Important declarations

Please remove this info from manuscript text if it is also present there.

Associated Data

Data supplied by the author:

SROIE Dataset: Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., & Jawahar, C. V. (2019, September). Icdar2019 competition on scanned receipt ocr and information extraction. In 2019 International Conference on Document Analysis and Recognition (ICDAR) (pp. 1516-1520). IEEE. https://doi.org/10.1109/ICDAR.2019.00244 CORD Dataset: Park, S., Shin, S., Lee, B., Lee, J., Surh, J., Seo, M., & Lee, H. 2019. CORD: a consolidated receipt dataset for post-OCR parsing. In Workshop on Document Intelligence at NeurIPS 2019 FUNSD Dataset: Jaume, G., Ekenel, H. K., & Thiran, J. P. (2019, September). Funsd: A dataset for form understanding in noisy scanned documents. In 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW) (Vol. 2, pp. 1-6). IEEE. https://doi.org/10.48550/arXiv.1905.13538 Raw data have also been uploaded to the supplemental files.

Required Statements

Competing Interest statement:

The authors declare that they have no competing interests.

Funding statement:

The authors received no funding for this work.

ConBGAT: A novel model combining CNN, BERT and graph attention network for information extraction from scanned image

Hung Thanh Bui Corresp., [1] , **Duy Ho Vo Hoang** [1] , **Huy Vo Quoc** [1]

[1] Data Science Laboratory/Data Science Department/Faculty of Information Technology, Industrial University of Ho Chi Minh city, Ho Chi Minh, VIETNAM, Vietnam

Corresponding Author: Hung Thanh Bui
Email address: buithanhhung@iuh.edu.vn

Extracting information from scanned images is a crucial task with numerous practical applications. Prior methods have often underutilized image and text features, resulting in suboptimal accuracy and efficiency. In this study, we introduce a novel model called ConBGAT, which integrates Convolutional Neural Networks (CNNs), Transformers, and Graph Attention Networks. Our approach involves constructing graphs from regions within the image text, utilizing Optical Character Recognition techniques to detect characters in images, and integrating CNNs and DistilBERT models to extract image and text features efficiently. Our experiments involve evaluating the versatility of our proposed ConBGAT model on the SROIE, FUNSD and CORD datasets. We then compare its performance against other existing methods. Furthermore, we optimize our algorithms to identify the most effective one for our proposed model. The experimental results demonstrate that our ConBGAT model outperforms the other compared models, showcasing its superior performance across various evaluation metrics.

ConBGAT: A novel model combining CNN, BERT and graph attention network for information extraction from scanned image

Bui Thanh Hung[1], Ho Vo Hoang Duy[1] , Vo Quoc Huy[1]
[1] Data Science Laboratory, Data Science Department, Faculty of Information Technology, Industrial University of Ho Chi Minh city, Ho Chi Minh city, Vietnam.

Corresponding Author: Bui Thanh Hung
12 Nguyen Van Bao, 4 ward, Go Vap district, Ho Chi Minh city, 700000, Vietnam
Email address: buithanhhung@iuh.edu.vn

Abstract
Extracting information from scanned images is a crucial task with numerous practical applications. Prior methods have often underutilized image and text features, resulting in suboptimal accuracy and efficiency. In this study, we introduce a novel model called ConBGAT, which integrates Convolutional Neural Networks (CNNs), Transformers, and Graph Attention Networks. Our approach involves constructing graphs from regions within the image text, utilizing Optical Character Recognition techniques to detect characters in images, and integrating CNNs and DistilBERT models to extract image and text features efficiently. Our experiments involve evaluating the versatility of our proposed ConBGAT model on the SROIE, FUNSD and CORD datasets. We then compare its performance against other existing methods. Furthermore, we optimize our algorithms to identify the most effective one for our proposed model. The experimental results demonstrate that our ConBGAT model outperforms the other compared models, showcasing its superior performance across various evaluation metrics.

Subjects Artificial Intelligence, Computer Vision, Data Science, Natural Language and Speech.
Keywords Information Extraction, CNN, Bert, GAT, Deep Learning, Scanned Image.

INTRODUCTION
The task of identifying information within scanned images poses a significant challenge with profound implications for both natural language processing and computer vision fields. In today's digital era, the ability to extract information from scanned images has become indispensable across various contexts. Deep learning methodologies have emerged as effective solutions to address this challenge. By leveraging deep learning models *(Nikhat Parveen et al., 2024*; *Bui Thanh Hung et*

al., 2024; Bui Thanh Hung et al., 2023), we can train systems to autonomously recognize and extract pertinent information from scanned images. These models acquire an understanding of natural language structures and image features, thereby enhancing the accuracy and efficiency of information identification. This automation not only streamlines processes but also mitigates the time and effort expended compared to manual approaches. Moreover, beyond enhancing efficiency and accuracy in scanned image processing, this capability unlocks diverse applications and research avenues, spanning from office automation to comprehensive management of textual information across image datasets, all while upholding standards of transparency and accuracy.

During the information extraction process, various obstacles may impede progress. Managing and analyzing vast volumes of unstructured data presents a formidable challenge in information extraction endeavors. To effectively derive meaningful insights from such datasets, robust techniques and scalable algorithms are imperative. However, the crux of the information extraction challenge lies in selecting and applying appropriate methodologies tailored to our specific data requirements. Different information extraction methods, including keyword extraction, sentiment analysis, text summarization, and question answering, serve distinct objectives, each governed by unique principles and outcomes. Thus, selecting the most suitable method hinges on aligning with our analytical objectives and understanding the inherent characteristics of our data. Furthermore, each method may employ diverse techniques, ranging from rule-based to statistical or machine learning approaches, each with its inherent strengths and limitations. Therefore, discerning the most suitable extraction technique for a given task is paramount. Evaluating and comparing these techniques to ascertain their efficacy and reliability presents a formidable challenge.

In this study, we propose an effective model to identify scanned image information. Our contributions in this research include the following:

- We propose a new model ConBGAT for the problem of extracting scanned image information.
- The ConBGAT model has features extracted from combining advanced models CNN in image feature extraction and Transformer-DistilBERT in text feature extraction.
- We utilize graph modeling techniques to depict the interrelations among regions within text images, encompassing attributes like position, size, and the interconnections between various objects.
- Employing the Graph Attention Network (GAT) model to glean insights from the graph structure, enabling the model to comprehend the intricate relationships among components.
- Conducting a comparative evaluation of performance across different Graph Neural Network (GNN) models.
- Optimizing our approach by employing optimal algorithms tailored for deep learning tasks, selecting the most effective algorithm to enhance the performance of the ConBGAT model.
- Performing comprehensive experiments and multidimensional evaluations on the SROIE, FUNSD and CORD datasets, juxtaposing the performance of our proposed model, ConBGAT, against other existing methods.

Beyond the introduction, the subsequent sections of the paper will be structured as follows: Session 2 delves into existing literature in the field, Session 3 outlines the architecture and methodology of our approach, Session 4 details the experimental setup, conducted experiments, and the evaluation of results, finally, in Section 5 presents our conclusions drawn from the findings and discuss potential avenues for future research.

MATERIALS AND METHODS

Identifying information from scanned images requires combining multiple image processing and natural language processing techniques. Our research is inspired by recent research on Graph Neural Networks and Information Extraction.

Graph Neural Network

Graph Neural Networks (GNNs) are a specialized class of machine learning models tailored for processing graph-structured data *(Zhou et al., 2020*; *Goyal et al., 2018*; *Allamanis et al., 2017*; *Kipf et al., 2017*; *Wu et al., 2020)*. These algorithms excel in learning representations of nodes, edges, and entire graphs. GNNs offer scalability and versatility, making them well-suited for analyzing complex structures found in various domains such as social networks, transportation networks, and systems with interconnected objects.

GNN methods typically fall into two primary categories based on their architectural principles: spatial and spectral. Spatial methods draw inspiration from the success of Convolutional Neural Networks (CNNs) in image processing. They operate by capturing local neighborhood interactions to update node representations *(D. K. Duvenaud et al., 2020*; *Y. Li et al., 2020)*. On the other hand, spectral methods rely on spectral graph theory *(D. I. Shuman et al., 2013)*, utilizing graph Laplacians to define convolution operations within the graph domain *(M. Defferrard et al., 2016*; *Sahbi H. et al., 2021)*.

In today's landscape, research on Graph Neural Networks (GNNs) continues to evolve, aiming to bolster the model's performance and broaden its applications across diverse domains. A plethora of GNN variants have emerged to tackle the complexities of graph data. Among them, Graph Convolutional Networks (GCNs) *(Zhang et al., 2019*; *Chen et al., 2020*; *Yang et al., 2020*; *Pei et al., 2020*; *Chen et al., 2020)* stand out as a fundamental and widely adopted form. GCNs employ a graph convolution mechanism to propagate information across vertices and edges within the graph. By integrating vertex features and graph structure, GCNs facilitate classification or prediction tasks on graphs.

