Summer Internship

Real Time Text Recognition from Instrument Display

Ashish Kumar Gupta
SC19B135 [SAC ID : RS00472]
B.Tech
Department of Avionics
Indian Institute of Space science and Technology
Thiruvananthapuram, Kerala

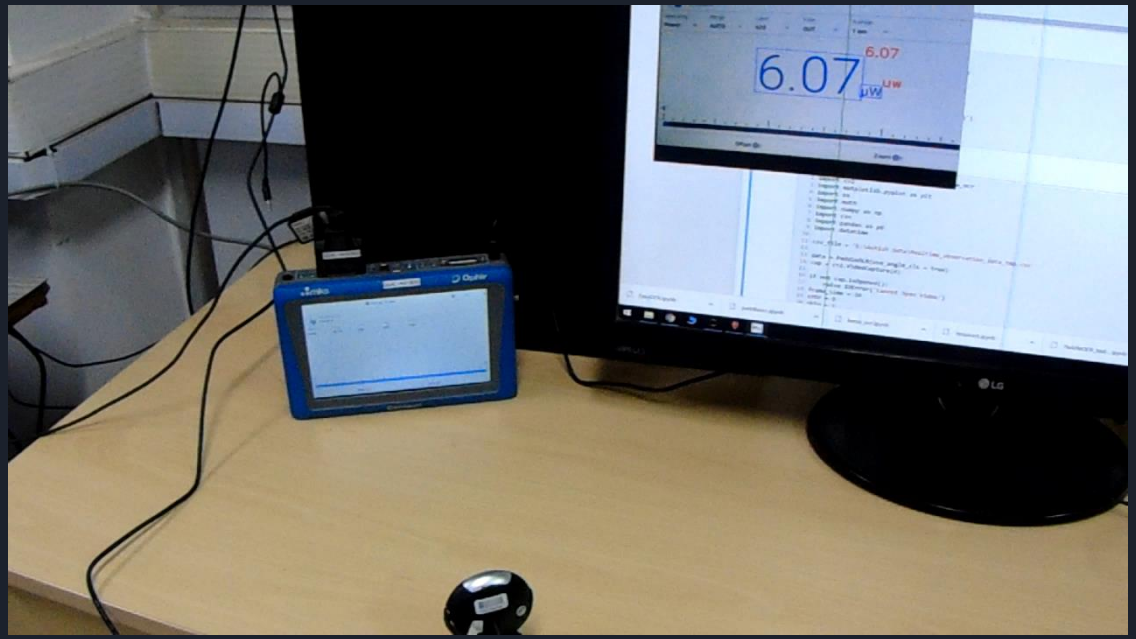
Shri Saket Chawla
Scientist/Engineer – SE
Space Applications Centre, Ahmedabad
Indian Space Research Organisation

Dr. Deepak Mishra
Professor & Head
Department of Avionics
Indian Institute of Space science and Technology
Thiruvananthapuram, Kerala

Content

Objective

Real-time Text Detection and Recognition from display devices



Applications

- Automation of data collection in scientific experiments
- Vehicle Number Plate Recognition
- Assisted-KYC

