TECHNICAL REPORT

Inès DURLENT

Loup DUMAS

Quentin FAURY

Théo DUBUY-MICHEL

Ali FAHS

Elisa MARTINELLI

ABSTRACT :

Identifying the value of electronics components, for this case resistors, from their visuals particularities ,like color bands, is a common yet error-prone job, especially for non-expert or beginners in a fast surrounding conditions.

In this project, we propose a deep learning approach to automatically recognize the value of an electronic resistance from an image, by doing that we will eliminate the need for manual decoding.

Using the Resistor Color Coding v5 dataset from Roboflow, which contains more than 7,000 annotated images of resistors, we trained a YOLO-based object detection model. The trained model (best.pt) is capable of detecting and classifying resistor color bands with high accuracy.

Our implementation demonstrates that combining computer vision with deep learning can provide a reliable, efficient, and user-friendly solution for resistor value recognition.

INTRODUCTION :

One of the most used electronic components is resistor. Their values is identified by understanding the sequence of color bands printed on it, this process is simple in theory but it can be time consuming and can lead to several error in an electronic project, especially for beginners, technicians in training or in a projects where you can not take time to calmly search for your electric components.

Recent advances in deep learning and computer vision have enabled the development of automated recognition systems for tasks previously requiring manual interpretation. In this work, we explore the use of YOLO (You Only Look Once), a state-of-the-art object detection model, to recognize resistor color bands and deduce their electrical value. Our goal is to build a tool that makes resistor identification fast, accurate, and accessible, by combining a robust dataset (Resistor Color Coding v5) with a high-performance deep learning model.

RESEARCH :

Architecture: Our YOLOv8 model has a real-time object detection architecture that combines speed and accuracy. With only 3.2 million parameters, it is a fairly lightweight model, making it quick to train and use even on machines with limited resources.

Types of tasks:

-Object detection: Automatically identify and locate different electronic components in images.

-Classification: Assign each detected object to a specific category.

Performance and efficiency: YOLOv8 offers very fast iteration speeds, suitable for systems with limited resources, while maintaining high detection accuracy. Its low memory requirements (compatible with low-capacity GPUs and even CPUs) allow for more flexible training and use.

Applications: YOLOv8 is particularly well suited for projects requiring real-time image processing. For our electronic component recognition project, it automates the detection of resistors in images, facilitating the tracking and analysis of measurements without manual intervention.

DATA ENGINEERING :

For the development of the resistance-reading AI, the main focus was on assembling and preparing the dataset.
Instead of creating our own dataset, we relied on an existing resource: the Resistor Color Coding – v5 dataset available on Roboflow, which contains around 7,000 labeled images of resistors.

This dataset was chosen because it provides a wide variety of resistor images, covering different values and including variations in appearance such as color shades, lighting conditions, and orientations. Its size and diversity ensured that the model had enough examples to train effectively and achieve good generalization.

The dataset was then used directly for supervised training, enabling the AI to learn how to recognize and interpret resistor color codes purely from images, without the need for manually engineered prompts or prior conceptual guidance.

ML TRAINING :

After compiling our dataset, we worked on adapting the pre-trained YOLOv8 model to address our electronic resistance detection issue. This lightweight model, optimized for real-time object detection, allowed us to combine speed and accuracy, even on machines with limited resources. We consulted the technical documentation and conducted several tests to identify the settings best suited to our project, paying particular attention to key parameters such as image size, number of training iterations, and network weight update speed. The goal was to find a good compromise between processing speed and prediction quality, which is essential for smooth integration into an interactive chatbot.

This training stage is based on transfer learning, which allowed us to reuse the knowledge already acquired by YOLOv8 on general datasets, while adapting it to our specific dataset. This reduced the learning time and resources required, while making resistance detection more accurate and efficient.

INTEGRATION :

In order to make the model usable by the entire group, we developed an inference pipeline and a user interface. We integrated the trained model into a simple web application with Gradio, allowing users to upload an image of an electronic component and enter a prompt to obtain a prediction in real time.

QUALITY :

Quality control was carried out continuously throughout the project.

From the earliest stages, coordination and brainstorming enabled us to define clear and achievable objectives.

