A computer vision approach with OpenCV and deep learning for determining inductance in planar coils

Younes Benazzouz[a], *Djilalia* Guendouz[b]

University of Oran 2 Mohamed Ben Ahmed, Industrial Maintenance and Safety Institute (IMSI), Department of instrumentation maintenance, Laboratory of Production Engineering and Industrial Maintenance (LGPMI), Oran, People's Democratic Republic of of Algeria

[a] e-mail: benazzouzyounes14@gmail.com, **corresponding author**, ORCID iD: https://orcid.org/0009-0004-8736-7610

[b] e-mail: lila.guen@yahoo.fr, ORCID iD: https://orcid.org/0009-0006-7129-5960

https://doi.org/10.5937/vojtehg72-51477

FIELD: computer sciences, electronics
ARTICLE TYPE: original scientific paper

Abstract:

Introduction/purpose: In the realm of development and use of computer vision and AI methodologies, this research introduces a combination and advanced method using YOLOv9, a deep learning concept of whole image processing in one pass through a convolutional neural network (CNN) and the OpenCV Python image processing library to determine the geometry of planar coils. These geometric parameters are the main parameters used to calculate the inductance value using Mohan's formula, which exclusively utilizes only geometric data to estimate inductance values. This method significantly speeds up the verification and calculation processes, while also playing a role in improving quality control after manufacturing.

Methods: The methodology is divided into two main phases. Initially, a YOLOv9 model was trained for object recognition using a generated synthetic dataset of coil shapes created with Python's Turtle graphics library. Then, after the detection phase, OpenCV was used to identify the geometric parameters of the images. The pixels were converted into millimeters using a ratio method to calculate the inductance value accurately.

Results: The YOLOv9 model successfully identified various planar coil shapes, and the geometric parameters were identified through OpenCV. Subsequently, the inductance was successfully calculated.

Conclusion: The results show that the proposed method is a novel and effective way of calculating inductance.

Key words: Convolutional Neural Networks (CNN), OpenCV, planar coil, inductance, YOLOv9, image processing.

Introduction

Planar coils are crucial components in electronic systems and can be found in the architecture of wireless power transfer (WPT) systems, ensuring energy management and transmission. They are useful in applications such as electric vehicle charging. Their operation is based on electromagnetic induction between two parallel coils; without physical contact, electrical energy is transferred when electrical current is passed through one of them (Luo & Wei, 2017). Additionally, they can be used for position sensing in permanent magnet synchronous motors, offering a resilient design compared with traditional sensors like Hall sensors. They are easy to produce without occupying a lot of space and function as converters of magnetic flux in the air gap of the motor into an electrical signal (Im & Hur, 2021). Moreover, planar coils are used as monolithic integrated circuits on chips to minimize planar surface area and are subsequently integrated into DC-DC Forward converters for both high and low frequency uses, allowing perfect miniaturization for modern electronic devices (Derkaoui et al, 2021). In the implementation of biomedical devices, they play a principal role, offering an ideal solution for wireless recharging of pacemakers' batteries using square-shaped printed spiral designs (Ahire et al, 2022). In NFC-enabled devices, planar coils play a crucial role, with those in the PICC (Proximity Integrated Circuit Card) harvesting energy from the magnetic field created by the PCD (Proximity Coupling Device). Thus, they perform a dual role: facilitating data transfer between the reader and the receiver, and serving as power supply for the NFC system (Couraud et al, 2020).

The inductance is the main parameter related to coils in the field of electromagnetics. Various estimation methods and calculations are used to calculate or estimate it with an objective of a low level of error. The formula of Mohan (Mohan et al, 1999) is one of them. What makes it different is that it is based solely on the geometrical parameters of the planar coil. Knowing these parameters will give an inductance value with a 3% error margin compared to field solver predictions. Researchers have successfully employed Mohan's formula to optimize the performance of planar inductors on Kapton (Kharbouch et al, 2017). It is not complex to estimate the inductance value using this method, and because the parameters used are purely geometrical and some constants, it facilitates the integration of computer vision technologies to capture these values and automate the inductance calculation.

Building upon the idea of capturing geometrical parameters, an image treatment process was employed. It can be divided into two main sections:

the first is coil shape recognition using a YOLOv9 deep learning object detection model (Wang et al, 2024), and the second is geometrical parameter capturing and calculating inductance using the OpenCV image treatment Python library.

A number of studies have started using YOLOv9 for different objectives. For example, YOLOv9 has been applied for fracture detection in pediatric wrist X-ray images (Chien et al, 2024). It has also been used for the objective of detecting small-scale (<20km) ocean eddies to monitor changes in the Earth's oceans and climate (Mostafa et al, 2024). To train the YOLOv9 coil shapes detection model, a synthetic dataset was created using the Python Turtle library. This allowed the generation of a varied dataset with multiple spiral configurations (square, hexagonal, octagonal, and circular). After the creation of the data, augmentation was applied using the Roboflow platform. Then, OpenCV was used to detect the contours with perfect accuracy, leading to precise distance measurements of the geometry. All values were in pixels, so we made a ratio based on the known diameter of a coin to convert all values to millimeters for inductance calculation. Figure 1 presents our work in the form of a chart flow visualization to summarize the entire process.

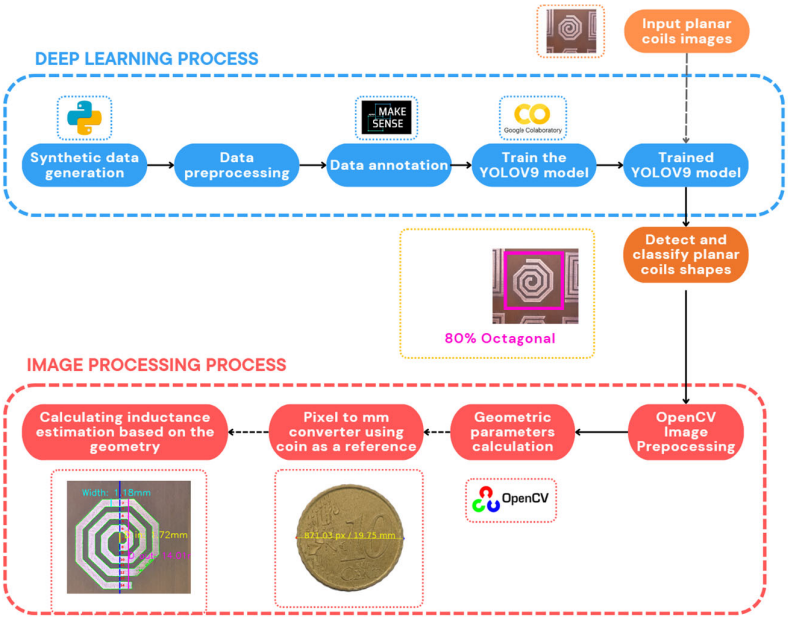

Figure 1 – Flowchart: Process for coil shape detection and inductance calculation using YOLOv9 and OpenCV

Overview of Mohan's formula

Mohan developed a simple and accurate expression for calculating the inductance of single-layer spiral shapes, including square, circular, hexagonal, and octagonal (Mohan et al, 1999). Since this expression relies directly on geometric parameters, the integration of computer vision is applied to automate inductance measurements.

