngang
formula
$R = |r_p|^2$ (9)
Với các thành phần d_{Cu} là độ dày của lớp Cu; ε_{core} là hằng số điện môi của lăng kính; ε_{Cu} là hằng số điện môi của lăng kính; ψ là góc tới của ánh sáng (tia laser); λ là bước sóng của ánh sáng (tia laser); ε_{core} là hằng số điện môi của dung dịch cần đo (môi trường cảm biến).

a) b)

FIGURE 9. **Visualization of some common failure cases of formula objects. Grey bounding boxes mark missing objects.**

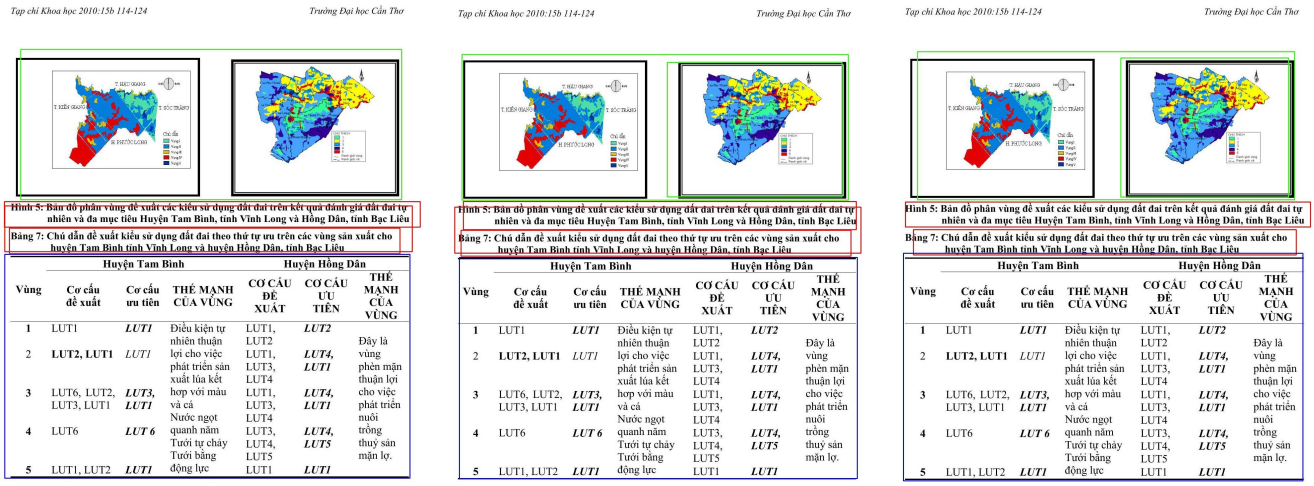
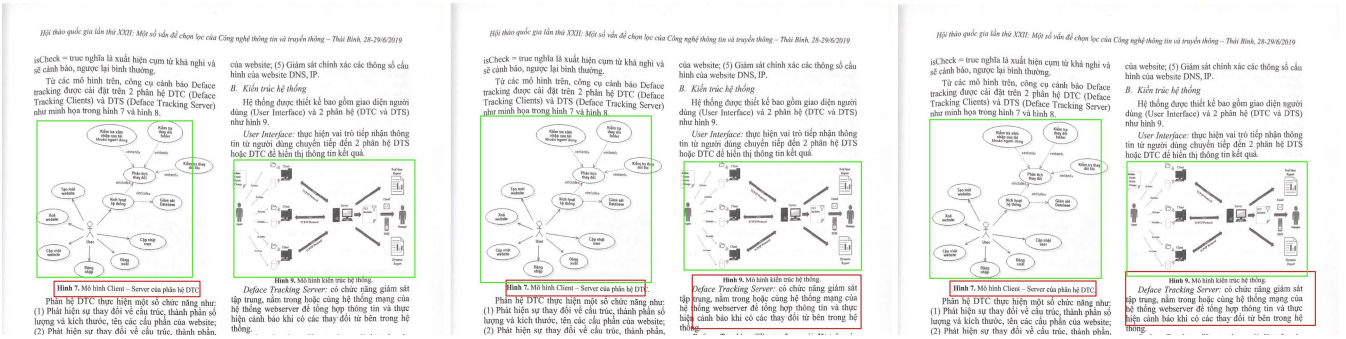


FIGURE 10. **Visualization results of different classification loss functions (best viewed in color with zoom). The results of the fused loss correctly localize and recognize the document objects.**

and formula while retaining the competitive results in the two existing classes. The AP on formula improves to nearly 1%, as predicted by Faster R-CNN, Generalized Attention, Libra R-CNN, and CARAFE. Moreover, GHM loss is also a good solution to classify page objects. GMM loss has competitive results in all classes, while comparing the detectors



DIoU GIoU CIoU

FIGURE 11. Visualization results of different regression loss functions.

has the best results. Libra R-CNN expresses its effectiveness with GHM loss of 64.7% at mAP. On the other hand, for regression problems. ATSS has better results than DIoU in three classes, caption, figure, and formula, compared with the default regression loss function (DIoU) in ATSS.

