

Transformer-Based Detection Model for Number Recognition on Electric kWh Meters

Leni Fitriani¹, Ahmad Sanusi², Rita Rismala³, Dewi Tresnawati⁴

^{1,2,4} *Department of Computer Science, Institut Teknologi Garut, Garut, Indonesia*

³ *School of Computing, Telkom University, Indonesia*

*corr-author: leni.fitriani@itg.ac.id

Abstract - Manual recording of analog kWh meters frequently results in user complaints due to discrepancies between recorded and actual electricity usage. These issues stem from the continued reliance on manual data collection. This study proposes a model that automatically detects and extracts numerical values from kWh electricity meters using the Detection Transformer (DETR) for object detection and EasyOCR for optical character recognition (OCR). The model was developed using the Machine Learning Life Cycle (MLLC) methodology, comprising data acquisition, preprocessing, modeling, evaluation, and deployment. Evaluation using the Mean Average Precision (mAP) metric yielded a score of 96.83%, demonstrating high object detection accuracy. The trained model was integrated into a simple web application built with the Flask framework. While the model performed well on high-quality images, its effectiveness declined on low-quality images, such as blurry or distant captures. This study highlights the potential of DETR for object detection and OCR-based text extraction in analog meter reading, while also identifying challenges in handling suboptimal image conditions for future improvements.

Keywords: detection transformer, kWh, MLLC, mean average precision, Optical Character Recognition

I. INTRODUCTION

Electricity is a fundamental resource that supports daily human life and business operations. It plays a central role in residential activities and is essential for industrial and office functions, including communication and information systems [1,2]. In Indonesia, electricity usage is measured using kilowatt-hour (kWh) meters provided by the State Electricity Company (PLN) [3]. These meters are categorized into two main types: prepaid digital meters and postpaid analog meters [4]. The analog meter system allows users to consume electricity first and pay afterward, based on readings shown on the meter display [5].

However, manual reading and recording of analog meters often lead to user complaints due to discrepancies

between billed and actual consumption [6]. These issues stem from the manual recording process, which is prone to human error and inconsistency [7,8]. To address this, image processing and machine learning techniques, particularly Optical Character Recognition (OCR), have been applied to convert meter images into text data automatically. CNN- and CRNN-based OCR models have been developed to recognize numeric characters in structured images [9]. In parallel, object detection models such as YOLO, Fast R-CNN, and Detection Transformer (DETR) have been explored for identifying regions of interest in images. DETR, in particular, has shown strong performance in object detection tasks by leveraging transformer-based self-attention mechanisms [10-13]. Studies such as those by [14] and [15] indicate that DETR can outperform traditional models like YOLOv3 in specific contexts due to its end-to-end architecture and ability to focus on image context.

Although DETR and EasyOCR have been widely used independently, their integration for analog electricity meter reading—particularly in the Indonesian context—remains underexplored. Previous studies in countries like Brazil and India applied YOLO combined with CNN or PaddleOCR for automated meter reading (AMR) [16,17]. These systems generally performed well but were developed under controlled lighting and stable conditions. In contrast, this study proposes a novel approach that integrates DETR and EasyOCR to handle real-world analog kWh meter images in Indonesia, which include variations in lighting, distance, and quality. The system is deployed via a lightweight web interface built with Flask, enabling practical implementation in low-resource environments. This context-aware integration addresses existing gaps in AMR research and provides a scalable solution tailored for developing countries where manual meter reading is still prevalent.

II. METHOD

The method used is the machine learning life cycle methodology. The machine learning life cycle is the

stage in developing and implementing a machine learning model. This Machine Learning Life Cycle method has six process stages in its development and implementation, including Acquisition, Preparation, Modeling, and Evaluation [18].

Fig. 1 is the flow of the methodology used and will explain the stages as follows :

A. Acquisition

At this stage, dataset collection is carried out, which aims to select data that will later be used in model training. In this process, a dataset that is real and follows what will be used is needed. The datasets used in this study were taken from kWh by taking pictures using a cellphone, and some were taken from the internet, which had images similar to the kWh used in this study. The photos taken are then validated again for suitability with the dataset needed for the modeling process. To ensure that the model generalizes well in real-world conditions, the dataset was collected with considerable variation in environmental and technical factors. The images were taken under different lighting conditions, including natural daylight (morning/afternoon), low-light indoor environments, and backlit scenarios, to simulate field conditions. Additionally, the camera angles vary from frontal (90°) to slightly tilted views (45°–70°), representing realistic positioning by human operators. The distance between the camera and the kWh meter ranged from approximately 20 cm to 1 meter, and the images were captured using a variety of smartphone models with different camera resolutions. To ensure character diversity, care was taken to include images displaying all digits from 0 to 9. Preliminary analysis showed that the digit distribution across the dataset is relatively balanced, minimizing bias during the training process. This diversity enhances the robustness of the model and supports better performance in unpredictable or suboptimal real-world scenarios.