The expression of the inductance is given in equation (1)

$$L=\frac{\mu_0 n^2 d_{avg}\, c_1}{2}\left(ln(c_2/\rho)+c_3\,\rho+c_4\rho^2\right) \tag{1}$$

The average diameter d_{avg} and the form factor ρ are generally defined as follows:

$$d_{avg}=\frac{d_{out}+d_{in}}{2} \tag{2}$$

$$\rho=\frac{d_{out}-d_{in}}{d_{out}+d_{in}} \tag{3}$$

where d_{in} is the inner diameter, d_{out} is the outer diameter, μ_0 is the vacuum permeability, and n is the number of turns. The coefficients from C_1 to C_4 vary based on the coil geometry and are listed in Table 1.

Table 1 – Coefficients for the analytical calculation of inductance

Form	C_1	C_2	C_3	C_4
square	1.27	2.07	0.18	0.13
hexagonal	1.09	2.23	0	0.17
Octagonal	1.07	2.29	0	0.19
Circular	1	2.46	0	0.2

Figure 2 illustrates the single-layer coil spiral shapes in various geometries, including square, circular, hexagonal, and octagonal forms.

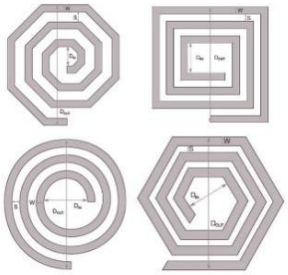

Figure 2 – Single-layer coil spiral shapes (Square, Circular, Hexagonal, And octagonal)

Dataset creation and pre-processing

Synthetic planar coil image dataset generation

Machine learning and deep learning fields latest research is highly focused on data generation to train models. The geographic information system (GIS) framework generates realistic pictures using deep neural networks and adversarial training for instance segmentation (Abu Alhaija et al, 2019; Paulin & Ivasic-Kos, 2023). Others render 3D geometric models using the ShapeNet dataset as a method for 3D object reconstruction (Richardson et al, 2016). A Python library called Turtle was utilized to generate the coils dataset. Different scripts were programmed to generate realistic pictures of planar spiral coils in hexagonal, octagonal, circular, and square geometries. This library was chosen for its capability and feature to design spiral shapes using coding. The code for each shape controlled the geometric parameters. Figure 3 is an example of how the Turtle library scripting plots the generation of spiral shapes. Each shape represents a distinct algorithmic approach to drawing patterns (Anderson, 2018).

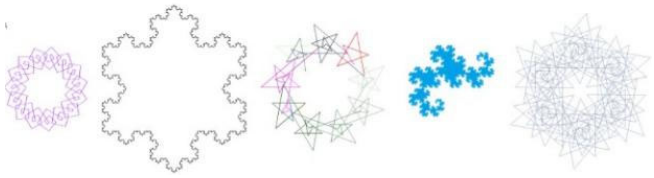

Figure 3 – Different patterned shapes plotted using the Turtle library (Anderson, 2018)

Four distinct scripts were developed to generate the dataset. The specific variable parameters included the number of spiral turns, the width

range, and the side lengths range, including inner and outer diameters. Using nested loops, each script iterates through a combination of these parameters, resulting in multiple outputs. The resulting pictures were created with the dimensions of 640 x 640 pixels, saved as EPS files, and then converted to JPEG. This automated approach ensures a perfect method to generate a comprehensive dataset. The focus was on creating a high-quality set of synthetic images to represent various geometries of planar spiral coils. The entire dataset consisted of binary color images, with each class designed to follow a clear and distinct pattern, with each single image represented in a simple 2D format with a single top-view angle. The key variations within each class were limited to the turn number and width as primary distinguishing features. This careful and clear diversity in class design ensured that even with a smaller dataset, the model had sufficient information to learn and differentiate between the four shape classes. The clear distinction between the classes made it easy for the model to recognize and learn the differences between them. Figure 4 shows a sample of generated spiral shapes using the Turtle library scripts.

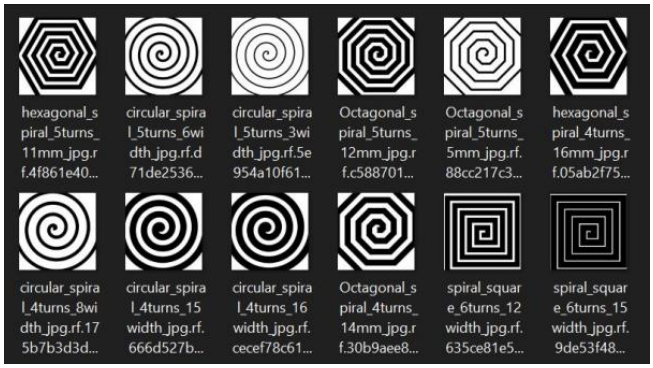

Figure 4 – Generated spiral shapes using the Turtle library scripts

Data labeling and annotation

Data labeling is an important step in dataset preparation tasks. After creating 230 different images, the annotation and labeling process was conducted using Makesense.ai, a manual online platform for annotation. Each spiral coil was labeled and annotated to categorize them into four classes. This involved perfect identification of shapes, leading to the division into specified classes, as illustrated in Figure 5, which shows the labeled classes: Hexagonal, Octagonal, Square, and Circular.