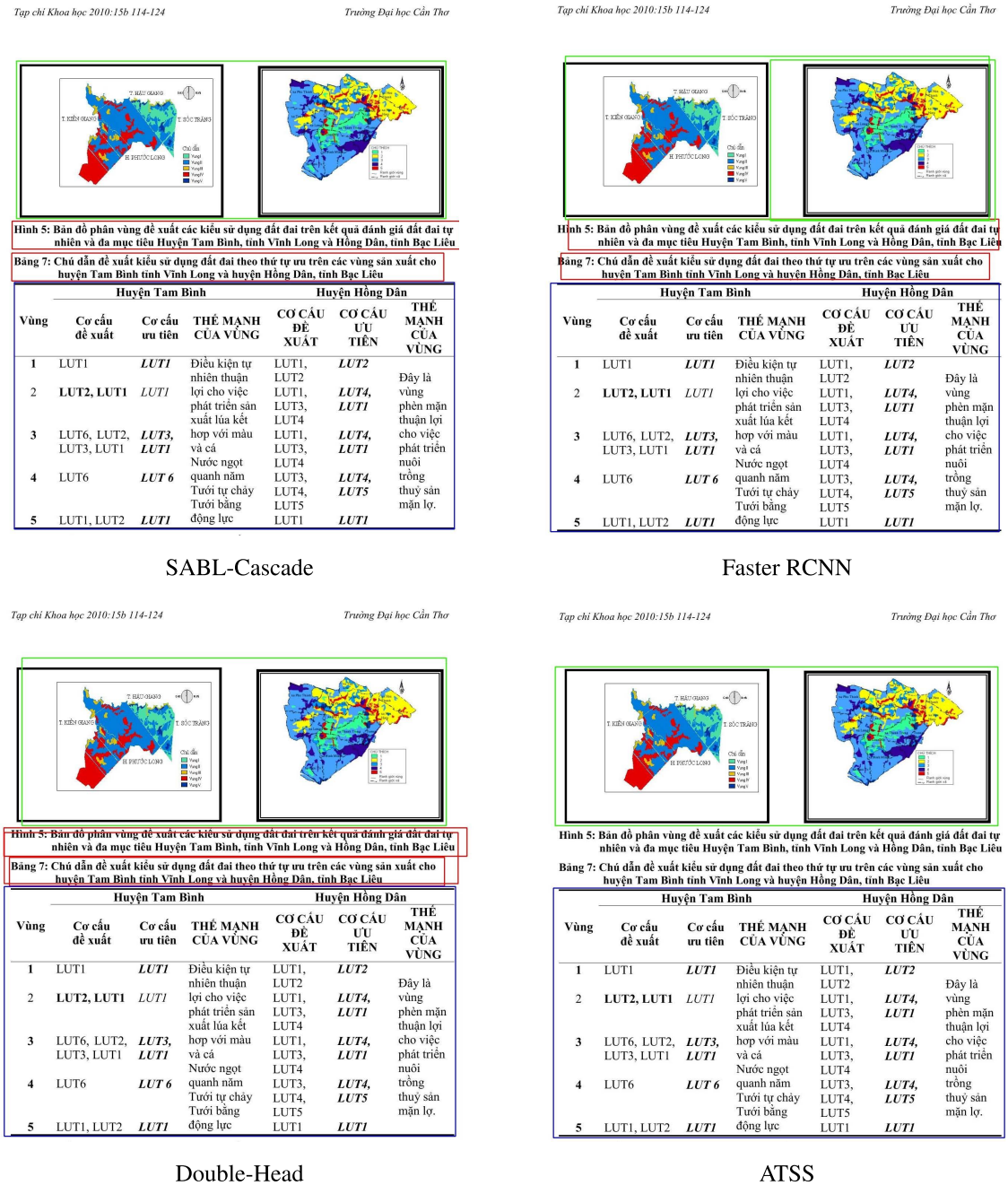
Since SABL Cascade achieves the best results, we further conduct experiments on SABL (Faster RCNN) with 2 different pooling methods: RoI Align and PrRoI. The evaluation results show that the application of PrRoI gives 1% higher results than RoIAlign. By using PrRoI Pooling, SABL (Faster RCNN) is slightly behind SABL (Cascade) with 74.5% AP. The results are reported in Table 3 and Figure 13. In this observation, using PrRoI as an RoI pooling module is more effective than RoI Align in the document images. Different from RoI Align, PrRoI uses full integration-based average pooling instead of sampling a constant number of points. Therefore, the localization task can be improved, helping the predicted boxes overlap more exactly with ground-truth boxes, leading to a better AP score.

Moreover, different loss functions have various impacts on detecting and classifying the model's results. We visualize the result in Figure 10 to show the impact of the classification loss functions. We observe that the fused loss function produces better localization performance than cross-entropy in some cases. Figure 11 illustrates the performance of different regression loss functions.