Another notable variant, Graph Sample and Aggregated (GraphSAGE) *(Hamilton et al., 2017*; *Ding et al., 2021*; *Xiao et al., 2019*; *Rong et al., 2019*; *Wang et al., 2021)*, employs a strategy of sampling and aggregating information from neighboring vertices to update each vertex's features. This approach enables GraphSAGE to glean insights into the overall graph representation, thus enhancing its capability to handle large-scale graphs effectively.

Furthermore, Graph Attention Networks (GATs) *(Veličković et al., 2017*; *Brody et al., 2021*; *He et al., 2023*; *Busbridge et al., 2019*; *Sun et al., 2023)* leverage the attention mechanism to assess the significance of neighboring vertices for each vertex in the graph. Through weighting the information from neighbors, GATs prioritize important vertices, enabling dynamic processing of the graph.

Information Extraction

In recent years, concurrent with the evolution of Graph Neural Network (GNN) models, there have been notable strides in the field of information extraction from scanned images. An increasing number of research endeavors leverage these GNN models in tandem with diverse methodologies to extract information from images. In our study, we draw upon several pertinent investigations in this domain.

One such study focuses on the extraction of information from invoices using a Spectral Graph Convolutional Network *(Bui Thanh Hung et al., 2022)*. This article offered a comprehensive overview of techniques for extracting information from invoices, encompassing template-based and natural language processing (NLP) approaches. Additionally, the article delineates the advantages of employing spectral graph convolutional networks and elucidates how they can effectively address the challenge of information extraction from invoices.

In their work, D. Lohani, A. Belaïd, and Y. Belaïd presented an innovative invoice reading system employing Graph Convolutional Networks (GCN) *(Lohani et al., 2019)*. This system demonstrates remarkable accuracy in reading invoices, even when confronted with diverse layouts. By harnessing GCN, the system effectively learns both the structural and semantic information inherent in invoice entities. Notably, the system operates without necessitating any predefined invoice format information.

Zhao, Xiaohui, et al. proposed the CUTIE model, a Universal Text Information Extractor utilizing Convolutional Neural Networks (CNNs) to comprehend document content *(Zhao et al., 2019)*. However, this model exhibits several limitations: it relies heavily on structured data, prioritizes textual content over graphical representations, struggles to adapt to new data, and lacks interpretational capabilities.

Yu, Wenwen, et al. introduced the PICK (Processing Key Information Extraction) model, designed specifically for extracting key information from text documents *(Yu et al,. 2021)*.

While many of the research articles and methodologies previously discussed employ Graph Convolutional Networks (GCNs) for information extraction and graph representation learning, there remain notable constraints in graph processing. These limitations often stem from dependencies on the input data's graph structure. In scenarios where input data lacks clear graph structures, establishing relationships between entities becomes challenging. Moreover, large training datasets may pose difficulties in pattern recognition, and biased training data can lead to the learning of inaccurate models.

In this study, we propose a new ConBGAT model for information extraction from scanned image. We use advanced models to extract image and text features with CNN and DistillBERT models and train on GNN models to solve the problem of extracting information from scanned image. We perform processing, identify regions containing text information and assign corresponding labels to each region. Then perform training of various deep learning models such as GCN *(Zhang et al., 2019)*, GraphSAGE *(Hamilton et al., 2017)*, GAT *(Veličković et al., 2017)*, GIN *(Xu et al. 2018)*, SGConv *(Wu et al. 2019)* to have an overview and conduct detailed evaluation and comparison between these models with our model on two datasets: SROIE invoice and our collected datasets.

The Proposed CONBGAT Model

We begin by identifying text regions within the input scanned image. Subsequently, we extract features by embedding both image and text features obtained from a model that combines CNN and DistilBERT. These features are then utilized to construct a graph representing the input data. The GNN model is trained on this graph-modeled data, enabling classification of nodes within the graph. Finally, we leverage the trained model to extract text entities classified and predicted by the GNN learning models, presenting the resulting text information as the output. Our proposed model is presented in *Figure 1*.

Word Recognition

This section comprises two primary tasks: Firstly, we identify the bounding boxes for each word region within the input scanned image, a process known as text detection. Subsequently, we extract the content of these words to facilitate feature extraction. The detailed steps are outlined below:

Text detection: We utilize the Character Region Awareness for Text Detection (CRAFT) model *(Baek et al., 2019)* in this study. CRAFT is specifically designed to identify text containers by leveraging character features, thereby achieving high performance, particularly with texts exhibiting complex shapes. The model employs an attention-based mechanism to predict the container for each character within the text.

Optical Character Recognition (OCR): OCR stands as a pivotal technology applied across various domains, ranging from natural language processing to office automation. Its capability to convert text images into machine-readable formats significantly reduces the time and labor involved in manual data entry processes. In this study, we leverage established OCR tools to identify text regions, as outlined in the text detection section. Specifically, we employ two prominent OCR engines: Tesseract *(Smith et al., 2007)* and EasyOCR *(Liao et al. 2022)*. These tools are renowned for their robust support in text recognition tasks.

- Tesseract, an open-source OCR engine developed by Google, stands out for its robustness and high accuracy in recognizing a wide range of languages. This versatile tool finds extensive application in various practical scenarios, including street sign recognition, digitizing paper documents, and streamlining business processes.
- EasyOCR, developed by OpenCV, is another open-source OCR engine known for its user-friendly interface and swift performance. Capable of recognizing multiple languages, EasyOCR is frequently employed for personal tasks such as converting physical documents into digital formats and QR code recognition.

Feature Extraction

Feature extraction here is to create features for the nodes (nodes, vertices) of the graph. In this study, the nodes of the graph are bounding boxes, which are areas containing text information identified in the text detection section above. We use two features for the nodes: image features and lexical features. We present details in *Figure 2*. First, from the bounding boxes defined above, we use two deep learning models to extract features: CNN model for extracting features from images, DistilBERT model *(Sanh et al., 2019)* for extracting text content features in the bounding box. Character is recognized by text recognition toolkits and integrate to nodes features. The definition of edges of the graph will be presented in detail in the next section.

Text Features Extraction: We employ word embeddings techniques to represent the words identified in the text. In our study, we utilize the pre-trained DistilBERT model to generate vector representations for text sentences. DistilBERT, introduced by Victor Sanh et al. *(Sanh et al., 2019)*, serves as a more compact alternative to BERT *(Devlin et al., 2018)*, offering comparable performance. DistilBERT is developed through a compression process known as 'distillation,' where a new model (referred to as the child model) is trained on predictions made by a larger model (the parent model). This process enables the child model to capture the essential features of the parent model without the need for extensive dimensions or parameters.

DistilBERT offers several advantages over BERT, including its reduced size, faster training and inference times, lower resource requirements, and cost-effectiveness, while maintaining equivalent performance capacity.

For a set of text in a document, we combine them according to coordinates from top to bottom and from left to right to form a character string. Given a string $tseq_k = (a^{(k)}_1,\ a^{(k)}_2,\ \ldots,\ a^{(k)}_i)$, text embed of the string $tseq_k$ is defined as follows:

$$TEmb^{(k)}_{1:i} = DistilBERT\left(a^{(k)}_{1:i}\,;\,\Theta_{DistilBERT}\right)\#\qquad(1)$$

Where $a^{(k)}_{1:i} = [a^{(k)}_1, a^{(k)}_2, \ldots, a^{(k)}_i]^i \in \mathbb{R}^{i\,*\,d_{model}}$ is the input string, $a^{(k)}_1 \in \mathbb{R}^{d_{model}}$ represents the embedded token of each character $a^{(k)}_1$, d_{model} is the size of the model. $TEmb^{(k)}_{1:i} = [TEmb^{(k)}_1, TEmb^{(k)}_2, \ldots, TEmb^{(k)}_i]^i \in \mathbb{R}^{i\,*\,d_{model}}$ represents the output embedded strings, $TEmb^{(k)}_i$ represents the i_{th} result of the pre-train model DistilBERT for k_{th} document. $\Theta_{DistilBERT}$ represents the parameters of the pre-train model DistilBERT. Each sentence is encoded independently, we get the text embeddings of document β with η sentences or text paragraphs. We define it as follows:

$$TFE = \left[TEmb^{(1)}_{1:i}, TEmb^{(2)}_{1:i}, \ldots, TEmb^{\eta}_{1:i}\right]\qquad(2)$$

Image Feature Extraction: We used CNN *(O'Shea et al., 2015)* for image embeddings. Given a set of image fragments created from these bounding boxes $iseq_k = \left(b^{(k)}_1, b^{(k)}_2, \ldots, b^{(k)}_i\right)$ for each text area in the pre-determined image, it will then be fed into the CNN model to perform feature representation and calculation for each box. Image embedding is defined as follows:

$$IEmb^{(k)}_{1:i} = CNN\left(b^{(k)}_{1:i};\,\Theta_{CNN}\right)\qquad(3)$$

Where $b_{1:i}^{(k)} = \left[b_1^{(k)}, b_2^{(k)}, \ldots, b_i^{(k)}\right] \epsilon\, \mathbb{R}^{H\,*\,W\,*\,3}$ are image region inputs, H and W are high and width of image respectively. $IEmb_{1:i}^{(k)}\, \epsilon\, \mathbb{R}^{H\,*\,W\,*\,d_{model}}$ is output of CNN model for i_{th} region image of a scanned image. Θ_{CNN} is parameters of CNN model. We use a variant of CNN for this image embeddings viz Resnet-50 *(He et al., 2016)* and a fully connect class that resizes the output according to the size of the d_{model}. By independent encoding, we can get the image embedding of the document β. We define it as follows:

$$IFE = \left[IEmb_{1:i}^{(1)}, IEmb_{1:i}^{(2)}, \ldots, IEmb_{1:i}^{(k)}\right] \tag{4}$$

After extracting text features (*TFE*) and image features (*IFE*), we combine these embeddings to create a new representation Ϋ by partially adding these features together. The features are then used as input nodes for the our GNN model GNN.

$$\ddot{Y} = TFE + IFE \tag{5}$$

Graph Modeling

As introduced above in this graph modeling section, we identify the edges of the graph and calculate the relative distances between boxes in the left, right, top and bottom directions if they exist, if they exist. Values that do not exist will be set to 0. *Figure 3* shows an overview of the relative distance between boxes on the image.

