Prognosis of Electronic Systems Based on Artificial Intelligence : Convolutional Neural Network

Abdelghani CHAHMI[1]

[1]Electrical Drives Laboratory University of Sciences and Technology Oran, Algeria

Email: chahmi.abdelghani@gmail.com

Abstract - This article explores the contribution and opportunity of implementing advanced techniques (Artificial Intelligence) in the field of prediction and maintenance of electronic charts. A smart model capable of detecting missing components on a faulty motherboard is presented. This skill is extremely valuable in the IT and electronics industry, where automation is common.

Keywords - Convolutional neural network, YOLOv8, Machine learning, Datasets.

I. INTRODUCTION

Large industrial systems are widely confronted to the challenges of reliability and availability of the production device [1]. In fact, it is to be found in all the industrial fields and, particularly, in the growth industries like chemistry, nuclear power, aeronautic, or railway transports [2]. This work of thesis is a part of the thematic of monitoring and fault diagnosis of the squirrel cage three-phase induction machine [3].

The primary manufactured neural network was created within the late 1950s with the coming of perceptron calculations [4]. Be that as it may, the confinements related to the computational taken a toll of these calculations amid this period, as well as the often-discussed attestation by Minsky and Papert [5] that their failures to memorize show that they cannot learn.

Learning isn't inconceivable. The lesson to be learned from this perception is that it would not have been conceivable without the expanding computing control of computers, without effective programming dialects that permitted the advancement of idealize programs. In expansion, all computer researchers know that the quality and amount of information utilized for learning are essential for the calculation to be solid and precise.

Innovative advancements within the upkeep of electronic sheets have made many opportunities for issue fathoming and numerous operational changes. The execution of progressed procedures (counterfeit insights) is always advancing within the field of computing control and data innovation. The topology of the learning design given by each layer and level is displayed [6] for an improved understanding. In this work, we propose a commonsense arrangement to upkeep issues (repair of electronic cards) due to need of specialized repair documentation. The prescribed arrangement is based on learning to superior coordinate the most effective calculations utilized within the areas of pharmaceutical and visual imaging. This learning makes a difference us characterize YOLOv8 (You See As it were Once) as one of the foremost prevalent question discovery algorithms within the field of Deep Learning. This information is critical to get it the design arrangement demonstrates. The significance of this original work is the usage of devices committed to vision and picture within the field of blame discovery in analog circuits of motherboards.

II. PROCEDURES AND APPROACHES

A) Convolutional Neural Network

CNN is very effective in image processing task and has been widely applied in fields such as computer vision, face recognition, image classification, etc.

CNN's architecture is based on the idea of simulating. the function of the human visual cortex, which specializes in processing visual information.The network is made up of many layers of neurons, with specific operations occurring at each layer. See the overview of CNN.

The architecture of a CNN is based on the idea of imitating the functioning of the human visual cortex, which is specialized in processing visual information. The network is composed of several layers of neurons, each performing specific operations. The main layers typically found in a CNN are

A.1. Convolution Layer

This layer performs a convolution operation on the input image using multiple filters. Each filter detects a specific characteristic of the image, such as edges, textures, or shapes. The filters are weight
matrices learned during the network's training phase.
After convolution, a pooling layer is generally added to reduce the image's spatial dimension by locally sampling regions of the image. This reduces the number of network parameters and makes the model more tolerant to translations and small variations in the data.

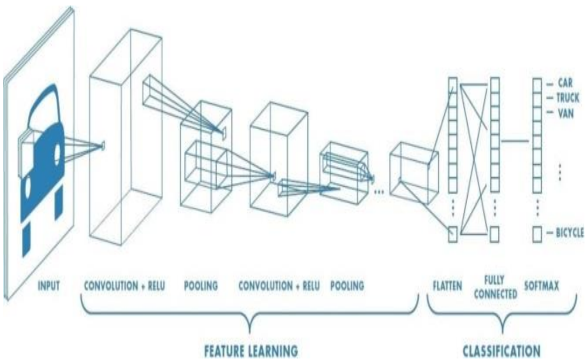

Fig. 1. General description of CNN.