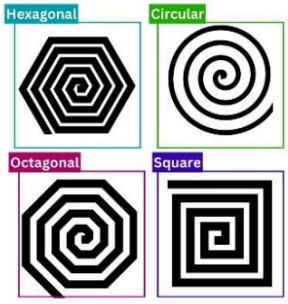

Figure 5 – Labeled classes (Hexagonal, Octagonal, Square, Circular)

Data augmentation and splitting

After the manual annotation, Roboflow, an open-source platform for data augmentation, was used to enlarge and diversify the dataset. This process expanded the dataset to a total of 547 different pictures. The techniques that were applied include blurring, rotation, flipping, and zooming. Specifically, horizontal and vertical flips, as well as 90° rotations in clockwise, counter-clockwise, and upside-down directions, were used. Preprocessing steps included auto-orientation and resizing with a center crop to 640x640 pixels. In the end, the augmented dataset was split into 87.4% for training (478 images), 8.4% for validation (46 images), and 4.2% for testing (23 images). This step is important to achieve a balanced and representative distribution of data, making it a perfect input for the AI model, as shown in Figure 6, which displays a screenshot from Roboflow showing the data split numbers with rounded percentages to 87%, 8%, and 4% for simplicity.

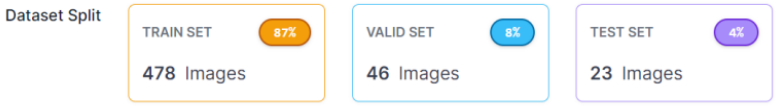

Figure 6 – Screenshot from Robotflow

YOLOv9 classification model implementation

Choice of a model

Deep learning model training requires high-performance computers to handle the training process. Google Colab found a solution for that by providing an online notebook environment that serves as an integrated

development environment (IDE) to execute the Python code directly from a browser. Colab is integrated directly with Google Drive, allowing for online file management, and comes with many machine learning libraries. The key point of the performance is the Tesla GPU acceleration, which is a suitable configuration for intensive deep learning training tasks. Figure 7 shows the specifications of the Tesla T6 GPU used.

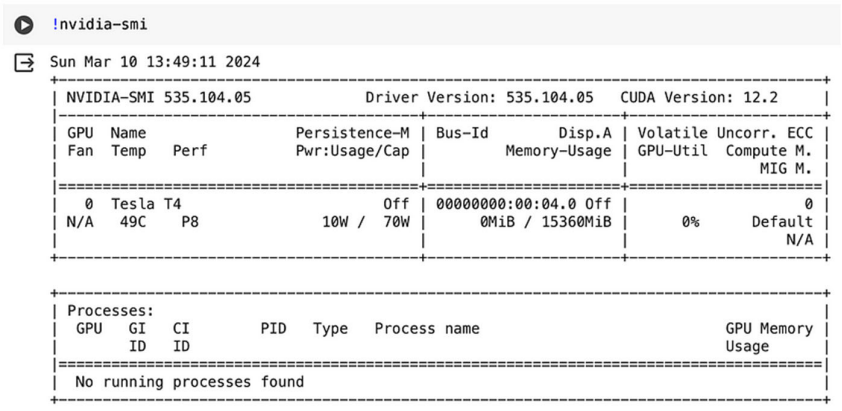

Figure 7 – Tesla T6 GPU specifications

The YOLOv9 model was implemented and programmed as a novel deep learning classification method for real-time object detection, released in February 2024. YOLOv9 came with Programmable Gradient Information (PGI), which addresses the issue of information loss through deep network layers. The new framework, PGI, generates reliable gradients using an auxiliary reversible branch, ensuring that deep features retain essential characteristics for accurate predictions. Another key improvement of this model compared to older models is the Generalized Efficient Layer Aggregation Network (GELAN), which optimizes parameter utilization and computational efficiency without affecting accuracy. It maintains lightweight and fast inference capabilities (Wang et al, 2024).

The decision to utilize YOLOv9 was made because of its status as the most recent version in the YOLO series, so it offers improved performance in terms of speed and accuracy compared to earlier versions and effectively handles complex patterned image data in classification and object detection, aligning perfectly with the needs of this study.

Training setup and hyperparameters

The model was trained with a configuration of a batch size N_{batch_size}=4, indicating that 4 data samples were processed simultaneously during each training iteration. After processing these 4 data samples, the model parameters were updated. This batch size was applied uniformly throughout each epoch, with a training duration of 50 epochs. The hyperparameters for Stochastic Gradient Descent (SGD) were as follows: momentum of 0.937 and weight decay of 0.0005. Other augmentation techniques include blur, median blur, gray conversion, and CLAHE (Contrast Limited Adaptive Histogram Equalization). This augmentation introduces further specified variations and it helps the model to better generalize new, unseen data by exposing it to a wider range of data distortions. It aims to reduce overfitting and it fine-tunes the model's ability to be adapted with different data conditions while this augmentation is often applied automatically as part of the data preprocessing pipeline of the YOLOV9 model. The model architecture comprised 930 layers with 60,804,152 parameters, and TensorBoard was used for process logging and visualization. The model has a perfect focus on the recognition of geometrical patterns over color. Even though the dataset consisted of black and white synthetic images, the model learned the spatial arrangements of pixels that form pattern shapes during training. Thus, it is capable of detecting shapes without color differences and can identify high-level features such as shapes, edges, and patterns.

Model evaluation indicators

To observe and evaluate the model's performance in detecting various classes of shapes, different evaluation indicators are used. The primary key metric is the Mean Average Precision (mAP), which yields a total value by averaging the aggregation of the precision-recall curves from each class. Precision is the measurement of positive predictions. Accuracy and recall detect all relevant instances of the coil shapes by measuring them. The F1 Score is also used, which is the harmonic mean of precision and recall, indicating how reliable the model is. It ranges between 0 and 1. mAP was used for the evaluation, computing the AP of each class and the average over several classes. These metrics are good indicators of the YOLOv9 model's ability to detect and classify objects, and they are represented mathematically as:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{4}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negatives} \quad (5)$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (6)$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \quad (7)$$

where N is the number of classes and AP_i represents the average precision for the class i.

Geometric parameter extraction using OpenCV

Overview of OpenCV

After using the deep learning model to detect the classes of different spiral coils, the next step is processing each image of the detected shape separately using OpenCV (Open-Source Computer Vision Library), a widely-used computer vision library (Xie & Lu, 2013). It provides a set of tools and functions for processing image and motion data (Mishra et al, 2022) at the pixel scale. OpenCV supports multiple techniques such as filtering, edge detection (Xie & Lu, 2013), and contour detection (Ni et al, 2016), leading to efficient extraction of geometric parameters. Images of the spiral coils were captured using a phone camera from the top view. The following image shows the setup with the phone camera capturing the spiral coils from above. These images were then processed using OpenCV. Figure 8 shows the setup with a fixed phone capturing the spiral coils, which were then processed using OpenCV to experiment with geometry extraction and calculation.