The relative distance indicators between boxes will be determined as follows:

$$\begin{aligned} D_L &= \frac{(RIGHT(Box_{left}) - LEFT(Box_{root}))}{WIDTH_{image}} \\ D_T &= \frac{(BOTTOM(Box_{top}) - TOP(Box_{root}))}{HEIGHT_{image}} \\ D_R &= \frac{(LEFT(Box_{right}) - RIGHT(Box_{root}))}{WIDTH_{image}} \\ D_B &= \frac{(TOP(Box_{bottom}) - BOTTOM(Box_{root}))}{HEIGHT_{image}} \end{aligned} \tag{6}$$

Where D_L, D_T, D_R, D_B corresponding to the relative distances left, above, right, and below the word Box_{root} (root box) to neighboring boxes.

The above parameters will be calculated based on the coordinates of the bounding boxes. These bounding boxes have been previously defined (in the text detection section). For example, with D_B will be equal to the distance from the original box to the box below and divided by the height of the image $HEIGHT_{image}$. With other parameters, perform similar calculations. *Figure 4* shows in detail the results we achieved after constructing graphs for the data.

Graph Attention Network Models

In this study, we adopt a variant of Graph Neural Networks (GNNs) known as the Graph Attention Network (GAT) for our analysis. The GAT model is selected for evaluation and comparison across both the SROIE dataset and our collected dataset.

GAT operates as a type of GNN that leverages the attention mechanism to ascertain the significance of neighboring vertices within the graph. Through weighted aggregation of neighbor information, GAT prioritizes crucial vertices and dynamically processes the graph. Notably, GAT demonstrates superior performance in various tasks, including classification, regression, and link prediction.

While GAT shares the foundational architecture of Graph Convolutional Networks (GCNs), it distinguishes itself by employing attention calculations instead of conventional convolutions to determine vertex importance. Below is the generalized formula of GAT for a propagation in the model.

Suppose the input of the model has N vertices and each vertex u will be represented by a feature vector $h_u \in \mathbb{R}^F$, with F is the dimensionality of the feature.

Compute the attention weight for each pair of vertices u and v.

$$e_{uv} = LeakyRELU\left(a^T[Wh_u \,||\, Wh_v]\right) \tag{7}$$

Where, $a \in \mathbb{R}^{2F}$, is a learned weight vector W is a learned weight matrix and || is the operator that joins two vectors. Soften the attention weights to sum to 1

$$\alpha_{uv} = \frac{exp\,(e_{uv})}{\sum_{v \in \mathcal{N}_u} exp\,(e_{uk})} \tag{8}$$

Where $\mathcal{N}_u$ is the set of adjacent vertices of vertex u.

Output of each u vertex.

$$h_u^{'} = \delta\left(\sum_{v \in \mathcal{N}_u} \alpha_{uv}(Wh_v)\right) \tag{9}$$

Where δ is a sigmod function.

This formula represents the propagation process through a graph. Each vertex calculates an attention weight with adjacent vertices, then smooths them and calculates a weighted sum of the adjacent vertex's features to obtain the final output of that vertex. This procedure is repeated for each vertex in the graph.

GAT is a powerful graph model that can be applied to many problems on graph data. GAT has proven its effectiveness in information extraction problems, thanks to the following outstanding points:

Cross-Attention: GAT employs the attention mechanism to compute attention weights between neighboring vertices. This enables the model to concentrate on pivotal nodes, thereby generating higher-quality representations for each node. Additionally, the attention mechanism enables the model to adeptly manage large-scale graphs with substantial structural variations.

Learnable Attention Weights: Attention weights are trainable through a linear function, allowing the model to dynamically prioritize important vertices during training. This enhances GAT's flexibility in determining the significance of vertices within the graph.

Ranking Mechanisms: GAT frequently integrates a rank accumulation mechanism into its attention weight learning process. This augmentation strengthens the model's proficiency in discerning the importance of connections between vertices within a graph, particularly within the realm of information extraction.

Efficiency and Flexibility: GAT demonstrates the ability to manage intricate and evolving graphs without compromising on model efficiency. This attribute proves invaluable in tasks like

information extraction, especially when dealing with intricate text graphs containing numerous relationships and information embedded within the vertices and edges.
Hence, we advocate for the adoption of GAT in our graph model, leading to notable outcomes. The ensuing section will delve into a detailed presentation of the results attained.

Loss function and Optimization

We used Cross Entropy Loss *(Mao et al., 2023)* in GNN classification task. This loss function is often favored in classification problems, especially when our model is faced with many different classes of objects.
Using the Cross Entropy Loss loss function helps us train our model so that it is capable of classifying graph objects accurately and efficiently. Specifically, Cross-Entropy loss is defined by the formula:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{N} y_i \log p_i \tag{10}$$

To optimize the loss function during model training, we use the AdamW *(Zhuang et al., 2022)* (Adam with Weight Decay) optimization algorithm. AdamW is a variant of Adam *(Kingma et al., 2014)*, a gradient-based optimization method commonly used in machine learning and deep learning tasks.
AdamW was designed to solve Adam's problem related to instability during training and unwanted growth of model weights. AdamW retains all the benefits of Adam, such as integrated adaptive learning rates and momentum, but adds a "weight decay" component.
Weight decay is a primary method to control overfitting in machine learning models by imposing a cost that depends on the weights of the parameters. In AdamW, the weight decay component is calculated and added to the weight update process. This helps prevent excessive growth of weights, minimizes the risk of overfitting, and improves the generalization ability of the model.
AdamW has demonstrated good performance in a variety of model training tasks and is generally popular among the research and development community in the field of machine learning. The combination of adaptive learning rates and weight decay helps improve model accuracy and stability, while minimizing the risk of overfitting during training.
Below are descriptions of the optimization steps of AdamW:

Algorithm 1: Adam with Weight Decay (AdamW)

Given: $\alpha, \beta_1, \beta_2, \varepsilon, \lambda \in \mathbb{R}$, $lr\ schedule\ \{\eta_t\}, t \geq 0$
Initialize: $\boldsymbol{a_0} \in \mathbb{R}^d$, $k_0 \leftarrow 0$, $l_0 \leftarrow 0$
for t = 1, 2, … , T **do**
Compute the stochastic gradient $\nabla q_t(a_{t-1})$
$\boldsymbol{h_t} \leftarrow \nabla q_t(a_{t-1}) + \lambda a_{t-1}$
$\boldsymbol{k_t} \leftarrow \beta_1 \boldsymbol{k_{t-1}} + (1-\beta_1)\boldsymbol{h_t}$, $\boldsymbol{l_t} \leftarrow \beta_2 \boldsymbol{l_{t-1}} + (1-\beta_2)\boldsymbol{h_t^2}$
$\hat{\boldsymbol{k}}_{\boldsymbol{t}} \leftarrow \boldsymbol{k_t} / (1-\beta_1^t)$, $\hat{\boldsymbol{l}}_{\boldsymbol{t}} \leftarrow \boldsymbol{l_t} / (1-\beta_2^t)$
$\boldsymbol{a_t} \leftarrow \boldsymbol{a_{t-1}} - \eta_t \lambda \boldsymbol{a_{t-1}} - \eta_t \alpha \hat{\boldsymbol{k}}_{\boldsymbol{t}} / (\sqrt{\hat{\boldsymbol{l}}_{\boldsymbol{t}}} + \varepsilon)$

end for

In our study, we applied the AdamW optimization method to our GAT model. When working with large graphs like scanned image datasets, there is a high risk of overfitting due to the diversity of the data and the complexity of the graph. AdamW with weight decay component helps control excessive growth of weights, reduces the risk of overfitting and improves the generalization ability of the model.

