

RESEARCH ARTICLE

End to End Invoice Processing Application Based on Key Fields Extraction

HALIL ARSLAN 

Department of Computer Engineering, Engineering Faculty, Sivas Cumhuriyet University, 58000 Sivas, Turkey


e-mail: harslan@cumhuriyet.edu.tr

ABSTRACT In this paper, an automatic invoice processing system, which is in great demand among private and public companies, was proposed. The proposed system supports all invoice file types that can be submitted by companies. Companies can easily submit invoices to the system via the web interface or email, and all invoices submitted to the system are queued and processed sequentially. If the invoice is a text file, the invoice information is extracted from the text by using template matching. If the invoice is an image, the text and table areas are detected and extracted. For table detection, we used both image processing based and YOLOv5-based deep learning method. Cell extraction was then performed from the extracted table images. As a result of these processes, all text and table cells were obtained as images and these images were converted into machine-readable text using the open-source software Tesseract OCR. Tesseract already provides trained models for English and Turkish. However, these models do not provide successful results for invoices submitted by companies in Turkish. Therefore, the new fine-tuned model trained with invoices in Turkish was used for OCR. The experimental results showed that the trained Turkish model was more accurate than the Turkish and English models provided by Tesseract. In addition, the YOLOv5-based table detection model was more accurate than the image-processing-based table detection method.

INDEX TERMS Invoice processing, key fields extraction, text detection, deep learning, table extraction, optical character recognition.

I. INTRODUCTION

With the increasing technological development and the reduction of costs, digitalization has not only changed the possibilities of people, but also the business world in terms of production, marketing and management [1]. Intelligent production according to Industry 4.0 standards, transportation tracking with IoT devices, and product recommendations through the use of personal data are examples of the positive effects of digitalization. On the other hand, management is being facilitated by applications, services and electronic transformations to meet needs such as managing human resources in an electronic environment, tracking goods and suppliers, and recording accounting processes. Enterprise Resource Planning (ERP) systems, in particular, offer companies a great advantage in management.

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

ERP systems help businesses with resource planning by integrating all core processes into a single system. These systems offer great convenience as they can manage the processes of many different departments such as manufacturing, purchasing, finance, engineering, and logistics through a single system. ERP systems, which also contribute to the electronic transformation process of businesses, can manage many processes automatically. However, these processes usually aim to solve the main needs of businesses such as accounting transactions and logistics processes. If another need arises specifically for electronic transformation, software or an extension should be developed and integrated into the business system. The management and recording of incoming invoices automatically are one of these requirements.

Invoices contain important information related to a particular business department, such as the total amount, taxpayer identification number (TIN), information about the material

sold, and the date. To ensure consistency between assets and liabilities, expedite accounting, and establish strategies for sales marketing, key fields of invoices are commonly used. Typically, most companies use a traditional invoice management system where invoice data is manually entered and stored in hard copy in archives, which takes up a lot of time and space. Moreover, businesses can get into financial and legal trouble if some of the archived documents are damaged or untraceable. Therefore, there is a need to manage invoices without software that automatically processes, extracts and stores important information in the ERP system.

Digitization of archived invoices has been studied in the literature since before the 1990s. There are many studies in the literature on digitization including not only on invoices but also documents such as historical books, business letters, and newspapers [2]–[4]. Many computer vision and machine learning-based techniques have been proposed in the literature for invoice processing. Invoice processing can be divided into four parts:

- Preprocessing: Invoices may contain noise because they are usually scanned, so preprocessing should be applied [5], [6].
- Text and Table Detection: Extraction of information from invoices is performed using text and table regions. Image processing techniques and convolutional neural networks are commonly used in the literature for this purpose [7]–[9].
- Optical Character Recognition (OCR): The obtained images are converted into digitized text by OCR. There are many studies on OCR in the literature and it is available in many open source libraries [10], [11].
- Information Extraction: Extracting information from invoices is one of the widespread topics in the literature. There are template-based and learning-based network methods for extracting information from documents [12]–[14].

Many methods have been proposed in the literature for table detection in document images. Nevertheless, deep learning models have become the state-of-the-art approach. As can be seen, object detection models in the literature were adapted with fine-tuning for table detection [15]. Object detection applications aim to detect objects in an image. By considering tables as objects, table detection can be performed directly in these models. Successful results are obtained in table detection are achieved with some minor adjustments and additional operations.

In this paper, we propose an end-to-end invoice digitization system that analyzes printed or digital invoices and extracts key information. Since the goal of our study is not only to find the important parts of invoices, but also to develop a system that meets the requirements of invoice digitization in enterprises, many different modules need to be processed. Our study includes various steps such as determining the text and table positions in the invoice image using deep learning models, smoothing and extracting areas using various image processing algorithms, digitizing the text of the extracted

areas using optical character recognition methods, and developing an integration/web interface. Due to serve fine-tunable models, we preferred the open-source application Tesseract OCR in the text recognition phase, and The Turkish language model was re-trained using 957 different texts in invoices. Moreover, the system supports various document formats such as PDF, XML, HTML and images, meeting almost all enterprise requirements for digitizing invoices.

Although many different studies have been published in the literature on digitising invoices as an academic study, an application that meets the needs of enterprises from field extraction to integration with ERP systems using object detection with image processing techniques together has not been intensively studied. Since the research and development centre where the study was conducted provides a large number of Turkish invoices and deep learning models require a large number of samples, our system works only on Turkish invoice sheets so far. However, our system can be easily integrated into another country's invoice digitization processes by simply changing the training datasets. So, the novelty of our study is that developing an end-to-end system that includes different processes instead of changing or improving the algorithm.

The rest of the paper is organized as follows: Section 2 provides information on the step-by-step process of invoice processing. Section 3 explains the materials and methods we used. The application and developed algorithms for extracting text and table parts are explained in detail in Section 4. Section 5 contains the evaluation results from invoice processing systems. Finally, Section 6 explains the conclusions and future work.

II. LITERATURE REVIEW

The structure of invoices can vary by country, by type of business, and even by company. Therefore, digitizing differently structured invoices and extracting the key information they contain requires applying some complex techniques to the invoices. In particular, the challenges of recognizing parts of tables and text, reducing noise on a document, and optical character recognition have led to many academic and sectoral studies on this topic. Gangal et. al. have applied various morphological techniques such as dilation, erosion and opening using the OpenCV library to remove blur and skewness from the document image. The result of the study is that the incoming image is preprocessed and prepared for the next stage such as OCR. Morphological operations are also commonly used to locate text containing parts in an image [16]. El Khattabi *et al.* extracted the text regions of the images using dilation and erosion operations. In order to determine whether the found part contains text or not, an algorithm was developed to check the structure of the part. [17]. In a similar study, text regions of document images are identified based on the differences between opening and closing operations [18].

There are also studies in the literature where image processing techniques are used in all stages of invoice and document processing systems. Y. Sun *et al.* have developed

a system that detects and extracts invoice information using the template matching method. First, the invoices, which are images, are preprocessed. At this stage, the unnecessary background is removed, then invoice angle correction is performed. The region with the requested information on the invoice is determined by template matching. Finally, the image is converted into machine text using OCR technology [19]. S. Bhowmik *et al.* performed document layout analysis for document image processing. The system called BINYAS is based on connected components and pixel analysis. The classification process such as paragraph, graphic, image and table are carried out with image processing methods [20].

In addition to image processing techniques, deep learning methods are also used to detect the tables on the document. Tables in the document may be available in different layouts and styles. It also happens that the borderlines of tables may be deleted or not found at all. For this purpose, object detection models can be used as well as deep learning models developed especially for table detection [21]. Y. Huang *et al.* modified the YOLOv3 [22] object detection model to make it suitable for table detection. In the proposed study, the anchor sizes used in YOLOv3 were adapted to be suitable for tables. Nine different table sizes were determined using the K-means clustering algorithm and the determined dimensions were used as anchor sizes. Another improvement is the post-processing of the obtained tables. This process ensures that the areas between the detected table and the actual table are eliminated. In the ICDAR 2017 dataset, an F-score value of 97.1% was obtained with a threshold of 0.8 IoU (Intersection over Union) [15]. In another study, D. Prasad *et al.* used Mask R-CNN [23] deep learning models for table detection. In this study, it is shown that Convolutional Neural Networks (CNN) based object detection algorithms are also very successful in detecting tables. State-of-the-art accuracy rates were achieved in experiments with ICDAR 2013 and ICDAR 2017 datasets [24]. This proposal is also supported by other studies in the literature [25], [26].

In addition, studies have been performed on the invoice processing system for business use. M. S. Satav *et al.* have designed a system for managing invoices. The proposed system, which is web-based, extract information from invoices image and store them categorically. Thus, a better system has been proposed for managing invoices. OpenCV library is used for image processing techniques and the open-source Tesseract library is used for OCR. The developed system is suitable for small businesses and small amount of information can be extracted from the invoice [27].

One of the most important stages of the invoice digitalization process is the Optical Character Recognition. Shi *et al.* used CNN to recognize character of Chinese invoices. They developed an algorithm that locates texts on the documents using template matching and segments them in order to piecewise processing. As a result, proposed system obtained more than 99% character recognition accuracy. [28]. Gui *et al.*, used Sobel filtering and residual network (ResNet)

to recognize only value-added taxes (VAT) in invoices. Using data augmentation methods, number of invoices in dataset is increased and models are tested. Results show that ResNet models obtained 99% accuracy while CNN stuck in 97% [29].