B. Inspection

After the dataset is collected or obtained, at the next stage, inspection or inspection is carried out on the collected dataset so that the dataset is following what is used, such as checking the image whether the image is correct and can be used or not and if the image is appropriate, it will enter the preparation stage later.

C. Preparation

After the image dataset has gone through the inspection stage, the next stage is the preparation stage. At this stage, several activities are carried out.

1) *Resize*: *Resize* image size is changing the original image's dimensions to the dimensions that match what is needed or what is used. The goal is that the images used are the same size and follow what the model requires. In this step, resize the size to make it easier for the model to train.

2) *Labeling*: *Labeling* is a process where raw data (images, files, text, videos, etc.) are added with one or more informative labels to provide characteristics so that the model can learn from the raw data that has been labeled [19]. Such labeling is needed or required in recognition or detection.

3) *Augmentation*: In this augmentation stage, data is added so that the data used is not tiny. This is commonly used in deep learning because deep learning is hungry for data, so augmentation is done to enlarge the data.

4) *Split Dataset*: *Split dataset* is a process performed in data analysis, especially in machine learning. The dataset is divided into two sets, and the purpose of this split data is to conduct training and testing on a model built with some training data to conduct training and test it with test data. The data split is generally at a ratio of 90% training data and 10% test data; the split for this ratio is better for extensive data such as one million data and also 80% to 20%, suitable for less large-scale data such as from 100 to 10,000 data [20].

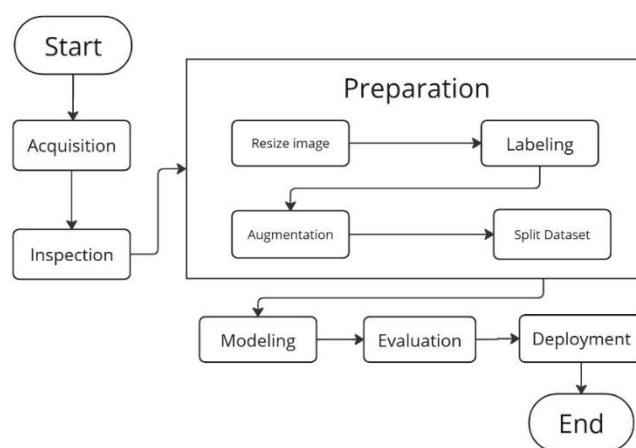


Fig. 1 The study methodology

D. Modeling

After data preparation, the next stage is the modeling stage. At this stage, the model implementation will be carried out using the Detection Transformer (DETR) model architecture; DETR will be trained to produce a model that can detect objects in the form of numbers on electric kWh meters. The Detection Transformer (DETR) model was chosen in this study due to its ability to perform end-to-end object detection without the need for region proposal networks or manual anchor box configurations, which are commonly required in YOLO and Fast R-CNN architectures. Unlike YOLO, which relies on a predefined grid and anchor boxes to localize objects, DETR utilizes a transformer-based attention mechanism that allows the model to learn object positions and relationships directly from the input features. This makes DETR particularly effective in handling small and closely spaced objects, such as the digits on analog kWh meters. Moreover, DETR's ability to incorporate global image context helps improve detection performance in cases with noisy or low-resolution input images—conditions that are common in field-captured meter images. These advantages support the selection of DETR over traditional detection models for this specific application.

E. Evaluation

The model evaluation aims to measure the extent to which the model functions properly in detecting objects in the form of electricity meter numbers and bounding the detected number box. The metric used to measure the performance of the DETR model, namely Mean Average Precision (mAP), is that the better the mAP value, the better the model detects objects [21]. With a value between 0 and 1. The higher the score or the closer to 1 the mapping score generated, the better or more accurate the model will detect [22].

DETR, in measuring the performance of its model, can use the Mean Average Precision (mAP) metric. MAP is usually used to measure object detection performance by considering classification and localization. Not only that, mAP consists of Precision-Recal (PR), Area Under the Curve (AUC), Multiple Object Categories (MOC), and also Intersection over Union (IoU). AUC is considered when calculating mAP to balance the trade-off of PR. In each category, the curve of PR is calculated by varying the prediction confidence of the model. Each class's average Precision (AP) will be calculated individually by sampling from the PR curve to classify and localize some ECG categories. Next, each class calculates the AP at the IoU threshold (AP50-90) and the Mean AP (mAP50-90). Finally, the overall AP is

calculated with the mAP at each IoU threshold [23]. With the formula shown in (1).