After each convolution or pooling operation, a non-linear activation function is applied to each neuron. This enhances the network's ability to learn complex relationships between the extracted features. Fully connected layers are also included. After several convolutional and pooling layers, fully connected layers can be added, similar to those of a classic neural network.

These layers take the previously extracted characteristics and use them to perform a classification or prediction specific to the task.

During the training phase, the convolutional neural network learns to adjust the weights of the filters to minimize a certain loss function. This is usually done using an optimization technique such as gradient backpropagation. The CNN used in this application is based on the famous YOLOv8 architecture developed by Ultralytics. You Only Look Once (YOLO) are a widely used real-time object detection algorithm in computer vision. Unlike traditional methods that perform object detection in multiple steps, YOLO performs detection in a single pass.

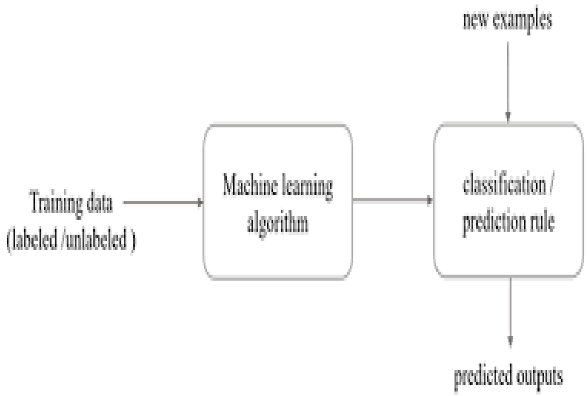

Fig. 2. General procedure for machine learning (4)

A.2. You Only Look Once (YOLO)

Is a real-time object detection algorithm widely used in the field of computer vision? You Only Look Once (YOLO) are a real-time object detection algorithm widely used in the field of computer vision. Unlike traditional methods that perform object detection in multiple stages, YOLO achieves detection in a single pass.
The YOLO approach divides the input image into a regular grid and predicts both the bounding boxes and the classes of the objects present in each cell of the grid. Each bounding box is represented by four values: the x and y coordinates of the upper-left corner of the box, its width, and its height. The model also predicts the probabilities of different object classes for each box. One key characteristic of YOLO is its speed.

Given that object detection is performed in a single pass, YOLO can detect objects in real-time with high speed.

In addition, YOLO incorporates advanced techniques such as the use of deep convolutional neural networks, pooling layers, and deconvolution layers to extract relevant features from the image and improve the accuracy of detection.

YOLO has been used in various applications, including object detection in surveillance videos, autonomous vehicle detection, object recognition in medical images, and more.

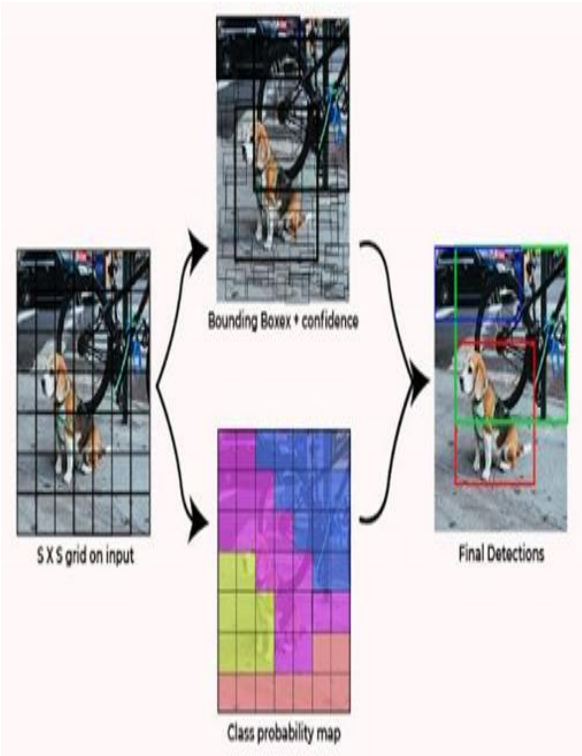

Fig. 3. Vision on YOLOv8.

