PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

Ministry of Higher Education and Scientific Research

UNIVERSITY OF ECHAHID HAMMA LAKHDAR – EL OUED

FACULTY OF EXACT SCIENCES

Computer Science Department

Presented by:

- Nour Elhouda Berrah
- Hmide El Romaissa
- Touati Amira

Under the supervision of:

- Hamoud Meriem

Title:

Aero-engines damages diagnosis approach based Deep Learning

This thesis was discussed on May 25th, 2025

Defense Committee :

DR.AHMED GHANABZAI

DR.FARIDA RTEMA

*Academic year : **2024/2025.***

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

الاهداء

﴿ و توكل على الله وكفى بالله وكيلا ﴾

الأحزاب :3

و من أصدق من الله قيلا الحمد لله الذي ألهمنا الصبر و اليقين بأن الله عز وجل على كل شيء قدير الحمدلله الذي ثبت خطانا عند كل عثرة و ألهمنا الدعاء

قال صلى الله عليه و سلم : "من سلك طريقًا يلتمس فيه علمًا، سهّل الله له به طريقًا إلى الجنة"

الحمدلله و الصلاة و السلام على نبينا محمد و على آله و سلم ومن تبعهم بإحسان إلى يوم الدين. أهدي تخرجي لمن كان سندا لي من بعد الله عز وجل عائلتي التي بسببها و بسبب دعمها لي لم أصل لما و صلت إليه اليوم , أمي العزيزة، أبعث لكِ هذا الإهداء، ممزوجًا بالعرفان والامتنان، فأنتِ نبض قلبي ونور دربي. وإلى من كان السند والعون، والدافع الأول للمضيّ قدمًا والدي الغالي، لك كل الشكر والحب، فما وصلت إليه اليوم إلا بفضلك بعد الله. نهدي تخرجنا إلى كل من علمنا حرفا **قال رسول الله صلى الله عليه وسلم : المرء على دين خَليله، فلينظر أحدُكم مَن يُخالِل** أتقدم بجزيل الشكر إلى صديقاتي اللواتي كانو داعمي طوال مسيرتي فقد كانوا خير الاصحاب نسأل الله عز وجل أن يجمعنا في جنته إلى كل من رافقني في رحلة العلم، من أصدقاء وزملاء وأساتذة،

نتقدم بشكر للاستاذة **مريم حمود** على مجودتها معنا و نقدر تعبها و توجيهها لنا خلال العمل بارك الله فيك و يسر لك أمرك و ثبت أجرك

(وَآخِرُ دَعْوَاهُمْ أَنِ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ)

يونس 10

الحمدلله دائما و ابدا الحمدلله حتى يبلغ الحمد منتهاه كما نهدي هذا العمل إلى شعوبٍ عظيمة صابرة، كُتب عليها الابتلاء فصبرت، وعانقت الألم بقلوبٍ موقنة بعدل الله، لا تنكسر. غزة الحبيبة نسأل الله أن يرفع عنك البلاء و أن ينزل سكينته عليك , و جعل نصرك قريب و رحم الله شهدائك إلى كل المستضعفين من المسلمين قوا كم الله و أعزكم بجنده

Abstract

Traditional aircraft engine inspection methods—relying primarily on visual examination and conventional diagnostic tools—are often time-consuming and may lack the sensitivity required to detect all potential forms of damage, thereby posing significant safety risks. To address these limitations, artificial intelligence (AI) techniques, particularly machine learning and deep learning in the domain of computer vision, have emerged as promising alternatives for enhancing the precision and efficiency of image-based diagnostics. This study proposes the development of a YOLO (You Only Look Once) deep learning model for the automated detection of aircraft engine damage. The objective is to expedite the inspection process, reduce maintenance costs, and improve the accuracy of damage detection, thereby contributing to safer and more efficient aircraft operations.

Keywords

Artificial Intelligence, Computer Vision, Image Processing, Deep Learning, YOLO, Aircraft Damage Detection, Automated Inspection.

الملخص

غالبًا ما تستغرق طرق فحص محركات الطائرات التقليدية - التي تعتمد بشكل أساسي على الفحص البصري وأدوات التشخيص التقليدية - وقتًا طويلاً، وقد تفتقر إلى الدقة اللازمة للكشف عن جميع أشكال الضرر المحتملة، مما يُشكل مخاطر سلامة كبيرة. ولمعالجة هذه القيود، برزت تقنيات الذكاء الاصطناعي، وخاصةً التعلم الآلي والتعلم العميق في مجال الرؤية الحاسوبية، كبدائل واعدة لتعزيز دقة وكفاءة التشخيص القائم على الصور. تقترح هذه الدراسة تطوير نموذج YOLO (You Only Look Once) للتعلم العميق للكشف الآلي عن أضرار محركات الطائرات. والهدف هو تسريع عملية الفحص، وخفض تكاليف الصيانة، وتحسين دقة الكشف عن الأضرار، مما يُسهم في عمليات طيران أكثر أمانًا وكفاءة.
الكلمات المفتاحية
الذكاء الاصطناعي | التعلم العميق | YOLO | صيانة الطائرات | الرؤية الحاسوبية | التشخيص الآلي

Contents

List of Figures

List of Tables

General Introduction

The significance of aviation security is paramount for protecting public health, enhancing economic growth, and ensuring military operations. Advances in technology have heightened interest in civil aviation, fueled by increased demand for air travel tied to the growth in the global middle class, the rise of low-cost airlines, and the growth in air freight transport. Irrespective of the negative impacts, the global active aircraft fleet numbered 25,900 in the year 2019, with estimates pointing towards growth to 47,081 units in the year 2041.

With the high number of aircraft in use, there is a need for strict control, not only with respect to air traffic but also to the assessment of the technical airworthiness of aircraft in order to maintain aviation safety [29]. Annual technical and operational examinations are carried out annually by each country's Civil Aviation Authority (CAA) to ensure compliance with high standards of maintenance and performance by aircraft. In addition, larger manufacturers like Boeing, Airbus, Embraer, and Bombardier depend upon the respective country's CAAs where they produce. The leading CAAs are the Federal Aviation Administration (United States), the European Union Aviation Safety Agency (EU), Agência Nacional de Aviação Civil (Brazil), and Transport Canada (Canada). The extensive operational lifespan of an aircraft makes it vulnerable to all manners of damage, ranging from blade degradation due to friction to coloration changes due to extreme-temperature combustion, among other causes [30]. One such example of a condition that may lead to an accident is the crash of the Algerian Air Force Ilyushin Il-76, which occurred on April 11, 2018, at around 6:50 AM. The crash involved an Ilyushin Il-76TD operated by the 347th Strategic Transport Squadron of the Algerian Air Force, occurring sadly just minutes after takeoff near the Boufarik Air Base located to the southeast of Algiers. Unfortunately, all 257 people on board this Algerian Air Force-owned transport plane died [31]. The incident stands as the deadliest aviation accident in Algeria and as the fourth deadliest in the world within the last twenty years. Witnesses reported that the left wing of the aircraft caught fire before crashing, an occurrence that they speculated was caused by an engine fire potentially triggered by a bird strike.