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (1)$$

As for the calculation of precision and recall (sensitivity), which is a standard metric used in arrhythmia classification [24] With the formula shown in (2) and (3).

$$Presisi = \frac{TP}{TP+FP'} \quad (2)$$

$$Recall = \frac{TP}{TP+FN'} \quad (3)$$

F. Deployment

This deployment aims to apply the results of the model performance on a simple web where the web only has image select and image upload features, and the framework used for the simple web uses the Flask framework from the Python programming language.

III. RESULT AND DISCUSSION

The research used the Machine Learning Life Cycle (MLLC) [25] Method with the results of object detection in numbers on kWh electricity meters using the Detection Transformer (DETR) architecture, a model used for object detection fields. The first layer is the CNN backbone, where this layer performs feature extraction from images with low resolution. Later, the features that have been extracted will be used by the encoder and decoder. By decreasing the image's resolution, the model can later reduce computation complexity by maintaining important information needed by object detection. The encoder layer has a Multi-head Self Attention architecture. The Feed Forward Network (FFN) will process the features that have been extracted by the backbone and change the resolution of the image to lower and also the feature map; after that, it will be processed in the decoder section, which will process the query object and present the encoder in parallel, this process will ensure the performance of the model in detecting objects accurately. The last layer is Prediction Heads, where the results that the decoder has processed will be predicted from both the class and the location of the bounding boxes. The dataset used in the study is an analog electricity kWh meter image collected by photographing kWh from house to house, which will later be processed for use in object detection training in the form of numbers on the electricity kWh meter. Then, the model trained with the data will go through the final stage, namely model evaluation with different datasets. These, namely test

data, have been prepared to see the performance results of the model that has been trained.

A. Acquisition

The stages carried out in the first stage is data collection, where the data used is analog kWh meter data generated by photographing kWh using a cellphone on each analog kWh in the environment around the researcher and validated by researchers whether object detection can be done or not. Here is a sample of the data that can be collected later in the next stage.

Fig. 2 is the raw data of 300 images, which will go through several stages to be done later for modeling.

B. Inspection

The raw data collected is inspected, and the data collected is by what is needed; this is where the 300 image data is an analog kwh in Indonesia and is used by PT. PLN Indonesia, and each image has an object in the form of numbers that can be detected; data on what is needed will be prepared first at the preparation stage.

C. Preparation

Data preparation will be carried out according to what is needed for training; at this stage, three stages are carried out, and the following are done at this stage.

1) *Resize*: In the first stage in the preparation section, a size resize is carried out where the raw data containing 300 image data is resized to a size of 640x640 pixels. This is done so that the data is the same size, which makes it easier for the model at the training stage. The size is based on previous images, mostly 640px, so the image is resized to a size of 640x640 pixels.

2) *Labeling*: At the dataset labeling stage, the data will be labeled using tools from roboflow, where the label is the location of the bounding boxes. The dataset labeling process is shown in Fig. 3.

In Fig. 3, the image is labeled by providing a bounding box, which will get an annotation of the

location of the bounding box created. The following annotation images are shown in Table I.

Table I shows the id, image, category, box, area, and crowd, which will be used in the training model stage.

3) *Augmentation*: At this stage, the data that has been annotated will be augmented to get more data. The augmentation stage has four stages, namely the first stage of cropping, the second stage of rotation with between -15% and +15%, Saturation with between -25% and +25%, and Exposure with between -10% and +10%. This is done because the previous data is 300 images. Data obtained after performing augmentation The total data generated by performing augmentation is 712.

4) *Split Dataset*: At this stage, the data will be divided into three parts where there is a train section for training, valid to validate the data that has been trained, and test to perform testing on data that has been trained and validated previously to see later how far the model works in detecting objects. The data split carried out in this study uses roboflow tools, with 80% training data, 10% validation data, and 10% test data.



Fig. 2 Analog kWh meter dataset



Fig. 3 Labeling dataset

TABLE I
ANNOTATIONS IMAGES

No	Id	Image Id	Category Id	Bbox	Area	Iscrowd
1	24	24	1	223, 231, 150, 30	4500	0
2	25	25	1	127, 109, 377, 84.5	31856.5	0
3	26	26	1	137, 40, 326, 82.5	26895	0
4	27	27	1	142, 77, 246, 54	13284	0
...
300	300	300	1	101, 147, 312.5, 63	19687.5	0

D. Modeling

The stage carried out at this stage is to create a model by training the model using the data prepared in the data process. The model architecture used is the Detection Transformer (DETR).