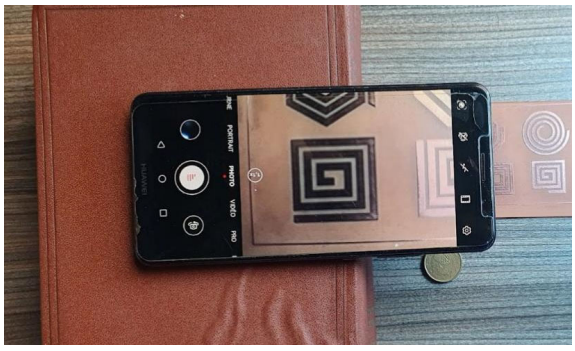

Figure 8 – Fixed phone capturing a spiral coil from above

Planar coils images preprocessing

OpenCV has a large collection of functions, with over 500 functions in computer vision that can be applied in different fields for various purposes (Mohamad et al, 2015). Specific functions are used in our research to calculate different distances and geometric parameters. These functions are described in the following steps:

Grayscale conversion and noise reduction with Gaussian blur

After capturing the image using a camera phone, the image will be loaded for processing. The first treatment involves converting this colored RGB image to a grayscale image by eliminating color information. This reduces unnecessary information from the image and decreases complexity, using the function cv2.cvtColor(). Once the image is converted to grayscale, Gaussian blur is applied to enhance the accuracy of the subsequent thresholding operation and minimize the impact of noise by smoothing the image. This is achieved using the function cv2.GaussianBlur(), which performs a convolution operation with a gaussian kernel of parameter size (9, 9) and a standard deviation (σ) of 10 to control the amount of blur. This value of kernel size was selected to achieve an optimal balance between noise reduction and detail preservation so it reduces high-frequency noise and prevents essential structural details, particularly important for the intricate patterns of planar coils. The standard deviation was set to 10 to ensure extensive smoothing for broader impact, minimizing digital noise distortions.

Figure 9 shows a grayscale image of the planar coil after the application of this process.

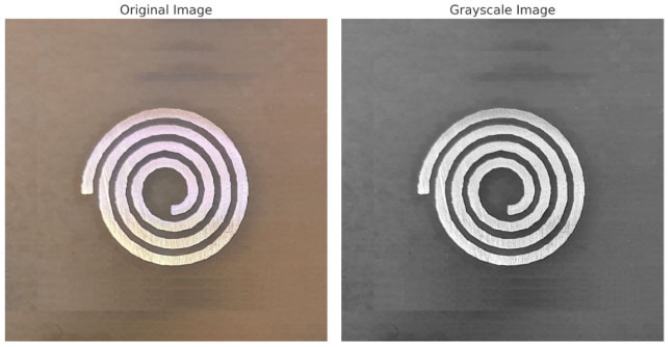

Figure 9 – Color image and its grayscale version

Binarization using Otsu's thresholding

The blurred grayscale image is then converted to a binary image using Otsu's thresholding method (Otsu, 1979). This adaptive thresholding technique calculates an optimal threshold value to separate the foreground coil image from the background, ensuring effective binarization. The function cv2.threshold() is used for this purpose, with the parameters including the blurred grayscale image an initial threshold value of 128 that serves as a starting point that typically works well for images with fair contrast and provides a balanced approach for separating the foreground from the background, and a maximum value of 255 for the binary thresholding type. Binary thresholding is applied to the grayscale image to produce a binary image where pixel values are set to either 0 or 255 based on the determined threshold. Otsu's method works by maximizing the between-class variance $\sigma_b^2(t)$, which is expressed as:

$$\sigma_b^2(t) = w_0(t)w_1(t)[\mu_0(t) - \mu_1(t)]^2 \tag{8}$$

where $w_0(t)$ and $w_1(t)$ are the class probabilities, and $\mu_0(t)$ and $\mu_1(t)$ are the means of the two classes. This method ensures the best separation between the background and the foreground in an image by maximizing the variance between these two classes.

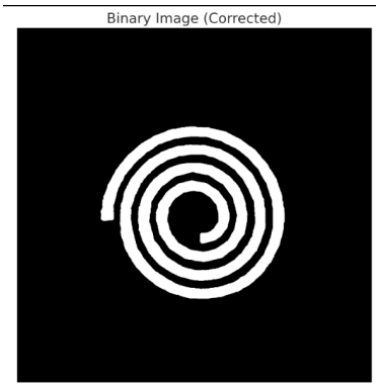

Figure 10 – Circular coil image after binarization

Contour detection and drawing

After obtaining a binary image with only white and black pixels, it is now easy to draw and detect contours. This helps identify the boundaries

of the spiral coil. The function cv2.findContours() is applied to the binary image, and then the contours are drawn and plotted on the original image using the cv2.drawContours() function, which highlights a green contour of the detected shape.

Figure 11 illustrates the circular coil image after the process of contour detection and drawing.

Figure 11 – Circular coil image after contour detection and drawing

Intersection points and parameters calculation

The measurement of specific geometries and extracting distances requires a logic to be implemented as an algorithm. For this purpose, guiding lines were designed to aid in calculating specific geometries by utilizing the cv2.line() drawing function. These lines make the intersection points with the contour. A custom-developed function detects these intersections between the guide line and the contour of the spiral coils, and these points are recorded and enumerated.

The first parameter is the width of the coil, which is calculated by the distance between the first and second intersection points along the guiding line. d_{out} represents the outer diameter of the spiral coil, which is calculated by the distance between the leftmost and rightmost intersection points. d_{in} is the inner diameter, and the logic to calculate it involves finding the first right point and the first left point near the middle of the guiding line. The number of turns n is calculated since each set of four intersection points makes one turn, so it is calculated by dividing the total number of points by four.

Figure 12 displays the measurements annotated in pixels.