The multimodal composition of aircraft engines poses a significant challenge to damage discovery. Imagery, the most natural medium for data acquisition, is widely applied in damage detecting and localizing [29]. In this, leading airlines often make use of a borescope to record the interior visuals of the engine, then visually examine and locate damages as may be necessary. Inspectors are authorized to do visual examinations of the interior structure of aircraft engines, including the blades. Borescope images form a vital technique which makes possible the inspection of an engine's interior; the borescope's flexible tube can be inserted into the engine's entry ports, hence enabling inspection with little disassembly needed [32]. For example, specialized entry plugs or sealing of an igniter opening is applied to make entry into the hot areas of a turbine possible. Inspectors analyze video images acquired through borescopes in real-time for any possible irregularities. When an essential defect is detected, the engine is sent to a maintenance, repair, and overhaul workshop, where additional investigation is done through engine disassembly. This is time-consuming and labor-intensive with much time invested, forming an appreciable part of the engine's catalog cost. However, the direct application of these damage detection methods faces several hurdles [32][33][34]: the borescope technique requires several hours of examination, calls for many skilled inspectors for the assessment of surface damage on blades with the use of artificial vision, consumes much time and labor, and is strongly dependent on specialized experience; issues involved are the small size of the damage, high-stress engine condition with background interference presence as well as other defects, tight precision standards, and complicated blade internal structures. Engine inspection also involves much manual labor, with many manual laborers requiring intensive and costly training, whereas their inexperienced counterparts do not have the essential knowledge and skills required to effectively spot and report defective engine parts. Human factors prove quite daunting in the maintenance process; the assessed severity of the diagnosed defect, regarding repair actions execution or deferral to future inspections, is dependent upon manual estimations of damage place and extent. These assessments are compared with tolerance standards of past case histories documented in relevant engine manuals. Therefore, all these factors make image quality inconsistent and enhance the difficulties involved with the identification. In this context, deep learning-based methods prove to be highly effective in identifying damage to blades and aircraft engine components due to their higher feature extraction and object recognition capabilities. This work proposes a system for the diagnosis of seven different types of damage that can occur in aircraft engines in three particular areas: compressor blades, turbine blades, and the

The structure of the thesis is outlined below:
Chapter I: Basic Concepts of Image Processing and Computer Vision Using Artificial Intelligence Techniques and Deep Learning Methods
In this section, we discuss several applications and theoretical models relevant to artificial intelligence, with special emphasis upon computer vision and image analysis.

ChaPter II: Exploring the Diagnostic Methods for Aircraft Engine Failure Through Artificial Intelligence Platforms: Modern Problems and Current Trends

This chapter outlines the underlying paradigms of artificial intelligence that have been developed for the independent diagnosis of damage in aircraft engines, an essential component that can trigger accidents. The more intricate aspects of this problem are to be considered in detail.

Chapter III: Review of the Suggested Artificial Intelligence-Based Approach to the Automation of Aircraft Engine Evaluation

This chapter assesses the effectiveness of the proposed system developed for identifying and classifying engine damage, highlighting the relevance and usefulness of the proposed approach

List of Abbreviations

AI	Artificial Intelligence
DL	Deep Learning
ML	Machine Learning
DNN	Deep Neural Network
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
FNN	Feedforward Neural Network
CV	Computer Vision
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DQN	Deep Q-Networks
MLP	Multi-Layer Perceptron
YOLO	You Only Look Once
AWS	Amazon Web Services
GCP	Google Cloud Platform
ONNX	Open Neural Network Exchange
NDI	Non-Destructive Inspection
NDT	Non-Destructive Testing
FAA	Federal Aviation Administration
FOD	Foreign Object Debris
IoU	Intersection over Union
mAP	mean Average Precision

Chapter I :Introduction to Image Processing and Computer Vision with a Focus on Artificial Intelligence Paradigms

I.1 Introduction

This chapter gives the theoretical and practical foundation for artificial intelligence models, specifically for the areas of computer vision and image processing. It begins by explaining what computer vision is and how it can mimic human vision in observing and interpreting pictures and videos, with the key role of Convolutional Neural Networks (CNNs) highlighted here. The chapter then outlines the fundamental difference between computer vision and image processing and explains that while image processing focuses on the technical enhancement of an image without recognizing what it portrays, computer vision aims to make sense of and interpret what appears in an image. This is then followed by a detailed explanation of machine learning, including its different types (supervised, unsupervised, semi-supervised, and reinforcement learning), and an introduction to deep learning and its use in handling big data using deep neural networks. The chapter also covers the types of deep neural networks, i.e., Feedforward Neural Networks (FNNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs), highlighting the role of CNNs in building efficient object detection models such as YOLO and its various versions. Finally, the chapter also recapitulates the evolution of the YOLO algorithm up to its latest iteration, YOLOv11, its enhancement in speed, accuracy, and ability to conduct complicated tasks such as instance segmentation, pose estimation, and object tracking, showing the continuous development of modern computer vision tools. This chapter lays the foundation for understanding modern developments in artificial intelligence and foretells future uses of computer vision and image processing across various scientific and industrial fields.

I.2 computer vision

Computer vision is a field of artificial intelligence that enables computers to understand, analyze, and recognize visual data such as images and videos in a way that mimics human vision. This technology relies on neural networks, particularly Convolutional Neural Networks (CNNs), which help detect patterns and details within images. Computer vision is used in various fields such as security, medicine, industry, and traffic management, contributing to improved performance and increased accuracy through tasks like object detection, image classification, instance segmentation, and pose estimation. The YOLO model series, especially the latest YOLO11, is among the most powerful tools in this field, offering high speed and accuracy with significant improvements in efficiently processing visual data as we present in Figure I.1 illustrates the difference between how images are understood by computer vision compared to human vision. 1

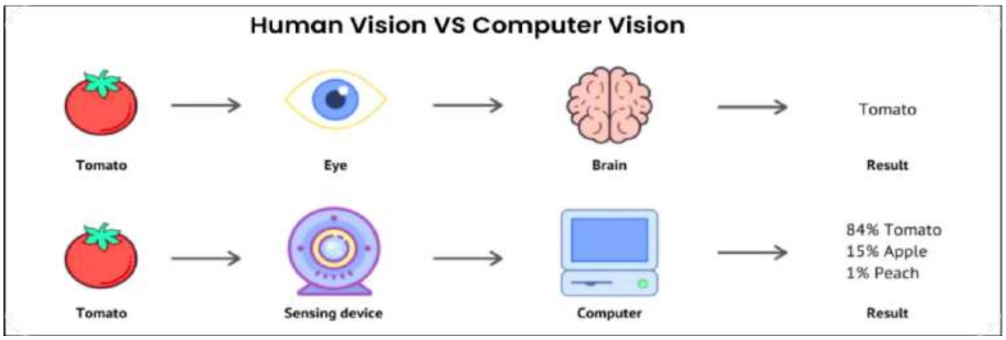


Figure I.1: Computer Vision vs Human Vision[1]

I.3 Image processing

Image processing relies on modification using mathematical functions that adjust pixel values without understanding the image as a whole. Techniques such as noise reduction, sharpening, and contrast enhancement are used to alter the colors and intensity of pixels to improve image quality, but they do not recognize objects, distinguish shapes, or understand meanings within images.2

I.3.1 The phases of image processing

I.3.1.1 Image Acquisition

is the first step in digital image processing, where an image is obtained and fed into the system for processing. This can be as simple as receiving an image that is already in digital format or may involve initial processing steps such as scaling and enhancing image quality before further processing.3

I.3.1.2 Image Enhancement

IS one of the most fundamental and visually appealing aspects of digital image processing. The primary goal of enhancement techniques is to reveal hidden details or emphasize specific features of interest within an image, improving its overall visibility and interpretability. 3

I.3.1.3 Image Restoration

is a field focused on improving the quality of an image, similar to image enhancement. However, unlike enhancement, which is subjective, image restoration is objective as it relies on mathematical and probabilistic models to correct distortions and recover the original appearance of a degraded image. 3

I.3.1.4 Color Image Processing

Dealing specifically with colour images, managing and improving colour representations for various applications. 3

I.3.1.5 Wavelets and Multiresolution Processing

Wavelets are the foundation for representing images in various degrees of resolution. Images subdivision successively into smaller regions for data compression and for pyramidal representation. 3

I.3.1.6 Compression

Compression deals with techniques for reducing the storage required to save an image or the bandwidth to transmit it. Particularly in the uses of internet it is very much necessary to compress data. 3

I.3.1.7 Morphological Processing

Morphological processing deals with tools for extracting image components that are useful in the representation and description of shape. 3

I.3.1.8 Segmentation

Segmentation techniques divide an image into its fundamental components or distinct objects. In general, achieving autonomous segmentation is one of the most challenging tasks in digital image processing. Implementing a robust segmentation method significantly enhances the ability to solve imaging problems that require the precise identification of individual objects. 3

I.3.1.9 Representation and Description

Representation defines how the data is organized after segmentation, while description focuses on extracting features that help distinguish between objects. Both are necessary to analyze images and make decisions based on their content. 3

I.3.1.10 Object recognition

Recognition is the final stage in image analysis, where objects are classified based on their characteristics. Its success depends on the quality of segmentation and the accuracy of the extracted features, which is one of the core areas of artificial intelligence and computer vision.