Data sets play an important role in training and evaluating convolutional neural networks (CNN). They provide examples where the model learns to detect patterns and make predictions. Below are the most commonly used datasets.

A.3. Hyperparameter regulation

The hyperparameters are settings that the user defines before the initiation process, rather than settings that the modem itself knows. The mode's behavior and performances are influenced by them. The learning rate is one of the most important hyperparameters.

It is possible that the mode may not converge or stabilize if the learning rate is too high and the weight adjustments are too significant. On the other hand, if the rate of learning is too low, training may be very slow and it may take a long time for the model to perform well.

It is common practice to modify the learning rate during training in order to enhance the model's performance. As training progresses, for instance, the learning rate can be decreased, allowing for a more precise affinement of the model's weight toward the end of training.

Fig. 4. Global view of the datasets.

This strategy might promote better convergence. prevent excessive oscillations and obtain an One crucial step in the model's entrainment process is the adjustment of the hyperparameters. To determine the ideal combination of hyperparameters that maximize the model's performance on the given game, it might be necessary to conduct iterative tests with varying values. This can be accomplished using methods like Bayesian optimization or grille search.

B) Motherboard

A motherboard, also known as a system board, is one of the essential components of a computer. It is the main hub from which all other hardware components are connected and interact with each other. The motherboard serves as a base plate on which the processor, RAM memory, expansion cards, storage peripherals such as hard drives and SSDs, as well as other essential hardware components are installed. It also provides the necessary connections and interfaces for data transfer and communication between these different components.

1. Power Connectors: Processor Socket : A location where the processor is installed.

2. RAM Memory Slots: The slots where RAM is installed. RAM Memory Slots:

3. The connectors that provide power to the motherboar

4. The chipset consists of chips that control communication between the processor, memory, and other peripherals. Additionally, there are expansion connectors.

5. The locations for connecting extension cards, such as graphics cards, sound cards, etc., and the connectors for SATA hard drives and SSDs are provided.

6. These are additional power connectors for graphics cards and other power-hungry peripherals.USB Connectors : These are connectors for USB peripherals.

7. Ethernet Connectors : These are connectors used for Ethernet network connections.

8. Audio Connectors : The connectors for the buttons, LEDs, and front panel ports of the computer case.

9. CMOS battery : A battery that powers the CMOS circuit, which stores the BIOS configuration parameters.

10. Real-time clock (RTC): Connectors for the cooling system fans are located at point.

11. The power circuit is located at point.

12. A circuit that maintains the current time and date. The power circuit is located at point.

13. The circuits that regulate and distribute electrical power to all components of the motherboard. Modifier.

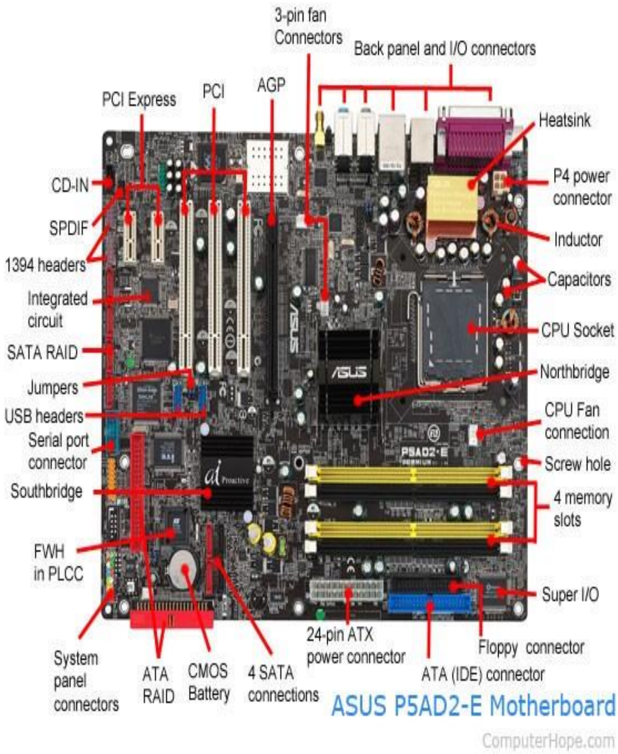

Fig. 5. Different components of a motherboard.