As a result of the OCR process, the data on the invoice is obtained. However, it is necessary to extract information from this data. M. Rusinol *et al.* have proposed the extraction of information from scanned invoices. The main purpose of the proposed system is to require minimum human intervention. Although the template model is learned using a single annotated image, the model can adapt to new situations with post-processed invoices. This ensures that different patterns of the invoice can be absorbed [13]. Machine learning-based approaches are also used for key field extraction. Basivkar *et al.* propose an algorithm that process multi-layout unstructured invoice documents without any template. Text regions which include key information are identified using machine learning and feature extraction methods such as Word2Vec, Glove and FastText. Besides, bidirectional long-short term memory neural model also utilized to finding key fields of invoice [30]. In another study, Graph based convolutional models that effective and robust in handling complex documents layout are used to extract and process key field information of any layout without ambiguity. Along with text and location box, original image is used as input for deep model which includes CNN, BiLSTM and Graph Convolution layers inside in. Models are tested over medical invoices and train ticket documents and result shows that proposed method obtained 87% and 98% mean entity F-1 score respectively [31].

The proposed end-to-end invoice processing system includes many stages. However, in the literature it is focused on a single topic such as table detection, text detection and OCR technologies on document image. In addition, the developed invoice processing systems do not cover all the requirements of enterprises. Inspired by this deficiency in the literature, a system was developed to cover all the needs of enterprises by integrating with ERP systems, using object detection and image processing techniques together. Another contribution of this paper is the ability to work with most incoming file types such as HTML, XML, PDF and image formats.

III. MATERIALS AND METHODS

A. INVOICE QUEUING

Due to data loss and latency, effectively managing incoming request from various clients in a web service is a crucial step for the service provider. To address that problems message brokers have been started to be used in order to manage incoming request efficiently. Message brokers are inter-application or inter-service communication technology that enable communication with exchanging information. By this way, applications that are written in different programming language can talk interdependently. In need of guaranteed delivery brokers stores requests in a queue until the consumer application process them. Our system process

invoices in two ways, through web interface or e-mail listener system. Therefore we used ActiveMQ Artemis (AMQA), [32] which is an open source queuing system.

The AMQA project is a high performance asynchronous messaging system and is an example of Message Oriented Middleware (MoM). The asynchronous system design allows efficient use of hardware resources and full utilization of network bandwidth. Another feature of the system is that it provides reliable messaging. Even in the event of system failure, it guarantees that the message will be delivered to consumer once and only once.

B. MORPHOLOGICAL OPERATIONS

In this paper, one of the image processing techniques used in the system is morphological operations. In morphological operations, an output image is obtained as a result of applying a structured element (SE) to the input image. The basic morphological operations are erosion and dilation. Other morphological operations consist of a combination of these two operations. In general, morphological operations are used for noise removal, cleaning of individual elements, filling holes and joining broken part of object. Erosion, dilation, opening, closing, morphological gradient (MG), hit and miss transform and thinning were used as morphological operations in this study.

C. TESSERACT OCR

OCR is a system that converts any type of printed text such as images, scanned or handwritten documents into a machine encoded format. There are many applications for OCR systems such as plate recognition, identity recognition, information extraction, converting handwritten books into digital form, and making scanned documents searchable. However, due to the variety of languages, fonts, styles, formats and differences between languages, the applicability of OCR has become difficult in some cases. Therefore, there are many proposed applications and studies in the literature and communities. One of them is Tesseract OCR.

Tesseract, which was developed between 1984 and 1994 by HP and released as open source in 2005, is an OCR library that contains more than 100 different fonts. This advanced solution includes some useful modules such as a layout analyzer and a connected component analysis to find the position of words in the printed text. It converts not only regular texts, but also uniform blocks and vertically aligned texts. In addition, the library allows training models and adding new characters that are not available in Tesseract by using custom images and text in a suitable format. After version 4, Tesseract began to use long-short term memory (LSTM) networks to increase the predictive accuracy of particularly long texts. For this reason, it started to be used more widely in the literature.

D. DATASET

Due to the obligations of digital transformation by the authorities, medium/large companies in Turkey have to use invoices in electronic form. Since the proposed system is a real-time

TABLE 1. File types of invoices with train test split.

Invoice Type	# Invoices	# Train Invoices	# Test Invoices
Image	140	112	28
Image PDF	1057	845	212
Vector PDF	2480	-	-
XML – HTML	1629	-	-
Total Invoices	5306	957	240

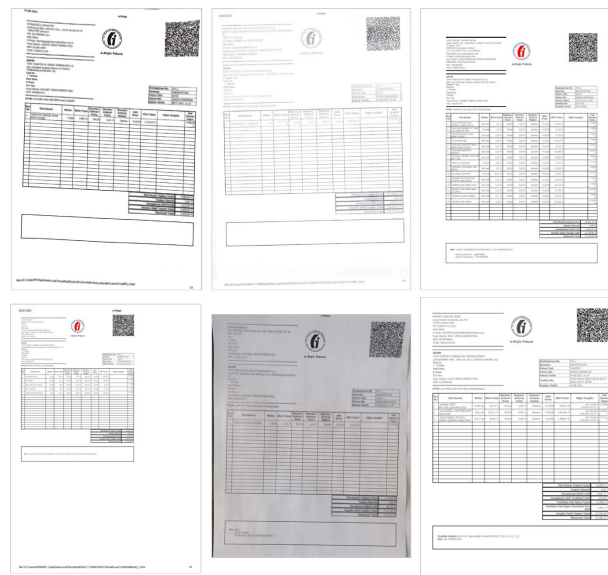


FIGURE 1. Some of the sample invoices included in the dataset. (Some parts of the invoices are frozen due to sensitive information.)

application, it should process invoices not only in image format but also in other formats. Therefore, our proposed system provides for the processing of invoices that can be in PDF, XML, and HTML formats as well as in image format.

In order to evaluate the performance of the proposed system, 5306 electronic archive invoices were collected from six different large companies. The quantitative information by type of invoices and the samples in the dataset are shown in Table 1 and Fig. 1, respectively. The PDF invoices were collected in two different formats: text embedded vector and a PDF image. If a PDF file contains embedded text, that file was treated as text and processed by the invoice reader module rather than the image module. Otherwise, the PDF file was converted to an image.

957 of the invoices in image and image PDF type listed in Table-1 were used for the training process of the deep learning and optical character recognition model to increase the performance. Those samples were not used in the testing phase. Deep learning and optical character recognition model was tested using the 240 test invoices. Since there is no training in the vector PDF, XML and HTML files, they are not separated as test and train.

IV. APPLICATION

The proposed application consists of five main modules: web and mail, invoice reading, image processing, OCR and

information extraction. The invoice processing operation begins with one or more requests from the web or email. These requests are received by the reading module to process the invoice. If target invoice is an image, the image processing module (IPM) is activated to extract the tables and text on the invoice. The table and text images are processed in the OCR module and converted to machine-readable text. Subsequently, information extraction is performed on the obtained text to extract necessary information from invoice. The extracted information is stored in the database and displayed to the users via web module. In addition, companies can manage incoming invoices automatically by integrating this incoming invoice information with the ERP systems in which they are used.

Different programming languages and libraries were used to develop each modules of the proposed system. In this context, React, Java and Python were used to develop web module, reading - email module and image processing - OCR modules respectively. In addition to programming languages or platforms, many additional libraries were used to develop the OCR modules. In this module, image processing methods were developed by using OpenCV libraries and images were converted to machine text using Pytesseract [33] library. All of these modules, which were developed by using several programming languages, have to work in integration with each other for a seamless system. However, it is inevitable to use many configurations for all these modules. Virtualization techniques are commonly using to solve problem in today's technology. In this study, each module was virtualized using Docker technology, which is a computer program that provides operating system level virtualization, also known as "containerization", to compile and deploy system quickly.

A. WEB AND E-MAIL MODULE

Our system accepts invoices via web or email. The web part is one of the parts that greet users and initiate invoice processing. Users can submit invoices as single or multiple invoices via the web interface or via email. After the invoices are processed, users are presented with information about the invoices. Invoices sent by email are listened in the background via a web socket to access information in the invoice that received by mail. The general system settings can also be adjusted via the web interface. For example, the email address to which the invoice should be sent can be edited via the web interface. There is also a web interface where all invoices can be listed, searched and analyzed.

Invoices can be delivered simultaneously via the web interface and via email. Incoming invoices from both the web and email modules are queued and processed one after the other. For this purpose, the AMQA message service is used in the end to end invoice processing and key fields extraction system. Here invoices can be considered as a message. In addition, after a system failure, undelivered invoices are automatically reloaded when the system is restarted.

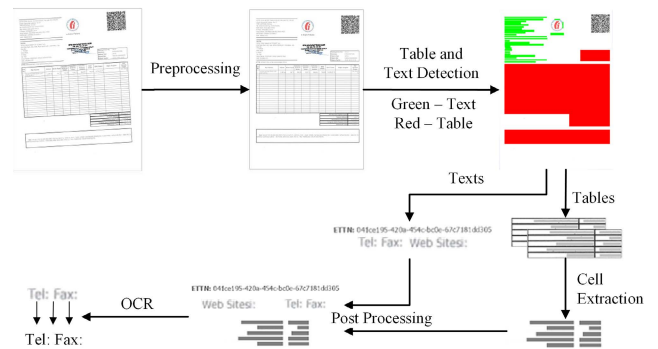


FIGURE 2. Computer vision based invoice processing system stages.

