www.revistageintec.net
ISSN: 2237-0722
revista GEINTEC
Gestão, Inovação e Tecnologias

Input Fields Recognition in Documents Using Deep Learning Techniques

Atharv Nagarikar[1]; Rahul Singh Dangi[2]; Samrit Kumar Maity[3]; Ashish Kuvelkar[4]; Sanjay Wandhekar[5]

[1]High-Performance Computing Technologies Group, Centre for Development of Advanced Computing (C-DAC) Pune, India.

[1]atharvn@cdac.in

[2]High-Performance Computing Technologies Group, Centre for Development of Advanced Computing (C-DAC) Pune, India.

[2]drahul@cdac.in

[3]High-Performance Computing Technologies Group, Centre for Development of Advanced Computing (C-DAC) Pune, India.

[3]samritm@cdac.in

[4]High-Performance Computing Technologies Group, Centre for Development of Advanced Computing (C-DAC) Pune, India.

[4]ashishk@cdac.in

[5]High-Performance Computing Technologies Group, Centre for Development of Advanced Computing (C-DAC) Pune, India.

[5]sanjayw@cdac.in

Abstract

Identification of input fields that appear on a document is a crucial requirement while digitizing any document. This paper presents a Deep Learning based approach to detect input fields from a form or document which consists of text, images and input fields like textbox, checkbox. The forms have been crawled and labelled manually to generate a dataset for training Deep Learning models. The YOLO V3 model is trained on the labelled dataset having four classes (static text, static image, input text, checkbox) with 1500 instances. We used bounding box techniques to label the dataset. The paper presents detection of limited types of input fields generally appearing on printed forms. We also discussed how such detection models can scale and sustain higher loads. If given the labelled dataset for other types of input fields, the existing YOLO V3 can be trained for them as well. The model is trained for 3500 iterations and the accuracy achieved is 71 percent.

Key-words: Deep Learning, YOLO, OCR, Forms, Document’s Input Fields.

Received: 06.07.2021 – Accepted: 05.08.2021

1. Introduction

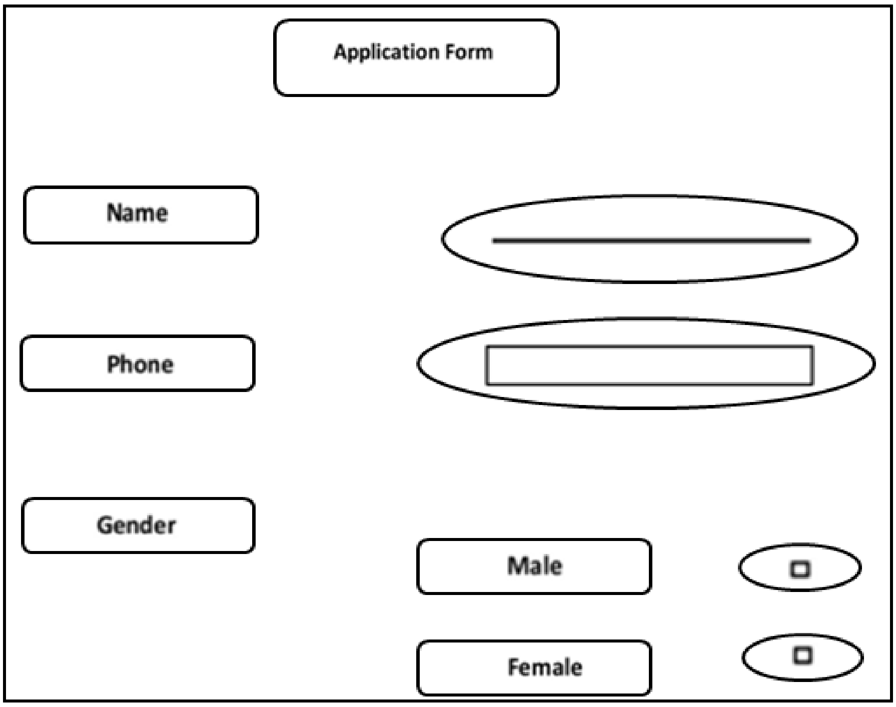
Today, artificial intelligence is a thriving field with many practical applications available and is considered a prime research domain [1]. Deep learning is an Artificial Intelligence(AI) function that imitates the workings of the human brain by processing data and identifying patterns to use in decision making [2]. Deep learning is a subset of machine learning in Artificial Intelligence which utilizes neural networks capabilities to learn from unsupervised data[3]. Unstructured or unlabeled data is categorized as unsupervised. In the early days of artificial intelligence, the field embraced and solved problems that are intellectually difficult for human beings but relatively straight-forward for computers such as problems which could be described by a list of formal, mathematical rules. The true challenge for Artificial Intelligence is to solve the tasks that are easy for humans to perform but hard to describe mathematically, problems that we solve intuitively like recognizing spoken words[4] or faces in images[5].