In Fig. 4, the first layer after the image is inputted, the image will enter the CNN backbone to extract low-resolution features from the image. The encoder and decoder will process the extracted features. This layer will produce a feature map with the input image's semantic and spatial representation. After the map features are generated, it will pass Positional Encoding; this is important because the transformer does not have position information by default. Positional encoding is used to provide position information for each map feature.

The next layer is the transformer encoder layer; this layer has self-attention layers that are used to pay attention to the global relationship between features in the entire image; in addition to self-attention, there is also a feed-forward network (FFN), which is used to process information and improve feature representation.

The next layer is the Transformer Decoder; this layer has object queries that detect specific objects in the image, which are then entered into cross-attention layers to make the object query interact with the features produced by the encoder. After passing through the cross-attention layers, it will enter the FFN, where it processes and creates the final representation of the query object.

After that, it will enter the Prediction Heads layer to predict the class and bounding box.

After creating the architecture, enter the model definition section with the hyperparameters used: the learning rate, how fast the model learns, weight decay to prevent the model from knowing too much training data, and the epoch for how long the model sees data samples.

After preparing everything, the data will be trained using the previously prepared model.

As for the DETR model that has been trained, it will be integrated with EasyOCR so that after DETR detects numbers, the numbers that have been detected will be extracted from the image into text in the form of strings. Figure 4 shows the integration of DETR and EasyOCR. Fig. 5 is the integration between DETR and EasyOCR, where when the image, after passing through the backbone, encoder, decoder, prediction heads, and produces an image with a bounding box, the image will be cropped based on the location of the bounding box, then grayscale, and the image will be blurred to remove noise, after that it is thresholded so that the image becomes binary which is black and white, and then extracted using EasyOCR into text in the form of a string.

E. Evaluation

After training, the next stage is the model evaluation stage; this stage will evaluate the test data that has been prepared previously and the evaluation results using the mapped metric with several trials. The results of the experiment shown in Table II.

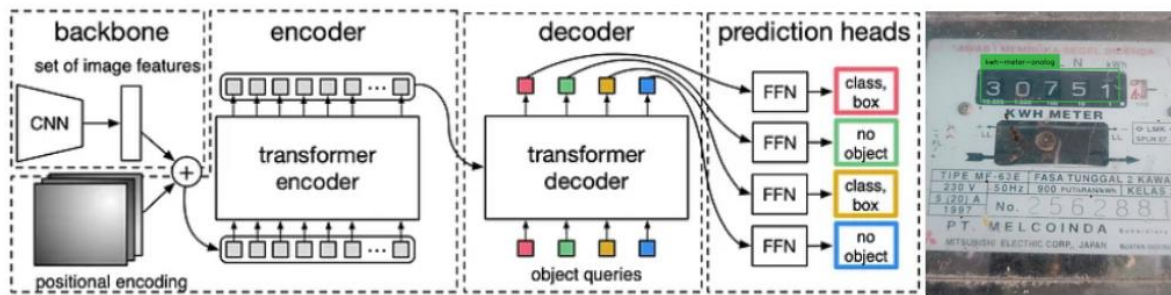


Fig. 4 Architecture model Detection Transformer (DETR)

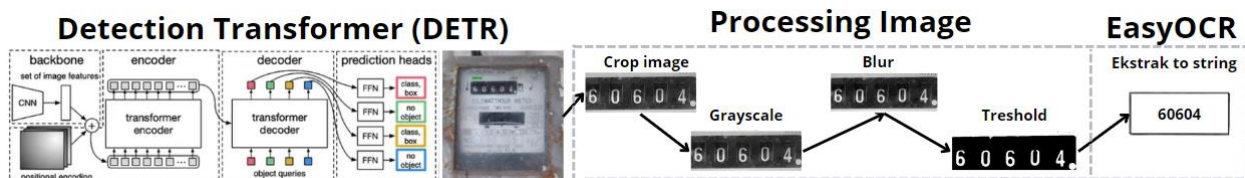


Fig. 5 Integration between DETR and EasyOCR

TABLE II
EVALUATION RESULTS

Hyperparameter	Experiment 1	Experiment 1
Epoch	100	200
Learning rate	1e-4	1e-4
Weight decay	1e-4	1e-4
Result mAP	0.9048	0.9683

Table II shows that the evaluation results with the mapping matrix obtained in the first experiment were 0.9048 and 0.9683 in the second. The results obtained by increasing the epoch size exceed the first trial mAP, with an mAP value of 0.9683. The mapped value is close to 1, which indicates that the model can work well and can detect objects consistently in various conditions, whether size, position, or light.