Dataset

We used SROIE dataset *(Huang et al., 2019)* with 973 receipts from stores. In this dataset we use 626 invoices for training and 347 invoices for testing. Each invoice has 4 main text fields including: Company, Address, Date and Total. The dataset is mainly English characters and numbers, the dataset exhibits variable layouts and complex structures. To facilitate research, the dataset also includes annotations for each text bounding box, including their corresponding coordinates and records. *Figure 5a* shows a sample of this dataset.

CORD (Consolidated Receipt Dataset for Post-OCR Parsing) *(Park et al., 2019)*. CORD is a groundbreaking dataset designed for invoice analysis, marking a significant milestone in publicly available resources for this purpose. It comprises meticulously annotated invoices, catering to both optical character recognition (OCR) and parsing. This dataset encompasses 1000 invoices, with 800 images allocated for the training set, 100 for validation, and another 100 for testing. The primary objective is to precisely categorize every word within the invoice into one of 30 fields across four distinct categories. Notably, our study utilized officially provided OCR images and annotations. The accompanying *Figure 5b* offers a glimpse into the invoice samples within the dataset.

FUNSD (Form Understanding in Noisy Scanned Documents) *(Jaume et al., 2019)* represents a publicly available dataset established to facilitate research and advancement in methods aimed at comprehending and extracting information from scanned forms plagued by noise. Encompassing a diverse array of form types such as applications, ballots, invoices, and more, this data stems from the RVL-CDIP dataset *(Harley et al., 2015)*. Within the FUNSD dataset, there exist 199 real-world scanned forms, meticulously annotated to delineate 9707 semantic entities. Among these, 149 images serve as training data, while 50 are earmarked for testing purposes. Offering versatility for a multitude of tasks, this dataset is particularly well-suited for the specific task undertaken in this study, which involves assigning each word a label selected from a predefined set of four categories: "Question," "Answer," "Header," or "Other." Illustrated in *Figure 5c* is a sample form from this dataset.

Hyperparameter

We did experiments on Pytorch with GPU Nvidia GeForce GTX 1650Ti (4GB of memory). We train the GAT model with 4 layers. For the correction factor, we use Dropout with ratio 0.2 for GAT. For pre-train DistilBERT model, we used the fix parameters on this model. The text embeddings has 512 dimension. Parameters of ResNet-50 model are the same with *(He et al., 2016)* . The fully connect layer is responsible for changing the output dimension 512.. Our model is trained on 2000 epochs, AdamW optimization function is used with learning rate 0.0001 for models to optimize Cross-Entropy loss and use batch size equal to 16 in the training phase.

Evaluation

We used F1-Score *(Harley et al., 2015)* to evaluate the performance and effectiveness of our model in a detailed and quantitative manner. For each scanned image in the test set, the extracted text is compared with reality. Extracted text is marked as correct if both the content and categories of the extracted text match reality; otherwise, it is marked as incorrect. F1-Score is an index that evaluates the performance of a classification model. It is a composite index of precision and recall. Precision is defined as the ratio of True Positive scores among all scores predicted by the model to be Positive (TP + FP). Meanwhile, Recall is defined as the ratio of True Positive scores among those that are actually Positive (TP + FN).

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{13}$$

Where TP, FP, FN represent for True Positive, False Positive, False Negative.

RESULTS AND DISCUSSION

Based on the method proposed above, we conduct experiments and evaluate our model on the SROIE dataset. Our proposed method yields impressive results.

First, we conduct experiments to choose the best optimization algorithm for our proposed model. We use some optimization algorithms such as: SGD *(Chase Lipton et al., 2014)*, RMSProp *(Liu et al., 2020)*, Adagrad *(Elshamy et al., 2023)*, Adam *(Zhang et al., 2018)*, AdamW *(Zhuang et al., 2022)* to compare the result and find the best optimization algorithm. Table 1 shows the comparison results of these optimization algorithms based on two measures: F1-score and Loss. Based on the result shown in Table 1, we can see that AdamW achieved the highest F1-Score (0.97), followed by Adam (0.94), RMSProp (0.90), SGD (0.86), and Adagrad (0.8643). Similarly, AdamW has the lowest Loss (0.1488), followed by Adam (0.2876), RMSProp (0.3632), Adagrad (0.4868) and SGD (0.5688). The results show that AdamW outperforms the remaining methods in this case, with higher F1-Score and significantly lower Loss. This also proves that it can help improve model accuracy and generalization better than other optimization algorithms.

Next, we select some basic GNNs models to experiment and evaluate with our proposed ConBGAT model: Graph Convolution Network (GCN) *(Zhang et al., 2019)*, GraphSAGE *(Hamilton et al., 2017)*, Graph Isomorphism Network (GIN) *(Xu et al., 2018)*, Simplifying Graph Convolutional Networks (SGConv) *(Wu et al., 2019)*. Table 2 shows the results of comparing the F1-Score measure of the proposed ConBGAT model with other GNNs models on the SROIE dataset about Company entities, addresses, dates and total invoices. Specifically, our ConBGAT model has a F1-Score measure 0.98, 0.98, 0.97 and 0.95 respectively for each entity: company, address, date and total invoice. The proposed ConBGAT model also achieved the highest accuracy for all entities, with Macro Average is 0.97. The result in Table 2 shows that the proposed model is an effective model for the task of entity classification on the real dataset. The model can learn complex relationships between entities, and has high accuracy for both each entity and all entities evaluation.

Finally, to prove the effectiveness of the proposed model, we have selected three previously introduced and developed models that have achieved good results for this problem to compare and evaluate with our proposed method. Our compared models are Spectral Graph Convolutional

Network *(Bui Thanh Hung et al., 2022)*, Bi-LSTM-CRF *(Huang et al., 2015)* and BERT-CRF *(Souza et al., 2019)*.

For the result in Table 3, we used the result of the research *(Bui Thanh Hung et al., 2022)* for Spectral Graph Convolutional Network and *(Hua et al., 2020)* for Bi-LSTM-CRF, BERT-CRF. In Table 4, we used the results of $LayoutLMv3_{BASE}$, BROS and PICK models are supplied by *(Huang et al., 2022)*, *(Hong et al., 2020)* and *(Bui Thanh Hung et al., 2022)* all other models by *(Xu et al., 2020)*.

Table 3 shows that our ConBGAT model outperforms the baseline models on all components of the dataset SROIE. Specically, our proposed model ConBGAT has the highest F1-Score for Company, Address and Date labels. For Total label, F1-Score of our proposed model is higher than Bi-LSTM-CRF and BERT-CRF, but lower than Spectral GCN model only 0.01 score. Overall, in both Table 3 and Table 4, our model always gives higher F1-Score results than all the baseline models.

When looking on the results shown in Table 2, Table 3 and Table 4, it can be seen that our proposed method produces better results than other models. Thanks to the attention mechanism of GAT combined with Resnet-50 to extract features for images and DistilBERT is used to perform text embeddings. Since then, the model has achieved positive results. Below are some images in the test dataset where we extracted entities shown in *Figure 6*.

However, when analyzing in detail, we can see that all labels in our method produce superior results compared to the rest, but only in the Total label, our model has lower results. compared to the model Spectral Graph Convolutional Network *(He et al., 2023)* to explain this problem, there are a few points as follows: First is the difference in labels, there is a quite large difference between labels in the data. Second, the Total label is quite limited in being able to extract detailed features, because it only contains very few numbers, making feature extraction more difficult and the above result tables also reflect clearly about this. The Total label always produces lower results than all other labels.

While our results and proposed method offer significant potential in addressing practical challenges associated with textual information identification, they also come with certain limitations:

Generalization: The model may struggle to generalize results to new datasets. This limitation arises from the model being trained on a specific dataset; if the characteristics of a new dataset differ significantly from those of the training dataset, the model may produce inaccurate results.

Data Characterization Challenges: The process of characterizing data can encounter several limitations, including unbalanced labels and information containers. This may cause the model to learn inaccurate features or fail to fully capture the characteristics of the actual data. Additionally, some containers may contain insufficient information about their characteristics, leading to inaccurate results for these containers.

These limitations present opportunities for future research and development aimed at enhancing and expanding the application of deep learning methods in the field of information identification from scanned images.

CONCLUSION

In our research, we introduced the ConBGAT model, which integrates CNN, DistillBert, and Graph Attention Network architectures to address the challenge of information identification in scanned images. Throughout our investigation, we proposed and explored novel methodologies to effectively tackle this problem. This endeavor not only expands the existing knowledge base in the field but also fosters advancements in image processing techniques and text-based information identification methods.

To ensure the practicality and efficacy of our proposed method, we conducted an extensive series of experiments comparing it with various existing methods on three datasets. The results obtained not only facilitate the evaluation of our model's performance but also offer valuable insights into understanding the disparities, advantages, and drawbacks among these methods.

Moving forward, we aim to enhance the model further by optimizing its performance and minimizing computational complexity. Additionally, we plan to explore methods to integrate multitasking capabilities into the model, enabling it to address multiple objectives or data types within a single image concurrently. We are committed to translating our research findings into practical implementations.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

On Behalf of all authors the corresponding author states that they did not receive any funds for this project.