Extensive research efforts were employed behind various computer based techniques to recognize character or handwritten text from any documents[6][7]. Optical Character Recognition or Optical Character Reader(OCR) is a technique to recognize the text content from a given image[8]. The output of the OCR becomes input to Natural Language Processing (NLP)[9][10] engines.

Limited research attention was given to recognize various kinds of text input fields that usually appear on a document. Identification of input fields that appear on a document is a crucial requirement while digitizing any document.

Usually a printed form or a document consists of two types of fields i.e. static and dynamic fields (also called input fields) as demonstrated in Figure 1. Static fields are where users don’t have to enter any details. Existing images, text or paragraphs is an example of such a static field. Dynamic fields are where a user has to enter some data, which can be image, text, check box etc. Dynamic fields are blank spaces, space with dashed / dotted underlines or collection of square shape boxes[Figure 1]. We are supposed to write down appropriate data inside dynamic field space. The static fields can be recognized using OCR[8] or any other handwriting recognition techniques[6][7]. But there is no method or technique to recognize the dynamic/input fields. To convert any such document or application form to digitized interactive web form it is important to recognize these input fields with high accuracy. In this work, we trained a deep learning model to recognize the static and dynamic fields in a form. The model can be utilized as a standalone or as an assistive engine based on need.

Fig. 1 - Recognizing static (bounded by Rectangle) and dynamic (bounded by Oval) fields in a sample form



2. Related Work

Microsoft Sketch2code - Skecth2Code[13] is an AI application developed by Microsoft to generate HTML code from handwritten design. User uploads the image via a website, the algorithm detects written characters within the image and it follows protocol to generate HTML design. In the end, the HTML engine detects all the parameters and generates the result. It is based on a set of rules to write input HTML fields, which can be further detected and converted to HTML. It is mainly used to generate working webpage from hand drawn web page designs. The algorithm is quick and accurate but has many limitations such as it works on forms which are drawn considering the guidelines and rules prescribed by sketch2code.

Shape detection algorithms - Image processing algorithms are useful to detect boxed, lines, circular shapes. They are employed to detect dynamic fields from documents [24]. These algorithms use edge detection and area calculation techniques to determine and identify the shapes. They make use of hardcoded thresholds, consume more time and are not very efficient when deployed.

Developing the input field recognition model using image processing techniques to identify and extract input fields out of a printed document has following drawbacks:

Received: 06.07.2021 – Accepted: 05.08.2021

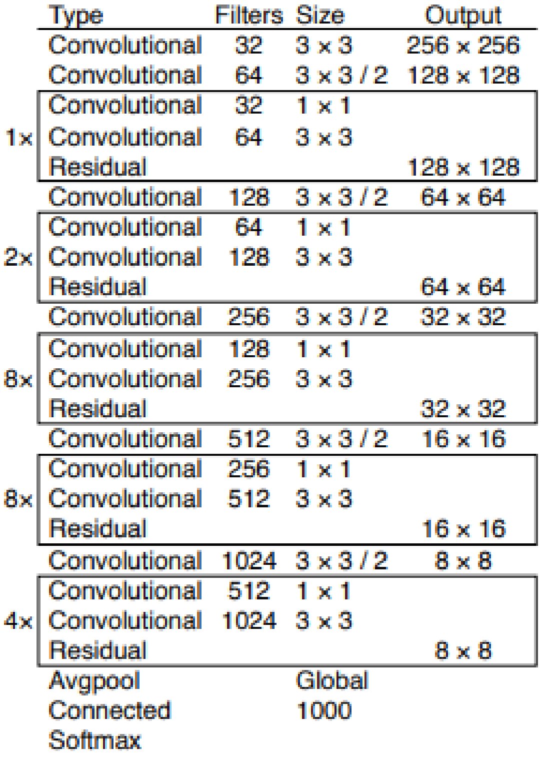
1. Different sets of rules have been used for detecting objects. These rules are hard thresholds which will run in linear time complexity for the objects present in the input image. Whereas deep learning techniques will detect objects with constant time complexity.
2. When we use threshold based rules to identify or classify objects, it may fail, when the detection threshold value is beyond limit.

3. Proposed Approach

To overcome the above challenges we used deep learning models, which are trained on a variety of forms and don’t use any rules to identify input fields.

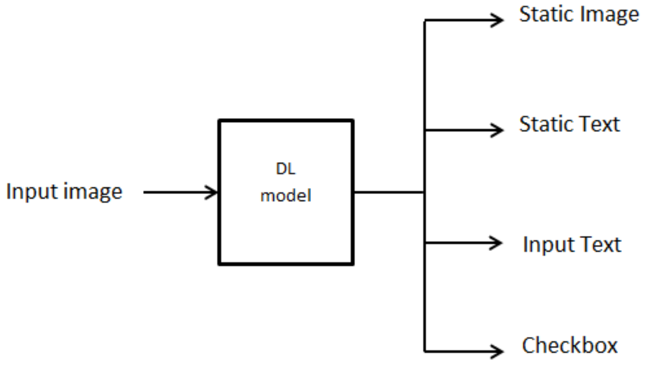
The dataset consists of distinct templates of images from governments application form, online forms, portals etc. The labeling of the dataset is done using Vott application[12]. YOLO V3 model (architecture shown in Fig. 2) is used to detect input fields from a document[11]. For training YOLO models, diversified datasets have been created by collecting various images from governmental portals [25]. The output of this model will result in recognizing and classifying static fields and types of dynamic fields which can be input text, check box, static image, static text as shown in Figure 3.

Fig. 2 - Yolo V3 Network



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig. 3 - DL Model Output Labels



Dataset Description, Labelling and Classes

The dataset for training had been crawled from the government website [25]. Various filters were applied, using OpenCV[16] to remove noise from input images. “Vott” is an open source application, used to label those images. “Vott” has a user-friendly GUI, enabling easy and effective labelling. It took us around 24-30 hours to label a dataset of 1450 image instances.

The fields on a hard copy document/form can be of static or dynamic type. Static fields don’t need any user input. Dynamic fields require users to write down data inside it.

The content in the static fields can be of 2 types: static images and static text. Static images are logos, pictures and static text [Fig. 4][Fig. 5]. Static textpart comprises existing textual content on the document like headings, title, instructions, etc. Static fields usually do not require any input by the end user.

Dynamic fields can be of 3 types: textbox, checkbox, imagebox.

The text box corresponds to the fields where users have to write textual details. The checkboxes are square shaped small boxes which the user has to select by ticking or marking on one or many of them. The image box is where the user has to paste an image. We have not used the Input Imagebox category in the dynamic labels for simplicity purpose.

In the process of converting an existing hardcopy document to digital form, the neural model has to recognize static and dynamic fields distinctly. This is more challenging because of the fact that, on a digital equivalent of an input document, dynamic fields become interactive input fields, where a user would be able to insert appropriate data (image, text) or would be able to select the dynamic field (checkboxes, radio buttons).