B. INVOICE READING MODULE

This module handles all invoice processing requests that come from both the web and mail modules. Invoice reading module is activated according to result of invoice filtering method where the type of file (image, PDF, XML and HTML) are determined by it. If the invoice type is an image, the invoice will be sent to the IPM, thus, invoice reading module are not activated. Other invoice types such as PDF, HTML, and XML are processed by the invoice reading module. Since HTML and XML files are text files, they can be processed directly. Therefore, the invoice information in these file types can be easily accessed.

In the first phase of invoice reading module, PDF files are tried to be read to find out if the PDF consist of and image or a text. If it is not possible to extract the text from the PDF, it is assumed that the invoice is an image, and the invoice is sent to IPM. In some cases, even if PDF files contain text, the desired information cannot always be obtained. In some invoices, some information such as universally unique identifier (UUID) and TIN cannot be read. In this case, a request is sent to the IPM only for information that cannot be obtained. Thus, the information extraction is performed by combining the texts received from the image processing and invoice reading modules.

C. IMAGE PROCESSING MODULE (IPM)

In this section, a computer vision based algorithm for invoice processing is proposed. The system consists of four main stages: preprocessing, table and text detection, cell extraction, and post processing. These steps are shown in Fig. 2.

1) PREPROCESSING

In the preprocessing stage, the general structure of the invoice is checked and all invoices are resized to the same width. Because some invoice images may contain large gaps, the invoice font size may decrease significantly while resizing. The invoice region is obtained by using the close operation with a large sized structured element to prevent font sizing decreasing problem.

In addition, scanned invoices are sometimes slightly rounded. In this case, the rotation angle of the invoice must

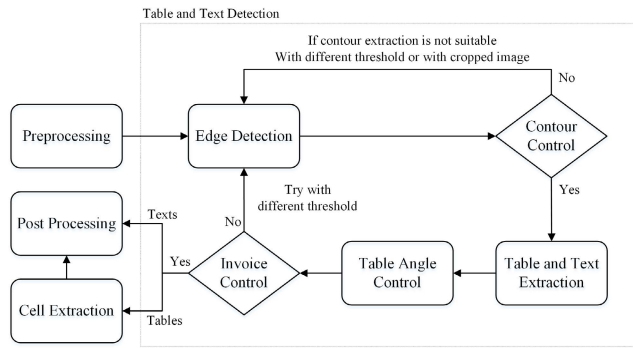


FIGURE 3. Table and text detection module sub-steps.

be determined. Generally, the detection of the image angle is performed by detecting the lines on the image, and lines are generally detected by Hough Line Transform [34]. However, this method works very slowly when the image resolution increases [35]. Since our invoices can have a large resolution, it is necessary to find invoice angle using a much faster method. In this paper, table edges were used to determine how many degrees to rotate the invoice. To do this, the largest table was determined and angle of rotation was found using lines of this table.

2) TABLE AND TEXT DETECTION WITH IMAGE PROCESSING

After preprocessing the invoice image, the next step is to find all the table and text expressions on the invoice. Some invoices may have noise and problems caused by user error such as blurring, illumination problems, and low resolution. In addition, undesirable shapes such as signatures and stamps on the invoice and undesirable situations such as pale or missing table edges are common. Therefore, it is important to select algorithms that eliminate these undesirable situations as much as possible and detect table edges in the best way.

This phase is divided into three sub-steps: edge detection, table and text extraction, and table angle control. These steps and the relationship between them are shown in Fig. 3. If insufficient contours, text, or tables are detected from the invoice as a result of table and text detection step, system returns the edge detection step to obtain clearer image.

In this section, the MG method is used to find edges. The MG is applied to grayscale images, then the obtained edge image is converted to binary image with the specified threshold value. By default, the threshold is the average intensity value of the MG operation. If the invoice processing is not good enough, the threshold value is determined according to the OTSU method and the invoice processing steps are performed again.

Since the preprocessing step is applied to the invoice, it can be assumed that the text and tables in the invoice are horizontal. Therefore, when a vertical closing operation is applied to the invoice, the text areas merge into one piece. Fig. 4 shows the results of the MG operation, edge detection, and close operation of the image with the UUID.

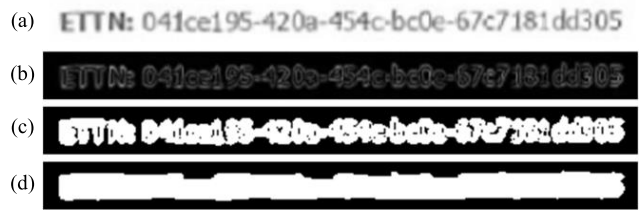


FIGURE 4. Detection of the text region on the invoice. (a) Original text region, (b) MG operation result, (c) edge detection result, (d) Close operation result.

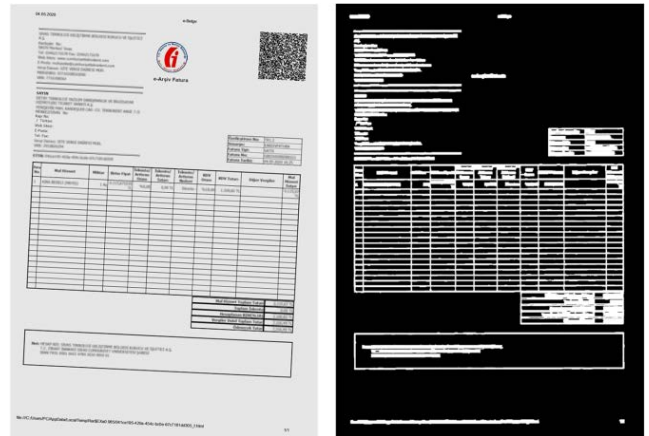


FIGURE 5. (left) Original invoice image (right) Result of edge extraction and noise removal. (Some parts of the invoices are frozen due to sensitive information.)

Since the process shown in Fig. 4 is applied to the entire invoice, all text and table contours in the invoice can be obtained. However, due to the unwanted parts on the invoice, there may still be a large number of contours that need to be eliminated. First, these contours are eliminated according to the defined rules that are given below:

- The area of the region must be of a specific size to eliminate small contours in the invoice.
- The aspect ratio, width, and height of the region must be within a certain range for it to be accepted as text or a table [17], [36].
- Within the bounding box of the image shown in Fig. 4 d, some white area is to be expected.
- The ratio of height and width between region and invoice should not exceed a certain size.
- The mean of the region and the median of the invoice are expected to be close to each other.
- If the size ratio of any contour is 85%, it can be assumed that the entire invoice consists of a single contour. In this case, the actual invoice is determined based on the detected contour coordinates and the process is restarted with edge detection.
- In cases where a sufficient number of contours cannot be obtained, it is assumed that the text and table parts are combined. In this case, the threshold value is determined by the OTSU method and the invoice processing is restarted from the edge detection step.

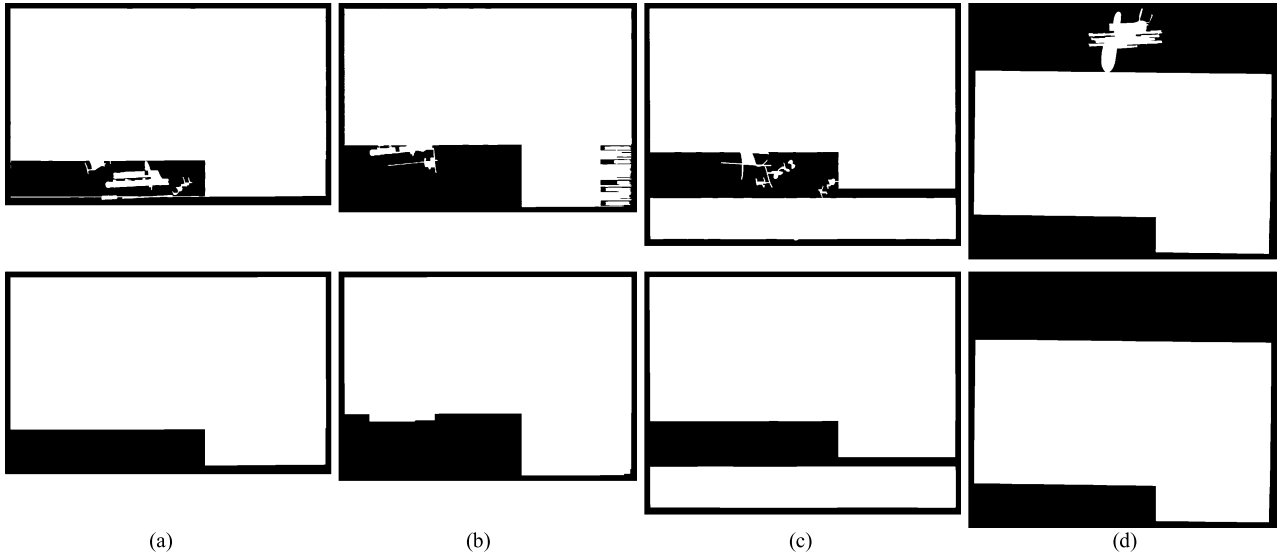


FIGURE 6. Noisy item and amount tables and their cleaned versions. a) Adjacent item and amount table with noise, b) Adjacent item and amount table with noise and without table border, c) Adjacent item, amount and note table, d) Adjacent item and amount table with signature noise.