Conflicts of Interest

The authors declare that we have no conflict of interest.

Competing Interests

The authors declare that we have no competing interest.

Data Availability Statement

The datasets are publicly available by *(Huang et al., 2019)* (SROIE), *(Park et al., 2019)* (CORD) and *(Jaume et al., 2019)* (FUNSD) at:

https://www.kaggle.com/datasets/urbikn/sroie-datasetv2

https://github.com/clovaai/cord

https://guillaumejaume.github.io/FUNSD/download/

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Author Contributions

Bui Thanh Hung conceived and designed the experiments, analyzed the data, performed the computation work, the experiments, prepared figures and/or tables, and approved the final draft.

Ho Vo Hoang Duy performed the experiments, performed the computation work, prepared figures and/or tables, and approved the final draft.

Vo Quoc Huy analyzed the data, authored or reviewed drafts of the article, and approved the final draft.

References

Allamanis, M., Brockschmidt, M., & Khademi, M. 2017. Learning to represent programs with graphs. arXiv preprint arXiv:1711.00740. https://doi.org/10.48550/arXiv.1711.00740

Baek, Y., Lee, B., Han, D., Yun, S., & Lee, H. 2019. Character region awareness for text detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9365-9374). https://doi.org/10.48550/arXiv.1904.01941

Brody, S., Alon, U., & Yahav, E. 2021. How attentive are graph attention networks? arXiv preprint arXiv:2105.14491. https://doi.org/10.48550/arXiv.2105.14491

Bui Thanh Hung, Nguyen Hoang Minh Thu. 2024. Novelty Fused Image and Text Models based on Deep Neural Network and Transformer for Multimodal Sentiment Analysis. Multimedia Tools and Applications Journal. https://doi.org/10.1007/s11042-023-18105-8

Bui Thanh Hung. 2022. Information Extraction from Receipts Using Spectral Graph Convolutional Network. Lecture Notes in Networks and Systems book series (LNNS, volume 371). https://doi.org/10.1007/978-3-030-93247-3_59

Bui Thanh Hung. 2023. Joining Aspect Detection and Opinion Target Expression Based on Multi-Deep Learning Models . Applications in Reliability and Statistical Computing. Springer Series in Reliability Engineering, pp 85-96. Springer. https://doi.org/10.1007/978-3-031-21232-1_4

Busbridge, D., Sherburn, D., Cavallo, P., & Hammerla, N. Y. 2019. Relational graph attention networks. arXiv preprint arXiv:1904.05811. https://doi.org/10.48550/arXiv.1904.05811

Chase Lipton, Z., Elkan, C., & Narayanaswamy, B. 2014..A2Thresholding classifiers to maximize F1 score. arXiv e-prints, arXiv-1402. https://doi.org/10.48550/arXiv.1402.1892

Chen, H., Yin, H., Sun, X., Chen, T., Gabrys, B., & Musial, K. 2020. Multi-level graph convolutional networks for cross-platform anchor link prediction. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 1503-1511). https://doi.org/10.48550/arXiv.2006.01963

Chen, M., Wei, Z., Huang, Z., Ding, B., & Li, Y. 2020. Simple and deep graph convolutional networks. In International conference on machine learning (pp. 1725-1735). PMLR. https://doi.org/10.48550/arXiv.2007.02133

D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE SPM, vol. 30, no. 3. https://doi.org/10.1109/MSP.2012.2235192

D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. 2015. Aspuru-Guzik, and R. P. Adams, Convolutional networks on graphs for learning molecular fingerprints, in NeurIPS. https://doi.org/10.48550/arXiv.1509.09292

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. https://doi.org/10.48550/arXiv.1810.04805

Ding, Y., Zhao, X., Zhang, Z., Cai, W., & Yang, N. 2021. Graph sample and aggregate-attention network for hyperspectral image classification. IEEE Geoscience and Remote Sensing Letters, 19, 1-5. https://doi.org/10.1109/LGRS.2021.3062944

Elshamy, R., Abu-Elnasr, O., Elhoseny, M., & Elmougy, S. 2023. Improving the efficiency of RMSProp optimizer by utilizing Nestrove in deep learning. Scientific Reports, 13(1), 8814. https://www.nature.com/articles/s41598-023-35663-x

Goyal, P., & Ferrara, E. 2018. Graph embedding techniques, applications, and performance: A survey. Knowledge-Based Systems, 151, 78-94.https://doi.org/10.1016/j.knosys.2018.03.022

Hamilton, W., Ying, Z., & Leskovec, J. 2017. Inductive representation learning on large graphs. Advances in neural information processing systems, 30. https://doi.org/10.48550/arXiv.1710.09471

Harley, A. W., Ufkes, A., & Derpanis, K. G. 2015. Evaluation of deep convolutional nets for document image classification and retrieval. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR) (pp. 991-995). IEEE. https://doi.org/10.48550/arXiv.1502.07058

He, K., Zhang, X., Ren, S., & Sun, J. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778), 2016. https://doi.org/10.48550/arXiv.1512.03385

He, L., Bai, L., Yang, X., Du, H., & Liang, J. 2023. High-order graph attention network. Information Sciences, 630, 222-234. https://doi.org/10.1016/j.ins.2023.02.054

Hong, T., Kim, D., Ji, M., Hwang, W., Nam, D., & Park, S. 2020. BROS: A pre-trained language model for understanding texts in document. https://openreview.net/forum?id=punMXQEsPr0

Hua, Y., Huang, Z., Guo, J., & Qiu, W. 2020. Attention-based graph neural network with global context awareness for document understanding. In Chinese Computational Linguistics: 19th China National Conference, CCL 2020, Hainan, China, Proceedings 19 (pp. 45-56). Springer International Publishing. https://link.springer.com/chapter/10.1007/978-3-030-63031-7_4

Huang, Y., Lv, T., Cui, L., Lu, Y., & Wei, F. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In Proceedings of the 30th ACM International Conference on Multimedia (pp. 4083-4091). https://doi.org/10.48550/arXiv.2204.08387

Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., & Jawahar, C. V. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In 2019 International Conference on Document Analysis and Recognition (ICDAR) (pp. 1516-1520). IEEE. https://doi.org/10.1109/ICDAR.2019.00244

Huang, Z., Xu, W., & Yu, K. 2015. Bidirectional LSTM-CRF models for sequence tagging. arXiv 2015. arXiv preprint arXiv:1508.01991. https://doi.org/10.48550/arXiv.1508.01991

Jaume, G., Ekenel, H. K., & Thiran, J. P. 2019. Funsd: A dataset for form understanding in noisy scanned documents. In 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW) (Vol. 2, pp. 1-6). IEEE. https://doi.org/10.48550/arXiv.1905.13538

Kingma, D. P., & Ba, J. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. https://doi.org/10.48550/arXiv.1412.6980

Kipf, T. N., & Welling, M. 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907. https://doi.org/10.48550/arXiv.1609.02907

Liao, M., Zou, Z., Wan, Z., Yao, C., & Bai, X. 2022. Real-time scene text detection with differentiable binarization and adaptive scale fusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(1), 919-931. https://doi.org/10.48550/arXiv.2202.10304

Liu, Y., Gao, Y., & Yin, W. 2020. An improved analysis of stochastic gradient descent with momentum. Advances in Neural Information Processing Systems, 33, 18261-18271. https://doi.org/10.48550/arXiv.2007.07989

Lohani, D., Belaïd, A., & Belaïd, Y. 2019. An invoice reading system using a graph convolutional network. In Asian Conference on Computer Vision (pp. 144-158). Springer, Cham. http://dx.doi.org/10.1007/978-3-030-21074-8_12

M. Defferrard, X. Bresson, and P. Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In NeurIPS. https://doi.org/10.48550/arXiv.1606.09375

Mao, A., Mohri, M., & Zhong, Y. 2023. Cross-entropy loss functions: Theoretical analysis and applications. arXiv preprint arXiv:2304.07288. https://doi.org/10.48550/arXiv.2304.07288

Nikhat Parveen, Prasun Chakrabarti, Bui Thanh Hung, Amjan Shaik. **2023.** Twitter Sentiment Analysis using Hybrid Gated Attention Recurrent Network. Journal of Big Data. https://doi.org/10.1186/s40537-023-00726-3

O'Shea, K., & Nash, R. 2015. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458. https://doi.org/10.48550/arXiv.1511.08458

Park, S., Shin, S., Lee, B., Lee, J., Surh, J., Seo, M., & Lee, H. 2019. CORD: a consolidated receipt dataset for post-OCR parsing. In Workshop on Document Intelligence at NeurIPS 2019. https://openreview.net/pdf?id=SJl3z659UH

Pei, H., Wei, B., Chang, K. C. C., Lei, Y., & Yang, B. 2020. Geom-gcn: Geometric graph convolutional networks. arXiv preprint arXiv:2002.05287. https://doi.org/10.48550/arXiv.2002.05287