As of now we trained the model for limited types of fields ie., static text, static image, dynamic text box and dynamic checkbox.

Received: 06.07.2021 – Accepted: 05.08.2021

Deep Learning Model used for Training

YOLO V3 is an improvement over previous YOLO detection networks. Compared to prior versions, it features multi-scale detection, stronger feature extractor network, and optimised loss function. This network can detect many more targets from big to small. Like other single-shot detectors, YOLO V3 also runs quite fast and makes real-time inference possible on GPU devices.

The output of our model will create a bounding box across the static part as seen in the [Fig. 4] [Fig. 5].

Inference Pipeline

The inference pipeline is first the input image of the form is resized and then passed via filters using OpenCV[14][15] library in order to remove noise. Then the YOLO v3[11] model is used to detect objects, which has been trained on various application templates such as government application forms, online google forms, etc. The model recognised and classified the content from the input image into 4 types viz. static image, static text, input text, and checkbox.

4. Experiment Setup

Training Details and Hyperparameters

This section explains the dataset creation, labeling and training hyperparameters.

Dataset - We crawled the dataset for Indian government forms from Indian government website [25] and labelled it manually. The dataset consisted of 950 instances of forms. Below is the category wise detail of type of fields appears on the input dataset:

- Static Image - 50 labels
- Static Text - 350 labels
- Textbox - 350 labels
- Checkbox - 200 labels

We also generated a handwritten dataset and labelled it manually. It has 500 images and following are number of labels:

- Static Image - 30 labels
- Static Text - 200 labels
- Textbox - 200 labels
- Checkbox - 70 labels

Received: 06.07.2021 – Accepted: 05.08.2021

Training hardware details - To train our model we used C-DAC's PARAM Shavak Deep Learning GPU System[17][18]. The system has dual socket 24 core, each x86_64 based Intel Xeon CPUs with 2.60GHz GHz frequency. It has an Nvidia Quadro P5000 GPU accelerator card. It has 64 GB of RAM and 8 TB of secondary storage. In dedicated mode, it took 2 hour as overall neural network model training time.

Hyperparameters - Hyperparameter selection as shown in Table 1], is based on multiple experiments performed over GPU using TensorFlow [19] framework and plotting the results in Tensorboard [20].

Table 1 - Hyperparameter Table

Hyperparameters	**Value**
Training data size	1450
Testing data size	290
Training batch size	32
Iterations per epochs	35
Total epochs	100

Procedure

The noise from crawled dataset is removed via OpenCV filters. The Dataset is then labelled using Vott application[12]. The Vott application provides tf records of given input images which consist of coordinates and label of an image. The YOLO Model is then trained with these tf record data as input. After training the YOLO model, we selected some random images in order to test the model and the result of the model is shown in Fig. 4. The same YOLO model is again re-trained from hand drawn forms and results of it are shown in Fig. 5.

5. Results and Inference

We provide various forms to recognize labels for static text, textbox and checkbox images as shown in Fig 4. We also trained the model to detect handwritten text and hand drawn input fields and the model detects most of the fields correctly as shown in Fig 5. Fig 6 shows a graph for 'Loss VS epoch' and it implies reduction in loss while training a YOLO model. Post training completion we achieved 0.67 as the Mean Average Precision (MAP) on the test dataset.

As seen and inferred from results that the YOLO V3 model trained on forms dataset works on handwritten data as well. The same model can be retrained with custom handwritten forms also in order to improve results. Though we don't claim that this model is best for document's input field detection as we have not demonstrated any comparison of the models. More attempts are required to try other similar object detection models in order to compare the results. However, it is to be noted that detection and recognition of document's fields using object detection models and automating this step in generating HTML forms using Artificial Intelligence is a new and innovative technique.