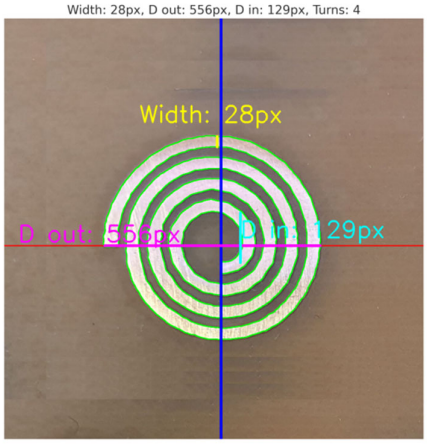

Figure 12 – Circular planar coil with pixel-extracted geometry

These geometric calculations serve as a conceptual framework for analyzing spiral coils:

$$Width = distance(i_1, i_2) \tag{9}$$

$$d_{in} = distance\left(i_{\,first\ left\ near\ middle}, i_{\,first\ right\ near\ middle}\right) \tag{10}$$

$$d_{out} = distance\left(i_{\,leftmost}, i_{\,rightmost}\right) \tag{11}$$

$$n = \left|\frac{number\ of\ intersection\ points}{4}\right| \tag{12}$$

where i_1 and i_2 represent the first and second intersection points along the guiding line, $i_{\,leftmost}$ and $i_{\,rightmost}$ are the extremal points on the left and right sides, $i_{\,first\ left\ near\ middle}$ and $i_{\,first\ right\ near\ middle}$ refer to the closest intersection points to the center of the object on each side.

Pixel-to-millimeter conversion

After extracting the required geometrical parameters of the spiral coil shapes in pixels, they should be converted to real-world units, which are millimeters. This conversion is done using a pixel-to-millimeter ratio. To calculate this ratio, a reference object with a known size is used. For this, a 10 euro cent coin was used; the diameter of the coin is 19.75 mm, so it serves as the basis for conversion. An important point to ensure the

consistency and reliability of the measurement taking is that both the reference object and the spiral coil should be captured in the same condition of the camera angle from the top and at the same distance too. The coin image processing begins by converting it to grayscale, then applying Gaussian Blur. This step reduces image noise and smooths the edges with a kernel size of (5, 5). This kernel size was selected based on preliminary tests that indicated it minimized edge distortion while adequately preserving the necessary details for accurate edge detection. Next, the Canny edge detector is applied to identify the strong edges while ignoring weaker ones with the parameters Lower Threshold: 50 and Upper Threshold: 150. These thresholds were empirically determined to provide the best balance between sensitivity to true edges and immunity to noise-induced false edges, as validated by repeated trials under varying imaging conditions. Then, the diameter can be calculated based on the last result of this image processing using a script that can detect the extreme left and right edges of the coin image. Measure the distance between these points using the Euclidean distance formula and finally, using the actual diameter of the coin and the measured diameter in pixels, calculate the conversion ratio from pixels to millimeters.

$$distance(pixels) = \sqrt{(x_{right} - x_{left})^2 + (y_{right} - y_{left})^2} \quad (13)$$

$$Ratio(px/mm) = \frac{distance(pixels)}{Actual\ Diameter\ (mm)} \quad (14)$$

where x_{right} and y_{right} are the coordinates of the rightmost point, and x_{left} and y_{left} are the coordinates of the leftmost point, measuring the maximum span across their widest points to get the diameter. Figure 13 demonstrates the coin image processing used for converting pixels to millimeters.

Figure 13 – Coin image processing for pixel-to-millimeter conversion

The obtained pixel-to-mm ratio is utilized to convert various parameter values of the spiral coil for all captured shapes. In Table 2, these values are presented with the conversions, allowing them to be used as inputs in Mohan's Law to compute the inductance.

Table 2 – Geometric values of different shapes: measurements in millimeters and pixels

Shape	Number of turns (n)	Width (mm)	Width (px)	d_{out} (mm)	d_{out} (px)	d_{in} (mm)	d_{in} (px)
Circular	4	0.63	28	12.61	556	2.92	129
Hexagonal	3	0.77	34	10.34	456	3.43	151
Octagonal	3	1.18	52	14.01	618	1.72	76
Square	3	1.09	48	11.95	527	2.86	126

Results

The results section is divided into two parts: the first evaluates the YOLOv9 classification model performance, and the second is about geometric parameter and inductance value calculation validation.

Model performance

The model indicates an improvement in the accuracy of bounding box coordinates and object class detection. This indication is proven by the consistent decrease in box loss, classification loss, and focal loss, across 50 epochs. In Table 3, the results demonstrate that the model is effectively learning, with performance improvement evident. The mAP values at an intersection over Union (IOU) 0.5 and other thresholds from 0.5 to 0.95 remain high and stable. The maximum recorded values for precision, recall, and F1 score are 0.989, 1.000, and 0.995, respectively.

Table 3 – Performance variability of the YOLOv9 model during training

Metric	25th Percentile	Median	75th Percentile	Max
Precision	0.579	0.881	0.982	0.989
Recall	0.541	0.973	1	1
F1 Score	0.532	0.882	0.991	0.995
mAP (IoU=0.5)	0.496	0.988	0.995	0.995
mAP (IoU=0.5:0.95)	0.264	0.83	0.89	0.913

The F1-Confidence curve reflects that the model has strong predictive capabilities for different shape classes. This curve is a graphical representation that illustrates the relationship between the confidence level of predictions and the corresponding F1 score, and it combines the precision level and the recall into a single metric, achieving an optimal macro-average F1 score of 0.98 at a confidence threshold of 0.501 with a balanced precision and recall. The Octagonal and Circular classes show a well-balanced trade-off between precision and recall. The Hexagonal class also performs impressively well, with high F1 scores at lower levels of confidence. The class of Square, on the other hand, has a bit of fluctuation at high confidence but gives good F1 scores. Figure 14 illustrates the F1-Confidence curves for various coil shapes.

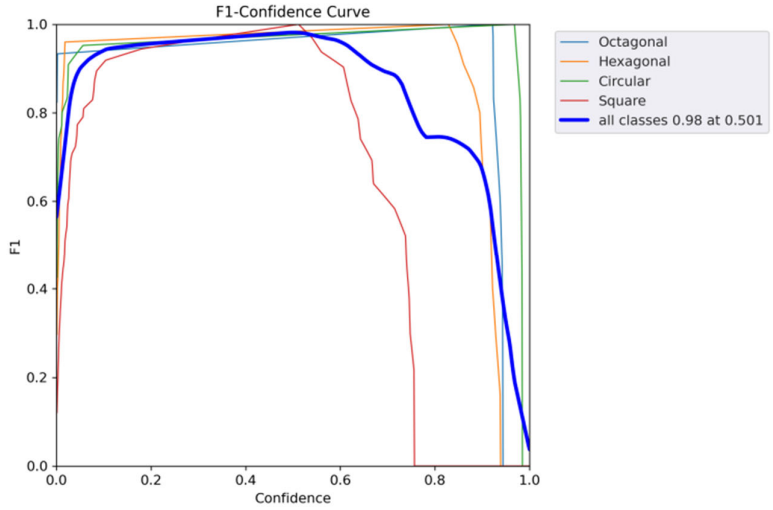

Figure 14 – F1-Confidence curves for different coil shapes

After experimental testing, the model demonstrated that a Square coil was detected with a confidence level of 0.91, a Circular coil with a confidence level of 0.80, a Hexagonal coil with a confidence level of 0.60, and an Octagonal coil with a confidence level of 0.65. In this context, the confidence level represents the model's degree of certainty in correctly identifying and classifying the detected object as belonging to a specific geometric shape class as a final prediction level. These confidence levels, which are directly related to the model's classification task in the image, indicate how certain the model is that the identified shape corresponds to

its labeled class, such as Square, Circular, Hexagonal, or Octagonal. These results confirm the robustness and reliability of the model in categorizing different geometric shapes. Figure 15 presents the detection confidence levels for various coil shapes.