Rong, Y., Huang, W., Xu, T., & Huang, J. 2019. Dropedge: Towards deep graph convolutional networks on node classification. arXiv preprint arXiv:1907.10903. https://doi.org/10.48550/arXiv.1907.10903

Sahbi, H. 2021. Learning laplacians in chebyshev graph convolutional networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 2064-2075). https://doi.org/10.48550/arXiv.2104.05482

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv 2019. arXiv preprint arXiv:1910.01108. https://doi.org/10.48550/arXiv.1910.01108

Smith, R. 2007. An overview of the Tesseract OCR engine. In Ninth international conference on document analysis and recognition (ICDAR 2007) (Vol. 2, pp. 629-633). IEEE. https://doi.org/10.1109/ICDAR.2007.4376991

Souza, F., Nogueira, R., & Lotufo, R. 2019. Portuguese named entity recognition using BERT-CRF. arXiv preprint arXiv:1909.10649. https://doi.org/10.48550/arXiv.1909.10649

Sun, C., Li, C., Lin, X., Zheng, T., Meng, F., Rui, X., & Wang, Z. 2023. Attention-based graph neural networks: a survey. Artificial Intelligence Review, 56(Suppl 2), 2263-2310. https://doi.org/10.1007/s10462-023-10577-2

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. 2017. Graph attention networks. arXiv preprint arXiv:1710.10903. https://doi.org/10.48550/arXiv.1710.10903

Wang, X., & Vinel, A. 2021. Benchmarking graph neural networks on link prediction. arXiv preprint arXiv:2102.12557, 2021. https://doi.org/10.48550/arXiv.2102.12557

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. 2019. Simplifying graph convolutional networks. In International conference on machine learning (pp. 6861-6871). PMLR. https://doi.org/10.48550/arXiv.1902.07153

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. 2020. A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems, 32(1), 4-24. https://doi.org/10.1109/TNNLS.2020.2978386

Xiao, L., Wu, X., & Wang, G. 2019. Social network analysis based on graph SAGE. In 12th international symposium on computational intelligence and design (ISCID) (Vol. 2, pp. 196-199). IEEE. https://doi.org/10.1109/ISCID.2019.10128

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. 2018. How powerful are graph neural networks? arXiv preprint arXiv:1810.00826. https://doi.org/10.48550/arXiv.1810.00826

Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., ... & Zhou, L. 2020.. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. arXiv preprint arXiv:2012.14740. https://doi.org/10.48550/arXiv.2012.14740

Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel. 2016. Gated graph sequence neural networks, in ICLR, 2016. https://doi.org/10.48550/arXiv.1511.05493

Yang, Y., Feng, Z., Song, M., & Wang, X. 2020. Factorizable graph convolutional networks. Advances in Neural Information Processing Systems, 33, 20286-20296. https://doi.org/10.48550/arXiv.2010.05421

Yu, W., Lu, N., Qi, X., Gong, P., & Xiao, R. 2021. PICK: processing key information extraction from documents using improved graph learning-convolutional networks. In 25th International Conference on Pattern Recognition (ICPR) (pp. 4363-4370). IEEE. https://doi.org/10.1109/ICPR48806.2021.9412927

Zhang, N., Lei, D., & Zhao, J. F.. 2018. An improved Adagrad gradient descent optimization algorithm. In 2018 Chinese Automation Congress (CAC) (pp. 2359-2362). IEEE, 2018. https://doi.org/10.1109/CAC.2018.8623271

Zhang, S., Tong, H., Xu, J., & Maciejewski, R. 2019. Graph convolutional networks: a comprehensive review. Computational Social Networks, 6(1), 1-23. https://doi.org/10.1186/s40649-019-0069-y

Zhao, X., Niu, E., Wu, Z., & Wang, X. 2019. Cutie: Learning to understand documents with convolutional universal text information extractor. arXiv preprint arXiv:1903.12363. https://doi.org/10.48550/arXiv.1903.12363

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., ... & Sun, M. 2020. Graph neural networks: A review of methods and applications. AI open, 1, 57-81. https://doi.org/10.1016/j.aiopen.2021.01.001

Zhuang, Z., Liu, M., Cutkosky, A., & Orabona, F. 2022. Understanding AdamW through Proximal Methods and Scale-Freeness. arXiv preprint arXiv:2202.0008. https://doi.org/10.48550/arXiv.2202.00089