As a result of all these processes, the invoice image given in Fig. 5 was obtained. As can be seen in Fig. 5, the edges of the table are also connected. In this way, tables can be easily extracted and processed. There are 4 tables in the sample invoice. From top to bottom, these tables are called the information table, item table, amount table, and note table. It is seen that the item and amount tables are adjacent in the sample invoice. The other white areas are processed as text regions.

The main problematic part of the invoices are the item and amount tables. Item and amount tables are usually adjacent on the invoice. These two tables must be separated in order to successfully read the tables. In addition, noises such as text, signature, and stamps that are later inserted into the invoices cause more than one table to merge or protrusion the edges of the tables. The item table and amount table was the largest contour determined on the invoice. In the further step, closing or opening processes are applied to remove noise from the table region. These processes filled the gaps and remove the noise, even if it is large. The item and amount tables and their cleaned versions are shown in Fig. 6.

Fig. 6 shows the original table images in the top row and the cleaned table images in the bottom row. An example that combine 3 tables is shown in Fig. 6 (c). Here the bottom table is stored as a separate table as it was separated after cleaning. An example where the edges of the table are not clear can be found in Fig. 6 (b). As can be seen in the figure, the gaps in the contour are filled using the close operation. As a result, both the table has been detected and the noise has been largely eliminated.

Now that the tables are cleared of noise, the item and amount tables can be separated. As can be seen in the cleaned images in Fig. 6, the adjacent tables consist of six vertices. In this paper, the Douglas-Peucker algorithm [37] is used to reduce the number of points on the contour. Under the given

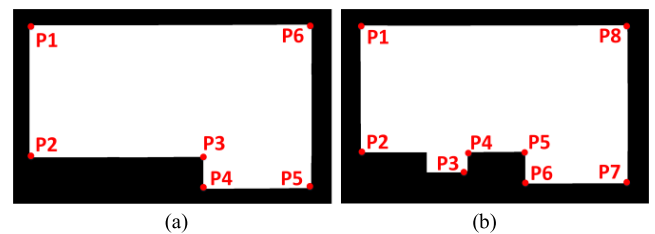


FIGURE 7. Points obtained by Douglas-Peucker algorithm. a) Adjacent tables consist of six vertices after Douglas-Peucker algorithm, b) Example case where table regions cannot be completely noise-free. As a result of the Douglas-Peucker algorithm, the number of points may be exceed six. In this case, the coordinates of the tables are estimated.

circumstances, the Douglas-Peucker algorithm reduces the number of points to six. However, in cases where the table regions cannot be completely cleared of noise, the number of points may exceed six. In this case, an attempt is made to estimate the coordinates of the table. The cases where the number of points as a result of the Douglas-Peucker algorithm is 6 and 8 are shown in Fig. 7. As can be seen in Fig.7 (b), the number of points can be more than six in cases where the noise cannot be completely eliminated.

When the number of edge points is six, the tables can be easily found. While the item table can be found with points P1, P2, P3, and P6, the amount table can be found with points P2, P3, P4, and P5. In cases where the number of points is more than six, the coordinates of the table were estimated. There are certain ratios in the width and height values of the item table and the amount table. When the invoices were examined, the width of the item table is approximately 2.6 times the width of the amount table. Similarly, the height of the item table is 2 to 7 times greater than the height of the amount table. This information can be used to calculate the start and end points of the tables. While the coordinates of the item table are calculated with the points P1, P2, P4,

P5 and P8, the coordinates of the amount table are calculated with the points P5, P6, P7 and P8.

Other contours determined as tables were preprocessed using the opening operation. After preprocessing, a point reduction was applied to the contour data using the Douglas-Peucker algorithm. Since these tables have a rectangular shape, the algorithm was expected to return four points that was used as a coordinates to extract table image. Finally, all remaining contours were stored as text.

Although, angle correction on invoices was applied in the preprocessing step, minor angle problems may occur in tables due to image distortions. Therefore, in the next step, the angle control is also performed for tables and the table is rotated according to the detected angle.

Finally, two checks were performed to understand whether or not the invoice has been processed correctly. The first one was the number of texts and tables that were recognized. Invoices usually have 4 tables and many text sections. Since our goal was to design a system that can generally handle all invoices, the minimum number of texts was set to 9 and the minimum number of tables was set to 2. The second check was about the maximum table height. Considering the general structure of the invoices, it is clear that there were not very large tables. However, in noisy invoices large contours can be gotten. In this paper, 70% of the invoice height was determined as the maximum table height. If this value was exceeded, the invoice processing was considered to be incorrect. In this case, the threshold value was determined according to the Otsu method and the invoice processing was restarted from the edge detection phase.

3) TABLE DETECTION WITH DEEP LEARNING

Table detection is difficult due to some problems such as adjacent tables, missing boundary lines and faintness. Despite these difficulties, table detection using only image processing techniques reduces the overall success of the system. For example, in cases where the item table and amount table are combined, determining table positions using only image processing methods makes the system sensitive to noise. Therefore, Deep Learning-based object detection models were used to detect the tables on the invoices. First, the tables on the invoices were labelled according to their types and the model training was performed.

In this study, the models YOLOv5 [38] and Mask R-CNN were compared with respect to the detection of tables over invoices. YOLO is a well-known deep learning model for object detection and localization tasks. Unlike other deep learning models, YOLO outputs the bounding box position and category through the neural network, increasing the final prediction speed of the model. Since 2015, researchers have evolved the YOLO model from v1 to v5 with various architectural improvements to achieve optimal performance and higher speed [39]. Due to its flexible architecture with multiple networks, lightweight pre-trained models and higher speed, we preferred YOLOv5 in our experiments.

Mask R-CNN, on the other hand, is a model based on Region-Based Convolutional Neural Network (R-CNN) that uses multiple bounding boxes over the object to classify multiple image regions into the proposed class. The unique feature of Mask R-CNN is that it outputs the object mask, which is crucial for segmentation, along with the class label and a bounding box offset. The mask information can be used to extract only the object region and not the bounding boxes.

Both models were trained with pre-labelled invoice examples containing tables with bounding boxes. The results of the experiment showed that both models achieved similar object detection accuracy, but the YOLOv5 model was preferred in the end-to-end system due to its higher speed and lower system requirements [40].

The table detection module was designed to find locations of 3 tables: Information, amount, and Item. Although the models can successfully detect the table position in the invoice image after the training process, some post-processing is required, such as angle control and gap filling, to improve the detection accuracy of the image. The angle control step is applied when a curvature is detected on the table, and the curvature is corrected. Due to Deep Learning, sometimes the object detection cannot find the exact position of the tables, and the edge lines of the tables cannot be detected. In addition, the lines may be erased due to noisy images. In both cases, the developed edge detection algorithm is applied to the images to prepare the tables for cell extraction.

4) CELL EXTRACTION

At this stage, each cell image in the tables was extracted and stored using row-column logic. In order to detect table cells, the table lines must first be obtained. It is possible to obtain the table lines using morphological open operations. The vertical kernel was used to find the horizontal lines of the table and the horizontal kernel was used to find the vertical lines of the table. The detected lines were then merged by combining the results of both kernels. However, the table lines were not always detected optimally. There may be missing edges, broken parts or noise in the table lines. Therefore, the table fix operation was applied to the table lines to remove these irregularities.

Morphological operations were applied to the detected table lines. Small broken parts in the table lines were repaired using the close operation and the opening operation was used to remove noise such as signatures, stamps, and lines. Because, the table line may become smaller after opening and closing operation, the table lines were expanded using the dilation operation and it was ensured that they fit exactly to the size of the table.

In the next phase, table edges were completed using the opposite edges, when tables have faded or missing edge lines. The table lines may be very thick after the morphological operation. Therefore, hit-and-miss transformation, which is also one of the morphological operations, was used to thin out these thick lines. The thinning process creates single-pixel

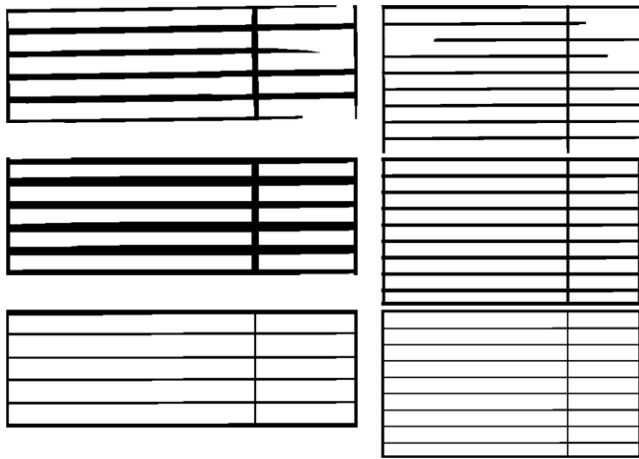


FIGURE 8. Table edge correction results.

edges. Therefore, gaps may appear at the edges of the cropped image. These gaps lead to incorrect cell detection results. To solve this problem, erosion and dilation operations were applied to the table image.