The integration results of DETR and EasyOCR, where EasyOCR is used to extract images to text. The results of the integration shown in Table III.

Table III shows the DETR model integrated with EasyOCR. In some images, the model can detect numbers well, the model can detect objects in the form of numbers, and EasyOCR can extract pictures into text, but in some images from image to text using EasyOCR has not been able to extract properly, object detection in the form of numbers and poor image extracts can be influenced by poor image quality and can also be due to the distance of the image that makes the detection model and also EasyPCR cannot extract properly. Although the DETR model achieved strong performance in detecting numerical regions on kWh meter images—demonstrated by an increase in mAP from 0.9048 to 0.9683—OCR accuracy using EasyOCR showed inconsistencies, particularly on low-quality images. Several factors contributed to the OCR failure, including image blur caused by hand movement, low contrast between digits and background due to lighting conditions, excessive image noise, and the presence of small-sized digits captured from a distance. These factors reduce the clarity of the cropped digit regions, thereby affecting EasyOCR’s ability to segment and recognize characters accurately. To mitigate these issues, future enhancements could incorporate preprocessing techniques such as image denoising, contrast enhancement, and super-resolution. Image denoising can help remove background artifacts, contrast enhancement can improve the distinction between digits and meter casing, while super-resolution techniques—such as ESRGAN or Real-ESRGAN—can upscale low-

resolution digit regions to improve clarity. Implementing these steps before feeding the cropped regions to EasyOCR is expected to improve recognition accuracy, especially under non-ideal field conditions.

F. Deployment

At this stage, the model evaluated with good results will be implemented into a simple web with flow in Fig. 6.

In Fig. 6, the flow of the deployment model that has been trained will be applied to the web using the flask framework. By starting and then entering the image input, after the image is inputted, object detection will be carried out using DETR; after finding the location results of the numbers listed on the kWh meter, image processing will be carried out so that the numbers listed can be extracted properly, the first stage in processing the image is to crop the image, grayscale, then blur and finally threshold. After the image goes through the image processing stage, the numbers in the image will be processed by EasyOCR for extraction from image to text and in the form of strings. The initial appearance of a simple web is shown in Fig. 7.

In Fig. 7, select and upload images are displayed on the initial display. When the image is uploaded, the image will be processed for number detection in the uploaded image. The results of the image that was detected in Fig. 8 are presented.

TABLE III
EASYOCR RESULTS





Image	Prediction Boxes	Results Number kWh	Confidence
	0,99	191447	0,47
	0,99	60624	0,40
	0,98	0215	0,70
	0,97	1169	0,41

Fig. 8 displays an image with a bounding box that has been processed and gets the detection results on the numbers. After the bounding box location is found in the image, it will be processed at the image processing stage; this stage aims to prepare the image so that it can produce an image-to-text extraction properly; the first step is to crop the image so that the image size becomes smaller, then do grayscale so that the image becomes 2-dimensional black and white or gray after that enter the blur stage to remove noise in the image, after doing blur, the image will be thresholding, which helps convert grayscale images into binary images, where each pixel has two values where the value zero is black, and one is white. This step makes it easier for EasyOCR to extract images to text. After going through the image processing stage, the image will enter the image-to-text extraction stage using EasyOCR. The following is the output of the processed image, as shown in Fig. 9. Fig. 9 (a) displays the output of the processed image results, where prediction results from the DETR model get 99% detection of numbers on the kWh meter and the results of image-to-text extraction. EasyOCR can extract the image and text it correctly. Fig. 9 (b) displays the results of the experiment process on images that have poor quality; in the image, the number detection model gets a prediction of 0.97, which is where the model can work well in detecting, but for image extracts, EasyOCR cannot extract properly, where the output produced is 7 &, with a confidence value of 0.46.

G. Discussion

This study demonstrates that the Detection Transformer (DETR) model can accurately detect numbers on kWh meters, achieving a high Mean Average Precision (mAP) of 96.83%. DETR outperforms other models, such as YOLO and Fast R-CNN, due to its simple end-to-end architecture and self-attention mechanism, which enables the model to focus on the context of the image. The integration with EasyOCR allows automatic extraction of numbers from images, although performance declines with low-quality images, such as blurry or poorly lit photos. This model has the potential to be implemented in an automated meter reading system, improving efficiency, reducing recording errors, and facilitating real-time customer data management. Future research could enhance the model's performance on low-quality images by expanding the training dataset, exploring advanced architectures like Deformable DETR, and incorporating AI-based image enhancement techniques such as super-resolution and deblurring to improve image clarity before detection and