The trade-off between performance and complexity is explored in Figure 1. Note that we only use the highest AP among loss functions in each object detection model to illustrate their trade-off. It is not difficult to recognize that the larger the model is, the higher the detection performance becomes. SABL-Cascade outperforms the others because it is a multistage model whose proposal boxes are refined within three stages. Moreover, the regression task is improved by the Side-aware boundary localization module. However, this is also a problem that increases the complexity. Among the one-stage or anchor-free methods, AutoAssign seems to operate quite well. However, the average precision cannot compare to the results achieved by two-stage methods; it is an acceptable selection if real-time speed is required. Two-stage methods tend to cluster each other because they are all modified versions of the Faster R-CNN; the difference lies in FPN, external learned branches or other modules. Double-Head with two branches for regression and classification seems to outperform the others among the two-stage models.

We also explore the performance of the transfer learning technique, which is training the detector on an existing benchmark dataset for page object detection and then calculating the AP score on the testing set of the UIT-DODV dataset. The DocBank dataset is selected for this experiment because it includes all four classes of the UIT-DODV dataset (caption, figure, table, formula), and it also includes document images from research papers; therefore, the AP scores for predicted detections in the UIT-DODV dataset on these four classes can be calculated. However, we only take 10,000 samples and use the ground-truth bounding boxes of objects belonging to four classes, as in the UIT-DODV dataset, for training. These 10,000 samples are split into training and testing sets, with each including 5,000 samples; this is known as the DocBank10K. Faster R-CNN using cross-entropy loss is selected to conduct the transfer learning experiment. First, we train the Faster R-CNN model on the DocBank10K dataset, and we use the trained weights on this dataset to evaluate the detection performance on the testing set of the UIT-DODV dataset. Second, we use the trained weights on DocBank10K as pretrained weights to continue to train on the training set of the UIT-DODV dataset, and then we evaluate again on the testing set of UIT-DODV. The results are reported in Table 4.

We note that training on DocBank10K and evaluating on UIT-DODV do not achieve the expected results. The AP scores are much lower than those of the model directly trained on the UIT-DODV training set. Moreover, fine-tuning the Faster R-CNN model on the UIT-DODV training set from the pretrained weights obtained by training on the DocBank10K dataset also does not perform well. The AP score is lower than that of the Faster R-CNN model reported in Table 2 (-12.5% AP). These observations prove that training from the existing POD document dataset in English and fine-tuning on the UIT-DODV are ineffective. The main reason may be the



SABL-Cascade

Faster RCNN

Double-Head

ATSS

FIGURE 12. Visualization results of different object detection methods: SABL Cascade, Faster R-CNN, DoubleHead and ATSS.

TABLE 4. Experimental results of transfer learning technique. DocBank10K → UIT-DODV means training on DocBank10K and evaluating on UIT-DODV.

Method	DocBank10K → UIT-DODV	DocBank10K pre-trained weights	$AP_{caption}$	AP_{figure}	AP_{table}	$AP_{formula}$	AP	AP@50	AP@75
Faster R-CNN	✓		1.1	41.4	35.2	0.3	19.5	34.4	20.3
		✓	48.2	70.3	80.8	31.4	57.7	80.9	64.5

differences in the script, table and formula styles between the two datasets. Regarding the script characteristic, DocBank includes document images edited in English, only Latin characters. At the same time, UIT-DODV contains Vietnamese document images, which additionally use UTF-8 characters to present various accents. This is why the performance of objects belonging to the 'caption' class is abysmal (1.1%). When trained on the DocBank10K dataset, the detector only

mô nuôi (Bảng 1). Thông qua các nghiên cứu trước đây cho thấy khu vực này được phân chia hành hai dạng nền đáy chính tùy thuộc hàm lượng chất hữu cơ xác định được: (1) Khu vực giàu dinh dưỡng (N >12 mg/kg; P ~ 0,2 mg/kg) và (2) Khu vực nghèo dinh dưỡng (N ~ 7,8 mg/kg; P ~ 2,48 mg/kg) (Tất Anh Thư, 2003 và Châu Minh Khôi, 2006).

Hình 1: Hệ thống thí nghiệm ở Vĩnh châu (trái); Cá kèo (*Pseudapocryptes lanceolatus* (trên phải) và cua biển (*Scylla paramamosain* (dưới, phải)

Kích thước ao nuôi: ao nuôi có hình chữ nhật với diện tích khoảng 800m^2 (20 m rộng x 40 m dài) cho cả hai khu vực giàu và nghèo dinh dưỡng.