Figure 15 – Detection confidence scores for coil shapes

Geometric parameter and inductance value validation

A comparison is made between the extracted inductance values calculated based on the design geometry created with the computer aided design (CAD) software, and the values computed from the results of OpenCV. Table 4 shows the difference between the value sets of different shapes.

Table 4 – Inductance measurement comparison across different geometric shapes based on CAD and computer vision (CV) data

	d_{out}		d_{in}			
Shape	CAD (mm)	CV (mm)	CAD (mm)	CV (mm)	Inductance CV (nH)	Inductance CAD (nH)
Circular	12.5	12.61	3	2.92	113.24	114.13
Square	11.8	11.95	2.8	2.86	73.09	72.80
Octagonal	14.07	14.01	1.73	1.72	56.73	55.20
Hexagonal	10.3	10.34	3.3	3.43	65.08	63.12

The accuracy of the geometric parameters extracted by OpenCV is remarkably high, leading to good accuracy in the estimated inductance value. These results underline the effectiveness of our high-resolution measurement techniques, which are essential for in-depth assessments in precision engineering contexts and quality control during construction. Figure 16 shows the inductance value for a circular coil geometry, where the CV system measured 113.24 nH, closely matching the CAD value of 114.13 nH. At the same time, Figure 17 presents the hexagonal coil geometry, with an inductance of 65.08 nH measured by CV, aligning well with the CAD value of 63.12 nH. Figures 18 and 19 display the octagonal and square geometries, with inductance values of 56.73 nH and 73.09 nH, respectively, both showing good agreement with their CAD values.

The difference with the values of the computer vision method and the inductance value of design geometry may be explained by the fact that, in realistic coil manufacturing, physical dimensions and software design values may not be exactly identical to each other. Deviations from the manufacturing process and small deviations in physical dimensions can result in theoretical design parameters that differ from actual measured values. Overall, the results show good agreement between the inductance values. Minor inconsistencies, as indicated, may be expected with the granularity of the pixel-based measurements too and have little or no impact on the overall inductance calculations while working with nanohenries. This means that the computer vision method can be relied upon to establish inductance values accurately. Figures 16 to 19 present the results of the analysis and inductance calculations for various planar coil geometries using OpenCV.

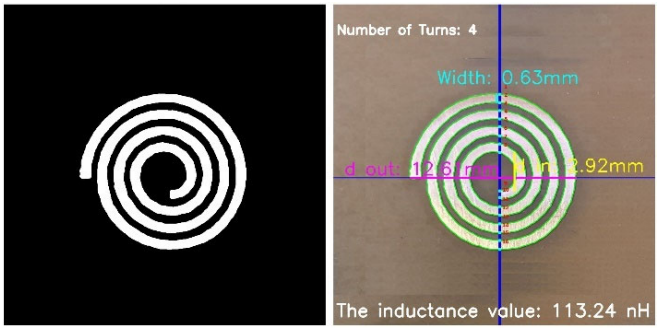

Figure 16 – Analysis of circular planar coil geometry and inductance calculation with OpenCV

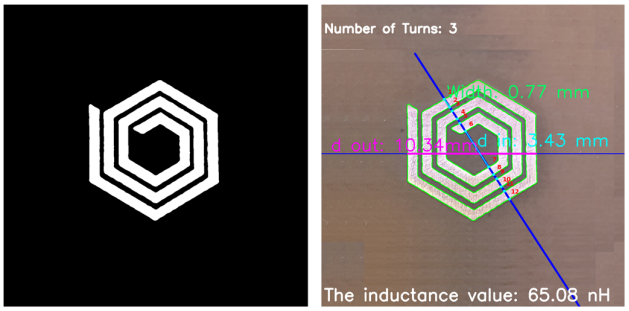

Figure 17 – Analysis of hexagonal planar coil geometry and inductance calculation with OpenCV

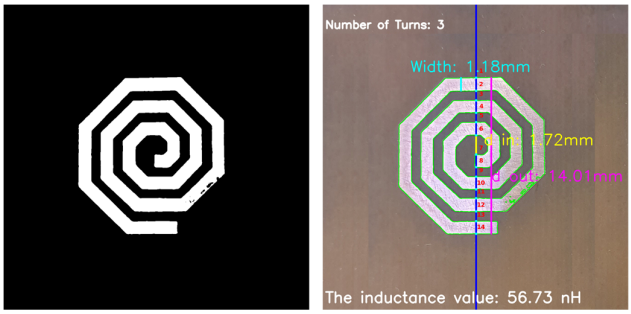

Figure 18 – Analysis of octagonal planar coil geometry and inductance calculation with OpenCV

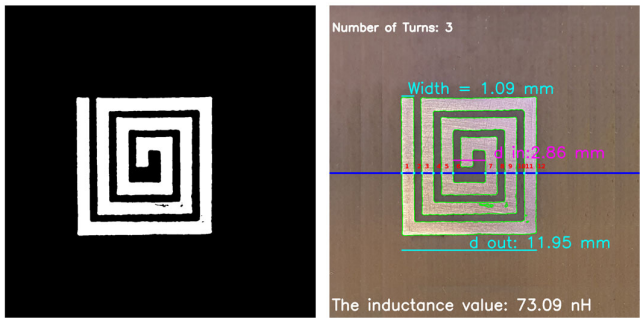

Figure 19 – Analysis of square planar coil geometry and inductance calculation with OpenCV

Conclusion

The work demonstrates how computer vision and deep learning techniques can be useful in assisting the calculation of inductance in the case of planar coils. Utilizing YOLOv9 for shape recognition and OpenCV to extract geometric parameters, the proposed methodology resulted in a way such that it performs with a high accuracy rating. This, in a way, makes the method reliable for industrial purposes where more precision in the manufacturing of a component is required. In further work, this model can be designed to work with more coil shapes, hence reducing the discrepancy in the measurements in the process. An extension of this work could be in integrating this computer vision with real-time monitoring systems for further advancement in automated manufacturing systems.