I.3.1.11 Knowledge Base

A knowledge base is an essential component of image processing, providing reference information that helps speed up and improve the accuracy of analysis and recognition processes. They can be simple, such as identifying regions of interest, or complex, such as large databases used in applications such as medical or environmental image analysis.3 We depict on the figure I.2 the fundamental steps in digital image proxessing.

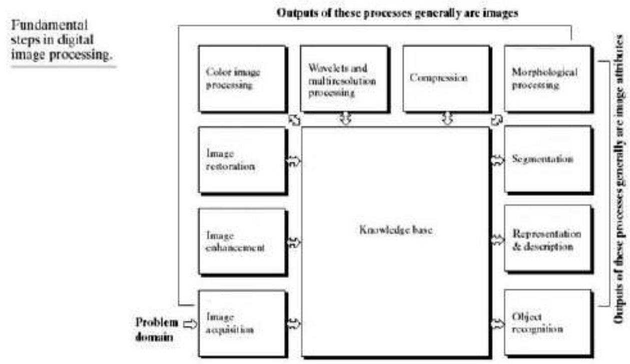


Figure I.2: Fundamental steps in digital image processing[2]

I.4 The difference between computer vision and image processing

table I.1

Aspect	Computer Vision	Image Processing
Purpose	Understanding & decision-making	Quality enhancement & preparation
Outputs	Semantic data (text/commands)	Enhanced/modified images
Complexity	High (AI-dependent)	Low (pixel-level focus)

table I.1: The difference between computer vision and image processing

The table clearly compares computer vision and image processing and how they differ in terms of purpose, output, and complexity level. In terms of purpose, computer vision is directed towards the interpretation of image contents and making intelligent decisions such as facial recognition or object detection, while image processing is committed to enhancing the quality of images such as noise reduction or image sharpening. In terms of output, computer vision produces semantic data like text and instructions, whereas image processing produces transformed and enhanced images. In terms of complexity, computer vision is more advanced and state-of-the-art as it relies on artificial intelligence and deep learning techniques, requiring huge datasets and high computing power. Image processing is relatively simpler, using traditional algorithms that operate at the pixel level. Computer vision is a must for applications like autonomous driving and smart surveillance, while image processing excels at smartphone image improvement and medical imaging applications. Such fundamental differences enable researchers and developers to choose the most suitable field based on project requirements and resource availability.

I.5 Machine Learning (ML)

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that enables computer systems to learn automatically from data without explicit programming. Instead of relying on predefined rules, ML algorithms detect patterns within data to make decisions or predictions. As they are exposed to more data over time, these systems continuously improve their performance4 .We present in the figure I.3 The relationship between artificial intelligence and machine learning

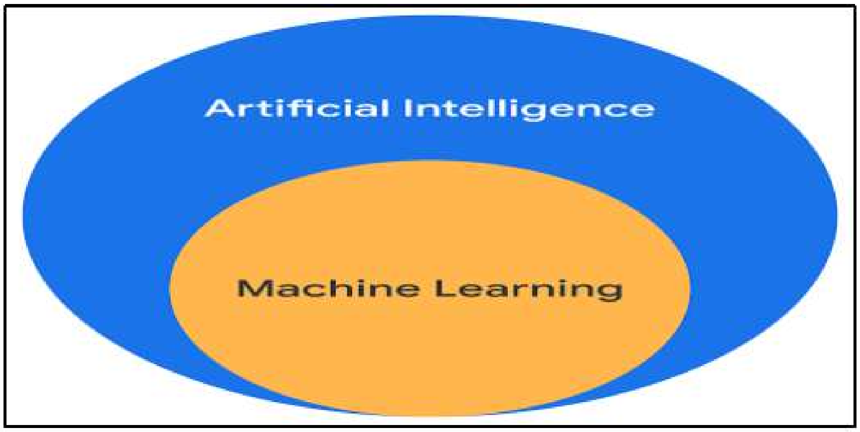


Figure I.3: The relationship of artificial intelligence to machine learning[3]

I.5.1 Types of Machine Learning

Machine Learning (ML) is categorized into 4 main type:

- Supervised Learning
- Unsupervised Learning
- semi-supervised learning
- Reinforcement Learning

I.5.2 Supervised Learning

Supervised machine learning involves human intervention, so that it takes place through two basic steps: first, giving the input data, and the second, naming the corresponding outputs, with the aim of developing a model that is trained using this classified data to make predictions or classifications of new data.5

Classification

In the classification stage, the system determines the type of information it receives. It must classify the input data into predefined categories or classes, based on criteria established during the training phase.Types of Classification 6

- Binary Classification
- Multiclass Classification
- Multi-Label Classification

Regression

Regression is a type of supervised learning task in which the objective is to predict a continuous numerical value. Examples include forecasting temperature, estimating stock prices, or determining house values. This is typically achieved using algorithms such as linear regression, ridge regression, or lasso regression. 7

I.5.3 Unsupervised Learning

Unsupervised learning is a type of machine learning (ML) in which algorithms analyze unlabeled data, meaning there are no predefined outputs or labels. Unlike supervised learning, which relies on data with known correct answers, unsupervised learning aims to uncover the inherent structure of the data by identifying patterns, clusters, or anomalies without prior guidance. This approach is particularly useful in Artificial Intelligence (AI) for initial data exploration and for understanding complex datasets where labeling is impractical or impossible. It allows models to automatically discover relationships and patterns directly from the data. 8

Clustering

This involves automatically grouping similar data points together based on certain characteristics. Popular algorithms include K-Means Clustering and DBSCAN. 8

Association Rule Learning

This method discovers interesting relationships or association rules between variables in large datasets. It's commonly applied in market basket analysis to find items frequently purchased together. 8

Dimensionality Reduction

This technique simplifies data by reducing the number of input variables or features while preserving essential information.Principal Component Analysis (PCA) is a widely used method for dimensionality reduction. 8

I.5.4 semi-supervised learning

It is a type of machine learning. Semi-supervised learning is used on two types of data: labeled and unlabeled. Unlabeled data is typically used more than labeled data, which helps improve learning accuracy. 9

Self-training

uses a classification method by using labeled data to classify unlabeled data. 9

Co-training

is a semi-supervised learning algorithm. This algorithm classifies data from multiple perspectives (or feature sets). It trains two models, each based on a different perspective, and then uses them to classify the unlabeled data. 9

Multi-view machine learning

is characterized by using multiple data. Joint training is similar to multi-view learning. For each view, a model is created to take advantage of the viewpoints 9

Graph-based Semi-Supervised Learning

4-is one of the approaches that uses a graph to represent the relationships between data points. As the relationships and shared features between data points increase, new features are constructed based on them.9

Generative models

are an essential technique in semi-supervised learning, used to represent the distribution of data. Once this distribution is known, the model can generate new data and determine which class it is likely to belong to. Techniques such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) are used for this purpose. 9

I.5.5 Reinforcement Learning

Reinforcement Learning (RL) is a type of machine learning where an agent learns to make decisions by interacting with its environment to achieve a specific goal. Instead of being given explicit instructions on which actions to take, the agent learns through trial and error, receiving rewards or penalties based on its actions. The goal is to develop a strategy (policy) that helps the agent maximize its cumulative reward over time.9

policy-based

These methods focus on directly learning the policy, where the agent determines the probability of taking each action in a given state without estimating state values. This approach is effective in environments with large or continuous action spaces. Notable examples include REINFORCE and Policy Gradient Methods. 11

value-based

These methods estimate the value of each state or state-action pair in the environment. The agent learns which actions are better by comparing the values. For example, Q-learning and Deep Q-Networks (DQN) 11

model-based

These methods involve building a model of the environment. The agent learns how the environment behaves and makes decisions using this model. This approach can be more sample efficient but requires a good model of the environment. 11

I.6 Deep learning (Dl)

Deep learning is a branch of machine learning that mimics the human brain through the use of multi-layered neural networks. It is characterized by its ability to analyze raw data and extract complex patterns without the need for manual feature selection.These networks process data hierarchically, moving from simple details to more abstract concepts, making deep learning highly effective in image, speech, and text recognition.Thanks to its accuracy and efficiency, deep learning is considered one of the most powerful tools in modern artificial intelligence, widely applied in fields such as computer vision, translation, and speech recognition. We depict in the figure 12 I.4 the relationship between Artificial Intelligence, Machine Learning and Deep Learning.