III. CASE STUDY

The images in our dataset were gathered from lab-ready motherboard images from Google Images motherboard searches and then manually annotated using Roboflow's polygon tool. Annotations are made by assigning different classes to the objects of interest, such as:

1. Fan, 2. CPU FAN NO Screws, 3. CPU FAN SCREW LOOSE, 4. CPU FAN SCREWS, 5. CPU FAN, 6. CPU FAN PORT, 7. CPU FAN PORT DETACHED, 8. INCORRECT SCREWS, 9. LOOSE SCREWS, 10. NO SCREWS, 11. SCRATCH, 12. SCREWS.

Our architecture has been simplified by adding/removing classes for the following reasons:

1. Simplify the model;
2. Reduce calculation time;
3. Layers are not needed compared to the lens.

Therefore, the final architecture is as follows

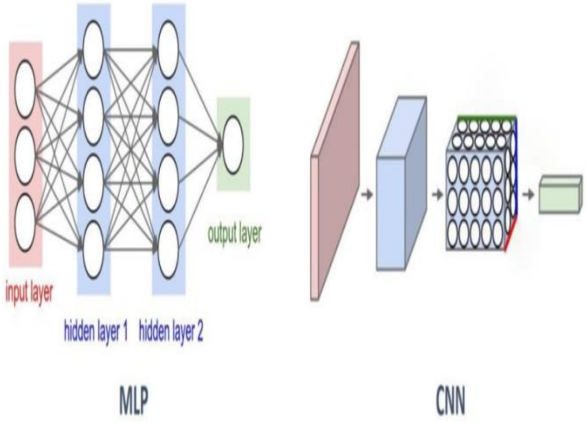

Fig. 6. Approximate structure of the model.

In this model, we have added batch normalization layers to speed up convergence and stabilize the network learning process. We have also introduced a maximum number of group layers to reduce the image size after each tap and capture the most important features. The final convolutional layer has an output of N channels, where N represents the number of detected feature classes. Each batch is passed through the neural network and the weight of the model is adjusted by gradient backpropagation to minimize the loss function. After each epoch or after a certain number of batches, the model is evaluated on the validation dataset.
This assessment involves running validation data through the model and measuring its performance using metrics such as accuracy, precision, and recall.

The goal is to understand how the model behaves on data it hasn't seen yet, in order to detect possible problems with overfitting. Overfitting occurs when the model fits too tightly to the training data and does not generalize well to the new data.

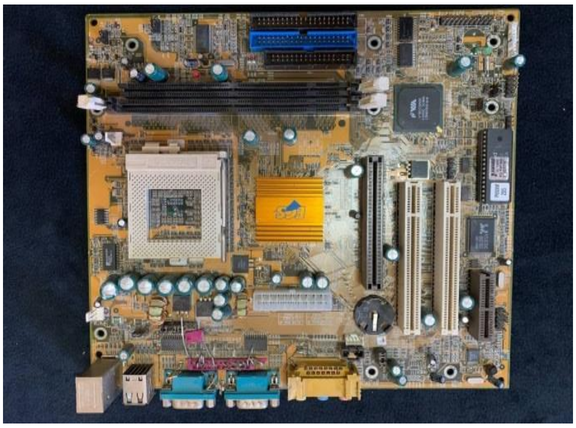

Fig. 7. Image of a Motherboard used in the dataset.