References

Abu Alhaija, H., Mustikovela, S.K., Geiger, A. & Rother, C. 2019. Geometric image synthesis. In: Jawahar, C., Li, H., Mori, G. & Schindler, K. (Eds.) *Computer Vision – ACCV 2018. ACCV 2018. Lecture Notes in Computer Science*, 11366, pp.85-100. Cham: Springer. Available at: https://doi.org/10.1007/978-3-030-20876-9_6.

Ahire, D.B., Gond, V.J. & Chopade, J.J. 2022. Geometrical parameter optimization of planner square-shaped printed spiral coil for efficient wireless power transfer system to biomedical implant application. *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, 2, art.number:100045. Available at: https://doi.org/10.1016/j.prime.2022.100045.

Anderson, E.F. 2018. Turtle Fractals and Spirolaterals: Effective Assignments for Novice Graphics Programmers. In: *Eurographics 2018*, Delft, The Netherlands, pp.39-42, April 20 [online]. Available at: https://eprints.bournemouth.ac.uk/30590/ [Accessed: 05.06.2024].

Chien, C.-T., Ju, R.-Y., Chou, K.-Y. & Chiang, J.-S. 2024. YOLOv9 for fracture detection in pediatric wrist trauma X-ray images. *Electronics Letters*, 60(11), e13248. Available at: https://doi.org/10.1049/ell2.13248.

Couraud, B., Deleruyelle, T., Vauche, R., Flynn, D. & Daskalakis, S.N. 2020. A low complexity design framework for nfc-rfid inductive coupled antennas. *IEEE Access*, 8, pp.111074-111088. Available at: https://doi.org/10.1109/ACCESS.2020.3001610.

Derkaoui, M., Benhadda, Y., Hamid, A. & Temmar, A. 2021. Design and Modeling of Octagonal Planar Inductor and Transformer in Monolithic Technology for RF Systems. *Journal of Electrical Engineering & Technology*, 16(3), pp.1481-1493. Available at: https://doi.org/10.1007/s42835-021-00692-x.

Im, J.-H. & Hur, J. 2021. Proposing new planar-type search coil for permanent magnet synchronous motor: Design and application for position estimation. *IEEE Access*, 9, pp.129078-129087. Available at: https://doi.org/10.1109/ACCESS.2021.3113384.

Kharbouch, H., Hamid, A., Lebey, T., Bley, V., Havez, L. & Combette, C. 2017. Using the variable width in a planar inductor on Kapton for optimizing its performance. *Turkish Journal of Electrical Engineering and Computer Sciences*, 25(5), pp.3798-3810. Available at: https://doi.org/10.3906/elk-1606-343.

Luo, Z. & Wei, X. 2017. Analysis of Square and Circular Planar Spiral Coils in Wireless Power Transfer System for Electric Vehicles. *IEEE Transactions on Industrial Electronics*, 65(1), pp.331-341. Available at: https://doi.org/10.1109/TIE.2017.2723867.

Mishra, S., Verma, V., Akhtar, N., Chaturvedi, S. & Perwej, Y. 2022. An Intelligent Motion Detection Using OpenCV. *International Journal of Scientific Research in Science, Engineering and Technology*, 9(2), pp.51-63. Available at: https://doi.org//10.32628/IJSRSET22925.

Mohamad, M., Saman, M.Y.M. & Hitam, M.S. 2015. A Review on OpenCV. *ResearchGate*, August Available at: https://doi.org/10.13140/RG.2.1.2269.8721.

Mohan, S.S., del Mar Hershenson, M., Boyd, S.P. & Lee, T.H. 1999. Simple accurate expressions for planar spiral inductances. *IEEE Journal of Solid-State Circuits*, 34(10), pp.1419-1424. Available at: https://doi.org/10.1109/4.792620.

Mostafa, S.A.M., Wang, J., Holt, B. & Wang, J. 2024. YOLO based Ocean Eddy Localization with AWS SageMaker. *arXiv:2404.06744v1*, 10 April. Available at: https://doi.org/10.48550/arXiv.2404.06744.

Ni, J., Khan, Z., Wang, S., Wang, K. & Haider, S.K. 2016. Automatic detection and counting of circular shaped overlapped objects using circular hough transform and contour detection. In: *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, Guilin, China, pp.2902-2906, June 12-15. Available at: https://doi.org/10.1109/WCICA.2016.7578268.

Otsu, N. 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), pp.62-66. Available at: https://www.doi.org/10.1109/TSMC.1979.4310076.

Paulin, G. & Ivasic-Kos, M. 2023. Review and analysis of synthetic dataset generation methods and techniques for application in computer vision. *Artificial Intelligence Review*, 56(9), pp.9221-9265. Available at: https://doi.org/10.1007/s10462-022-10358-3.

Richardson, E., Sela, M. & Kimmel, R. 2016. 3D Face Reconstruction by Learning from Synthetic Data. In: *2016 Fourth International Conference on 3D Vision (3DV)*, Stanford, CA, USA, pp.460-469, October 25-28. Available at: https://doi.org/10.1109/3DV.2016.56.

Wang, C.-Y., Yeh, I.-H. & Liao, H.-Y.M. 2024. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. *arXiv:2402.13616v2*, 29 February. Available at: https://doi.org/10.48550/arXiv.2402.13616.

Xie, G. & Lu, W. 2013. Image Edge Detection Based On Opencv. *International Journal of Electronics and Electrical Engineering*, 1(2), pp.104-106. Available at: https://doi.org/10.12720/ijeee.1.2.104-106.