Text Detection and Recognition

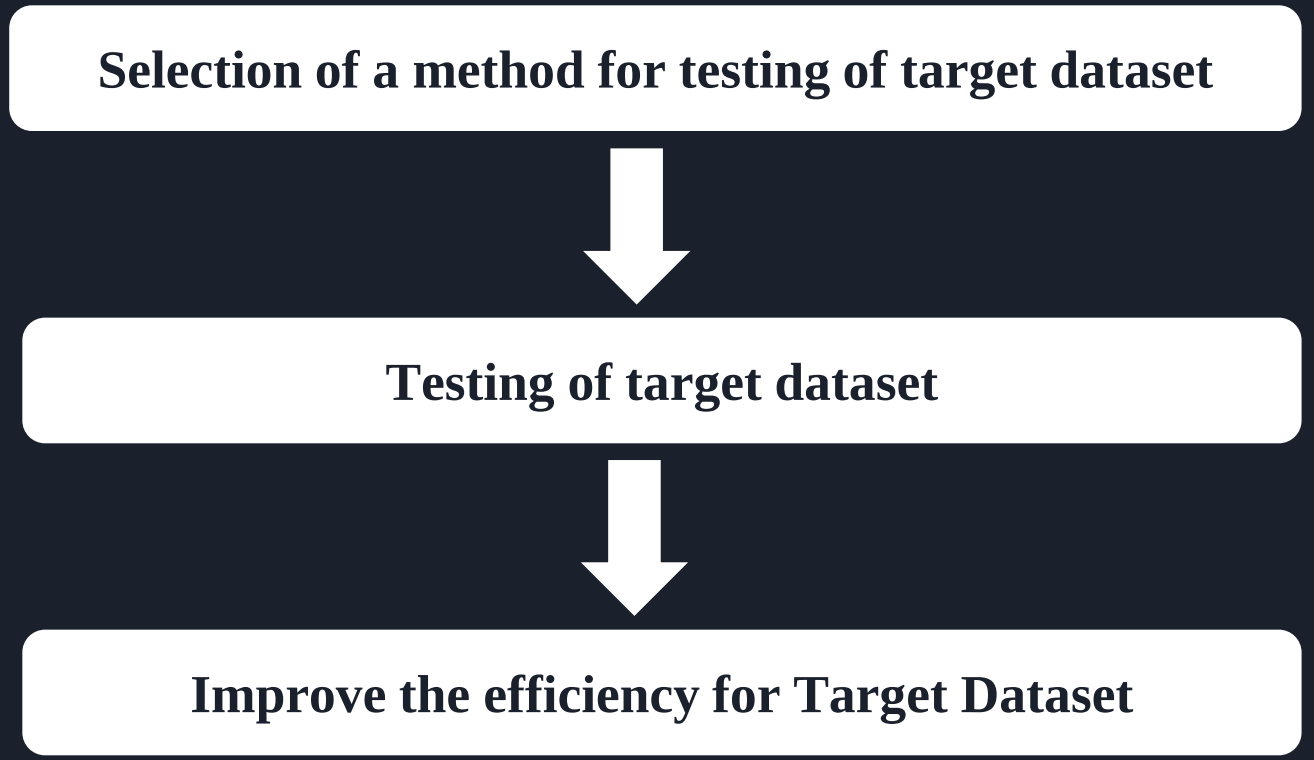
Text Detection → 2.460 → Text Recognition → 2.460

Optical Character Recognition (OCR) :

Printed text ⟶ Machine-encoded text.

Y. Du, C. Li, R. Guo, X. Yin, W. Liu, J. Zhou, Y. Bai, Z. Yu, Y. Yang, Q. Dang *et al.*,"Pp-ocr: A practical ultra lightweight ocr system," *arXiv preprint arXiv:2009.09941*,2020

Work Procedure



STEP : 1

Select one of the following methods to test target dataset.

OCR methods

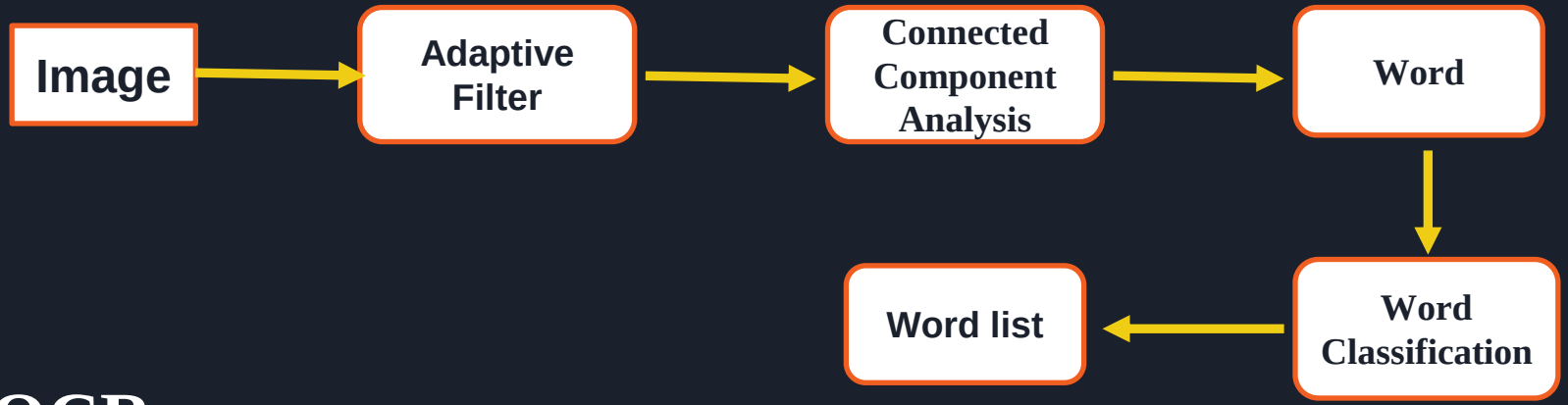
1. Tesseract [2]
2. EasyOCR [3]
3. Paddle OCR [4]
4. Keras-OCR [5]