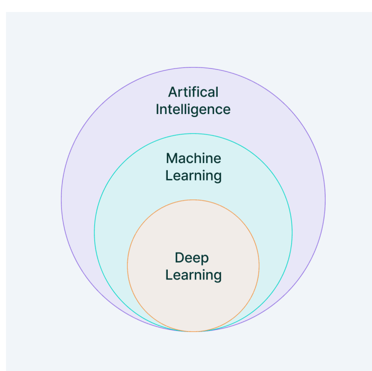


Figure I.4: The Relationship Between Artificial Intelligence, Machine Learning, and Deep Learning [4]

I.7 The difference between machine learning and deep learning

13

	ML	DL
Human intervention	Requires regular tuning and model updates	Minimal intervention once the model is trained
Data	Works with moderately sized datasets	Requires very large datasets
Training	Involves standard computational training	Demands extensive time and computing power for training
Engineering	Based on statistical methods and known algorithms	Relies on neural network architecture
Use	Performs specific tasks based on data and predictions	Handles complex tasks that require advanced pattern recognition

table I.2: Comparison between Machine Learning (ML) and Deep Learning (DL)

The table succinctly distinguishes the fundamental differences between Machine Learning (ML) and Deep Learning (DL) in some of the most significant aspects. On the human involvement factor, ML models require constant maintenance and updating, while DL models depict high autonomy upon training completion. On the data requirement factor, ML functions well with medium-sized datasets, while DL needs massive amounts of data in order to produce high accuracy. In terms of training, ML relies on less complex traditional techniques, while DL needs powerful computational hardware and longer training times due to the depth and complexity of neural networks. Structurally, ML is based on established statistical algorithms, while DL is based on deep and complex neural network architecture. As for uses, ML is able to handle simple and clearly specified tasks while DL does its best when it has to conduct complex tasks involving deep pattern detection such as image and natural language processing.

I.8.1 How Deep Learning Works

Deep learning is a major branch of artificial intelligence, and it relies on what's known as deep neural networks. These networks are made up of layers of interconnected artificial neurons, where each neuron performs a simple mathematical operation—typically a linear function—that helps analyze and represent data within the system.A deep neural network is structured around three main types of layers: Input Layer: This is where the raw data enters the network, such as an image or a piece of text Hidden Layers: These are the core of the network. They process the data step by step, detecting patterns and complex relationships. Output Layer: This layer produces the final result—whether it's a prediction, classification, or decision—based on all the internal processing.Thanks to this multi-layered as demonstrated in the figure I.5 architecture, deep learning models are capable of handling large volumes of data and extracting meaningful insights with high accuracy. This is what makes deep learning so powerful in tasks like image recognition, machine translation, and intelligent assistants as demonstrated in the figure 14

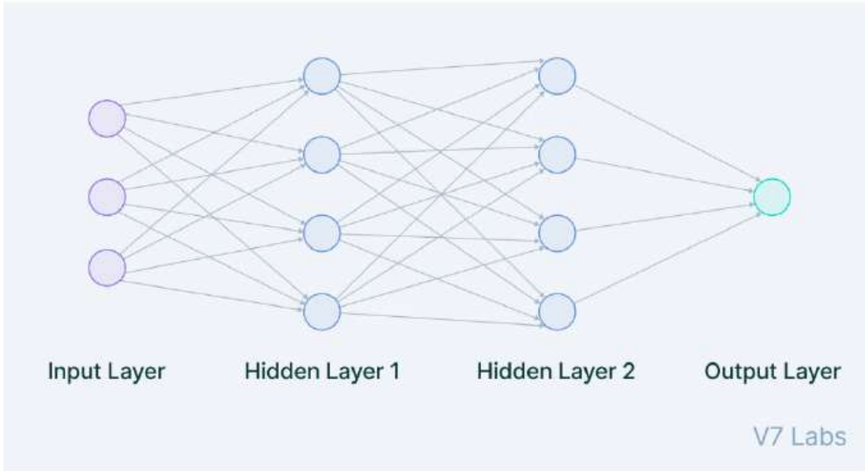


Figure I.5: Illustration of a Deep Neural Network with Input, Hidden, and Output Layers [5]

I.9 Type of deep Neural Network

‘

I.9.1 Feedforward Neural Networks

One of the simplest neural network models is the Feedforward Neural Network (FNN), also known as the Multi-Layer Perceptron (MLP). It is trained using the backpropagation algorithm. The network is made up of neurons, where each neuron receives outputs from the previous layer and processes them using specific weights and biases to generate the outputs of the next layer.It consists of three main layers: an input layer, a set of hidden layers, and an output layer. Each layer contains multiple neurons. This model is commonly used in applications such as image classification and credit scoring. 15

I.9.2 convolutional Neural Networks

Multi-layer perceptrons (MLPs) are not well-suited for image processing because they treat data as vectors, which causes the loss of important spatial information such as shapes. Before deep learning emerged, image analysis relied on manually extracting features, which required a lot of expertise. However, with the introduction of Convolutional Neural Networks (CNNs) by LeCun, this changed dramatically. CNNs can process images directly as matrices or tensors, eliminating the need for manual feature extraction. Today, CNNs are widely used in image classification, segmentation, object detection, and face recognition 15 The Convolutional Neural Network (CNN) consists of three main layers: the convolutional layer, which detects local features in data such as images; the pooling layer, which reduces data dimensions and improves model efficiency; and the fully connected layer, which combines the features for classification or prediction. Among its key advantages are local connectivity and weight sharing, making it efficient in reducing parameters and enhancing performance. It also minimizes the need for manual feature extraction and offers greater stability against slight changes in data. CNNs are widely used in applications such as computer vision (image classification, face recognition, and object detection) as well as in medical image analysis to diagnose diseases and accurately identify lesion locations. 16

I.9.2.1 HOW CNN WORK

The Figure I.6 illustrates how the Convolutional Neural Network (CNN) works The figure I.6 demonstrate the CNN architecture.

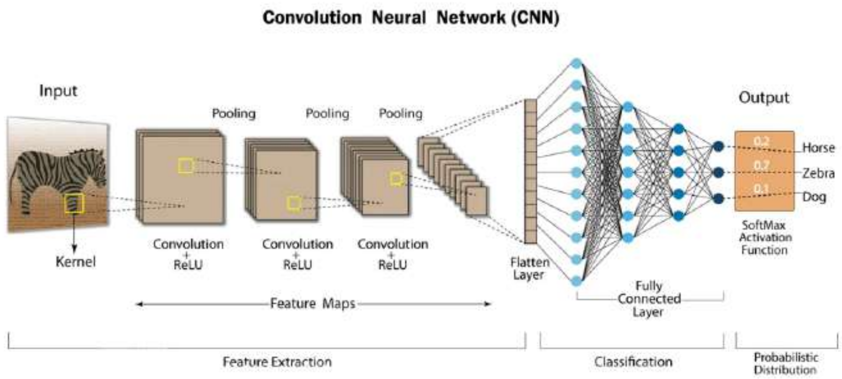


Figure I.6: CNN Architecture[6]

I.9.2.2 CNN-Based Object Detection Models

Some CNN-based models have achievedi ncredible enhancements in object detection:

- R-CNN: It uses region proposals and applies the individual region separately, so it's extremely slow.
- Fast R-CNN: Instead of separately examining every region, it uses the CNN once for the entire image and then region pooling. Therefore, it's 45 times faster at test time and 9 times faster at training time compared to R-CNN.
- Faster R-CNN: It includes a Region Proposal Network (RPN) which produces object proposals directly in the model. It enhances the overall process and also makes it more efficient and fast.
- YOLO (You Only Look Once): YOLO reduces object detection to a single regression problem. It predicts bounding boxes and class probabilities at once. It is extremely fast and is great for real-time systems.
- SSD (Single Shot Detector): Similar to YOLO, but it uses multiple feature maps from different CNN layers to detect objects of varying scales, thus accurate and well-balanced in terms of speed.