There were several stages: research, dataset creation, model training, and integration. Regular monitoring ensured that the technical choices made met the defined requirements. We then verified the consistency between the results obtained and the objectives set,

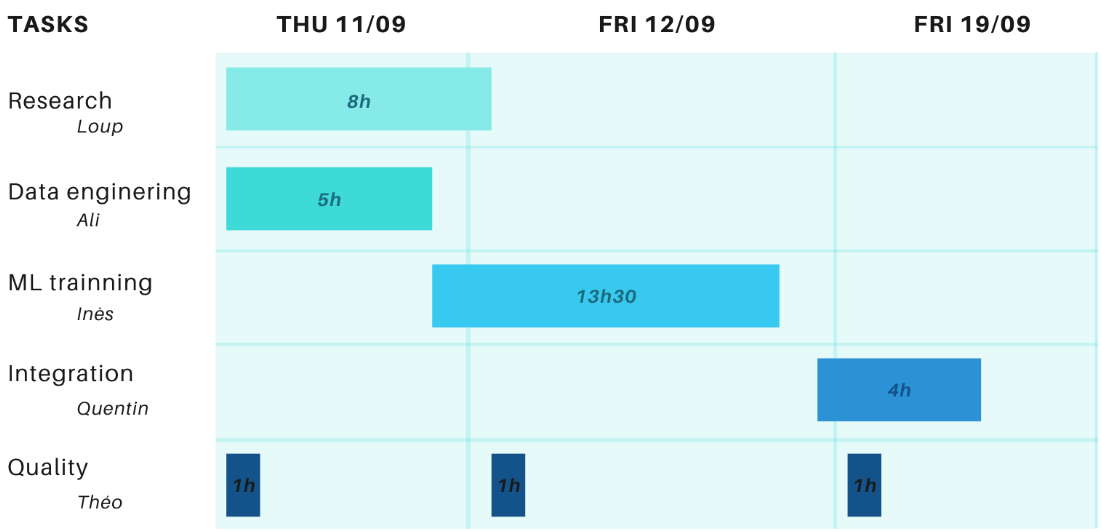
quickly identified bottlenecks or performance gaps,

proposed adjustments to improve the reliability and relevance of the model, and established good communication within the team to maintain smooth progress.

It was thanks to this quality approach that we were able to move forward in a structured manner despite the difficulties encountered, and achieve a functional and effective application. Regular monitoring optimized development time and ensured the overall consistency of the project.

BUDGET & SCHEDULE:

Target: Friday 19/09, 2025



The project was initially estimated at 20 hours of work, corresponding to a budget of 2,000 € (100 €/h). However, the actual workload reached 33.5 hours, for a final budget of 3,350 €.
The main cause of this overrun was the training phase (ML training), which required 13.5 hours due to multiple iterations and parameter adjustments to optimize performance.

Despite this increase (+67.5 % compared to the initial budget), the additional investment allowed us to achieve a functional and efficient application. The time spent on training ensured better accuracy and robustness of the model, which is essential for the reliability of the system.

CONCLUSION

This project demonstrated the feasibility and efficiency of using deep learning techniques for the automated recognition of resistor values from images. By leveraging the YOLOv8 architecture and the Resistor Color Coding v5 dataset, we developed a solution that combines accuracy, speed, and usability. The integration of the trained model into a simple user interface further confirmed the potential for practical applications, making resistor identification more accessible and reducing human error.

Although the project exceeded the initial time and budget estimates, this investment resulted in a more robust and reliable system. The work carried out highlights the relevance of combining computer vision and deep learning to address recurring challenges in electronics.

FUTURE IMPROVMENT :

We have considered grouping other types of electronic components such as AOs to facilitate the search for datasheets, or for coils, capacitors, etc. The possibilities are quite broad and could make work in the field of electronics simpler and faster. We also thought that adding a chatbot could improve our interface and contribute to a better environment for the person using it but we could not make it works as fast as it was without it, this possibility would totally be possible with more time.

ANNEXE

Article 1 :

The paper Attention Is All You Need presents the Transformer, a new architecture for machine translation and more generally all sequence transduction tasks. Unlike the previous approaches, which relied on recurrent (RNN, LSTM) or convolutional (CNN) networks, the Transformer relies solely on attention mechanisms, which completely removes dependence on the sequential nature of time, and thus ram much faster in training.

The impetus behind the architecture is that traditional sequential models processed words in a sequence one at a time in order to predict the next time step, meaning that these models did not allow for parallelization and weren't effective for capturing long-term dependencies. Therefore, the authors took the approach to develop a fully-parallel architecture, which allows for the simultaneous processing of all elements in a sequence, while still allowing for the modeling of complex relationships between words.