Figure 1

The proposed ConBGAT architecture

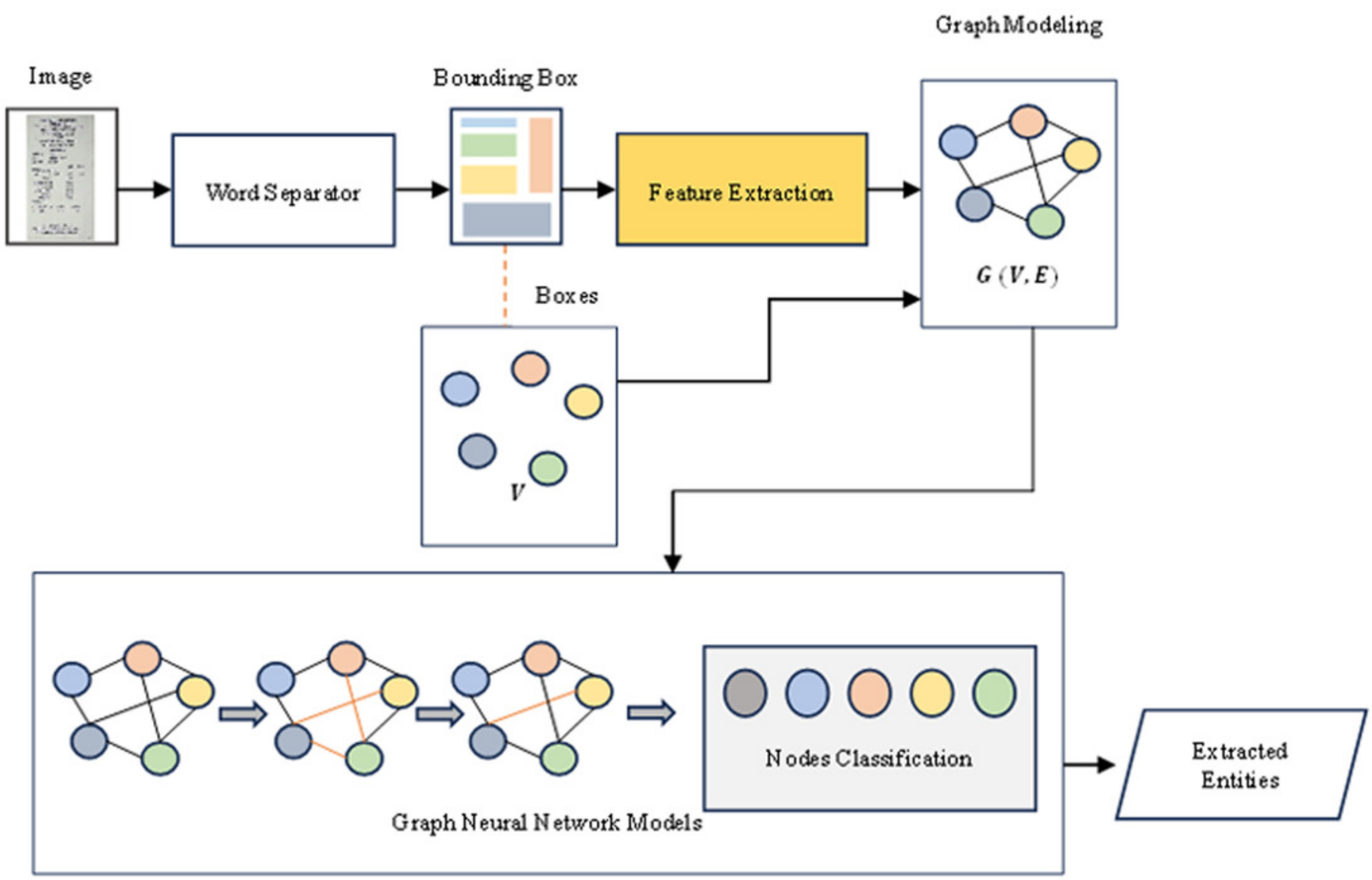


Figure 2

Features Extraction method

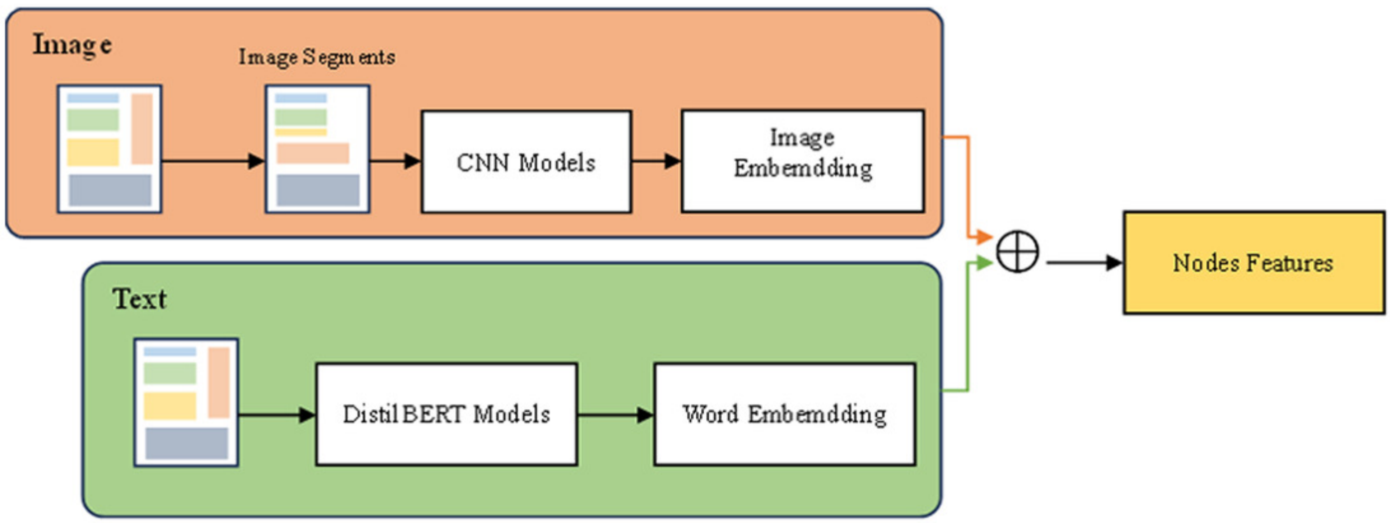


Figure 3

Relative distance of boxes on the image

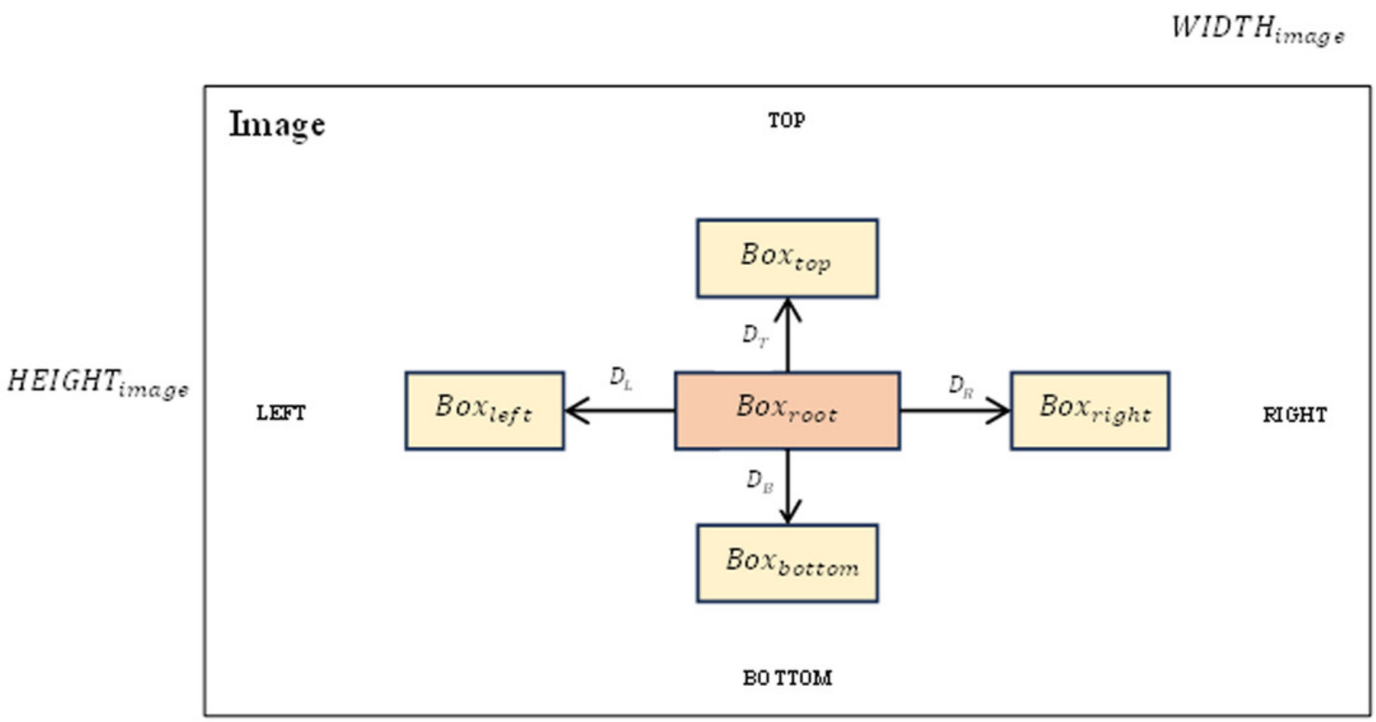


Figure 4

Scanned image after Graph Modeling processing

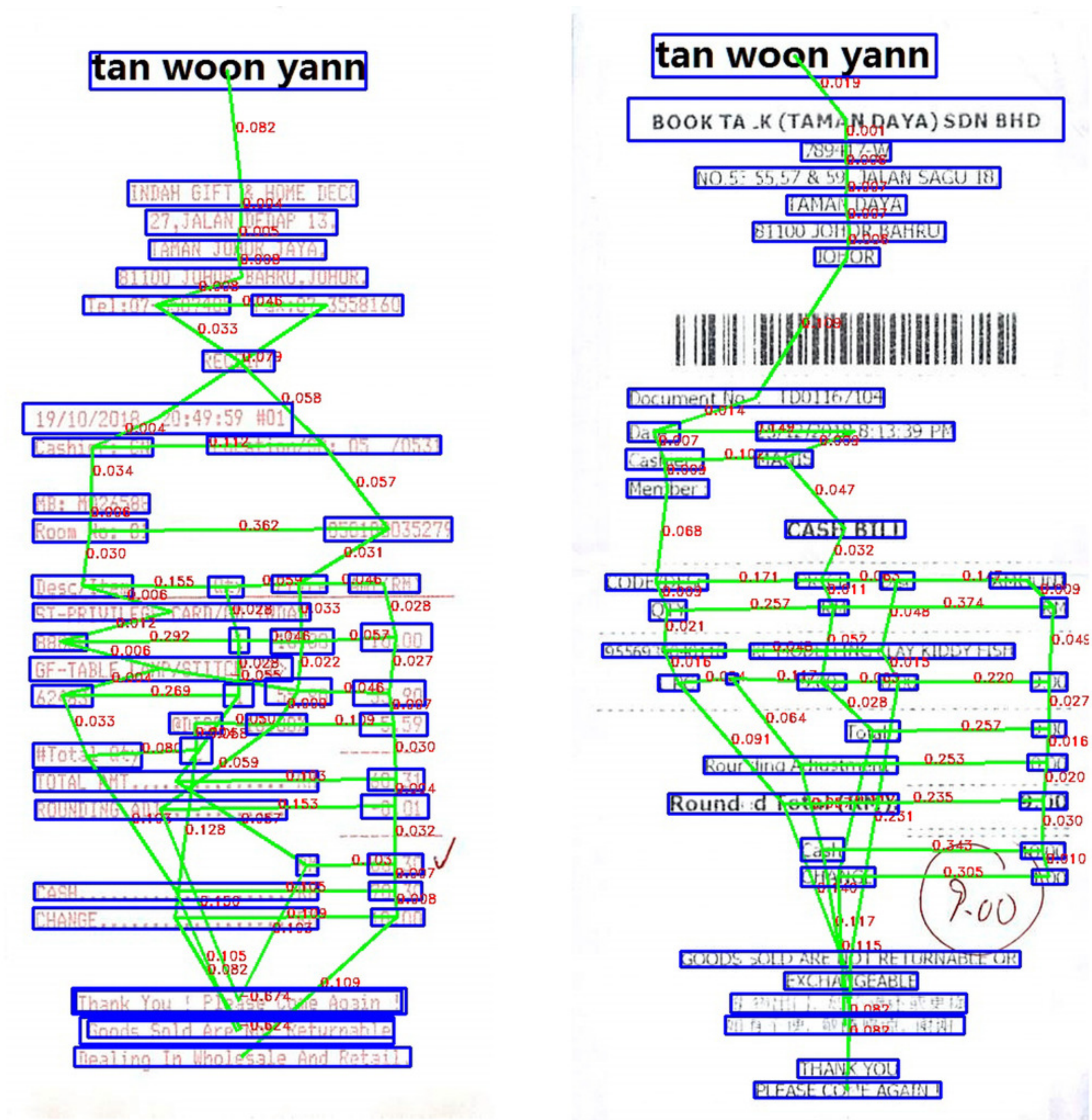
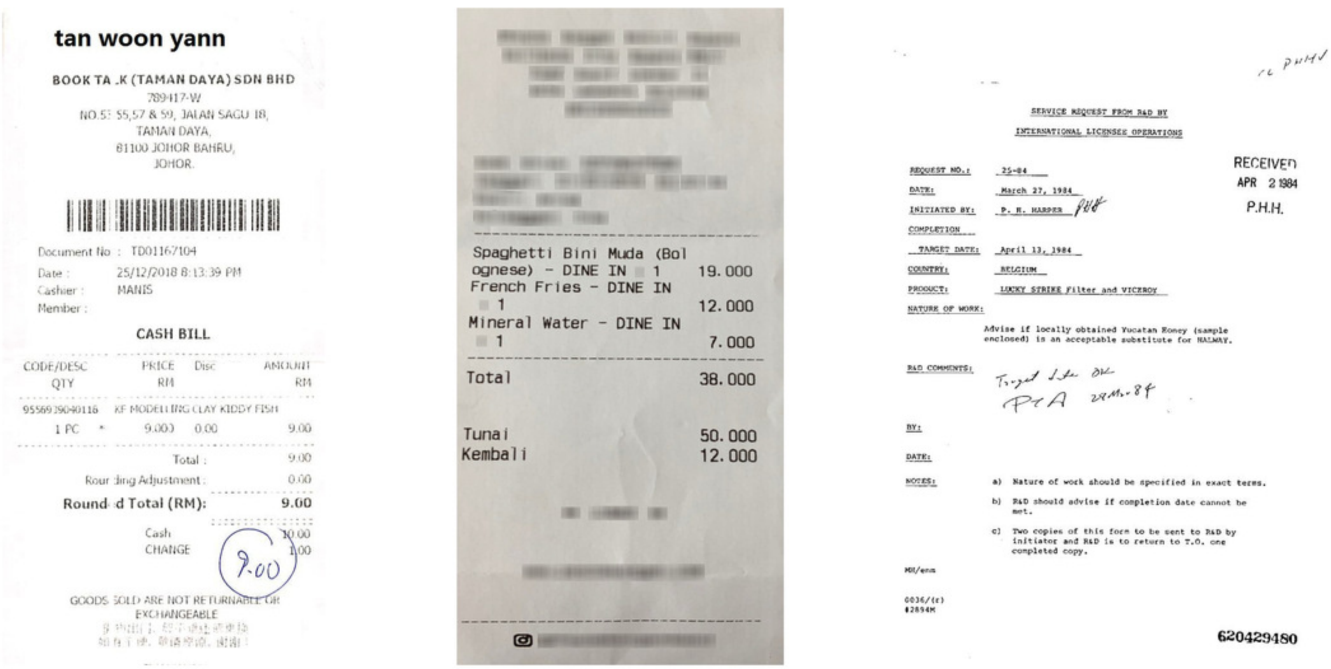