17

I.9.3 Recurrent Neural Networks

A recurrent neural network (RNN) is a type of neural network designed to process sequential (time-series) data, as it contains cyclic connections that allow it to retain past information. It consists of several main components: the input cell, which receives data step by step; the hidden cell, which stores previous states and learns short- and long-term dependencies using units such as simple RNNs, LSTMs, or GRUs; the recurrent cell, which processes sequences and updates the hidden state based on the current input and past state; and the output cell, which generates the final output and can feed it back into the network for generation tasks. RNNs are widely used in applications like machine translation, speech recognition, and text summarization, where their memory capabilities allow them to understand and generate sequences effectively.16. The figure I.7 depict the RNN Architecture.

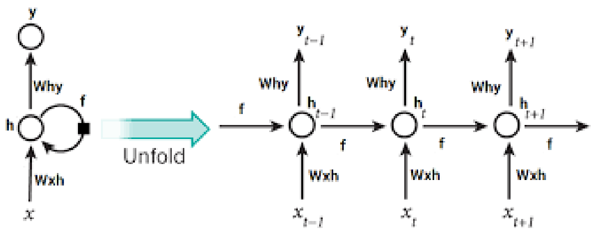


Figure I.7: RNN Architecture[7]

I.9.3.1 Applications of Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a very important part of artificial intelligence, and they are being applied in:

- Natural Language Processing (NLP): machine translation, sentiment analysis, and text generation
- Speech Recognition: audio to text translation for use in virtual assistants.
- Time Series Prediction:stock price prediction, weather forecasting, and sensor data analysis.

18

I.10 YOLO

YOLO is a deep learning-based algorithm , when it was proposed as a One-Stage Object Detection Network. Upon testing, it demonstrated outstanding performance by being able to process 45 frames per second, with ease of performing real-time detection. Due to its high speed and unique usage approach, it was given the name YOLO (You Only Look Once).The YOLO algorithm divides the input image into a grid of cells, where each cell predicts the probability of an object's presence, the bounding box coordinates, and the class of the object. YOLO stands out from two-stage object detectors (such as R-CNN) by processing the entire image in a single pass, making it faster and more efficient, particularly for real-time applications.Following its success, a series of improved YOLO versions have been released, as illustrated in Figure 1, with each version building upon its predecessor by introducing enhancements in speed, accuracy, and the detection of small objects.19 The figure I.8 demonstrates Timeline of YOLO models starting from 2015 to 2024 Architecture.

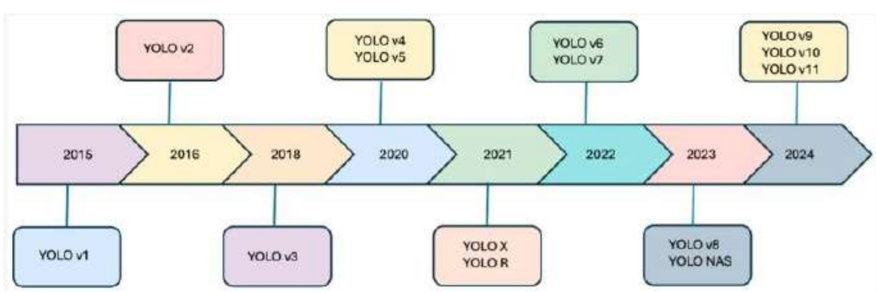


Figure I.8: Timeline of YOLO models starting from 2015 to 2023 Architecture[8]

I.10.1 Evolution of YOLO models

Table I.3 illustrates the progression of YOLO models from their inception to the most recent versions. Each iteration has brought significant improvements in object detection capabilities, computational efficiency, and versatility in handling various CV tasks 20.

table I.3:omparison of YOLO Version

YOLO Version	Year	Tasks	Contributions	Framework
YOLOv1	2015	Object Detection, Basic Classification	Single-stage object detector	Darknet
YOLOv2	2016	Object Detection, Improved Classification	Multi-scale training, dimension clustering	Darknet
YOLOv3	2018	Object Detection, Multi-scale Detection	Supports detection at 3 scales, Darknet-53 backbone	Darknet
YOLOv4	2020	Object Detection, Object Tracking	Mish activation, CSPDarknet53 backbone	Darknet
YOLOv5	2020	Object Detection, Instance Segmentation	Backbone selection, SIOU loss, activation, PAN	PyTorch
YOLOv6	2022	Object Detection, Instance Segmentation	Self-attention, anchor-free OD	PyTorch
YOLOv7	2022	Object Detection, Object Tracking, Instance Segmentation	Transformers, E-ELAN	PyTorch
YOLOv8	2023	Object Detection, Instance Segmentation, Panoptic Segmentation	Pseudo-labeling, anchor-free detection	PyTorch
YOLOv9	2024	Object Detection, Instance Segmentation	Padded GEAR	PyTorch
YOLOv10	2024	Object Detection	Consistent dual assignments for NMS-free training	PyTorch
YOLOv11	2024	Object Detection ,Instance Segmentation, Image Classification , Pose Estimation,Oriented Bounding Boxes	Unified architecture for multi-task learning, enhanced efficiency, and extended output support	PyTorch

We present in the figure I.9 a comparison of YOLO Versions and Their variants.

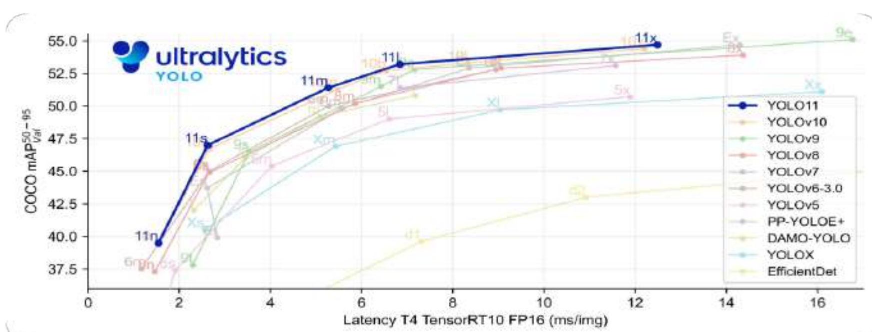


Figure I.9: Comparison of YOLO Models and Other Object Detection Models in Terms of Accuracy and Speed [9]

In this work, we relied on the most recent version in the YOLO object detection serie:

I.10.2 yolov11

The YOLO algorithm has reached a new level with the release of YOLOv11, marking a major leap forward in real-time object detection. This version builds on the successes of earlier models while introducing new features that enhance its usefulness across a wide range of computer vision (CV) applications YOLOv11 stands out due to its increased flexibility and ability to handle more complex CV tasks beyond basic object detection. This includes capabilities like posture estimation and instance segmentation, making the model more adaptable for different use cases. It is specifically designed to strike a balance between high performance and practical deployment, offering improved precision and efficiency.This new release reflects the continuous progress in object detection technology. With its enhanced capabilities and broader applicability,YOLOv11 pushes the limits of what's achievable in CV, and positions itself as a key innovation for future applications across various sectors. 20

I.11 Computer vision tasks supported by YOLO11

I.11.1 Object Detection

Identifying the locations of objects within images or video frames by drawing bounding boxes around them.

I.11.2 Instance Segmentation

Separating each object in an image precisely at the pixel level.

I.11.3 Image Classification

Assigning an entire image to a predefined category.

I.11.4 Pose Estimation

Detecting key points on the body to analyze motion or posture.

I.11.5 Oriented Object Detection

Locating objects while accounting for their rotation angle to achieve more precise positioning.