Bảng 1: Bố trí thí nghiệm theo mô hình nuôi ở các dạng đáy ao khác nhau

Ao	Nền đáy[1]	N,Pdễ tiêu[1] (mg/kg)	Mùa mưa	Mùa khô
T1 (T11, T12)	Giàu dinh dưỡng	N:10,30 P:1,72	Cá kèo-Cua biển (BTC[2])	Artemia
T2 (T21, T22)	Giàu dinh dưỡng	N:13,80 P:0,20	Cá kèo-Cua biển (QCCT[3])	Artemia
T3 (T31, T32)	Nghèo dinh dưỡng	N:1,20 P:0,77	Cá kèo-Cua biển (BTC)	Artemia
T4 (T41, T42)	Nghèo dinh dưỡng	N:7,80 P:2,48	Cá kèo-Cua biển (QCCT)	Artemia
ĐC (T9, T10)	Nghèo dinh dưỡng	N:4,70 P:1,21	Không nuôi	Artemia

Ghi chú: [1]Theo Tất Anh Thư et al. (2006); Chau Minh Khoi et al. (2006); [2]Bán thâm canh (BTC): (30 cá kèo + 0,5 cua biển)/m^2; [3]Quảng canh cải tiến (QCCT): (15 cá kèo + 0,5 cua biển)/m^2

mô nuôi (Bảng 1). Thông qua các nghiên cứu trước đây cho thấy khu vực này được phân chia hành hai dạng nền đáy chính tùy thuộc hàm lượng chất hữu cơ xác định được: (1) Khu vực giàu dinh dưỡng (N >12 mg/kg; P ~ 0,2 mg/kg) và (2) Khu vực nghèo dinh dưỡng (N ~ 7,8 mg/kg; P ~ 2,48 mg/kg) (Tất Anh Thư, 2003 và Châu Minh Khôi, 2006).

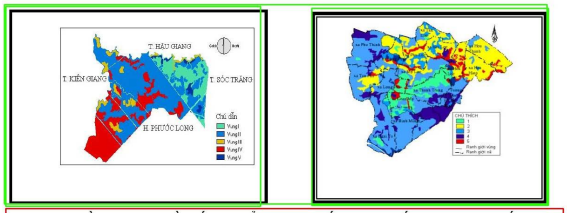
Hình 1: Hệ thống thí nghiệm ở Vĩnh châu (trái); Cá kèo (*Pseudapocryptes lanceolatus* (trên phải) và cua biển (*Scylla paramamosain* (dưới, phải)

Kích thước ao nuôi: ao nuôi có hình chữ nhật với diện tích khoảng 800m^2 (20 m rộng x 40 m dài) cho cả hai khu vực giàu và nghèo dinh dưỡng.

Bảng 1: Bố trí thí nghiệm theo mô hình nuôi ở các dạng đáy ao khác nhau

Ao	Nền đáy[1]	N,Pdễ tiêu[1] (mg/kg)	Mùa mưa	Mùa khô
T1 (T11, T12)	Giàu dinh dưỡng	N:10,30 P:1,72	Cá kèo-Cua biển (BTC[2])	Artemia
T2 (T21, T22)	Giàu dinh dưỡng	N:13,80 P:0,20	Cá kèo-Cua biển (QCCT[3])	Artemia
T3 (T31, T32)	Nghèo dinh dưỡng	N:1,20 P:0,77	Cá kèo-Cua biển (BTC)	Artemia
T4 (T41, T42)	Nghèo dinh dưỡng	N:7,80 P:2,48	Cá kèo-Cua biển (QCCT)	Artemia
ĐC (T9, T10)	Nghèo dinh dưỡng	N:4,70 P:1,21	Không nuôi	Artemia

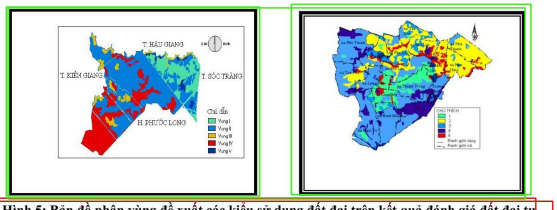
Ghi chú: [1]Theo Tất Anh Thư et al. (2006); Chau Minh Khoi et al. (2006); [2]Bán thâm canh (BTC): (30 cá kèo + 0,5 cua biển)/m^2; [3]Quảng canh cải tiến (QCCT): (15 cá kèo + 0,5 cua biển)/m^2



Hình 5: Bản đồ phân vùng đề xuất các kiểu sử dụng đất đai trên kết quả đánh giá đất đai tự nhiên và đa mục tiêu Huyện Tam Bình, tỉnh Vĩnh Long và Hồng Dân, tỉnh Bạc Liêu

Bảng 7: Chú dẫn đề xuất kiểu sử dụng đất đai theo thứ tự ưu trên các vùng sản xuất cho huyện Tam Bình tỉnh Vĩnh Long và huyện Hồng Dân, tỉnh Bạc Liêu

Vùng	Huyện Tam Bình			Huyện Hồng Dân		
	Cơ cấu đề xuất	Cơ cấu ưu tiên	THẾ MẠNH CỦA VÙNG	CƠ CẤU ĐỀ XUẤT	CƠ CẤU ƯU TIÊN	THẾ MẠNH CỦA VÙNG
1	LUT1	***LUT1***	Điều kiện tự nhiên thuận lợi cho việc phát triển sản xuất lúa kết hợp với màu và cá Nước ngọt quanh năm Tưới tự chảy Tưới bằng động lực	LUT1, LUT2	***LUT2***	Đây là vùng phèn mặn thuận lợi cho việc phát triển nuôi trồng thuỷ sản mặn lợ.
2	**LUT2, LUT1**	*LUT1*		LUT1, LUT3, LUT4	***LUT4, LUT1***	
3	LUT6, LUT2, LUT3, LUT1	***LUT3, LUT1***		LUT1, LUT3, LUT4	***LUT4, LUT1***	
4	LUT6	***LUT 6***		LUT3, LUT4, LUT5	***LUT4, LUT5***	
5	LUT1, LUT2	***LUT1***		LUT1	***LUT1***	