Fig. 4 - Sample Detection Results on Computer Generated Forms from our DL Mode (Figure on the Left is an Image Provided to our Model and Figure on Right is an Image from the Model which Detect Field such as Input Text, Static Text and checkbox)

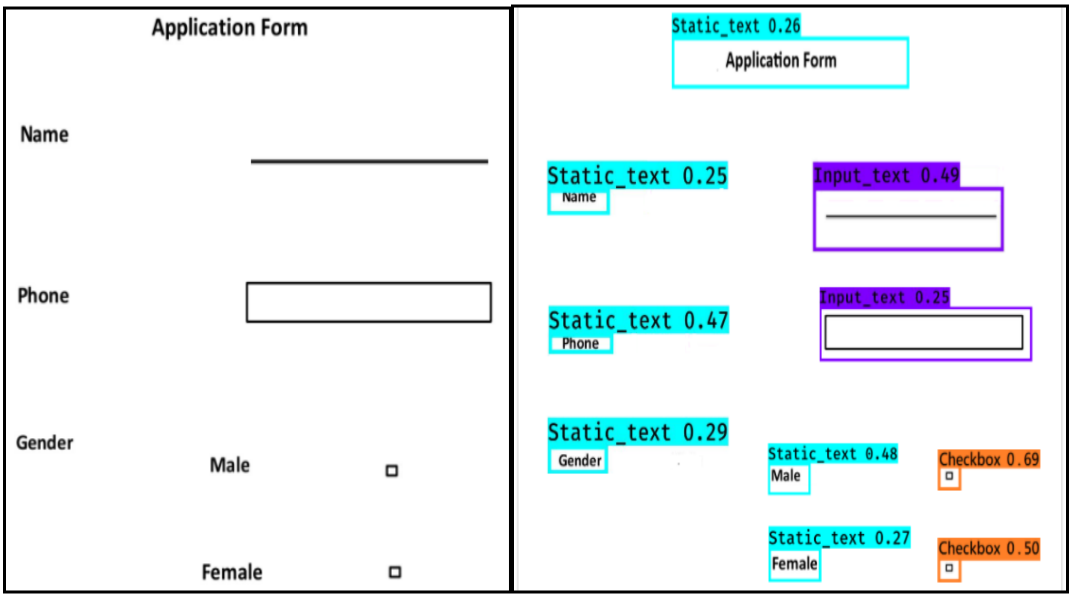


Fig. 5 - Sample Detection Results on Hand Written Forms from our DL Model (An Image on the Left Side an Input Image to our Model and an Image on Side Shows Output of our Model which Detects Field such as Input Text and Static Text)

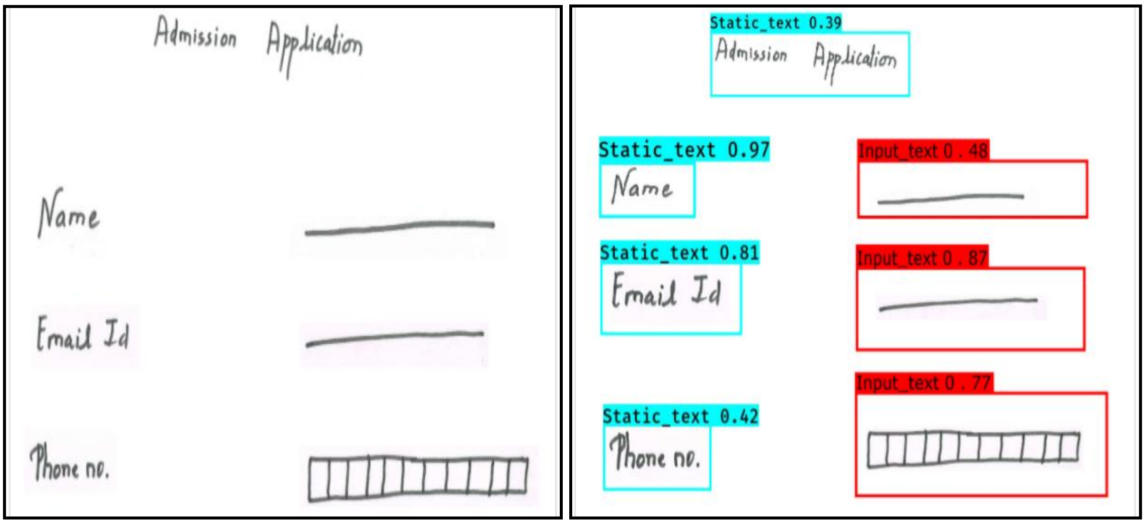
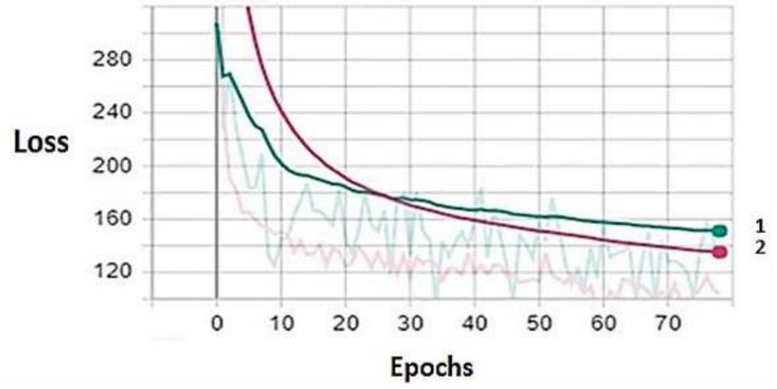