I.11.6 Object Tracking

Continuously monitoring object movement across video frames

I.12 Architecture of YOLOv11

The YOLO (You Only Look Once) framework brought a major shift to the field of object detection by introducing a unified neural network that performs object localization and classification simultaneously. Unlike traditional two-stage detectors, YOLO's singlestage approach allows for fully end-to-end training, thanks to its fully differentiable architecture. The YOLO model is composed of three main components: Backbone – This is the core feature extractor, which uses convolutional neural networks (CNNs) to process raw input images and generate multi-scale feature representations. Neck – This module acts as a bridge between the backbone and head, combining and refining features from different levels using structures such as Feature Pyramid Networks (FPN) or Path Aggregation Networks (PAN).Head – The head is responsible for the final predictions, generating bounding boxes and class scores from the processed features. Building upon this standard design, YOLO11 advances the capabilities of YOLOv8 through several architectural improvements and optimized parameters. These enhancements contribute to higher detection accuracy and better overall performance, as illustrated in Figure. The following sections provide a detailed explanation of the key changes and innovations introduced in YOLO11. 19 We expose in the figure I.1 the Yolov11 architecture.

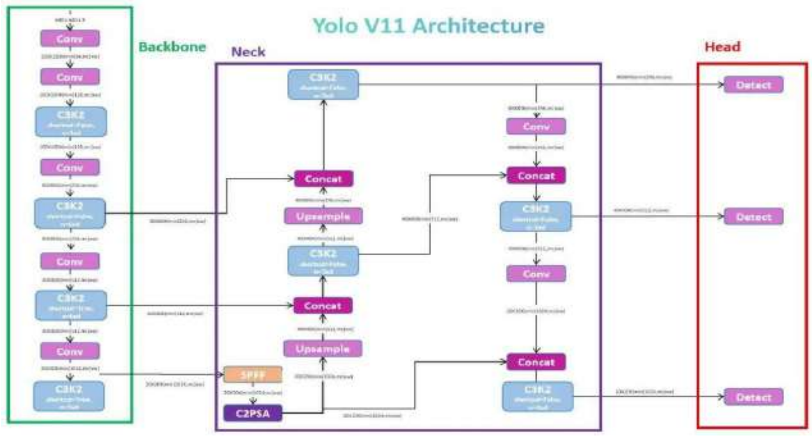


Figure I.10: Architecture yolov11[10]

I.12.1 Backbone

The backbone in YOLO is responsible for extracting features from the input image using convolutional layers and specialized blocks. It produces multi-scale feature maps that help detect objects of different sizes. Common backbones include Darknet-53 (YOLOv3) and CSPDarknet53 (YOLOv4/YOLOv5). These features are then passed to the neck and head for object detection.

I.12.1 Convolutional Layers (Simplified)

YOLOv11 follows a similar structure to earlier versions, starting with convolutional layers that reduce the image size while increasing feature depth. A major update is the new C3k2 block, which replaces the older C2f block. The C3k2 block is a more efficient version of the CSP Bottleneck, using two small convolutions instead of one large one, as in YOLOv8. The smaller kernel size ("k2") helps speed up processing without sacrificing performance.

I.12.1.2 SPPF and C2PSA

YOLO11 keeps the Spatial Pyramid Pooling – Fast (SPPF) block to capture multi-scale features efficiently. It also introduces a new Cross Stage Partial with Spatial Attention (C2PSA) block, which improves the model's focus on important areas in an image. By using spatial attention, the C2PSA block helps YOLO11 better detect objects of different sizes and locations, leading to more accurate results.

I.12.2 Neck

The neck plays a key role in fusing multi-scale features and delivering them to the head for object detection. It typically involves upsampling and merging feature maps from different stages of the network, allowing the model to better recognize objects of various sizes

I.12.2.1 C3k2 Block

YOLO11 replaces the C2f block in the neck with the faster and more efficient C3k2 block. This upgrade improves feature processing speed and overall model performance after up sampling and merging.

I.12.2.2 Attention Mechanism

YOLO11 improves detection by adding the C2PSA module, which uses spatial attention to focus on important image regions. This helps detect small or hidden objects better and sets YOLO11 apart from YOLOv8.

I.12.3 Head

The head of YOLOv11 is responsible for generating the final predictions in terms of object detection and classification. It processes the feature maps passed from the neck, ultimately outputting bounding boxes and class labels for objects within the image.

I.12.3.1 C3k2 Block

In the head section, YOLOv11 utilizes multiple C3k2 blocks to efficiently process and refine the feature maps. The C3k2 blocks are placed in several pathways within the head, functioning to process multi-scale features at different depths. The C3k2 block exhibits flexibility depending on the value of the c3k parameter: When c3k = False, the C3k2 module behaves similarly to the C2f block, utilizing a standard bottleneck structure. When c3k = True, the bottleneck structure is replaced by the C3 module, which allows for deeper and more complex feature extraction. Key characteristics of the C3k2 block: -Faster processing: The use of two smaller convolutions reduces the computational overhead compared to a single large convolution, leading to quicker feature extraction.
- Parameter efficiency: C3k2 is a more compact version of the CSP bottleneck, making the architecture more efficient in terms of the number of trainable parameters.
Another notable addition is the C3k block, which offers enhanced flexibility by allowing customizable kernel sizes. The adaptability of C3k is particularly useful for extracting more detailed features from images, contributing to improved detection accuracy.

I.12.3.2 CBS Blocks

The head of YOLOv11 includes several CBS (Convolution-Batch NormSilu) [19] layers after the C3k2 blocks.

-Extracting relevant features for accurate object detection.

-Stabilizing and normalizing the data flow through batch normalization.

- Utilizing the Sigmoid Linear Unit (SiLU) activation function for non

- linearity, which improves model performance.

-CBS blocks serve as foundational components in both feature extraction and the detection process, ensuring that the refined feature maps are passed to the subsequent layers for bounding box and classification predictions.18

I.12.4 Final Convolutional Layers and Detect Layer

Each detection branch ends with a set of Conv2D layers, which reduce the features to the required number of outputs for bounding box coordinates and class predictions. The final Detect layer consolidates these predictions, which include:

-Bounding box coordinates for localizing objects in the image.

-Objectness scores that indicate the presence of objects.

Class scores for determining the class of the detected object

I.13 Conclusion

In this chapter, we have presented the general principles of artificial intelligence, and more specifically, the fields of computer vision and image processing.We talked about how machine learning and deep learning have been applied in image analysis and what they can accomplish. Additionally, we have discussed deep neural networks categories, with particular emphasis on Convolutional Neural Networks (CNNs) and their application in object detection model building such as YOLO. At the end of the chapter, we focused on several YOLO models and versions, with specific focus on YOLOv11 and explaining how it functions.

Chapter II: Aircraft engine damage diagnosis based on AI paradigms

II.1 Introduction

Aircraft engine inspections begin with an internal- borescope -examination to detect faults in non-visible internal components. The technician inserts the device through engine openings to inspect rotating parts like blades, searching for any damage. When a defect is found, it is measured and assessed for acceptability. This process faces challenges such as: - Limited visibility
- Inadequate lighting
- Low image quality
- Time pressures
- Potential human error
Since results depend on human expertise, the need for automated detection systems has grown to prevent error-related accidents.

II.2 Aircraft Engines: Types and Components

II.2.1 Aircraft Engines Types

The engine is one of the main components of an aircraft, providing the necessary thrust (by pulling in air and pushing it backward forcefully to propel the aircraft forward). There are two types of engines

II.2.1.1 Piston Engine (Reciprocating Engine):

An internal combustion engine (similar to those in cars) that drives a propeller at the front of the aircraft or multiple propellers on the wings. The propeller works like a household fan, but instead of pushing air forward, it pulls air in and pushes it backward with force to move the aircraft forward28. We depict on the figure II.1 a piston engine.

Figure II.1: Piston Engine (Reciprocating Engine)[11]

II.2.1.2 Turbine Engine

This comes in two forms:

1. Some use rotational energy to drive the aircraft's propellers, similar to piston engines.
2. Others utilize the force of hot air being ejected backward to propel the aircraft (in this case, no propellers are needed).

We present in the figure II.2 a turbine engine.