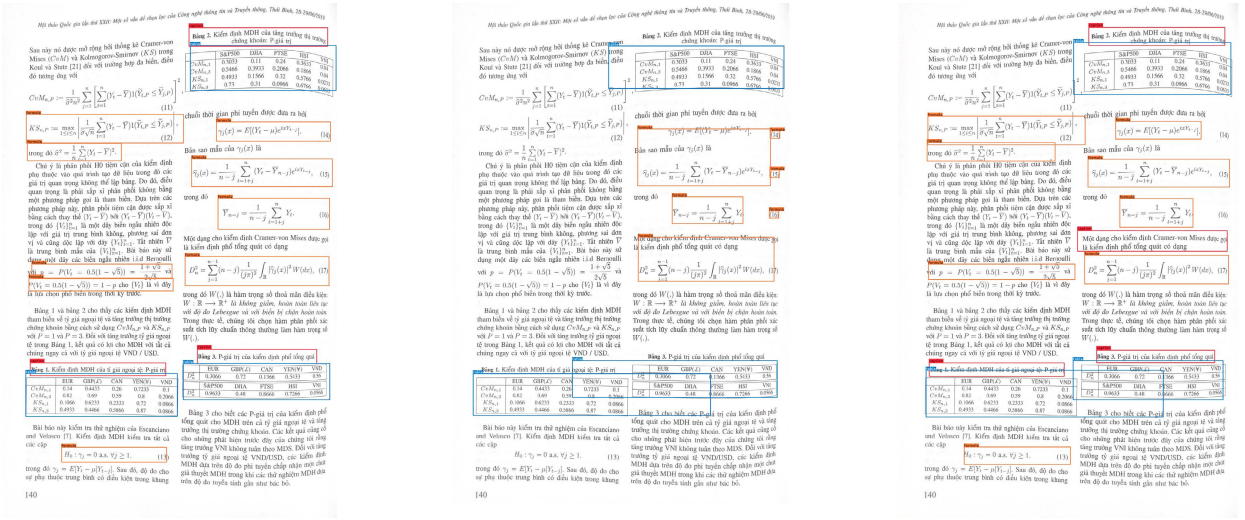
Hình 5: Bản đồ phân vùng đề xuất các kiểu sử dụng đất đai trên kết quả đánh giá đất đai tự nhiên và đa mục tiêu Huyện Tam Bình, tỉnh Vĩnh Long và Hồng Dân, tỉnh Bạc Liêu

Bảng 7: Chú dẫn đề xuất kiểu sử dụng đất đai theo thứ tự ưu trên các vùng sản xuất cho huyện Tam Bình tỉnh Vĩnh Long và huyện Hồng Dân, tỉnh Bạc Liêu

Vùng	Huyện Tam Bình			Huyện Hồng Dân		
	Cơ cấu đề xuất	Cơ cấu ưu tiên	THẾ MẠNH CỦA VÙNG	CƠ CẤU ĐỀ XUẤT	CƠ CẤU ƯU TIÊN	THẾ MẠNH CỦA VÙNG
1	LUT1	***LUT1***	Điều kiện tự nhiên thuận lợi cho việc phát triển sản xuất lúa kết hợp với màu và cá Nước ngọt quanh năm Tưới tự chảy Tưới bằng động lực	LUT1, LUT2	***LUT2***	Đây là vùng phèn mặn thuận lợi cho việc phát triển nuôi trồng thuỷ sản mặn lợ.
2	**LUT2, LUT1**	*LUT1*		LUT1, LUT3, LUT4	***LUT4, LUT1***	
3	LUT6, LUT2, LUT3, LUT1	***LUT3, LUT1***		LUT1, LUT3, LUT4	***LUT4, LUT1***	
4	LUT6	***LUT 6***		LUT3, LUT4, LUT5	***LUT4, LUT5***	
5	LUT1, LUT2	***LUT1***		LUT1	***LUT1***	