Hyper-parameters are parameters that are not learned by the model but are defined by the user prior to the training process. They affect the performance and behavior of the model. One of the important hyperparameters is the learning rate. If the learning rate is too high, the weight adjustment may be too large and the model may have difficulty converging or stabilizing. On the other hand, if the learning rate is too low, the training can be very slow and the model may take a long time to achieve good performance. It is common to adjust the learning rate during training to improve the performance of the model. For example, the learning rate can decrease as training progresses, allowing for more precise model weighting at the end of training. This approach can help avoid excessive oscillation and achieve better convergence.

Hyper-parameter tuning is an important step in the model training process. This may require repeated testing with different values to find the optimal combination of hyperparameters to maximize model performance on a particular data set. This can be done using techniques such as grid search or Bayesian optimization. Once the model has achieved satisfactory performance, it can be exported for further use, especially for detecting objects in real time or on new images. General learning process diagram.

A) *Performance and Characteristics*

The normalized confusion matrix is a tabular representation used to evaluate the performance of a classification model for each class. It shows how many samples belong to each class and how they are predicted by the model. By examining this matrix, we can identify false classifiers, such as false positives (samples incorrectly predicted to belong to a class) and false negatives (templates). Sample is incorrectly predicted not to belong to a class). By normalizing the matrix, the values are converted to percentages, giving a more meaningful view of the error distribution. By analyzing this matrix, it is possible to identify the classes where the model performs well and those where the model has difficulty. This can help identify areas where improvement can be made, such as adjusting the model's decision thresholds or collecting more specific data for underperforming classes. By understanding specific classification errors, corrective actions can be taken to improve the overall accuracy of the model.

IV. RESULTS AND SIMULATION

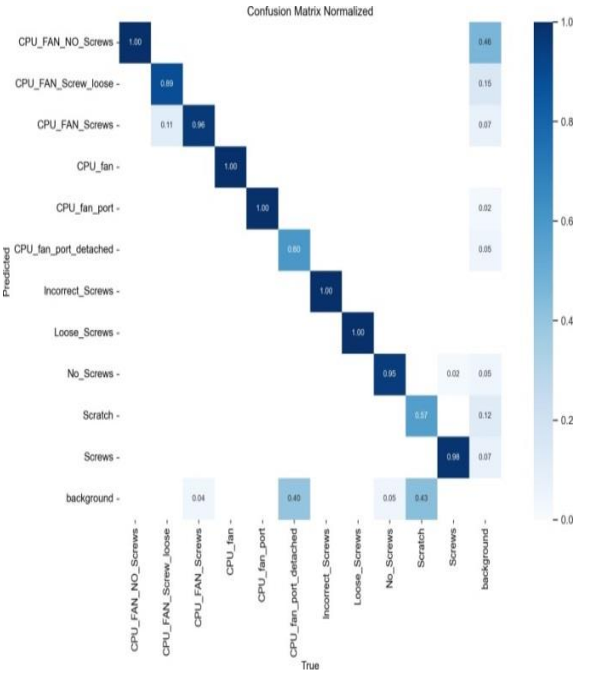

Fig. 8. Confusion matrix normalized.

F1-1 Confidence Curve

The F1-confidence curve is a g3raphical representation used to evaluate the performance of a classification model based on 2 F1-Confidence different confidence thresholds. This curve is based on the F1 metric, which combines a model's precision and recall into a single harmonic score. The F1-score is calculated by taking into account true positives, false positives, and false negatives. It is often used when classes are imbalanced.

Examining the F1-confidence curve allows for visualisation of how the F1score changes based on the confidence threshold used for classification decisions. An ideal curve would be increasing, reaching a maximum F1-score at an optimal confidence threshold. However, in practice, the shape of the curve may vary depending on the characteristics of the dataset and the model. Adjusting the confidence threshold allows for finding a compromise between precision and recall, and identifying the thresholds that maximize the overall model performance.

The threshold is the minimum probability required for a class to be predicted by the model. Plotting this curve allows for visualization of how the model's accuracy varies depending on the confidence threshold. Precision measures the accuracy of the model in terms of correct predictions of positive and negative classes.