Figure 5

Some images from three datasets



a.SROIE dataset b.CORD dataset c.FUNSD dataset

Figure 6

Example results of extracted entities

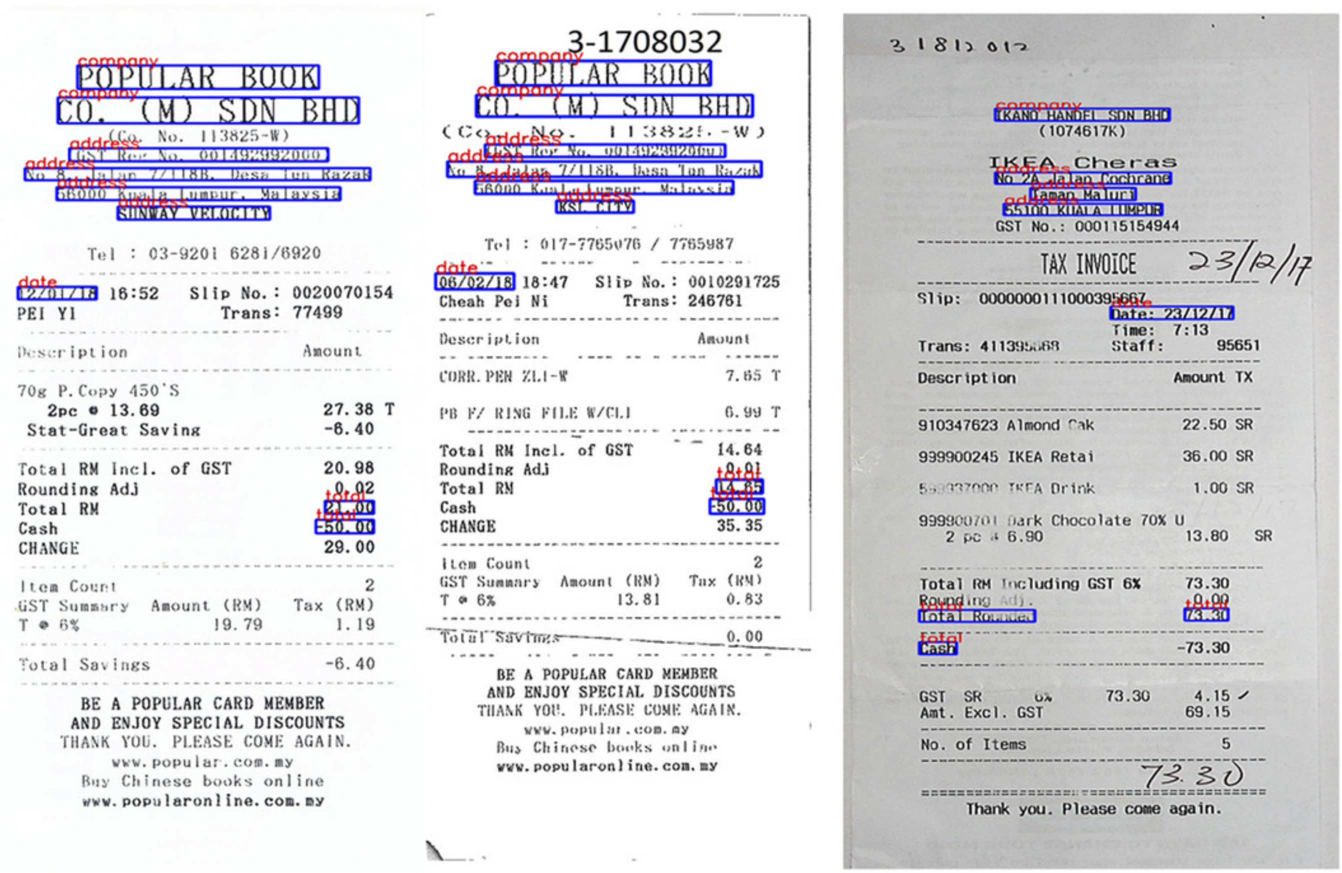


Table 1(on next page)

Comparison results of optimization functions based on F1-Score and Loss measurements

Optimization Algorithm	F1-Score	Loss
SGD	0.86	0.5688
RMSProp	0.90	0.3632
Adagrad	0.8643	0.4868
Adam	0.94	0.2876
AdamW	**0.97**	**0.1488**

Table 2 (on next page)

Results of comparing the F1-Score measure with the GNNs models

Entities	GCN	SAGE	SGConv	GIN	**ConBGAT**
Company	0.8	0.85	0.68	0.73	**0.98**
Address	0.8	0.88	0.73	0.76	**0.98**
Date	0.76	0.8	0.77	0.68	**0.97**
Total	0.72	0.75	0.8	0.71	**0.95**
Macro Average	0.77	0.82	0.745	0.72	**0.97**

Table 3 (on next page)

Results of comparing the F1-Score measure on each label of our proposed ConBGAT model with the baseline models on the SROIE dataset

Entities	Spectral Graph Convolutional Network	Bi-LSTM-CRF	BERT-CRF	**ConBGAT**
Company	0.85	0.851	0.868	**0.97**
Address	0.93	0.883	0.891	**0.96**
Date	0.95	0.942	0.962	**0.96**
Total	**0.93**	0.835	0.847	0.92
Macro Average	0.915	0.878	0.892	**0.9525**

Table 4 (on next page)

Results comparing the F1-Score measure of our model with the baseline models on the SROIE, FUNSD and CORD datasets

Model	SROIE	FUNSD	CORD
$BERT_{BASE}$	90.99	60.26	89.68
$BERT_{LARGE}$	92.00	65.63	90.25
$UniLMv2_{BASE}$	94.59	68.90	90.92
$UniLMv2_{LARGE}$	94.88	72.57	92.05
$LayoutLM_{BASE}$	94.38	78.66	94.72
$LayoutLM_{LARGE}$	95.24	78.95	94.93
$LayoutLMv2_{BASE}$	96.25	82.76	94.95
$LayoutLMv2_{LARGE}$	96.61	84.20	96.01
$LayoutLMv3_{BASE}$	-	**90.29**	96.56
BROS	94.93	81.21	95.58
PICK	96.79	-	-
ConBGAT	**97.00**	89.61	**96.72**