Types of Images

1. Sharp
2. Blur
3. Tilted
4. Tilted Blur
5. Haze

“Airport timetable hi-res stock photography and images - alamy,” https://www.alamy.com/stock-photo/airport-timetable.html,

Tesseract



EasyOCR

Craft Algorithm

ResNet
LSTM

Implementation of Methods on sample dataset

- The code can take input as images, video , Google Drive link , url link of video, images or Image Folder

Observation

Sharp Images

Methods	Total Images	Execution Time	Text Detection	Text recognition	Detection %	Recognition %
Tesseract	41	0:01:48	13	1	31.71	2.44
EasyOCR	41	0:06:40	27	6	65.85	14.63
PaddleOCR	41	0:03:37	32	24	78.05	58.54
Keras-OCR	41	00:17:56	34	3	82.93	7.32

Tilted Images

Method	Total	Execution Time	Text Detection	Text recognition	Detection %	Recognition %
Tesseract	26	0:01:27	5	0	19.23	0.00
EasyOCR	26	0:04:33	17	3	65.38	11.54
PaddleOCR	26	0:02:55	15	8	57.69	30.77
Keras-OCR	26	0:14:13	19	1	73.08	3.85

1. Tesseract is taking least execution time while Keras-OCR is taking most execution time.
2. Keras-OCR has high Text Detection efficiency.
3. Tesseract fails in Text Recognition in tilted images.
4. Paddle OCR has high Text Recognition efficiency.

Observation

Blur Images

Method	Total	Execution Time	Text Detection	Text recognition	Detection %	Recognition %
Tesseract	73	0:02:41	9	0	12.33	0.00
EasyOCR	73	0:12:01	36	1	49.32	1.37
PaddleOCR	73	0:05:09	40	26	54.79	35.62
Keras-OCR	73	0:37:44	37	2	50.68	2.74

Tilted Blur Images

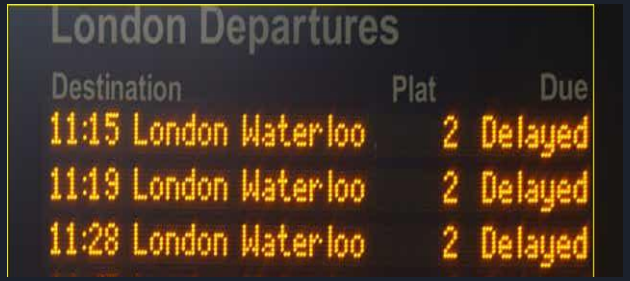
Method	Total	Execution Time	Text Detection	Text recognition	Detection %	Recognition %
Tesseract	31	0:01:17	1	0	3.23	0.00
EasyOCR	31	0:05:50	13	0	41.94	0.00
PaddleOCR	31	0:02:28	15	9	48.39	29.03
Keras-OCR	31	0:15:01	9	1	29.03	3.23

1. Tesseract is taking least execution time while Keras-OCR is taking most execution.
2. Tesseract and EasyOCR fail in Text Recognition of tilted Blur Images.
3. Paddle OCR has high Text Detection and Recognition efficiency.

Sharp Image

- Tesseract failed in detection.
- EasyOCR failed in Special character Recognition.
- KerasOCR performed well in Detection but failed in recognition

Tesseract



EasyOCR

PaddleOCR

Keras-OCR

Tilted Image

- EasyOCR and KerasOCR both failed in special Character Recognition.

Tesseract

EasyOCR

Paddle OCR

Keras-OCR

Observation

Method	Total	Execution Time	Text Detection	Text recognition	Detection %	Recognition %
Tesseract	140	0:05:56	27	1	19.29	0.71
EasyOCR	140	0:23:14	80	10	57.14	7.14
PaddleOCR	140	0:11:41	87	58	62.14	41.43
Keras-OCR	140	1:09:53	90	6	64.29	4.29

1. Tesseract is taking least execution Time while Keras -OCR is taking most execution time.
2. Keras-OCR have high text Detection efficiency.
3. Paddle OCR have high text recognition efficiency.

Conclusion

Paddle OCR is the most suitable method for testing of target dataset.

STEP : 2

Testing of PaddleOCR on Target Dataset

Target Data : Video of instrument shown on right.