Fig. 6 - Training versus validation loss. #1 is train loss and #2 is validation loss



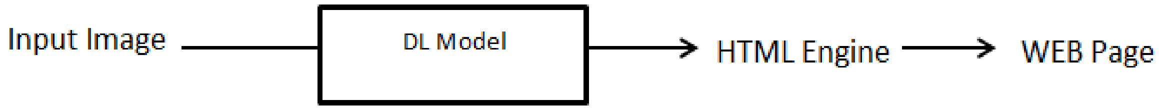
6. Conclusion and Future Work

In this paper we presented a technique to detect static and dynamic input fields from an image of a document. We performed an experiment, considering limited type input fields but given the dataset the same model can be trained for other fields.

In future we plan to use our work in generating HTML codes [Fig.6] based on the type of input field detected. This work is beneficial while generating dynamic interactive web pages from and hardcopy input form.

In this experiment we did not perform a comparative study of performances of multiple object detection models while detecting and identifying dynamic fields on a document. We plan to perform detail study on this front to decide upon best suitable model for such work and integrated them in our dynamic webform generation pipeline as show picture (Figure- 7).

Fig. 7 - Pipeline to Demonstrate the use of this Model for Creating Dynamic Web Pages



References

J. Estevez, G. Garate and M. Graña, "Gentle Introduction to Artificial Intelligence for High-School Students Using Scratch," *In IEEE Access*, 7, 179027-179036, 2019. doi:10.1109/ACCESS.2019.2956136

F.Q. Lauzon, "An introduction to deep learning," *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, Montreal, QC, Canada, 2012, 1438-1439. doi: 10.1109/ISSPA.2012.6310529

M. Girolami, A. Cichocki and S.I. Amari, "A common neural-network model for unsupervised exploratory data analysis and independent component analysis," *In IEEE Transactions on Neural Networks,* 9(6), 1495-1501, Nov. 1998, doi: 10.1109/72.728398.

J. Meng, J. Zhang and H. Zhao, "Overview of the Speech Recognition Technology," *2012 Fourth International Conference on Computational and Information Sciences, Chongqing, China,* 2012, 199-202, doi: 10.1109/ICCIS.2012.202.

X. Han and Q. Du, "Research on face recognition based on deep learning," *2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications (DINWC),* Beirut, Lebanon, 2018, pp. 53-58, doi: 10.1109/DINWC.2018.8356995.