The Transformer features an encoder-decoder design with six identical layers at both ends. The encoder converts the input sequence to contextual representations using a multi-head self-attention vulnerability and a feed-forward network at each position. The decoder generates the output sequence using self-attention with masking to maintain the order of generation, plus cross-attention over the encoder output. Since attention does not attend to the order, the authors introduce positional encoding with sinusoidal functions to provide that information.

The heart of the model is Scaled Dot-Product Attention, which computes weights based upon contextual relevance between words. Multi-head attention allows that computation to be simultaneously considered in multiple representation subspaces, helping the model capture varied relationships. The entire model is stabilized by residual connections and layer normalizations.

To train the Transformer, the authors utilize public translation corpora, which include WMT 2014 English–German (4.5 million sentence pairs) and WMT 2014 English–French (36 million pairs). The optimization uses Adam, a warmup of learning rate, dropout, and label smoothing for regularization, which allows for regularization in the optimization process. Training the “base” model takes approximately twelve hours on eight P100 GPUs, while training the “big” model takes about three and a half days.

The final results lead to new records on translation tasks, such as a BLEU score of 28.4 for English–German, and BLEU score of 41.0 for English–French. Models that were previously trained were unable to accomplish the above examples, while also lowering the costs of training. Hence, the Transformer has shown that with a much simpler organization, an architecture can perform better and train faster.

In summary, it has truly revolutionized natural language processing, and technically allowed a means to efficiently model long dependencies. The speed and flexibility, also, will mean it will form the core of almost all modern models such as BERT, GPT, or T5 and will allow applications well above text, such as images, audio, or video.

Article 3:

The article “Training Compute-Optimal Large Language Models” challenges the traditional strategy for training large language models, which consisted of constantly increasing the number of parameters. The authors show that this approach is not optimal if the amount of training data is not also taken into account.

By analyzing different configurations, they propose a more accurate scaling law that links three key factors: model size (number of parameters), dataset size (number of tokens), and available computing budget.

To test their idea, the researchers trained a model they call Chinchilla. It is much smaller than GPT-3 (70 billion parameters versus 175 billion), but the difference is that they fed it with much more data, about four times as much. And that's what makes all the difference: even though it is smaller, Chinchilla manages to beat GPT-3 on most tests.

In conclusion, the article shows us that just because a model is bigger doesn't necessarily mean it's better. What it shows is that it's important to find the right balance between model size and data quantity, depending on the resources available. This is what they call compute-optimal training.

Article 4:

This article addresses the problems of adapting large LLMs such as gPT-3, which have the disadvantage of requiring billions of parameters. In addition, fine-tuning them requires enormous amounts of money in terms of computing power and storage in order to adapt them to specific tasks. Finally, completely retraining an LLM for each task is impractical and inaccessible.

That is why this article proposes using LORA (Low-Rank Adaptation of Large Language Models), a fine-tuning method that uses fewer parameters. The goal here is to insert small, low-rank matrices into existing layers instead of updating all of the model's weights. The matrices, which are more compact than the original weights, are therefore trained. In summary, updates to the weights, which are very large, can be approximated by decomposing them into low-rank matrices. This approach targets, in particular, the projection matrices of attention mechanisms and MLPs in Transformers. Instead of recalculating an entire matrix, the update is factorized as $\Delta W = A \times B$, where A and B are two small matrices. This approach makes it possible to drastically reduce the number of parameters adjusted while maintaining the quality of the results.

Thus, reducing the number of parameters to be trained by several orders of magnitude greatly reduces the amount of GPU memory required and allows training on more accessible hardware. This results in low storage usage, as LORA matrices are saved for each task. Finally, in terms of performance, LoRA achieves results comparable to those of full fine-tuning.

Experiments conducted on GPT-2 and GPT-3 on several NLP benchmarks confirm this efficiency. For example, a model with 175 billion parameters now requires only a few million parameters trained with LoRA. Each task can now have its own LoRA module, which preserves the versatility of the base model.

In conclusion, LoRA represents a major advance in the field of Parameter-Efficient Fine-Tuning (PEFT). It democratizes the adaptation of large language models by making it accessible to a greater number of researchers and practitioners, while maintaining high performance. This method is now widely adopted in the artificial intelligence community.