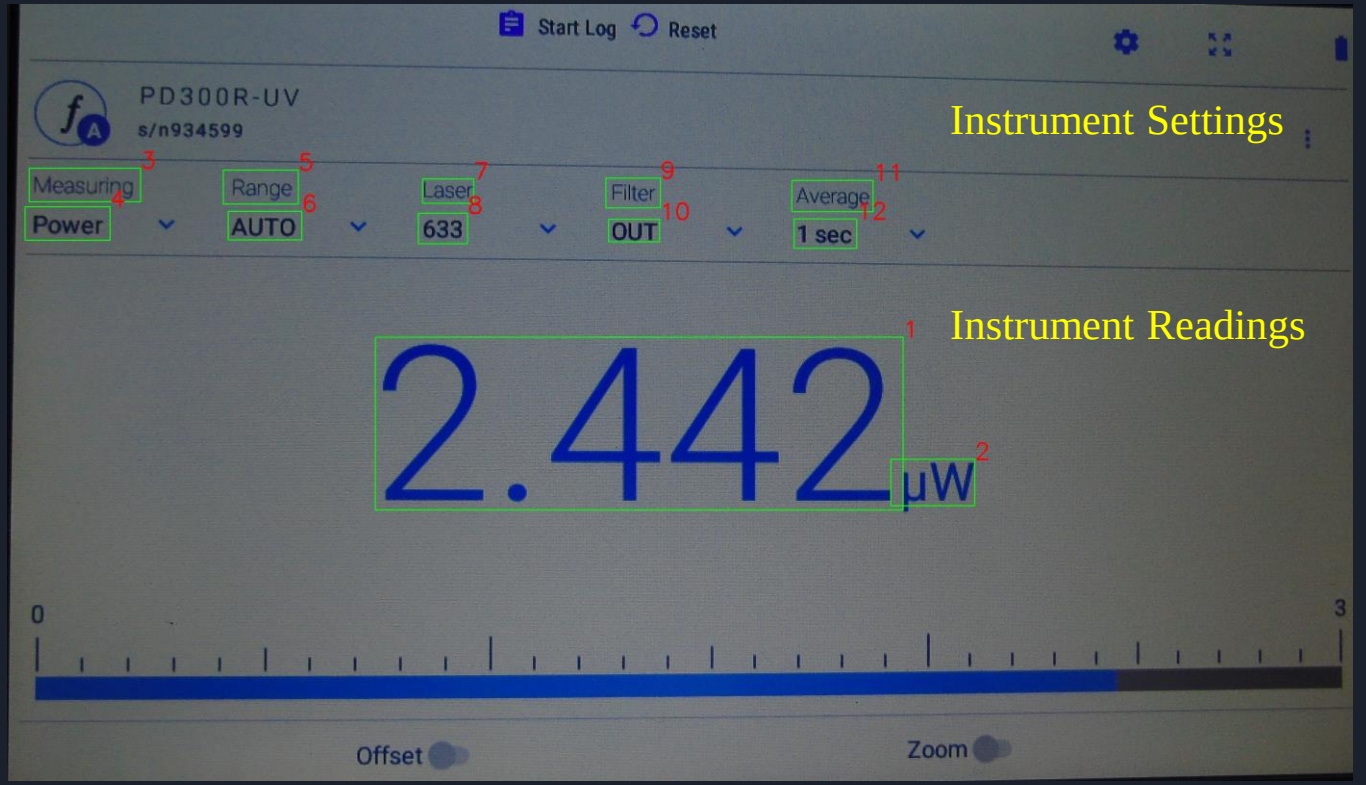
I have collected sample data of Target system for Statistical evaluation.

"Paddlepaddle/paddleocr: Awesome multilingual ocr toolkits based on paddlepaddle." https://github.com/PaddlePaddle/PaddleOCR,

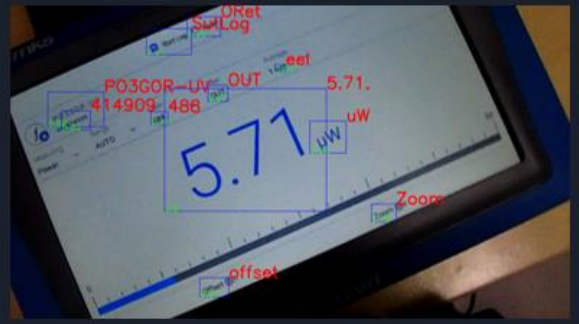
Target Data

Video Condition:

- Sharp
- Tilted
- Blur
- Haze
- Cropped



Problems in Recognizing target data



Observation

In the videos of target data, from duration of approx. 430 sec, I pick up one frame after 5 seconds for evaluation