Un enfoque de visión por computadora con OpenCV y aprendizaje profundo para determinar la inductancia en bobinas planas

Younes Benazzouz, **autor de correspondencia**, *Djilalia* Guendouz

Universidad de Orán 2 Mohamed Ben Ahmed, Instituto de Seguridad y Mantenimiento Industrial (IMSI), Departamento de mantenimiento de instrumentación, Laboratorio de Ingeniería de Producción y Mantenimiento Industrial (LGPMI), Orán, República Argelina Democrática y Popular

CAMPO: ciencias de la computación, electrónica
TIPO DE ARTÍCULO: artículo científico original

Resumen:

Introducción/propósito: En el ámbito del desarrollo y uso de metodologías de visión por computadora e inteligencia artificial, esta investigación presenta una combinación y un método avanzado que utiliza YOLOv9, un concepto de aprendizaje profundo de procesamiento de imágenes completas en un solo paso a través de una red neuronal convolucional (CNN) y la Biblioteca de procesamiento de imágenes OpenCV Python para determinar la geometría de bobinas planas. Estos parámetros geométricos son los principales parámetros utilizados para calcular el valor de inductancia utilizando la fórmula de Mohan, que utiliza exclusivamente datos geométricos para calcular los valores de inductancia. Este método acelera significativamente los procesos de verificación y cálculo, al mismo tiempo que contribuye a mejorar el control de calidad después de la fabricación.

Métodos: La metodología se divide en dos fases principales. Inicialmente, se entrenó un modelo YOLOv9 para el reconocimiento de objetos utilizando un conjunto de datos sintéticos generados de formas de bobinas creadas con la biblioteca de gráficos Turtle de Python. Luego, después de la fase de detección, se utilizó OpenCV para identificar los parámetros geométricos de las imágenes. Los píxeles se convirtieron a milímetros utilizando un método de relación para calcular el valor de inductancia con precisión.

Resultados: El modelo YOLOv9 identificó con éxito varias formas de bobinas planas y los parámetros geométricos se identificaron mediante OpenCV. Posteriormente, se calculó con éxito la inductancia.

Conclusión: Los resultados muestran que el método propuesto es una forma novedosa y eficaz de calcular la inductancia.

Palabras claves: redes neuronales convolucionales (CNN), OpenCV, bobina plana, inductancia, YOLOv9, procesamiento de imágenes.

Применение компьютерного зрения с использованием OpenCV и глубокого обучения для определения индуктивности плоских катушек

Юнез Беназзоуз, **корреспондент**, Джилалия Гуендоуз

Университет Орана 2 Мохамед Бен Ахмед, Институт технического обслуживания и промышленной безопасности (IMSI), кафедра технического обслуживания контрольно-измерительных приборов, Лаборатория производственного инжиниринга и промышленного обслуживания (LGPMI),
г. Оран, Алжирская Народная Демократическая Республика

РУБРИКА ГРНТИ: 28.23.37 Нейронные сети,
20.23.25 Информационные системы с базами знаний,
47.09.29 Полупроводниковые материалы
ВИД СТАТЬИ: оригинальная научная статья

Резюме:

Введение/цель: В свете развития и использования методологий компьютерного зрения и искусственного интеллекта данное исследование представляет собой комбинированный и продвинутый метод, использующий YOLOv9, концепцию глубокого обучения для обработки полного изображения за один проход через свёрточную нейронную сеть (CNN) и библиотеку обработки изображений OpenCV Python для определения геометрии плоских катушек. Данные геометрические параметры являются главными в расчёте значений индуктивности с использованием формулы Мохана, которая использует геометрические данные исключительно для оценки значений индуктивности. Этот метод значительно ускоряет процессы верификации и расчёта, а также способствует повышению качества контроля производства.

Методы: Методология исследования распределена на два основных этапа. Сначала была обучена модель YOLOv9 для распознавания объектов, используя сгенерированный синтетический набор данных форм катушек, созданный с помощью библиотеки Turtle Graphics в Python. Затем, после этапа обнаружения, с помощью OpenCV были идентифицированы геометрические параметры изображений. Пиксели были преобразованы в миллиметры, используя метод соотношения для точного расчета значения индуктивности.

Результаты: Модель YOLOv9 успешно идентифицировала различные формы плоских катушек, а геометрические параметры были определены с помощью OpenCV. Индуктивность также была успешно рассчитана.

Выводы: Результаты показывают, что предложенный метод является инновационным и эффективным способом расчета индуктивности.

Ключевые слова: сверточные нейронные сети (CNN), OpenCV, плоская катушка, индуктивность, YOLOv9, обработка изображений.

Примена рачунарског вида помоћу OpenCV и дубоког учења за одређивање индуктивности у плоснатим намотајима

Јунез Беназуз, **аутор за преписку**, *Дајлија* Гундуз

Универзитет Оран 2 Мохамед Бен Ахмед, Институт за индустријско одржавање и безбедност (IMSI), Одељење за одржавање инструментације, Лабораторија за производно машинство и индустријско одржавање (LGPMI), Оран, Народна Демократска Република Алжир

ОБЛАСТ: рачунарске науке, електроника
КАТЕГОРИЈА (ТИП) ЧЛАНКА: оригинални научни рад

Сажетак:

Увод/циљ: У области развоја и примене методологија рачунарског вида и вештачке интелигенције, ово истраживање представља комбинацију и напредну методу која користи YOLOv9 – концепт дубоког учења за обраду целокупне слике у једном пролазу кроз конволуциону неуронску мрежу (CNN) и библиотеку за обраду слика OpenCV на Pythonu за одређивање геометрије плоснатих намотаја. Ови основни геометријски параметри користе се за израчунавање вредности индуктивности помоћу Моханове формуле, која искључиво употребљава геометријске податке за процену вредности индуктивности. Ова метода знатно убрзава процесе верификације и израчунавања, а побољшава и контролу квалитета после производње.

Методе: Методологија је подељена на две главне фазе. У почетку је модел YOLOv9 био конструисан за препознавање објеката коришћењем генерисаног синтетичког скупа података облика намотаја створеног помоћу Pythonove библиотеке Turtle Graphics. Затим, након фазе детекције, OpenCV је коришћен за идентификацију геометријских параметара слика. Пиксели су претворени у милиметре применомметоде пропорција за тачно израчунавање вредности индуктивности.

Резултати: Модел YOLOv9 је успешно идентификовао различите облике плоснатих намотаја, а геометријски параметри су идентификовани путем OpenCV. Након тога, индуктивност је успешно израчуната.

Закључак: Резултати показују да је предложена метода нов и ефикасан начин за израчунавање индуктивности.

Кључне речи: конволуционе неуронске мреже (CNN), OpenCV, плоснати намотај, индуктивност, YOLOv9, обрада слика.

Benazzouz, Y. et al, A computer vision approach with OpenCV and deep learning for determining inductance in planar coils, pp.1645-1670

Paper received on: 06.06.2024.
Manuscript corrections submitted on: 16.11.2024.
Paper accepted for publishing on: 18.11.2024.