P. Parvathi and T. S. Jyothis, "Identifying Relevant Text from Text Document Using Deep Learning," *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), Kottayam,* India, 2018, 1-4. doi: 10.1109/ICCSDET.2018.8821192

A. Yuan, G. Bai, P. Yang, Y. Guo and X. Zhao, "Handwritten English Word Recognition Based on Convolutional Neural Networks," *2012 International Conference on Frontiers in Handwriting Recognition, Bari, Italy,* 207-212, doi: 10.1109/ICFHR.2012.210.

Selva Kumari, R.Shantha & Sangeetha, R.. (2015). Optical character recognition for document and newspaper. *International Journal of Applied Engineering Research, 10,* 15279-15285.

Q. Ma, "Natural language processing with neural networks," *Language Engineering Conference, 2002. Proceedings, Hyderabad, India,* 2002, 45-56, doi:10.1109/LEC.2002.1182290.

A. Gelbukh, "Natural language processing," Fifth International Conference on Hybrid Intelligent Systems (HIS'05), Rio de Janeiro, Brazil, 2005, 1 pp. doi:10.1109/ICHIS.2005.79.

Redmon, Joseph & Farhadi, Ali. (2018). *YOLOv3: An Incremental Improvement.*

https://github.com/microsoft/VoTT

Robinson, Alex. (2019). *Sketch2code: Generating a website from a paper mockup.*

W. Hongzhi, L. Meijing and Z. Liwei, "The distortion correction of large view wide-angle lens for image mosaic based on OpenCV," *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), Jilin, China,* 2011, 1074-1077, doi: 10.1109/MEC.2011.6025652

S. Guennouni, A. Ahaitouf and A. Mansouri, "Multiple object detection using OpenCV on an embedded platform," *2014 Third IEEE International Colloquium in Information Science and Technology (CIST), Tetouan, Morocco,* 2014, 374-377, doi: 10.1109/CIST.2014.7016649.

N. Habibunnisha, K. Sivamani, R. Seetharaman and D. Nedumaran, "Reduction of Noises from Degraded Document Images Using Image Enhancement Techniques," *2019 Third International Conference on Inventive Systems and Control (ICISC),* Coimbatore, India, 2019, 522-525. doi: 10.1109/ICISC44355.2019.9036418.

S. Agrawal, S. Das, M. Valmiki, S. Wandhekar and R. Moona, "A Case for PARAM Shavak: Ready-to-Use and Affordable Supercomputing Solution," *2017 International Conference on High Performance Computing & Simulation (HPCS),* Genoa, 2017, pp. 396-401

https://www.cdac.in/index.aspx?id=lu_param_shavakDL_launch

F. Ertam and G. Aydın, "Data classification with deep learning using Tensorflow," *2017 International Conference on Computer Science and Engineering (UBMK), Antalya, Turkey,* 2017, 755-758, doi: 10.1109/UBMK.2017.8093521.

Received: 06.07.2021 – Accepted: 05.08.2021

H. Ben Braiek and F. Khomh, "TFCheck: A TensorFlow Library for Detecting Training Issues in Neural Network Programs," *2019 IEEE 19th International Conference on Software Quality, Reliability and Security (QRS), Sofia, Bulgaria*, 2019, 426-433, doi: 10.1109/QRS.2019.00059.

Beigi, Homayoon. (1997). *An Overview of Handwriting Recognition.*

S. Arif and F. Shafait, "Table Detection in Document Images using Foreground and Background Features," 2018 *Digital Image Computing: Techniques and Applications (DICTA), Canberra, ACT, Australia,* 2018, pp. 1-8, doi: 10.1109/DICTA.2018.8615795.

T. Lu and A. Dooms, "A Deep Transfer Learning Approach to Document Image Quality Assessment," *2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, NSW,* Australia, 2019, pp. 1372-1377, doi: 10.1109/ICDAR.2019.00221.

Kumar, Vivek & Pandey, Sumit & Pal, Amrindra & Sharma, Sandeep. (2016). *Edge Detection Based Shape Identification.* https://www.india.gov.in/my-government/forms

Received: 06.07.2021 – Accepted: 05.08.2021