extraction. Additionally, future work could explore integration with IoT devices for real-time meter reading automation and develop post-processing algorithms to improve OCR accuracy for challenging characters. In comparative object detection studies, DETR has shown competitive or superior performance to conventional models such as YOLOv5, especially in structured environments with densely packed or overlapping objects. For instance, Chen et al. (2022) demonstrated that DETR achieved more stable precision in detecting small targets compared to YOLOv5 and Faster R-CNN in industrial image contexts. In the context of this study, DETR's transformer-based architecture—with its global attention mechanism—enables the model to capture spatial relationships and contextual patterns more effectively, making it well-suited for detecting digits on analog kWh meters, which often appear with minimal spacing and variable placement. Another advantage of DETR is its robustness with limited training data [26]. While many deep learning models require thousands of images to generalize effectively, DETR benefits from its attention-based mechanism that learns holistic representations across the entire image. This allows the model to maintain high detection performance (mAP 96.83%) even with a relatively small dataset (712 images after augmentation). In contrast, models like YOLOv5 rely more heavily on dense anchor-based localization, which can be sensitive to small dataset variations and require extensive fine-tuning. These architectural advantages explain DETR's strong performance in this study, making it a suitable choice for object detection tasks involving localized digit patterns with limited training samples.

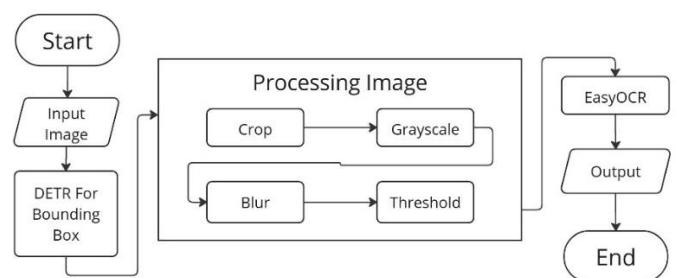


Fig. 6 Flow website

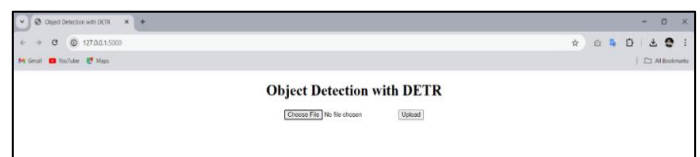


Fig. 7 Web start view

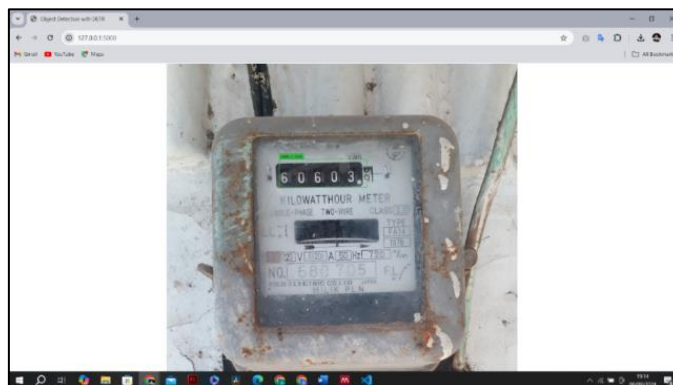


Fig. 8 Bounding box

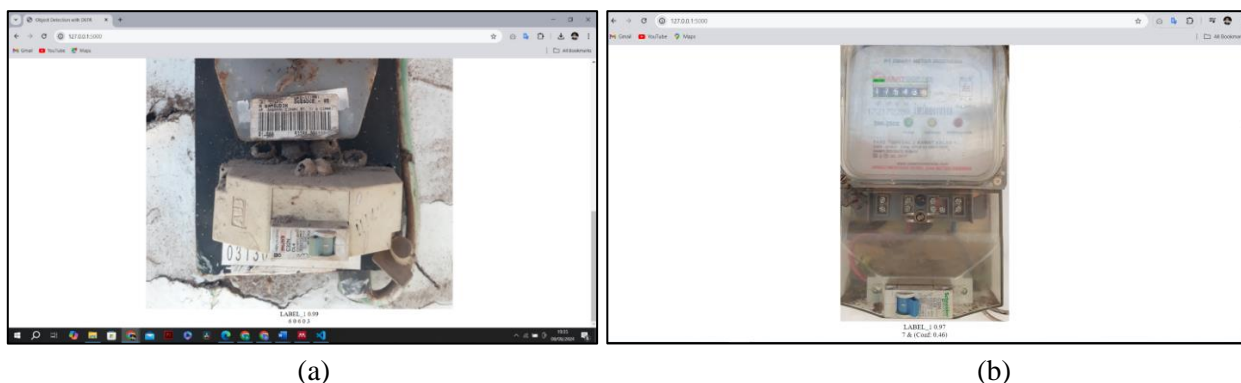


Fig. 9 Output (a) and Poor quality image output (b)