Parameters		1	2	3	4	5	6	7	8	9	10	11	12
Total Sample		81	81	81	81	81	81	81	81	81	81	81	81
Detection		75	66	39	52	51	63	42	59	47	57	47	54
Detection Efficiency		0.93	0.81	0.48	0.64	0.63	0.78	0.52	0.73	0.58	0.70	0.58	0.67
True Recognition	Number	74	58	22	50	33	60	25	56	4	47	38	51
	Min confidence	0.62	0.51	0.69	0.82	0.71	0.83	0.81	0.68	0.79	0.6	0.79	0.53
	Max confidence	1	0.96	0.99	1	0.99	1	0.99	1	0.87	1	0.99	0.99
	Mean confidence	0.91	0.7	0.95	0.93	0.93	0.92	0.97	0.93	0.84	0.93	0.92	0.94
False Recognition	Number	1	8	17	2	18	3	17	3	43	10	9	3
	Min confidence	0.83	0.51	0.5	0.62	0.52	0.6	0.51	0.68	0.53	0.51	0.56	0.62
	Max confidence	0.83	0.88	0.93	0.65	0.93	0.98	0.97	0.83	0.98	0.78	0.83	0.82
	Mean confidence	0.83	0.66	0.62	0.64	0.68	0.83	0.67	0.76	0.78	0.62	0.8	0.71
Recognition Efficiency		**0.91**	**0.72**	**0.27**	**0.62**	**0.41**	**0.74**	**0.31**	**0.69**	**0.05**	**0.58**	**0.47**	**0.63**

Conclusion

To completely automate data gathering from instrument, we will try to achieve 100% efficiency for instrument reading and measurement unit

STEP : 3

Improve the efficiency for Target Dataset

Improving efficiency :

- Software Implementation
- Train the model for Target Dataset

1. Software Implementation

By Evaluating based on conditions implementation in Software.

Condition Evaluated on a frame for accepting the result

- If Number of Detection boxes $\geq$ 2.
- Identification of Numerical value and units of Instrument reading.
- Based on evaluation of sample target dataset. Identify the Threshold of confidence level, which is used for box detection.

1. Number of Detection boxes ≥ 2

Number Detection boxes : 17

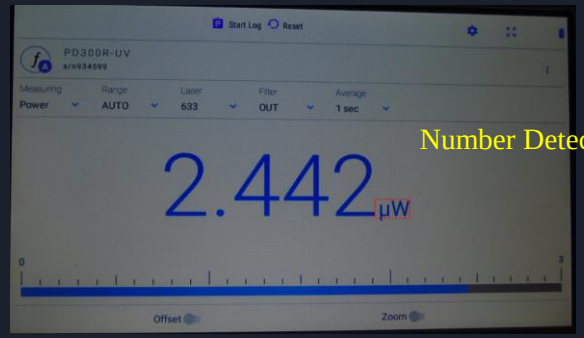
Accepted

Number Detection boxes : 11

Accepted

Number Detection boxes : 2

Accepted



Number Detection boxes : 1

Rejected

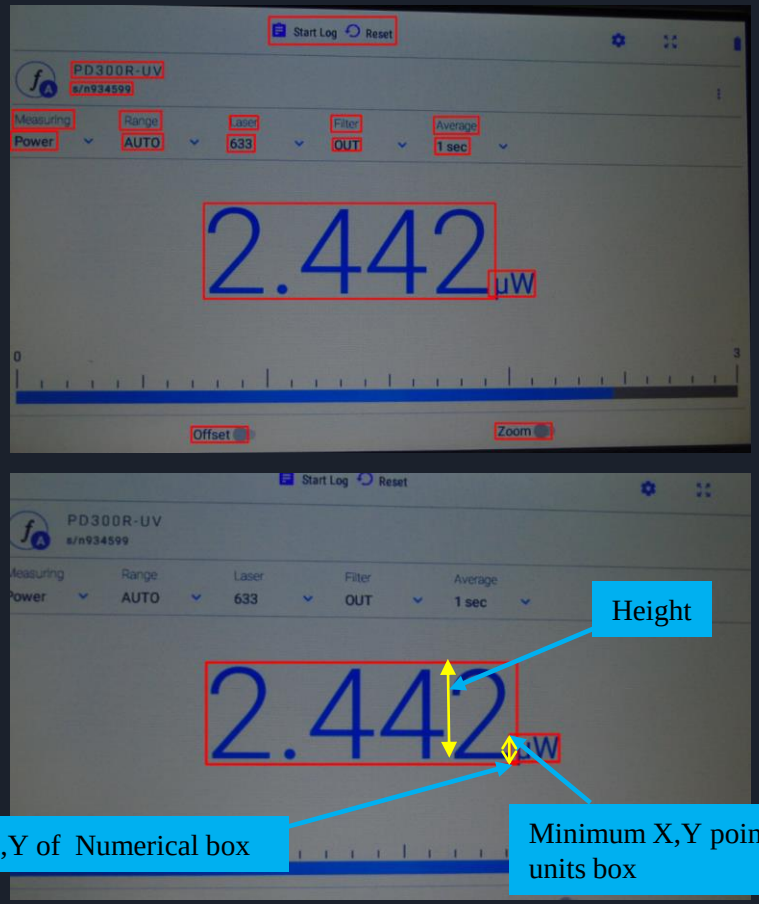
2. Identification of Numerical value and units of Instrument reading.