After table fix process was completed, cell extraction was performed. The results of the table fix algorithm are shown in Fig. 8 where the top row shows the original images of the table rows, the middle row shows the repaired images of the table rows and the bottom row shows the result of the thinning process. As can be seen in Fig. 8, thick lines that were obtained after morphological operations were thinned out in the last phase.

5) POST PROCESSING

After extracting the cell images from the table, some cell images may have table lines. Cleaning up the edges in the image is necessary for a successful OCR operation. By examining the edge pixels, the table lines were detected and the cell images were cropped based on the detected coordinates. After this process, the cells were extracted and stored in matrices according to the table structures.

Before the OCR process, it should be checked whether the table rows contain text or not. Trying to read cells in all rows with OCR causes a high demand on processing power. Therefore, the blank rows in the table were determined and not included in the OCR process. Peak-to-peak intensity was used to determine the empty rows and was calculated using equation (1).

$$\frac{\sum_i^n [\max(\text{cell}_i) - \min(\text{cell}_i)]}{n} > T \quad (1)$$

where n is the number of cells in a row and T is the threshold value indicating the minimum intensity value. In this study, the T value was set to 70. If the result was True, it means that there was text in the row. The operation given in equation (1) was applied to all the rows in the tables. Thus, only the filled rows were processed. In this way, both text images and cell images were extracted from the invoice. Submitting

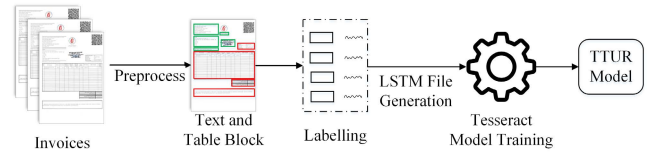


FIGURE 9. Training process of TTUR model.

the obtained images directly to the OCR module decreased the success rate because cropping operations generally separated text regions exactly. In addition, table cells can be right-aligned or left-aligned. Therefore, adding small padding to each image increased OCR success.

D. OCR MODULE

After post-processing the invoice, the images are sent to the OCR module to recognize the text they contain. Due to the low OCR performance in long texts, text blocks and table cells in the invoice were tried to be processed by dividing them into separate images. The disadvantage of this approach is that processing time increases greatly because the optical character recognition module is called frequently. For these reasons, the OCR module was designed to process incoming images in parallel to reduce runtime while maintaining the highest level of performance.

The language models included by default in the Tesseract library have been trained with more than 100 vectorial fonts such as Arial, Times New Roman, Calibri, etc. Therefore, the default models can achieve a lower error rate when vector (PDF-like) invoices are processed. However, our model should be able to process and convert scanned or photographic images that contain noise such as pepper and salt, which reduces character recognition performance. In addition, images containing special characters such as percent signs (%) and at-signs (@), where the Tesseract model has low performance, need to be better recognized. Since the data to be processed in the system are invoices with financial information, the accuracy of text prediction is crucial. Otherwise, inaccurate prediction of a fiscal value can lead to losses or confusion in financial reports. Therefore, it was necessary to optimize the optical character model to ensure the most successful recognition results.

To meet these requirements and improve the recognition of invoice text, the models should be re-trained with Turkish text and special characters. Thanks to the Tesseract library, it is possible to fine-tune the existing models using images and text in images. Training process of Tesseract requires images and annotation files, that shows text location on images, for both train and validation of model. Therefore, the texts and their positions were extracted from 957 invoices and converted to a suitable file format. Then, the Tesseract training script was run with the invoice images, annotation files, and default Turkish model. The script generated an optimal Turkish model called “TrainedTUR (TTUR)” by updating the network weights within the model to maximize

the accuracy of the validation data. The general flow of the Tesseract training process is shown in Figure 9.

E. INFORMATION EXTRACTION MODULE

The information extraction module extracts key field information from digitized text converted by the OCR module. The information in the invoices can be grouped into 4 main topics;

- General information of sender and receiver such as TIN, UUID, company name, date and number of invoice.
 - Taxpayer Identification Number (TIN): The tax identification number is the number used when paying taxes. Companies handle their tax transactions with this number.
 - Universally Unique Identifier (UUID): It is a mandatory field on e-invoices in Turkey. ETTN becomes a unique value on each invoice.
 - Company Name: Name of the company that sent the invoice.
 - Invoice Id: Invoice Id is the number used to track the Invoice.
 - Invoice Date: It includes the date and time the invoice was issued.
 - Total Amount: Indicates the total amount to be paid.
- Amount information such as payable amount, tax amount, tax ratio, discount ratio etc.,
- Sold item information like products or services,
- Notes and inconsiderable information like phone number, web address etc. if available.

The proposed system tries to convert printed text and table parts in the image of invoice documents to digital text as much as possible. However, a full text search should be performed to determine which of the texts contain the required information. Since the information required may vary from company to company, the keywords to be used to extract the information from the text must be specified by the users. Therefore, an interface for uploading templates within the web module has been developed, and users can manually edit the template keywords via the web module. Depending on the template keywords, information such as TIN, UUID, etc. were extracted by text search and returned to validation.

As a result of the OCR process, the exact correct result cannot be obtained. For example, although TIN consists of a total of 10 characters, there may be additional characters. Similarly, all other invoice fields may be incorrect as a result of the OCR process. Validation processes were integrated into the module that controls the schematic structure of special information to ensure the accuracy of the system and to correct errors. If the extracted information passes the validation, the system stores it in the database and returns it to the ERP system.

V. EXPERIMENTAL RESULT

A. PERFORMANCE METRICS

Many metrics are used in the literature to measure the accuracy of object detection applications. Intersection over

Union (IoU) is one of the most commonly used metrics in object detection applications. It is also used to calculate average precision (AP) and mean average precision (mAP). IoU is the measure of how much the predicted region and ground truth overlap and is calculated by equation (2).

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (2)$$

The value of 1 IoU indicates that the prediction and ground truth match exactly. Of course, it is very difficult to achieve a value of 1 in real life. Therefore, a threshold is set and values that exceed this value are considered successful. Here are some metrics used for an object class in the COCO dataset.

- AP: Average Precision at IoU=.50:.05:.95 (primary COCO challenge metric)
- $AP^{IoU=.5}$: Average Precision at IoU=.50
- $AP^{IoU=.75}$: Average Precision at IoU=.75

Precision is a metric that measures how accurate your predictions are. The mAP value is determined by calculating the mean AP value for all classes in the entire dataset. In this study, AP and mAP metrics are used to measure the success of table detection methods.

Measuring the success of the proposed system, it is also necessary to measure the accuracy of OCR. Character errors may occur when converting images to machine text. In this paper, character error rate (CER), normalized CER (NCER), and word error rate (WER) were calculated for OCR error [41], [42]. There were 4 situations used in the evaluation of OCR metrics. These were:

- Hit (H): Number of correct characters/words in the OCR text.
- Substitution errors (S): Number of misspelled characters/words in the OCR text.
- Deletion errors (D): Number of missing characters/words in the OCR text.
- Insertion error (I): Number of incorrectly inserted characters/words in the OCR text.

These expressions were used to calculate the CER using equation (3).

$$CER = 100 \frac{S + D + I}{N} \quad (3)$$

where N is the number of characters in the reference text. In equation (3), the CER value can exceed 100, especially as a result of addition. Therefore, if the evaluation result was to be between 0-100, the NCER, which were calculated using equation (4), value was used. The equation for the WER metric is the same as for CER. However, a word-based comparison was performed in WER.

$$NCER = 100 \frac{S + D + I}{H + S + D + I} \quad (4)$$

In addition to evaluating OCR, the success of the system in invoice processing were also measured. Precision, Recall, F-Score, and Accuracy, which are commonly used in the literature, are used as a performance metrics to evaluate the

success of the system. The Precision, Recall, F-Score, and Accuracy values were calculated using equation (5), (6), (7), and (8) respectively.

$$Precision = \frac{num_C}{num_C + num_{OCR}} \quad (5)$$

$$Recall = \frac{num_C}{num_C + num_{Err}} \quad (6)$$

$$FScore = \frac{2 \cdot P \cdot R}{P + R} \quad (7)$$

$$Accuracy = \frac{num_C}{num_C + num_{OCR} + num_{Err}} \quad (8)$$

where num_C represents the total number of correctly extracted and converted machine texts, num_{OCR} represents the total number of OCR errors and, num_{Err} represents the total number of undetected or incorrectly detected/converted fields on the invoices. In this study, the NCER value was expected to be zero for the OCR result to be considered successful. Values between 0 and 20 were classified as OCR errors. In cases where the NCER result was greater than 20, the OCR result was considered to be erroneous. Finally, character correction was applied to all identified fields in the invoice. As a result of character correction, the accuracy increases significantly. In this paper, the accuracy rate after character correction was also given as an evaluation criterion.

All experiments were performed on a computer with Intel Core i7-11800H CPU, 16 GB of memory, and NVIDIA GeForce RTX 3050 Ti graphics card. The YOLOv5 and Mask R-CNN models were trained using the default settings for each model. It is not preferred for the developed system to work with GPU in terms of cost. Therefore, the training was performed on the GPU and the tests on the CPU.

B. EXPERIMENTAL RESULT OF TABLE DETECTION

Table detection on invoices was performed using both image processing techniques and deep learning models. The comprehensive results of the table detection methods are given in this section. Table 2 shows the results of table detection using image processing techniques. Table 3 shows the result of YOLOv5 model and Table 4 shows the result of Mask R-CNN model.