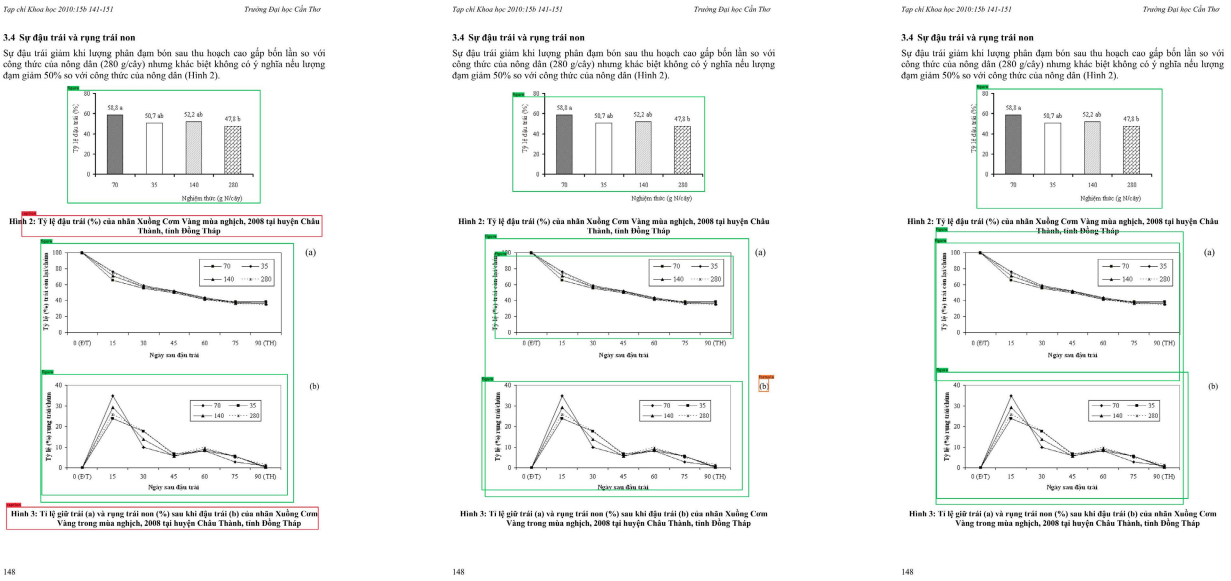
FIGURE 13. **Visualization results of PrRoI (top row) and RoiAlign (bottom row) in Table 3.**

sees Latin characters; it cannot observe samples containing UTF-8 characters used in the Vietnamese language, creating confusion for the pretrained model. In addition, there are various formula writing styles in English papers, such as the position of indexing numbers and indexing style. We observe that the indexing numbers can be put at the formula's right and left. The indexing style is also slightly different: they can be numbers only or include numbers and one character. Meanwhile, Vietnamese documents usually use only numbers to denote formulas and put them on the right. These differences cause poor performance in formula objects' class when recorded at only 0.3%. Regarding document style, English research papers commonly contain non-bordered tables, while Vietnamese researchers habitually use bordered tables. This aspect leads to the low AP_{table} score (35.2%). We provide some qualitative results in Figure 14 and Figure 15, which compare the performance between three versions of Faster R-CNN: directly trained on UIT-DODV (reported in Table 2), trained on the DocBank10K dataset, and fine-tuned on UIT-DODV using pretrained weights from a model trained on DocBank10K. Based on visualizations, directly using pretrained weights on the DocBank10K dataset cannot detect captions' objects (Figure 15b). Even if we fine-tuned the UIT-DODV dataset, the detector still predicted some false-positive captioning objects (Figure 14c). Besides, the detection performance on equations' objects seems of



a) b) c)

FIGURE 14. Visualization results (on scanned images) with and without transfer learning on the DocBank10K dataset. a) Faster R-CNN trained on the UIT-DODV training set; b) Faster R-CNN trained on DocBank10K; c) Faster R-CNN fine-tuned on the UIT-DODV dataset using pretrained weights on DocBank10K.



a) b) c)

FIGURE 15. Visualization results (on PDF images) with and without transfer learning on the DocBank10K dataset. a) Faster R-CNN trained on the UIT-DODV training set; b) Faster R-CNN trained on DocBank10K; c) Faster R-CNN fine-tuned on the UIT-DODV dataset using pretrained weights on DocBank10K.

utmost bad. While Faster R-CNN trained on the UIT-DODV training set predicts almost correctly (Figure 14a), its counterpart trained on the DocBank10K dataset not only misses objects but also regresses bounding boxes inaccurately (Figure 14b). Even after fine-tuning the UIT-DODV training set, the detector still misses one formula object (Figure 14c). However, Faster R-CNN trained on DocBank10K shows quite good performance on table objects (Figure 14b). Nevertheless, the predicted bounding boxes occupy many redundant parts, leading to a low AP_{table} score. The same problem also appears on Faster R-CNN fine-tuned on the UIT-DODV training set (Figure 14c).

E. ERROR ANALYSIS

To address the current problems that state-of-the-art detectors face in the page object detection problem, we visualize the

TABLE 5. List of abbreviations which are used in the paper.

Abbreviation	Explanation
DIU	Document Image Understanding
POD	Page Object Detection
ICDAR	International Conference on Document Analysis and Recognition
SSD	Single Shot Detection
YOLO	You Only Look Once
GTE	Global Table Extractor
CDeCNet	Composite Deformable Cascade Network
R-CNN	Regional-based Convolutional Neural Network
ROI	Region of Interests
FPN	Feature Pyramid Network
RPN	Regional Proposal Network
ATSS	Adaptive Training Sample Selection
FCOS	Fully Convolutional One-stage
GRoIE	Generic Region of Interests Extractor
SABL	Side-Aware Boundary Localization
CARAFE	Content-aware Reassembly of Features
IoU	Intersection Over Union
mAP	Mean Average Precision
AP	Average Precision
GHM	Gradient Harmonized Mechanism
PrRoI	Precise RoI Pooling

detection results from the best performing model (SABL-Cascade) and find some cases in which the detector commonly fails. Since the SABL-Cascade performs well in detecting table objects 95.9% AP), we focus only on the performance in detecting other types of objects. After visualization of all images in the testing set, we observe some problems.