Identification of box for numerical value

- The bounding box with maximum area have numerical value if its area is at least 5 time greater than others.
- Bounding box should contain float.

Identification of box for units

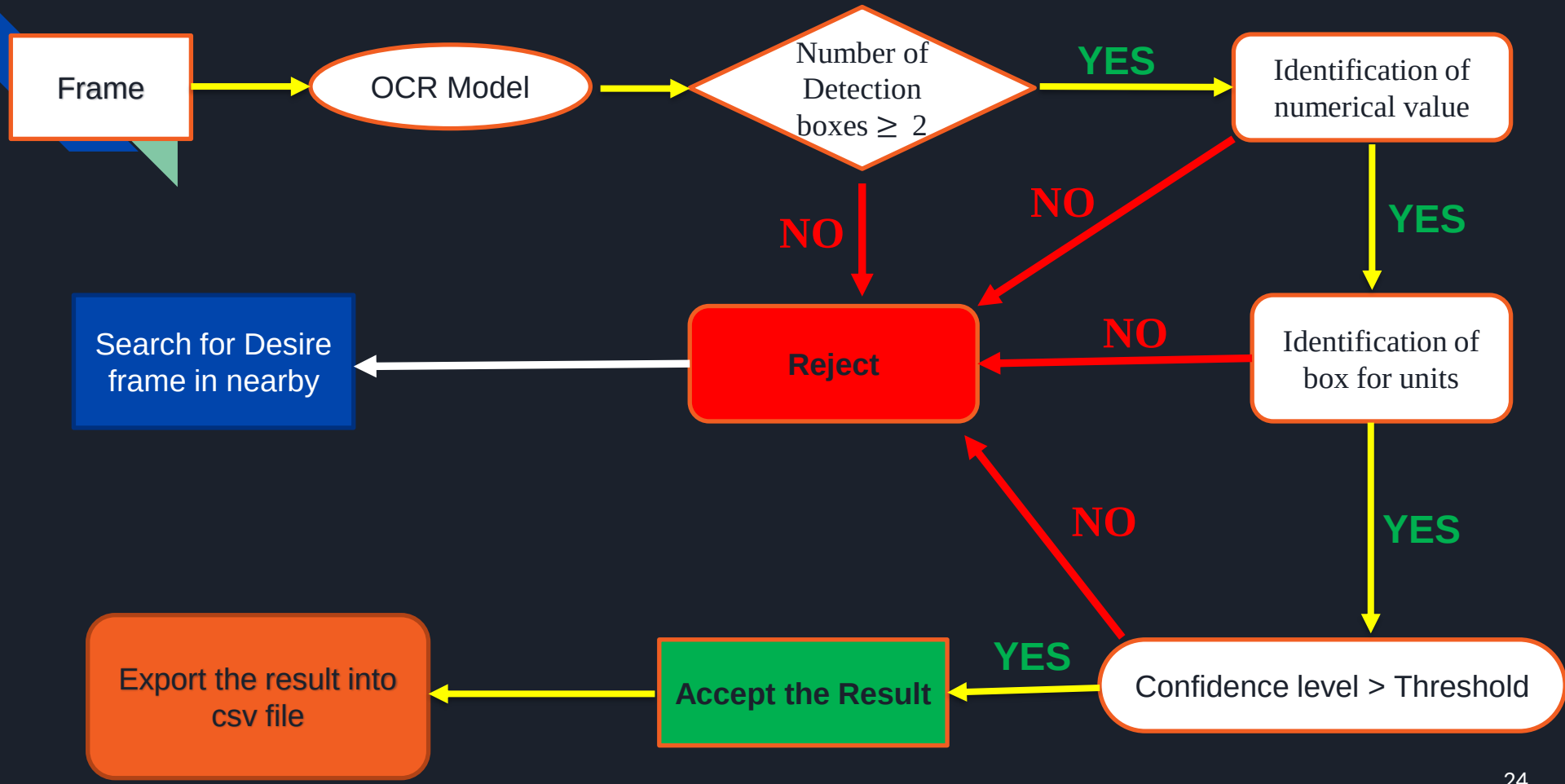
- If distance between maximum point of numerical box and minimum points of nearby box of numerical box is less than half of width.