As can be seen in Table 2-4, the detection accuracy of tables with Deep Learning models improved significantly. In all table classes, the mAP value was 95.14% with image processing techniques, 98.07% with YOLOv5, and 98.19% with Mask R-CNN. In the extreme case where the IoU value is 0.95, the Mask R-CNN model provides much more successful results than the other methods. However, the Mask R-CNN model requires much more memory and CPU in terms of system requirements. In addition, the YOLOv5 model detects tables in invoices in 0.27 seconds on average, while the Mask R-CNN model can detect tables in 2.5 seconds on average. For these reasons, the YOLOv5 model was preferred in the end-to-end invoice processing system.

TABLE 2. Table detection result with image processing techniques.

Table Class	AP ^{IoU=0.5}	AP ^{IoU=0.75}	AP ^{IoU=0.95}	AP
Item Table	100	100	94.82	99.19
Amount Table	98.12	97.19	83.83	95.62
Information Table	95.96	94.15	72.40	92.11
mAP	98.02	97.12	83.68	95.64

TABLE 3. Table detection result with YOLOv5.

Table Class	AP ^{IoU=0.5}	AP ^{IoU=0.75}	AP ^{IoU=0.95}	AP
Item Table	100	100	97.92	99.71
Amount Table	100	100	87.01	98.07
Information Table	100	98.42	79.64	96.44
mAP	100	99.47	88.19	98.07

TABLE 4. Table detection result with Mask R-CNN.

Table Class	AP ^{IoU=0.5}	AP ^{IoU=0.75}	AP ^{IoU=0.95}	AP
Item Table	100	100	97.92	99.71
Amount Table	100	99.20	87.19	97.83
Information Table	99.58	98.01	89.31	97.04
mAP	99.86	99.07	91.47	98.19

C. EXPERIMENTAL RESULT OF TESSERACT MODELS

Tesseract 4 offers OCR support in more than 100 languages including Turkish. The pre-trained model offered by Tesseract called 'TUR' was first used in the system for invoice processing. Although the 'TUR' model was developed for the Turkish language, it was not sufficiently accurate in invoices. As a result, a new Turkish model, which was named TTUR, was trained.

One of the models where Tesseract OCR works best is the English model that called as ENG. However, it is not possible to use the ENG model directly in Turkish invoices due to character and numeric differences. The ENG model can only be used in fields where Turkish characters are not available. These fields on the invoice are TIN, UUID, invoice number, and invoice date.

OCR results for the UUID, TIN, Invoice ID, and Invoice Date fields were obtained separately using the TUR, ENG, and TTUR models and the results were compared. There can be Turkish characters in the total amount and company name. Therefore, these areas were compared only with the TUR and TTUR models. In addition, the invoice ID, invoice date, and total amount fields are retrieved from the tables. The results for these fields are obtained by using image processing and YOLOv5 based table detection algorithms.

OCR results for the image processing-based table detection application are shown in Table 5. As shown in the Table 5, proposed TTUR model was more successful than other models. Comparing the TTUR model with other OCR models,

TABLE 5. Experimental result of OCR with image processing-based table detection. (Best values are shown in bold.)

Metrics	CER			NCER			WER		
	ENG	TUR	TTUR	ENG	TUR	TTUR	ENG	TUR	TTUR
Invoice Fields	ENG	TUR	TTUR	ENG	TUR	TTUR	ENG	TUR	TTUR
UUID	18.42	19.94	2.90	18.42	19.94	2.90	36.38	37.10	13.36
Customer TIN	4.82	4.04	2.44	4.82	4.04	2.44	8.79	9.84	6.15
Supplier TIN	9.51	8.33	6.34	9.51	8.31	6.34	11.95	11.78	7.38
Company Name	-	13.02	6.29	-	11.47	4.79	-	21.77	9.38
Invoice Id	11.69	12.72	9.07	11.67	12.69	9.06	22.32	26.57	12.48
Invoice Date	6.99	8.50	5.13	6.99	8.50	5.13	7.82	8.79	5.45
Total Amount	-	10.32	10.34	-	10.11	10.25	-	16.34	13.18
Average	10.29	10.98	6.07	10.28	10.72	5.84	17.45	18.88	9.63

TABLE 6. Experimental result of OCR with YOLOv5-based table detection. (Best values are shown in bold.)

Metrics	CER			NCER			WER		
	ENG	TUR	TTUR	ENG	TUR	TTUR	ENG	TUR	TTUR
Invoice Fields	ENG	TUR	TTUR	ENG	TUR	TTUR	ENG	TUR	TTUR
Invoice Id	9.79	10.90	7.30	9.78	10.87	7.30	19.86	26.40	10.02
Invoice Date	5.09	6.07	2.87	5.09	6.07	2.87	6.06	6.50	3.25
Total Amount	-	10.38	9.91	-	10.16	9.82	-	16.17	13.01
Average of All Fields	9.53	10.38	5.44	9.52	10.12	5.21	16.61	18.51	8.94

it is seen that generally low error rate were obtained. When the average values are examined, it is seen that there is a 5% improvement in the CER and NCER values and an 8% improvement in the WER value.

Table 6 shows the OCR results obtained using the YOLOv5-based table detection application. In Table 6, only the data related to the information retrieved from the table in the invoices are given. The average value given in the table is the average of all fields in the invoice. As seen in Table 6, a significant decrease was achieved in CER, NCER and WER values. The TTUR model with image processing-based table detection 9.07, 5.13, and 10.34 CER values are obtained in the invoice ID, invoice date, and total amount fields, respectively. The YOLOv5 model yielded CER values of 7.3, 2.87, and 9.91, respectively. As you can see, the error rates have decreased significantly.

Table 6 shows the OCR results obtained using the YOLOv5-based table detection application. In Table 6, only the data related to the information retrieved from the table in the invoices are given. The average value given in the table is the average of all fields in the invoice. As seen in Table 6, a significant decrease was achieved in CER, NCER and WER values. The TTUR model with image processing-based table detection 9.07, 5.13, and 10.34 CER values are obtained in the invoice ID, invoice date, and total amount

fields, respectively. The YOLOv5 model yielded CER values of 7.3, 2.87, and 9.91, respectively. As you can see, the error rates have decreased significantly.

The results of Precision, Recall, F-Score, Accuracy, and Accuracy with Correction metrics are shown in Table 7 and Table 8 for the complete evaluation of proposed system. Image processing based results are given in Table 7, and YOLOv5 based results are given in Table 8. For the sake of simplicity, only the results of effected fields and varying average values are given in Table 8.

The TTUR, TUR, and ENG models achieved an average accuracy rate of 89.41, 80.47, and 82.39 percent, respectively. Compared to the other models, the fine-tuned TTUR model achieved a higher accuracy rate for all invoice fields. An additional character can be detected in the OCR models, especially in the Invoice Id fields. This extra character was identified and eliminated based on the special format of the invoice ID. As a result of the character correction, the TTUR accuracy rate in the Invoice ID field was increased from 87.52% to 89.28%. As can be seen in Table 7, the average success rate increases by 1% to 3% as a result of the character correction. As a result of all these steps, the average accuracy of our system for invoice fields is 90.35% for the TTUR model, 83.08% for the TUR model, and 84.7% for the ENG model.

TABLE 7. Experimental result of proposed image processed-based application with TUR, TTUR and ENG models for scanned invoices. (Best F-Score and Accuracy values are shown in bold.)

Invoice Fields	Model	Precision	Recall	F-Score	Accuracy	Accuracy with Correction
UUID	ENG	82.27	73.72	77.77	63.62	67.60
	TUR	74.38	80.90	77.50	63.27	67.84
	TTUR	87.93	96.23	91.89	85.00	86.67
Customer TIN	ENG	96.29	94.54	95.40	91.21	91.21
	TUR	95.71	93.96	94.82	90.16	90.16
	TTUR	96.04	97.62	96.83	93.85	93.85
Supplier TIN	ENG	98.62	89.15	93.64	88.05	88.05
	TUR	97.86	89.96	93.74	88.22	88.58
	TTUR	99.43	93.11	96.17	92.62	92.62
Company Name	ENG	-	-	-	-	-
	TUR	80.65	83.54	82.07	69.60	79.26
	TTUR	91.56	93.21	92.38	85.83	88.33
Invoice Id	ENG	88.05	86.84	87.44	77.68	85.24
	TUR	89.09	85.63	87.33	77.50	79.96
	TTUR	96.14	90.71	93.35	87.52	89.28
Invoice Date	ENG	98.11	93.02	95.50	91.39	91.39
	TUR	99.42	91.34	95.21	90.86	90.86
	TTUR	99.26	94.87	97.01	94.20	94.38
Total Amount	ENG	-	-	-	-	-
	TUR	96.75	86.08	91.10	83.66	84.89
	TTUR	99.20	87.43	92.94	86.82	87.35
Average	ENG	92.67	87.45	89.95	82.39	84.70
	TUR	90.55	87.34	88.82	80.47	83.08
	TTUR	95.65	93.31	94.37	89.41	90.35

TABLE 8. Experimental result of proposed YOLOv5-based application with TUR, TTUR and ENG models for scanned invoices. (Best F-Score and Accuracy values are shown in bold.)