IV. CONCLUSION

This study demonstrates that the Detection Transformer (DETR) model, developed using the Machine Learning Life Cycle (MLLC) methodology and trained on 712 augmented analog kWh meter images, achieved a high object detection performance with a mean average precision (mAP) of 96.83%. The model was effectively integrated with EasyOCR and deployed through a lightweight web interface using the Flask framework. While the system performs well on high-quality images, it faces challenges when processing low-quality inputs such as blurred, noisy, or distant images. Overall, this integrated approach offers a promising solution for automated meter reading in real-world environments, though further improvement is needed to enhance performance under suboptimal image conditions. To enhance system robustness and scalability, future work may explore the integration of IoT-based devices for real-time and remote meter reading, eliminating the need for manual field data collection. In addition, applying image enhancement techniques such as super-resolution or contrast normalization could improve OCR accuracy on low-quality inputs. From the model perspective, advanced architectures like

Deformable DETR may provide better adaptability to small or irregularly spaced digits. Furthermore, while EasyOCR was chosen for its simplicity, exploring alternative OCR engines such as Tesseract or PaddleOCR could offer improved performance, especially under more complex or noisy conditions. Comparative evaluation of these alternatives may help determine the most suitable OCR engine for large-scale deployment in diverse operational settings.

REFERENCES

- [1] Napis, M. Farhan, Rahmatulloh, A. R. Hakim, and M. T. Apriyanto, "MENINGKATKAN KESADARAN MASYARAKAT DALAM BUDAYA HEMAT ENERGI MELALUI PENYULUHAN EFISIENSI PENGGUNAAN LISTRIK RUMAH TANGGA," *J. Pendidik. Dan Pengabd. Masy.*, vol. 6, no. 2, pp. 107–115, May 2023, doi: 10.29303/jppm.v6i2.4980.
- [2] D. S. Markovic, D. Zivkovic, I. Branovic, R. Popovic, and D. Cvetkovic, "Smart power grid and cloud computing," *Renew. Sustain. Energy Rev.*, vol. 24, pp. 566–577, 2013, doi: <https://doi.org/10.1016/j.rser.2013.03.068>.