3. Based on evaluation of sample target dataset, a threshold for confidence level over instrument reading and unit was identified.

- After identifying the text-box for instrument reading and unit, if the confidence level of instrument reading is greater than their threshold, Then frame is accepted.

If a frame is rejected, we search for a desired frame with high confidence level on all parameters in nearby frame.


Frame
OCR Model
Number of Detection boxes ≥ 2
YES
Identification of numerical value
NO
NO
YES
Search for Desire frame in nearby
Reject
NO
Identification of box for units
NO
YES
Export the result into csv file
Accept the Result
YES
Confidence level > Threshold

Algorithm 5.0: Condition for Correct Output

```
Input:Image Frame;
Output : Caution( Condition satisfied or not);
Output = PaddleOCR(Image Frame);
Para1 = Output which contain highest area Para2 = Nearest neighbourhood of
  Para1;
if length(Output) ≥ 2  distance(para1,para2)≤ height(para1)/2 then
    if Confidence(each parameter of Output) ≥ Threshold then
        Caution = True
    else
        Caution = False
    end
else
    Caution = False
end
```

Algorithm 5.0: Real-Time Text Recognition from Display Screen

```
Input:Frame of video;
Output : Detected Text with confidence level;
frame No = 0;
frameset = Interval of frame after which we want to process the frame;
Caution = False;
while True do
    frame No ← frame No + 1;
    frame = Single frame from camera;
    if (frame No / frameset == 0) then
        Caution = Condition Algorithm  1;
        if Caution == True then
            Process the frame and Recognize the text
        else
            Move to next frame
        end
    end
    else if frame No / frameset == 1& Caution == False then
        Caution = Condition Algorithm  1;
        if Caution == True then
            Process the frame and Recognize the text
        else
            Move to next frame
        end
    end
    else if loop is continue till( frame No /frameset) = frameset-1;
    else if frame No / frameset == frameset-1 & Caution == False then
        Caution = Condition Algorithm  1;
        if Caution == True then
            Process the frame and Recognize the text
        else
            Recognize Text = None
            Move to next frame
        end
    end
end
```

Observation

In the videos of target data, from duration of approx. 430 sec, We are choosing one frame in the interval of 5 seconds for evaluation and exporting the data into .csv file.

Table : Result of Implemented Software			
Instrument readings		1 (Numerical value)	2 (Units)
Number of Sample		89	89
Skip Samples		1	1
Detected Sample		88	88
Correct Recognized Sample		88	88
Detection efficiency		98.87	98.87
Recognition Efficiency (On Detected Samples)		100	100
Confidence Level	Mean	0.902	0.687
	Maximum	1.000	0.932
	Minimum	0.678	0.527

Software shows 98.87% accuracy on detection and 100% recognition accuracy on detected frame.

Result

We achieve 100% efficiency on recognized data.

2. Training

Input Data

Images
Labels in text file

Package of PaddleOCR

https://github.com/PaddlePaddle/PaddleOCR

Training

Dataset :

100 Training Data
50 Testing Data

Train the model in Google-Colab for epoch 10 and observe the loss.

Epoch	Loss
1	7.62
2	6.87
3	6.66
4	5.91
5	6.61
6	5.04
7	5.59
8	4.82
9	4.59
10	5.53

Optimization the execution time

Train the model in Google-Colab to optimize the execution time

No. of CPU : 1 No. of Core : 2

Train Size	Test Size	No. of Epoch	Batch Size	No. of worker	Execution Time
100	50	1	16	1	00:09:29
100	50	1	16	2	00:09:39
100	50	1	16	4	00:09:30
100	50	1	16	6	00:09:38
100	50	1	16	8	00:10:07
100	50	1	16	16	00:09:49

Train Size	Test Size	No. of Epoch	No. of worker	Batch Size	Execution Time
100	50	1	1	1	00:06:44
100	50	1	1	2	00:06:33
100	50	1	1	4	00:07:13
100	50	1	1	8	00:07:19
100	50	1	1	16	00:09:29
100	50	1	1	32	> 00:30:00

Train Size	Test Size	No. of Epoch	No. of worker	Batch Size	No. of worker(Test Data)	Execution Time
100	50	1	1	2	1	00:06:33
100	50	1	1	2	2	00:06:36
100	50	1	1	2	4	00:06:42
100	50	1	1	2	8	00:06:37
100	50	1	1	2	16	00:06:39

Batch Size : Number of sample used together to train model.