Invoice Fields	Model	Precision	Recall	F-Score	Accuracy	Accuracy with Correction
Invoice Id	ENG	88.72	89.24	88.98	80.14	87.35
	TUR	90.18	86.54	88.32	79.09	81.55
	TTUR	96.97	92.59	94.73	89.99	91.56
Invoice Date	ENG	97.78	94.97	96.36	92.97	92.97
	TUR	99.06	93.79	96.36	92.97	92.97
	TTUR	99.10	97.16	98.12	96.31	96.31
Total Amount	ENG	-	-	-	-	-
	TUR	97.15	85.95	91.20	83.83	84.71
	TTUR	98.80	87.92	93.04	86.99	87.52
Average of All Fields	ENG	92.74	88.32	90.43	83.20	85.44
	TUR	90.71	87.81	89.14	81.02	83.58
	TTUR	95.69	93.98	94.74	90.08	90.98

The YOLOv5-based table detection method is more accurate in detecting tables than the image processing-based method. Thus, by using the YOLOv5 model, the end-to-end

invoice processing system becomes more successful. For the image processing-based method, success rates of 89.28%, 94.38%, and 87.35% were achieved using the TTUR

TABLE 9. Invoice processing results with TUR, TTUR and ENG models for invoices in PDF file type with text.

	# invoices to send IPM	Accuracy without IPM (%)	# corrected invoices	Accuracy of IPM (%)	Accuracy after IPM (%)
TTUR	659	73.43	611	92.72	98.06
TUR	659	73.43	600	91.05	97.62
ENG	659	73.43	540	81.94	95.20

model for the invoice ID, invoice date, and total amount fields, respectively. Success rates of 91.56, 96.31, and 87.52 were obtained for the YOLOv5-based method, respectively. Although the YOLOv5 model was used for only three invoice fields, the experimental results showed that it increased the overall success of the end-to-end invoice processing system by 0.55%.

Although an attempt was made to directly process 2480 text-containing PDF invoices, information extraction cannot be performed or was incomplete due to character or format errors that may occur in some PDF files. Incomplete information was detected in 659 out of 2480 invoices. This study attempts to correct the deficiencies by sending these 659 invoices to IPM. The text PDF files were also tested using the ENG, TUR and TTUR models with the YOLOv5-based table detection application. The results obtained are shown in Table 9. According to the results obtained on a total of 659 PDF invoices, ENG, TUR and TTUR models successfully eliminated deficiencies 540, 600, and 611 invoices respectively. By applying this method to PDF files, the rate of successfully processed invoices with the TTUR model increased from 73.43% to 98.06%.

VI. CONCLUSION

In this work, it was proposed to develop an end-to-end system to transfer invoices to the electronic environment. The proposed system extracts information from the invoice and records it systematically to the ERP system. There is no criterion in the system for the type of invoice and it is submitted to the system in the desired file type (HTML, XML, PDF, and image formats). Invoices that are in text format are relatively easy to process. However, if the invoices are scanned images, the invoices have gone through 5 steps and were then used for information extraction. These steps were pre-processing, table and text detection, table cell extraction, post-processing, and OCR.

Scanned invoices are first pre-processed and made suitable for table and text detection. Text detection on invoices was performed using image processing techniques. Table detection was performed by using both image processing methods and deep learning models. In this study, YOLOv5 and Mask R-CNN models, which are object detection models, were selected. As a result of the experiments, it is seen that deep learning-based table detection is more accurate. YOLOv5 and Mask R-CNN mAP values were determined as

98.07 and 98.19. Since the success rates are similar, YOLOv5 was preferred because it is faster and requires less system requirements. After the tables in the invoice were detected, cell detection was performed in the tables. Thus, table cells as well as other text expressions on the invoice are detected. Finally, before the OCR process, all image parts are prepared for OCR by post-processing.

The open-source library Tesseract 4.0 was used for the OCR process. Our experimental result reveal that our newly trained TTUR model outperforms the TUR and ENG models provided by the Tesseract OCR. The accuracy rate has increased significantly with the TTUR model. While the average accuracy rate for scanned invoices using the TUR and ENG models were 83.08% and 84.70%, the accuracy rate increased to 90.35% using the TTUR model. In addition, this accuracy rate has increased to 90.98% with deep learning-based table detection.

Another contribution of this paper is the ability to send the desired number of invoices to the system via the web interface or by email. These invoices were queued and processed sequentially. This allows companies to send invoices in bulk or individually. The developed system processes HTML and XML invoices quickly and without errors. However, it cannot achieve 100% accuracy with vector PDF files. Invoices with missing information were sent to IPM and the missing information was completed. Our experiment showed that out of 659 invoices with missing information were corrected with ENG, TUR, and TTUR models. ENG, TUR, and TTUR models successfully corrected 495, 593, and 603 missing information respectively. As can be seen from the experimental results, the ENG model had a low accuracy rate since our dataset consists of Turkish invoices.

It is recommended that further research is needed on table and text recognition. The use of machine learning based methods, especially deep learning methods, could improve the accuracy of table and text detection. Furthermore, the development of a new model with much more invoices for OCR is being considered as a solution to the duplicate character problem in the TIN.

VII. THREATS TO VALIDITY

There are factors that threaten the results of our proposed study. The first threat comes from the database. The invoice database used was obtained from companies and contains sensitive information due to its nature. Since it is not possible

to share invoices, no method can be suggested to verify the database. In addition, the invoice structure may have minor changes from company to company. For this reason, the analysis results may differ in different companies. In order to detect tables for different companies and different documents, additional training should be considered.

ACKNOWLEDGMENT

The author appreciate their support. This study is an output of studies conducted in Detay Teknoloji I. C. Research and Development Center. The numerical calculations reported in this paper were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA Resources).