1) CAPTION

The detector mainly ignores captions if they appear on the left or right of tables or figures. In Figure 7a, Captions are much longer sentences and lie to the right of figure objects; in this case, the detector cannot detect the bounding boxes of the captions. The reason may be explained as these samples are outliers because there are not too many samples whose object captions lie beside object figures. On the other hand, in scanned images, the captions may be wavy. In these cases, caption objects are easily ignored (Figure 7b).

2) FIGURE

Predicted bounding boxes of figure objects are commonly overlapped in the cases in which subimages appear in a figure or many figures are placed beside each other (Figure 8a). At the same time, abnormal figure objects are easily ignored. In Figure 8b, the figure object contains a long string, which makes it extremely difficult for the model to recognize whether this is a figure, table, or caption.

3) FORMULA

Formula objects may include the formula numbers. In our opinion, this is the pattern that helps the model recognize whether the object is a formula or not. Some formulas that do not contain the numbers may become hard objects, which are commonly not detected by the detector (Figure 9a). However, formula objects are the same as figure objects; if these objects are located too close to each other, the model also predicts some redundant boxes (Figure 9b).

V. CONCLUSION AND FUTURE WORK

In this paper, we conduct comprehensive assessments of the UIT-DODV dataset with ten state-of-the-art object detection methods for Vietnamese document analysis. We train from two to three different loss functions for each of these methods in the proposal boxes classification task. In addition to cross-entropy loss, focal loss, and GHMC loss, we conduct experiments with fused loss (a combination of cross-entropy and focal loss). We assess not only the impact of different loss functions but also the impact of two different RoI pooling. In particular, we replace the default RoI Align with PrRoI to further improve the performance.

In the future, we will diversify the UIT-DODV dataset by collecting more images from lectures, textbooks, and receipts. In addition, we aim to address more problems in the document understanding problem, such as recognizing captions below figure or table objects and visual question answering based on text contents in document images.

ABBREVIATIONS

The following abbreviations are used in this manuscript: See Table 5.

REFERENCES

[1] V. Alcácer and V. Cruz-Machado, "Scanning the industry 4.0: A literature review on technologies for manufacturing systems," *Eng. Sci. Technol., Int. J.*, vol. 22, no. 3, pp. 899–919, Jun. 2019.

[2] T. A. Tran, H. T. Tran, I. S. Na, G. S. Lee, H. J. Yang, and S. H. Kim, "A mixture model using random rotation bounding box to detect table region in document image," *J. Vis. Commun. Image Represent.*, vol. 39, pp. 196–208, Aug. 2016.

[3] S. Bakkali, Z. Ming, M. Coustaty, and M. Rusinol, "Visual and textual deep feature fusion for document image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 562–563.

[4] K. Laven, S. Leishman, and S. Roweis, "A statistical learning approach to document image analysis," in *Proc. 8th Int. Conf. Document Anal. Recognit. (ICDAR)*, 2005, pp. 357–361.

[5] M. Bibi, A. Hamid, M. Moetesum, and I. Siddiqi, "Document forgery detection using source printer identification: A comparative study of text-dependent versus text-independent analysis," *Exp. Syst.*, vol. 39, no. 8, Sep. 2022, doi: 10.1111/exsy.13020.

[6] C.-Y. Shiah, "Content-based document image retrieval based on document modeling," *J. Intell. Inf. Syst.*, vol. 55, no. 2, pp. 287–306, Oct. 2020.

[7] A. Mishra, S. Shekhar, A. K. Singh, and A. Chakraborty, "OCR-VQA: Visual question answering by reading text in images," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 947–952.

[8] L. T. Dieu, T. T. Nguyen, N. D. Vo, T. V. Nguyen, and K. Nguyen, "Parsing digitized Vietnamese paper documents," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, 2021, pp. 382–392.

[9] B. Zhu, J. Wang, Z. Jiang, F. Zong, S. Liu, Z. Li, and J. Sun, "AutoAssign: Differentiable label assignment for dense object detection," 2020, *arXiv:2007.03496*.

[10] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," 2019, *arXiv:1912.02424*.

[11] Y. Wu, Y. Chen, L. Yuan, Z. Liu, L. Wang, H. Li, and Y. Fu, "Rethinking classification and localization for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* 2020, pp. 10186–10195.

[12] L. Rossi, A. Karimi, and A. Prati, "A novel region of interest extraction layer for instance segmentation," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 2203–2209.

[13] J. Wang, W. Zhang, Y. Cao, K. Chen, J. Pang, T. Gong, J. Shi, C. C. Loy, and D. Lin, "Side-aware boundary localization for more precise object detection," in *Proc. ECCV*, 2020, pp. 403–419.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[15] X. Zhu, D. Cheng, Z. Zhang, S. Lin, and J. Dai, "An empirical study of spatial attention mechanisms in deep networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6688–6697.