Number of workers : Number of threads used for data processing

Conclusion

Less Execution Time :

No. of CPU : 1 No. of Core : 2

Training Dataset :

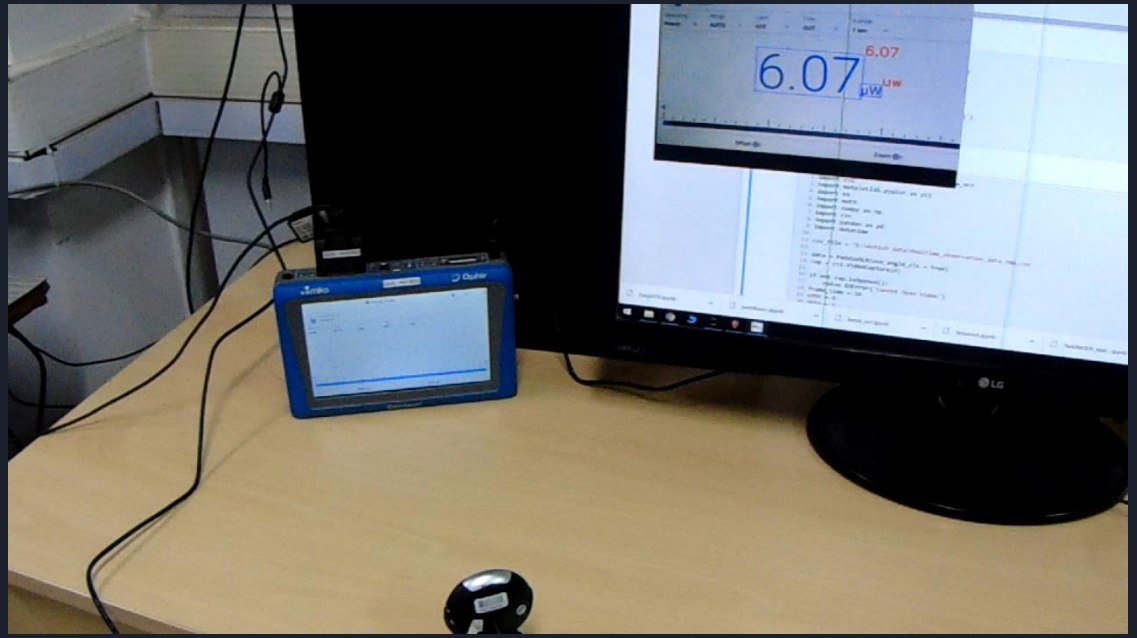
- Batch Size : 2
- No of Worker : 1

Testing Dataset :

- No. of Worker : 1

Result

- We evaluated different libraries for Text detection and recognition
- We have successfully completed Real-time text detection with 100 % recognition efficiency in target environment.
- We have established the process for training the model for further use.



Future Works

Design a website which contain :

- Image to Word converter
- PDF to word Converter

Training of Model for Different Language

Improve the efficiency for EasyOCR and KerasOCR

References

1. Y. Du, C. Li, R. Guo, X. Yin, W. Liu, J. Zhou, Y. Bai, Z. Yu, Y. Yang, Q. Dang *et al.*, "Pp-ocr: A practical ultra lightweight ocr system," *arXiv preprint arXiv:2009.09941*, 2020
2. "Tesseract open source ocr engine (main repository)," https://github.com/tesseract-ocr/tesseract,
3. "Easyocr: Ready-to-use ocr with 80+ supported languages and all popular writing scripts including latin, chinese, arabic, devanagari, cyrillic and etc." https://github.com/JaidedAI/EasyOCR,
4. "Paddlepaddle/paddleocr: Awesome multilingual ocr toolkits based on paddlepaddle."https://github.com/PaddlePaddle/PaddleOCR,
5. "Keras-ocr: A packaged and flexible version of the craft text detector and keras crnn recognition model." https://github.com/faustomorales/keras-ocr,
   "Airport timetable hi-res stock photography and images - alamy," https://www.alamy.com/stock-photo/airport-timetable.html,
6. "Airport timetable hi-res stock photography and images - alamy," https://www.alamy.com/stock-photo/airport-timetable.html,
7. Train timetable usa hi-res stock photography and images - alamy," https://www.alamy.com/stock-photo/train-timetable-usa.html,

Thank You