Figure II.2: Turbine engine [12]

II.2.2 Aircraft Engines Components

- Conventional Engine Components
  1. Fan (the plase to air inter).
  2. Compressor.
  3. Combustor (Combustion Chamber).
  4. Turbine
  5. Exhaust Nozzle

As shown in the following figure: II.3

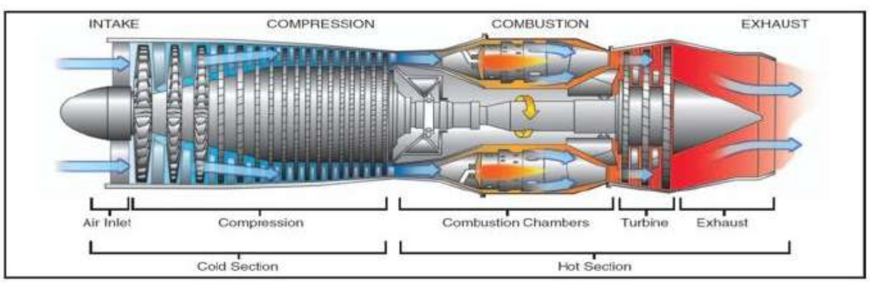


Figure II.3:Aircraft Jet Engine: Main Sections and Components[13]

II.3 Causes of Aircraft Engine Failures

II.3.1 Structural Failures

- Broken engine components (connecting rod, valves, camshaft)
- Engine part wear and tear

II.3.2 Mechanical Failures

- Engine mount issues

- Bolt and nut failures
- Fuel pump malfunctions
- Accessory component problems

II.3.3 Fuel-Related Issues

- Fuel exhaustion due to
- Insufficient refueling
- Fuel leaks
- Fuel contamination with
- Water
- Dirt/debris
- Fuel delivery failure caused by
- Incorrect tank selection
- Clogged fuel lines
- Faulty fuel pump

II.3.4 Other Causes

- Neglected maintenance
- Failure to replace worn/damaged parts
- Clogged air filter
- Engine icing

Tip: Thorough pre-flight checks and regular maintenance significantly reduce the risk of sudden failures.21

II.4 Non-Destructive Inspection (Non-Destructive Inspection/Non-Destructive Testing.) for Aircraft: Purpose, Requirements, and Techniques

35

II.4.1 Purpose

Non-destructive inspection (NDI/NDT) is performed on aircraft to assess the airworthiness of components without causing damage. Techniques range from simple (visual inspection) to advanced (X-rays, eddy current), requiring highly trained and certified technicians.

II.4.2 Inspection Requirements

- The manufacturer or aviation authorities (such as the FAA) specify the appropriate inspection method, as outlined in maintenance manuals.
- Inspectors must be qualified and aware of potential defects and their impact on structural safety.
- Inspection equipment must be regularly calibrated, with proper cleaning and preparation (e.g., paint or part removal) before testing

II.4.3 Inspection Types

II.4.3.1 Visual Inspection

- Uses bright light and magnifying lenses to detect surface cracks.
- Light is directed at a 5-45° angle to enhance defect visibility.

II.4.3.2 Industrial Endoscopes (Borescope) Inspection

Industrial endoscopes are used to inspect tight and hard-to-reach areas inside aircraft engines where traditional methods like visual inspection or X-rays fail. These endoscopes feature high-resolution cameras and powerful LED lighting, providing clear images without the need to disassemble the engine, saving both time and costs. The figure II.4 depicts a Borescope.

Figure II.4: Borescope[14]

The figure II.5 presents an aircraft Engine Inspection by Maintenance Technician Using a Borescope[

Figure II.5: Aircraft Engine Inspection by Maintenance Technician Using a Borescope[15]

II.4.3.2.1 Advantages of Industrial Endoscopes

- Efficiency: Enable fast and precise engine inspections, reducing aircraft downtime.
- Safety:Help detect early faults such as cracks or corrosion before they become serious issues.
- Documentation: Allow recording of images and videos to share with technicians and experts.
- Versatility: Can be used to inspect various components like turbines, fuel systems, and gears.

II.4.3.2.2 Future Developments

With technological advancements, endoscopes will integrate AI for automated data analysis and become even more accurate and durable. They will also become a key part of smart maintenance systems, enhancing the efficiency and safety of the aviation industry. In short, industrial endoscopes are a vital tool for engine inspections, ensuring efficiency, safety, and cost savings.

II.4.3.3 Other Methods:

Ultrasonic testing and magnetic particle inspection detect hidden flaws.

II.4.4 Importance of Routine Maintenance

- Regular inspections reduce unexpected failures and extend aircraft lifespan.
- Inspection intervals are based on flight hours or calendar time, following manufacturer guidelines.

II.5 Automatic Detection of Aircraft Engine Damage Using Machine Learning and Deep Learning

[ref] Damage detection in aircraft engines is one of the important applications of computer vision in quality assurance. Deep learning techniques can be used to automatically detect defects and damage in engines.

II.5.1 Technologies Used

II.5.1.1 Data Collection

- Using high-resolution images of aircraft engines
- Images of various types of damage (scratches, cracks, corrosion, etc.)
- Balanced dataset between intact and defective images for model training

II.5.1.2 Image Preprocessing

- Enhancing contrast and lighting
- Normalizing image sizes
- Reducing noise in images
- Data augmentation to improve model performance

II.5.1.3 Deep Learning Models Used

- YOLO (You Only Look Once): Real-time object detection model
- U-Net: For precise detection of damage areas
- ResNet and EfficientNet: For classifying images as intact/defective

II.5.1.4 Model Training

- Using transfer learning with pre-trained models
- Hyperparameter tuning to optimize performance
- Model evaluation using metrics like accuracy, recall, and F1-score

II.5.2 Practical Application

II.5.2.1 Process Steps

- Image Capture: Using high-resolution cameras or fiber-optic scopes
- Initial Detection: Identifying regions of interest in the engine
- Damage Analysis: Classifying the type and severity of damage
- Results Reporting: Generating automatic reports with damage area identification

II.5.2.2 Challenges and Solutions

- Varying Lighting Conditions: Using image processing to enhance conditions
- Small Damages: Using high-resolution models with increased regions of interest
- Complex Backgrounds: Using background subtraction techniques

II.5.2.3 System Benefits

- Higher inspection accuracy compared to human inspection
- Significant reduction in inspection time
- Early detection of damages before they worsen
- Cost reduction by preventing major failures

II.6 Conclusion

Deep learning models have demonstrated high effectiveness in detecting aircraft engine faults with great accuracy. However, they face a major limitation: their inability to recognize new types of faults they weren't previously trained on.
The core challenge lies in obtaining sufficient data on all potential fault types. The proposed solution involves developing adaptive systems that can learn during operation, immediately collecting and analyzing data on newly encountered faults.
In the following section, we will examine a novel model specifically designed to detect unexpected aircraft engine faults using advanced artificial intelligence techniques.

Chapter III: Assessment of the Proposed AI-Driven Method for Automated Aircraft Engine Inspection

III.1 Introduction

In this chapter, we study the performance of the YOLOv11 model in detecting and classifying damages in aircraft engines. We begin by explaining the development environment and the tools used for training and evaluating the model. Then, we present the evaluation results of the proposed approach based on the experiments conducted. We also assess the model's effectiveness and accuracy in identifying various types of engine damage. The data used for training and evaluation was sourced from the work `ref 1`. We aim to compare the results of `yolov11s.pt` with `YOLOv8s-seg`.

III.2 Development Environment and Tools Used

III.2.1 Hardware Used

We present in the table III.1 the specifications of the hardware used

table III.1: Specifications of the Hardware Used

PC Name	Processor	Operating System	Memory (RAM)	Graphics Card
DESKTOP-OHSL78J	Intel(R) Core(TM) i5-6300U CPU @ 2.40GHz 2.50GHz	Windows 10 Pro	8.0 GB	Intel(R) HD Graphics 520

III.2.2 software used

III.2.2.1 Google Colab

Google Colab is a cloud service that provides a Jupyter Notebook environment that allows users to easily write and run Python code. Colab allows users to improve their Python programming skills and supports the development of deep learning applications using popular libraries such as TensorFlow, Keras, PyTorch, and OpenCV. Additionally, Google Colab offers free access to graphics processing units (GPUs) for a limited time, making it easier to perform computationally intensive operations such as training AI models. 22. Figure III.1 depicts an overview of Google Colab.