[16] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 821–830.

[17] S. Qiao, H. Wang, C. Liu, W. Shen, and A. Yuille, "Micro-batch training with batch-channel normalization and weight standardization," 2019, *arXiv:1903.10520*.

[18] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, "CARAFE: Content-aware ReAssembly of FEatures," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3007–3016.

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[20] F. Cesarini, S. Marinai, L. Sarti, and G. Soda, "Trainable table location in document images," in *Proc. Object Recognit. Supported User Interact. Service Robots*, vol. 3, 2002, pp. 236–240.

[21] J. Fang, X. Tao, Z. Tang, R. Qiu, and Y. Liu, "Dataset, ground-truth and performance metrics for table detection evaluation," in *Proc. 10th IAPR Int. Workshop Document Anal. Syst.*, Mar. 2012, pp. 445–449.

[22] L. Gao, X. Yi, Z. Jiang, L. Hao, and Z. Tang, "ICDAR 2017 competition on page object detection," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 1417–1422.

[23] M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, and Z. Li, "TableBank: A benchmark dataset for table detection and recognition," 2019, *arXiv:1903.01949*.

[24] X. Zhong, J. Tang, and A. J. Yepes, "PubLayNet: Largest dataset ever for document layout analysis," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1015–1022.

[25] L. Gao, Y. Huang, H. Dejean, J.-L. Meunier, Q. Yan, Y. Fang, F. Kleber, and E. Lang, "ICDAR 2019 competition on table detection and recognition (cTDaR)," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1510–1515.

[26] M. Li, Y. Xu, L. Cui, S. Huang, F. Wei, Z. Li, and M. Zhou, "DocBank: A benchmark dataset for document layout analysis," 2020, *arXiv:2006.01038*.

[27] M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, and Z. Li, "TableBank: Table benchmark for image-based table detection and recognition," in *Proc. 12th Lang. Resour. Eval. Conf.*, 2020, pp. 1918–1925.

[28] M. Kerwat, R. George, and K. Shujaee, "Detecting knowledge artifacts in scientific document Images—Comparing deep learning architectures," in *Proc. 5th Int. Conf. Social Netw. Anal., Manag. Secur. (SNAMS)*, Oct. 2018, pp. 147–152.

[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[30] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[31] Y. Huang, Q. Yan, Y. Li, Y. Chen, X. Wang, L. Gao, and Z. Tang, "A YOLO-based table detection method," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 813–818.

[32] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 91–99.

[33] N. Sun, Y. Zhu, and X. Hu, "Faster R-CNN based table detection combining corner locating," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1314–1319.

[34] S. A. Siddiqui, I. A. Fateh, S. T. R. Rizvi, A. Dengel, and S. Ahmed, "DeepTabStR: Deep learning based table structure recognition," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1403–1409.

[35] X. Zhong, E. ShafieiBavani, and A. J. Yepes, "Image-based table recognition: Data, model, and evaluation," in *Proc. ECCV*, 2020, pp. 564–580.

[36] X. Zheng, D. Burdick, L. Popa, X. Zhong, and N. X. R. Wang, "Global table extractor (GTE): A framework for joint table identification and cell structure recognition using visual context," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 697–706.

[37] M. Agarwal, A. Mondal, and C. V. Jawahar, "CDeC-Net: Composite deformable cascade network for table detection in document images," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 9491–9498.

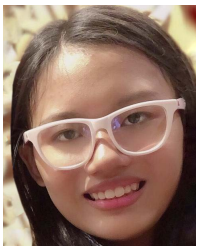
[38] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[39] Y. Liu, "CBNet: A novel composite backbone network architecture for object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11653–11660.

[40] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9627–9636.

[41] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, and Z. Zhang, "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*.

KHANG NGUYEN received the B.S. and M.S. degrees in computer science and the Ph.D. degree from the University of Science, VNUHCM, Vietnam, in 1990, 1995, and 2012, respectively. Currently, he is the Vice-President of the University of Information Technology-VNUHCM. His research interests include artificial intelligence and computer vision.



AN NGUYEN is currently a Research Student with the Faculty of Computer Science, University of Information Technology-VNUHCM. Her research interests include computer vision and machine learning.

NGUYEN D. VO received the B.Sc. and M.Sc. degrees in computer science from the University of Science, VNUHCM, in 2013 and 2018, respectively. Since 2017, he has been with the University of Information Technology-VNUHCM. His research interests include computer vision and deep learning.

TAM V. NGUYEN (Senior Member, IEEE) received the Ph.D. degree from the National University of Singapore, in 2013. He was a Research Scientist and a Principal Investigator at the ARTIC Research Centre, Singapore Polytechnic. He was also an Adjunct Lecturer at the National University of Singapore. He is currently an Associate Professor with the Department of Computer Science, University of Dayton. His research interests include computer vision, applied deep learning, multimedia content analysis, and mixed reality.