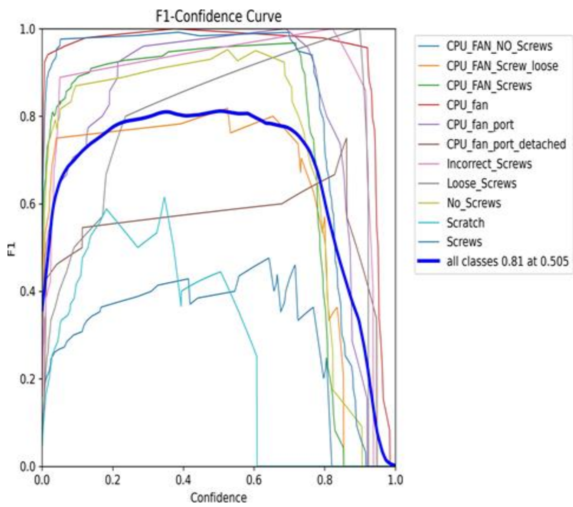

Fig. 9. Confidence curve.

It is calculated by dividing the number of true positives by the sum of true positives and false positives. An ideal precision-confidence curve should be a rising curve that reaches maximum precision at... However, in reality, the shape of the curve may vary depending on the characteristics of the dataset and the model used.

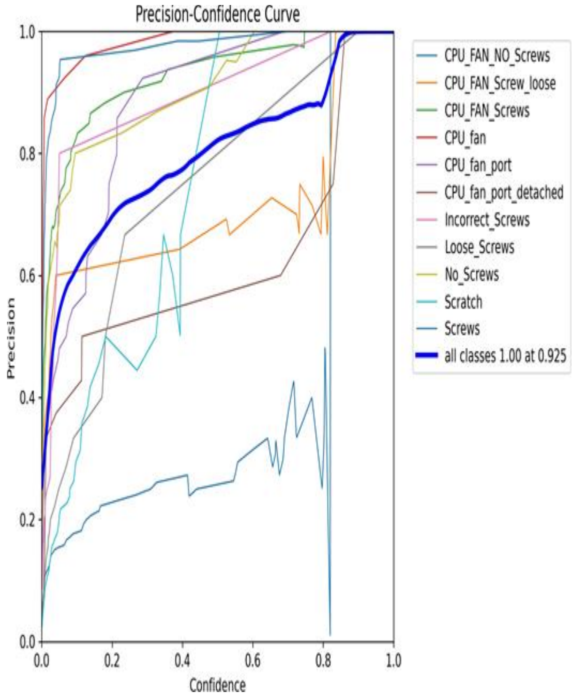

Fig. 10. Precision confidence curve.

The curve also enables the identification of confidence thresholds that maximize the overall accuracy of the model. Analysing this curve can assist in finding a balance between the precision and recall of the model by adjusting the confidence threshold. It also enables the identification of confidence thresholds that maximise the overall precision of the model. It is important to note that this curve should be used to make objective evaluations and avoid subjective interpretations. Additionally, it is recommended to use clear and concise language with a logical flow of information and causal connections between statements.

Technical term abbreviations should be explained when first used. The language should be formal, free from bias, and grammatically correct. Finally, it is important to adhere to style guides and maintain consistent formatting features. The Precision-Recall curve is a commonly used evaluation class to be predicted by the model. Plotting this curve allows for visualization of how the model's accuracy varies depending on the confidence threshold. Precision measures the accuracy of the model in terms of correct predictions of positive and negative classes.

It is calculated by dividing the number of true positives by the sum of true positives and false positives.

An ideal precision-confidence curve should be a rising curve that reaches maximum precision at... However, in reality, the shape of the curve may vary depending on the characteristics of the dataset and the model used. The curve also enables the identification of confidence thresholds that maximize the overall accuracy of the model.

Analysing this curve can assist in finding a balance between the precision and recall of the model by adjusting the confidence threshold. It also enables the identification of confidence thresholds that maximise the overall precision of the model. It is important to note that this curve should be used to make objective evaluations and avoid subjective interpretations. Additionally, it is recommended to use clear and concise language with a logical flow of information and causal connections between statements.