REFERENCES

- [1] T. Ritter and C. L. Pedersen, "Digitization capability and the digitalization of business models in business-to-business firms: Past, present, and future," *Ind. Marketing Manage.*, vol. 86, pp. 180–190, Apr. 2020, doi: [10.1016/j.indmarm.2019.11.019](https://doi.org/10.1016/j.indmarm.2019.11.019).
- [2] D. Doermann and K. Tombre, *Handbook of Document Image Processing and Recognition*. Accessed: May 6, 2022. [Online]. Available: <https://dl.acm.org/doi/abs/10.5555/2632841>
- [3] C. Grana, G. Serra, M. Manfredi, D. Coppi, and R. Cucchiara, "Layout analysis and content enrichment of digitized books," *Multimedia Tools Appl.*, vol. 75, no. 7, pp. 3879–3900, Apr. 2016, doi: [10.1007/s11042-014-2360-0](https://doi.org/10.1007/s11042-014-2360-0).
- [4] M. Mehri, P. Héroux, P. Gomez-Krämer, and R. Mullot, "Texture feature benchmarking and evaluation for historical document image analysis," *Int. J. Document Anal. Recognit.*, vol. 20, no. 1, pp. 1–35, 2017, doi: [10.1007/s10032-016-0278-y](https://doi.org/10.1007/s10032-016-0278-y).
- [5] R. Keefer and N. Bourbakis, "A survey on document image processing methods useful for assistive technology for the blind," *Int. J. Image Graph.*, vol. 15, no. 1, Jan. 2015, Art. no. 1550005, doi: [10.1142/S0219467815500059](https://doi.org/10.1142/S0219467815500059).
- [6] M. K. Tekleyohannes, V. Rybalkin, M. M. Ghaffar, J. A. Varela, N. Wehn, and A. Dengel, "iDocChip: A configurable hardware accelerator for an end-to-end historical document image processing," *J. Imag.*, vol. 7, no. 9, p. 175, Sep. 2021, doi: [10.3390/jimaging7090175](https://doi.org/10.3390/jimaging7090175).
- [7] S. Bhowmik, S. Kundu, and R. Sarkar, "BINYAS: A complex document layout analysis system," *Multimedia Tools Appl.*, vol. 80, no. 6, pp. 8471–8504, Mar. 2021, doi: [10.1007/s11042-020-09832-3](https://doi.org/10.1007/s11042-020-09832-3).
- [8] B. Gatos, D. Danatsas, I. Pratikakis, and S. J. Perantonis, "Automatic table detection in document images," in *Pattern Recognition and Data Mining (Lecture Notes in Computer Science)*. Berlin, Germany: Springer, 2005, pp. 609–618, doi: [10.1007/11551188_67](https://doi.org/10.1007/11551188_67).
- [9] L. Hao, L. Gao, X. Yi, and Z. Tang, "A table detection method for PDF documents based on convolutional neural networks," in *Proc. 12th IAPR Workshop Document Anal. Syst. (DAS)*, Apr. 2016, pp. 287–292, doi: [10.1109/DAS.2016.23](https://doi.org/10.1109/DAS.2016.23).
- [10] N. Islam, Z. Islam, and N. Noor, "A survey on optical character recognition system," 2017, *arXiv:1710.05703*.
- [11] B. Subedi, J. Yunusov, A. Gaybulayev, and T.-H. Kim, "Development of a low-cost industrial OCR system with an end-to-end deep learning technology," *IEMEK J. Embedded Syst. Appl.*, vol. 15, no. 2, pp. 51–60, 2020, doi: [10.14372/IEMEK.2020.15.2.51](https://doi.org/10.14372/IEMEK.2020.15.2.51).
- [12] N. Rahal, M. Tounsi, M. Benjlaiel, and A. M. Alimi, "Information extraction from Arabic and Latin scanned invoices," in *Proc. IEEE 2nd Int. Workshop Arabic Derived Script Anal. Recognit. (ASAR)*, Mar. 2018, pp. 145–150, doi: [10.1109/ASAR.2018.8480221](https://doi.org/10.1109/ASAR.2018.8480221).
- [13] M. Rusinol, T. Benkhelfallah, and V. P. dAndecy, "Field extraction from administrative documents by incremental structural templates," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1100–1104, doi: [10.1109/ICDAR.2013.223](https://doi.org/10.1109/ICDAR.2013.223).
- [14] P. Zhang, Y. Xu, Z. Cheng, S. Pu, J. Lu, L. Qiao, Y. Niu, and F. Wu, "TRIE: End-to-end text reading and information extraction for document understanding," in *Proc. 28th ACM Int. Conf. Multimedia*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1413–1422. Accessed: May 6, 2022, doi: [10.1145/3394171.3413900](https://doi.org/10.1145/3394171.3413900).
- [15] 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, doi: [10.1109/ICDAR.2019.00135](https://doi.org/10.1109/ICDAR.2019.00135).
- [16] A. Gangal, P. Kumar, and S. Kumari, "Complete scanning application using OpenCv," 2021, *arXiv:2107.03700*.
- [17] Z. E. Khattabi, Y. Tabii, and A. Benkaddour, "A new morphology-based method for text detection in image and video," *Int. J. Comput. Appl.*, vol. 103, no. 13, pp. 1–3, 2014.
- [18] T. Pratheeba, V. Kavitha, and S. R. Rajeswari, "Morphology based text detection and extraction from complex video scene," *Int. J. Eng. Technol.*, vol. 2, no. 3, pp. 200–206, 2010.
- [19] Y. Sun, X. Mao, S. Hong, W. Xu, and G. Gui, "Template matching-based method for intelligent invoice information identification," *IEEE Access*, vol. 7, pp. 28392–28401, 2019, doi: [10.1109/ACCESS.2019.2901943](https://doi.org/10.1109/ACCESS.2019.2901943).
- [20] S. Bhowmik, R. Sarkar, M. Nasipuri, and D. Doermann, "Text and non-text separation in offline document images: A survey," *Int. J. Document Anal. Recognit. (IJDR)*, vol. 21, nos. 1–2, pp. 1–20, Jun. 2018, doi: [10.1007/s10032-018-0296-z](https://doi.org/10.1007/s10032-018-0296-z).
- [21] D. Nazir, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Z. Afzal, "HybridTabNet: Towards better table detection in scanned document images," *Appl. Sci.*, vol. 11, no. 18, p. 8396, Sep. 2021, doi: [10.3390/app11188396](https://doi.org/10.3390/app11188396).
- [22] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [23] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969. Accessed: May 6, 2022. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2017/html/He_Mask_R-CNN_ICCV_2017_paper.html
- [24] D. Prasad, A. Gadpal, K. Kapadni, M. Visave, and K. Sultanpure, "CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 572–573. Accessed: May 06, 2022. [Online]. Available: https://openaccess.thecvf.com/content_CVPRW_2020/html/w34/Prasad_CascadeTabNet_An_Approach_for_End_to_End_Table_Detection_and_CVPRW_2020_paper.html
- [25] Á. Casado-García, C. Domínguez, J. Heras, E. Mata, and V. Pascual, "The benefits of close-domain fine-tuning for table detection in document images," in *Document Analysis System (Lecture Notes in Computer Science)*. Cham, Switzerland: Springer, 2020, pp. 199–215, doi: [10.1007/978-3-030-57058-3_15](https://doi.org/10.1007/978-3-030-57058-3_15).
- [26] K. A. Hashmi, M. Liwicki, D. Stricker, M. A. Afzal, M. A. Afzal, and M. Z. Afzal, "Current status and performance analysis of table recognition in document images with deep neural networks," *IEEE Access*, vol. 9, pp. 87663–87685, 2021, doi: [10.1109/ACCESS.2021.3087865](https://doi.org/10.1109/ACCESS.2021.3087865).
- [27] M. S. Satav, T. Varade, D. Kothavale, S. Thombare, and P. Lokhande, "Data extraction from invoices using computer vision," in *Proc. IEEE 15th Int. Conf. Ind. Inf. Syst. (ICIIS)*, Nov. 2020, pp. 316–320, doi: [10.1109/ICIIS51140.2020.9342722](https://doi.org/10.1109/ICIIS51140.2020.9342722).
- [28] S. Shi, C. Cui, and Y. Xiao, "An invoice recognition system using deep learning," in *Proc. Int. Conf. Intell. Comput., Autom. Syst. (ICICAS)*, Dec. 2020, pp. 416–423, doi: [10.1109/ICICAS51530.2020.00093](https://doi.org/10.1109/ICICAS51530.2020.00093).
- [29] Y. Wang, G. Gui, N. Zhao, Y. Yin, H. Huang, Y. Li, J. Wang, J. Yang, and H. Zhang, "Deep learning for optical character recognition and its application to VAT invoice recognition," in *Communications, Signal Processing, and Systems (Lecture Notes in Electrical Engineering)*, Singapore: Springer, 2020, pp. 87–95, doi: [10.1007/978-981-13-6508-9_12](https://doi.org/10.1007/978-981-13-6508-9_12).
- [30] D. Baviskar, S. Ahirrao, and K. Kotecha, "Multi-layout unstructured invoice documents dataset: A dataset for template-free invoice processing and its evaluation using AI approaches," *IEEE Access*, vol. 9, pp. 101494–101512, 2021, doi: [10.1109/ACCESS.2021.3096739](https://doi.org/10.1109/ACCESS.2021.3096739).
- [31] W. Yu, N. Lu, X. Qi, P. Gong, and R. Xiao, "PICK: Processing key information extraction from documents using improved graph learning-convolutional networks," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, 2021, pp. 4363–4370, doi: [10.1109/ICPR48806.2021.9412927](https://doi.org/10.1109/ICPR48806.2021.9412927).
- [32] (2021). *ActiveMQ Artemis*. [Online]. Available: <https://activemq.apache.org/components/artemis/>
- [33] M. Lee, *Pytesseract: Python-Tesseract is a Python Wrapper for Google's Tesseract-OCR*. Accessed: May 6, 2022. [Online]. Available: <https://github.com/madmaze/pytesseract>

- [34] J. Illingworth and J. Kittler, "A survey of the Hough transform," *Comput. Vis., Graph., Image Process.*, vol. 44, no. 1, pp. 87–116, 1988, doi: [10.1016/S0734-189X\(88\)80033-1](https://doi.org/10.1016/S0734-189X(88)80033-1).
- [35] D. Antolovic, "Review of the Hough transform method, with an implementation of the fast Hough variant for line detection," Dept. Comput. Sci., Indiana Univ., IBM Corp., Tech. Rep. TR663, 2008.
- [36] S. Lee, J. Seok, K. Min, and J. Kim, "Scene text extraction using image intensity and color information," in *Proc. Chin. Conf. Pattern Recognit.*, Nov. 2009, pp. 1–5, doi: [10.1109/CCPR.2009.5343971](https://doi.org/10.1109/CCPR.2009.5343971).
- [37] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica, Int. J. Geographic Inf. Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973, doi: [10.3138/FM57-6770-U75U-7727](https://doi.org/10.3138/FM57-6770-U75U-7727).
- [38] G. Jocher *et al.*, "Ultralytics/YOLOv5: V6.0—YOLOv5n 'nano' models, roboflow integration, TensorFlow export, OpenCV DNN support," Zenodo, Organisation Européenne pour la Recherche Nucléaire, Switzerland, 2021. [Online]. Available: https://glenn_jocher_2021_5563715, doi: [10.5281/zenodo.5563715](https://doi.org/10.5281/zenodo.5563715).
- [39] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of Yolo algorithm developments," *Proc. Comput. Sci.*, vol. 199, pp. 1066–1073, Jan. 2022, doi: [10.1016/j.procs.2022.01.135](https://doi.org/10.1016/j.procs.2022.01.135).
- [40] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
- [41] A. C. Morris, V. Maier, and P. Green, "From WER and RIL to MER and WIL: Improved evaluation measures for connected speech recognition," in *Proc. Interspeech*, Oct. 2004, pp. 2765–2768, doi: [10.21437/Interspeech.2004-668](https://doi.org/10.21437/Interspeech.2004-668).
- [42] C. Neudecker, K. Baierer, M. Gerber, C. Christian, A. Apostolos, and P. Stefan, "A survey of OCR evaluation tools and metrics," in *Proc. 6th Int. Workshop Historical Document Imag. Process.* New York, NY, USA: Association for Computing Machinery, 2021, pp. 13–18. Accessed: May 6, 2022, doi: [10.1145/3476887.3476888](https://doi.org/10.1145/3476887.3476888).



HALIL ARSLAN received the B.S., M.S., and Ph.D. degrees in electronic and computer education from Sakarya University, Turkey, in 2004, 2008, and 2016, respectively. Since 2017, he has been teaching software engineering and computer networks courses with the Computer Engineering Department as an Assistant Professor. His research interests include computer networks, cyber security, and software engineering.

...