- [3] A. N. Walidi, "Akurasi Pengukuran kWh Meter Analog Terhadap Losses Energi Listrik," *Sutet*, vol. 11, no. 2, pp. 105–113, 2021, doi: 10.33322/sutet.v11i2.1577.
- [4] J. Son, "Analisa Data Hasil Pengukuran Beban Motor Listrik 1 Fasa pada kWh Analog dan kWh Digital," *Electr. J. Rekayasa Dan Teknol. Elektro*, vol. 15, no. 3, Art. no. 3, Sep. 2021, doi: 10.23960/elc.v15n3.2219.
- [5] D. Permata *et al.*, "Sistem Perhitungan kW Meter Listrik Prabayar (LPB) untuk Pelanggan Daya 900 Va PT. PLN (Persero) Area Palembang," *J. Teliska ISSN*, vol. 5, no. 2, pp. 53–61, 2013.
- [6] D. D. Prihartomo, R. D. Nyoto, and A. S. Sukamto, "Rancang Bangun Aplikasi Pencatatan dan Pengolahan Data Pemakaian KWH (Kilowatt Hour) Listrik Digital," *Justin J. Sist. Dan Teknol. Inf.*, vol. 4, no. 2, pp. 1–5, 2016.
- [7] I. Wijaya and C. Lubis, "Pengimplementasian Ocr Menggunakan Cnn Untuk Ekstraksi Teks Pada Gambar," *J. Ilmu Komput. Dan Sist. Inf.*, vol. 10, no. 1, 2022, doi: 10.24912/jiksi.v10i1.17836.
- [8] A. Haris, "Sistem Pencatat Kwh Meter Terintegrasi Komputer Untuk Meningkatkan Layanan Pada Pelanggan," *KILAT*, vol. 6, no. 1, Art. no. 1, 2017, doi: 10.33322/kilat.v6i1.664.
- [9] N. Kholis and F. Baskoro, "Rancang Bangun Sistem Deteksi Label Kardus Berbasis Model Kecerdasan Buatan YOLO dan EasyOCR serta ESP32-CAM Rancang Bangun Sistem Deteksi Label Kardus Berbasis Model Kecerdasan Buatan YOLO dan EasyOCR serta ESP32-CAM Stefanus Adhie Nugroho Abstrak," pp. 190–200, 2022, doi: https://doi.org/10.26740/jte.v11n2.p190-200.
- [10] M. Tanzib Hosain, A. Zaman, M. R. Abir, S. Akter, S. Mursalin, and S. S. Khan, "Synchronizing Object Detection: Applications, Advancements and Existing Challenges," *IEEE Access*, vol. 12, no. April, pp. 54129–54167, 2024, doi: 10.1109/ACCESS.2024.3388889.
- [11] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," *Lect. Notes Comput. Sci. Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinforma.*, vol. 12346 LNCS, pp. 213–229, 2020, doi: 10.1007/978-3-030-58452-8_13.
- [12] E. Suherman, B. Rahman, D. Hindarto, and H. Santoso, "Implementation of ResNet-50 on End-to-End Object Detection (DETR) on Objects," *Sinkron*, vol. 8, no. 2, pp. 1085–1096, 2023, doi: 10.33395/sinkron.v8i2.12378.
- [13] X. Chen, F. Wei, G. Zeng, and J. Wang, "Conditional DETR V2: Efficient Detection Transformer with Box Queries," Jul. 18, 2022, *arXiv*: arXiv:2207.08914. doi: 10.48550/arXiv.2207.08914.
- [14] J. Yang, S. Deng, F. Zhang, A. Pan, and Y. Yang, "FATCNet: Feature Adaptive Transformer and CNN for Infrared Small Target Detection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 6, pp. 9231–9246, Dec. 2024, doi: 10.1109/TAES.2024.3441551.
- [15] F. Luo, Y. Dai, J. Fuentes, W. Ding, and X. Zhang, "M-DETR: Multi-scale DETR for Optical Music Recognition," *Expert Syst. Appl.*, vol. 249, 2024, doi: 10.1016/j.eswa.2024.123664.
- [16] R. Laroca, V. Barroso, M. A. Diniz, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, "Convolutional Neural Networks for Automatic Meter Reading," *J. Electron. Imaging*, vol. 28, no. 01, p. 1, Feb. 2019, doi: 10.1117/1.JEI.28.1.013023.
- [17] F. Khan, S. Rafique, and G. M. Khan, "Low-Cost Smart Metering Using Deep Learning," *Int. J. Innov. Sci. Technol.*, vol. 6, no. 5, Art. no. 5, May 2024.
- [18] A. Combs, *Python Machine Learning Blueprints Second Edition*, no. July. 2020.
- [19] L. Fitriani, D. Tresnawati, and M. B. Sukriyansah, "Image Classification On Garutan Batik Using Convolutional Neural Network with Data Augmentation," *JUITA J. Inform.*, vol. 11, no. 1, p. 107, 2023, doi: 10.30595/juita.v11i1.16166.
- [20] V. R. Joseph and A. Vakayil, "SPlit: An Optimal Method for Data Splitting," *Technometrics*, vol. 64, no. 2, pp. 166–176, 2022, doi: 10.1080/00401706.2021.1921037.
- [21] J. Terven and D. Cordova-Esparza, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680–1716, Nov. 2023, doi: 10.3390/make5040083.
- [22] W. Muldayani, "Implementasi Sistem Object Tracking Untuk Mendeteksi Dua Objek Berbasis Deep Learning," *Simetris J. Tek. Mesin Elektro Dan Ilmu Komput.*, vol. 14, no. 1, pp. 1–14, 2023, doi: 10.24176/simet.v14i1.9236.
- [23] G. J. N. Ang *et al.*, "A Novel real-time arrhythmia detection model using YOLOv8," Jan. 08, 2024, *arXiv*: arXiv:2305.16727. doi: 10.48550/arXiv.2305.16727.
- [24] Z. Ebrahimi, M. Loni, M. Daneshlab, and A. Gharehbaghi, "A review on deep learning methods for ECG arrhythmia classification," *Expert Syst. Appl. X*, vol. 7, p. 100033, 2020, doi: 10.1016/j.eswax.2020.100033.
- [25] L. Fitriani, A. Latifah, and M. R. Cahyadi Putra, "Image Classification of Room Tidiness Using VGGNet with Data Augmentation," *JUITA J. Inform.*, vol. 12, no. 1, p. 111, May 2024, doi: 10.30595/juita.v12i1.21204.
- [26] X. Chen, F. Wei, G. Zeng, and J. Wang, "Conditional DETR V2: Efficient Detection Transformer with Box Queries," Jul. 18, 2022, *arXiv*: arXiv:2207.08914. doi: 10.48550/arXiv.2207.08914.