Technical term abbreviations should be explained when first used. The language should be formal, free from bias, and grammatically correct. Finally, it is important to adhere to style guides and maintain consistent formatting features. The Precision-Recall curve is a commonly used evaluation tool.

To evaluate the performance of a classification model, especially in cases where the classes are... The graph is unbalanced. The graph visually represents the relationship between model precision and recall at different classification thresholds. Recall measures the model's ability to correctly identify positive examples among all real positive examples, while precision measures. Is ability to correctly predict positive examples among all positive predictions? By plotting this curve, we can observe how precision and recall evolve together depending on the classification threshold. An ideal precision-recall curve would be one that achieves maximum precision and maximum recall simultaneously.

However, in reality, there is usually a trade-off between precision and recall, where increasing the classification threshold results in an increase in precision but a decrease in recall, and vice versa.

In the field of object detection and classification. It allows you to visualize the relationship between the recall of the model and the confidence associated with its predictions.

Recall measures the model's ability to correctly identify all positive examples among all real positive examples, while confidence represents the certainty attributed by the model to its predictions. By plotting this curve, we can analyze how recall evolves as a function of confidence, which makes it possible to determine the level of confidence necessary to detect the majority of examples positive.

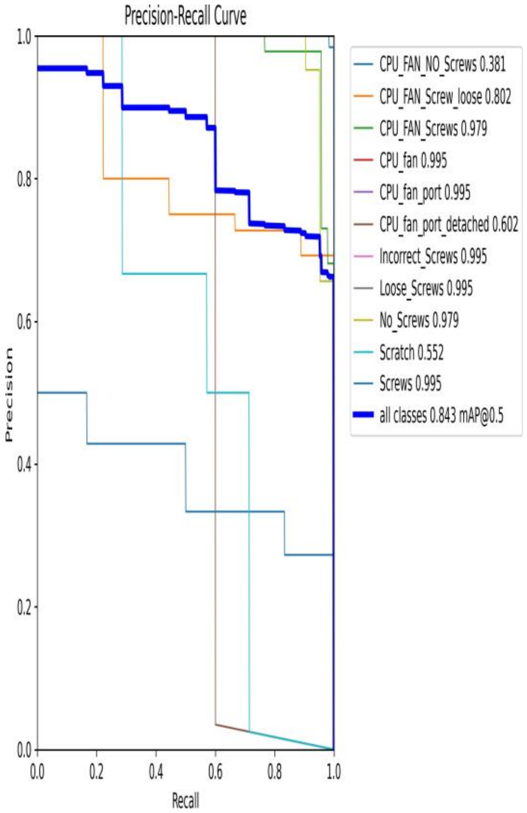

Fig. 11. Precision-recall curve.

This curve is useful for making decisions about the confidence threshold to use for model predictions, balancing precision and recall. It also makes it possible to evaluate the reliability of the model in terms of confidence placed in its predictions, and to observe whether the model improves its predictions with increasing confidence.

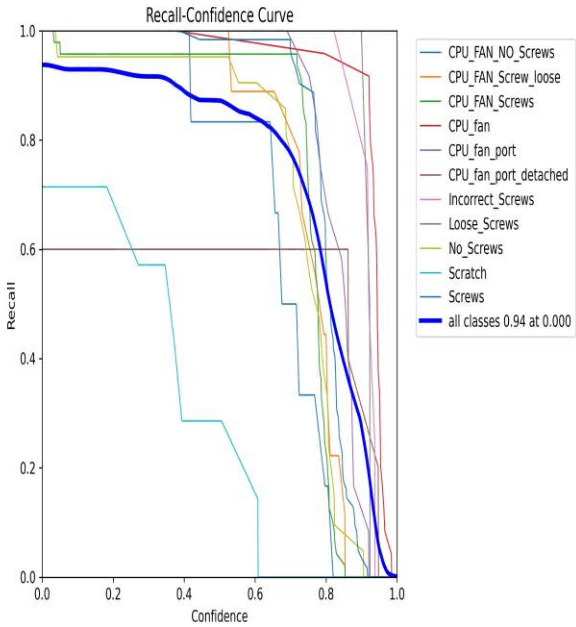

Fig. 12. Recall-confidence curve.

V. MODEL TEST

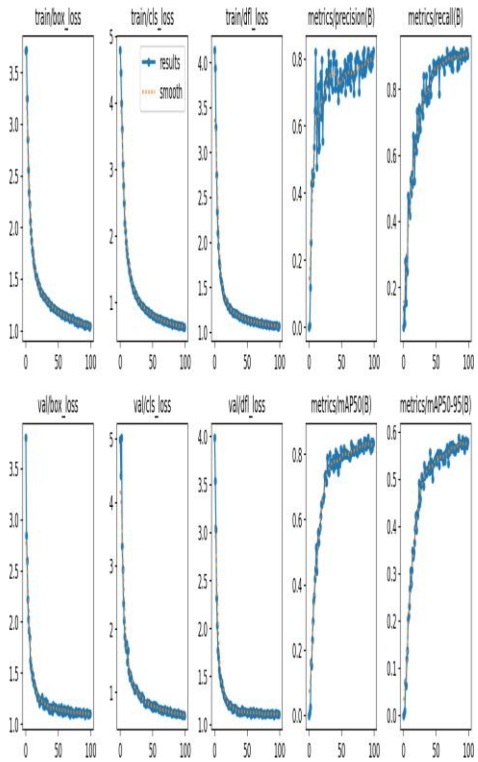

Fig. 13. Model test.

VI. CONCLUSION

This model is successfully implemented and does its job with outstanding performance and outstanding reliability. Although his forecast may have shortcomings or limitations, it can still yield positive results in most cases. Thanks to its reliable operation, it can make a significant contribution to the assigned task. While improvement can be considered, it is important to understand that the current model can perform its functions with a certain degree of accuracy and commendable consistency.

We learned a lot about the maintenance aspect of the model using algorithmic methods for the electronic circuits on the motherboard.

Notwithstanding its flaws, the deployed model is able to carry out its duties with notable effectiveness and dependability. Even though its predictions might have flaws or restrictions, it typically produces results that are satisfactory. It can contribute significantly to the task given to it because of its dependable operation. Even though there is room for improvement, it is crucial to acknowledge that the current model is capable of carrying out its tasks with a certain level of accuracy and admirable consistency.

VII. REFERENCE

[1] Abdelghani Chahmi, M.Bendjebbar, B.Raison, M.Benbouzid “An extender kalman filter-based induction machines faults detection” International Journal of Electrical and Computer Engineering (IJECE), ISSN: 2088-8708, V6 (2) April 2016.

[2] Abdelghani Chahmi, M.Birane « Detection of electrical faults by the Time-Frequency Approach », the first international conference on communications, control systems and signal processing -CCSSP2020, 16-18 May 2020, El-Oued, Algeria. DOI: 10.1109/CCSSP49278.2020.9151762 Publication Year: 2020, Page(s): 1 – 6. IEEE Conference Publication.

[3] Abdelghani Chahmi, A.Djoudi « Diagnosis of the machine by the Kalman Filter », 5 th International Conference on Electrical Engineering- ICEE’2017, October 29- 31, Boumerdes, Algeria. DOI: 10.1109/ICEE-B.2017.8192063 Publication Year: 2017, Page(s): 1 – 6. IEEE Conference Publication

[4] Rosenblatt F (1957) The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautical Laboratory.

[5] Minsky M, Papert S (1969) Perceptron: an introduction to computational geometry.

[6] Maria Vakalopoulou, Stergios Christodoulidis, Ninon Burgos, Olivier Colliot, Vincent Lepetit., Deep learning: basics and convolutional neural networks (CNN)

[7] Machine Learning in Agriculture: A Review Konstantinos G. Liakos 1,Patrizia Busato Patrizia Busato Scilit ,DimitriosMoshou 1, Simon Pearson and Dionysis Bochtis.