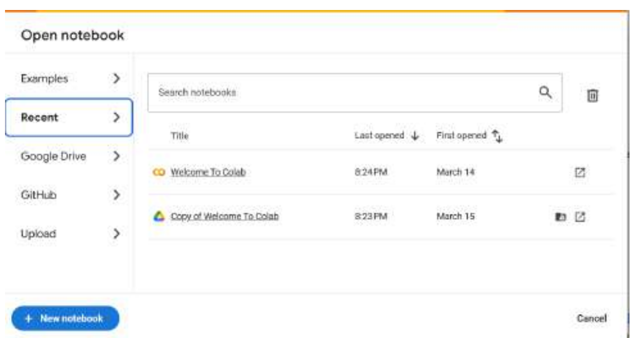


Figure III.1: Google Colab

III.2.2.2 Why Use Google Colab

Designed specifically for machine learning and data analysis, Colab provides users with complimentary access to high-performance hardware such as GPUs and TPUs, which are essential for efficiently training complex models. In deep learning and artificial intelligence, training a model requires feeding it large amounts of test data. The more data the model processes, the more accurate it becomes. However, handling such vast datasets demands significant computational power. This is where Google's cloud infrastructure plays a crucial role. By leveraging Google's powerful GPUs and TPUs, users can train their models directly on Google's servers, significantly enhancing efficiency and performance. 22

III.2.2.3 The Necessity of Working with a GPU

Graphics Processing Units (GPUs) are a critical component in machine learning and deep learning due to their high ability to perform complex computations quickly through parallel processing. This makes them far more efficient than Central Processing Units (CPUs), which may take much longer to perform similar tasks. While the cost of GPUs and cloud computing platforms like AWS and GCP can be high, free tools such as Google Colab offer an excellent alternative, providing users with access to an Nvidia Tesla K80 GPU for free, allowing them to train models efficiently without a significant investment in hardware.22

III.2.2.4 The Python Language

Python is a high-level, versatile programming language known for its readability and simplicity. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming, making it suitable for both beginners and experienced developers. Developed by Guido van Rossum in 1991, Python was named after the British comedy group Monty Python. It has become the most widely used language for teaching programming, as it abstracts much of the complexity, allowing beginners to focus on core programming concepts. Python is widely used in server-side web development, software development, mathematical computing, system scripting, and rapid application development (RAD). Additionally, it serves as a scripting language for integrating software components. Being an open-source language, Python supports modules and packages, enabling modular programming and easy code reuse. Its simple syntax and strong community support contribute to its widespread adoption, with developers continuously expanding its functionality through new libraries and tools 23
The figure III.2 illustrates the concept in the form of Python code

Figure III.2:PYTHON[16]

III.2.2.3 Libraries Used

1-Pytorch : PyTorch is a powerful Python library designed to simplify the development of deep learning projects. Python is favored for its readability and clarity, making PyTorch a natural fit for machine learning practitioners. What sets PyTorch apart is its focus on flexibility, allowing developers to define deep learning models using native, idiomatic Python code.To put it simply, imagine NumPy with the added advantage of GPU acceleration. Even more impressively, PyTorch supports dynamic computation graphs, enabling real-time

changes in the behavior of a network during execution, a feature that distinguishes it from frameworks like TensorFlow, which rely on static computation graphs 25.
2-Ultralytics Ultralytics is an open-source library specialized in object detection, built upon the YOLO (You Only Look Once) models, and is considered one of the leading tools in the field of computer vision. The library offers a comprehensive set of functionalities, including model training, inference, and performance evaluation. It also supports exporting models to various formats such as ONNX and TorchScript, making it easy to deploy across different platforms—from cloud servers to edge devices. Ultralytics is known for its ease of use and efficient performance, and it supports multiple tasks such as image classification, object tracking, instance segmentation, and pose estimation. Additionally, it provides various user interfaces, including a Command Line Interface (CLI) and a Python API, along with an active community and thorough documentation to help developers quickly and efficiently build powerful AI vision solutions. 26

III.2.3 Performance Evaluation Metrics Used

In YOLOv11, several metrics are used to evaluate the model's performance in object detection and image segmentation tasks. Among the most prominent metrics are Precision, which measures the proportion of correctly detected objects compared to all objects predicted as positive, and Recall, which reflects the model's ability to detect all actual objects present in the image. Additionally, mean Average Precision (mAP) is the primary evaluation metric in YOLOv11, measured through mAP@0.5, which represents the accuracy at a 50% Intersection over Union (IoU) threshold, and mAP@0.5:0.95, which averages the performance across multiple IoU thresholds from 0.5 to 0.95. Furthermore, Intersection over Union (IoU) is used to measure the overlap between the predicted bounding box and the ground truth box, reflecting the model's accuracy in detecting object locations.Frames Per Second (FPS) is also measured to assess the model's speed in real-time applications like video surveillance and autonomous driving. Moreover, classification loss and regression loss are calculated to measure errors in object classification and location prediction, helping to improve the model's accuracy **REF**
1-Precision

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

- TP = True Positives
- FP = False Positives

2-Recall

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

- FN = False Negatives

3-Intersection over Union (IoU)

$$\text{IoU (A,B)} = \frac{|A \bigcap B|}{|A \bigcup B|} \tag{3}$$

4-Average Precision (AP) To compute the Average Precision (AP) for a single class, predictions are sorted by their confidence scores, and the area under the precision-recall curve is calculated using the following equation

$$\text{AP} = \sum_{n} (Recalln - Recall(n-1).Precisionn \tag{4}$$

Since our model is designed to detect seven object classes, we compute the mean Average Precision (mAP) as the average of the AP values across all classes. We further employ two common versions of mAP for a more comprehensive evaluation:
°mAP@50: the AP calculated at a fixed IoU threshold of 0.5.
° mAP@50:95: the mean of AP values computed over multiple IoU thresholds ranging from 0.5 to 0.95 with a step size of 0.05, as follows

$$\text{mAP} = \frac{APIoU = 0.5 + APIoU = 0.55 + ... + APIoU = 0.95}{k} \tag{5}$$

Where k is the total number of IoU thresholds considered (typically 10). These metrics were used to evaluate the model's ability to accurately detect and segment objects at different levels of spatial overlap. Among them, mAP was the principal metric used for multi-class object detection performance analysis. In addition to bounding boxes, we also evaluated instance segmentation results, using the same metrics (mAP, precision, and recall) to assess both bounding boxes and segmentation masks.

III.2.4 Experiments

We have identified seven types of damage that are found in aircraft engines, which are:
1-Broken
Broken or damaged mechanical parts As shown in the figure III.3

Figure III.3: Broken

2-Burned
Deterioration or damage caused by high heat may occur, which may affect engine components such as metal surfaces or other parts that have been exposed to extreme heat. parts As shown in the figure III.4

Figure III.4: Burned

3-Corrosion

Corrosion refers to the interaction of materials with the surrounding environment (such as moisture or chemicals) that results in the deterioration of metal parts or other components in the engine.parts As shown in the figure III.5

Figure III.5: Corrosion

4-Tip curl

This phenomenon occurs when the tips of the fan blades are bent or deformed.parts As shown in the figure III.6

Figure III.6: Tip curl

5-Crackk

Small cracks in metal components. parts As shown in the figure III.7

Figure III.7:CRACKK

6-Nick

It means a small scratch or crack in the surface of a metal component. parts As shown in the figure III.8

Figure III.8: Nick

7-Overhated

Components exposed to excessive temperatures deform or degrade. parts As shown in the figure III.9

Figure III.9: Overhated

III.3 Evaluation of the Performance of the Proposed Automated Aircraft Engine Damage Detection System

To train our detection model, we used a dataset containing approximately **3,984 images for training**, **357 images for validation**, and **47 images for testing**. This dataset includes a wide variety of damage types. The images were obtained from **the Algerian airline**, while the videos were sourced from a **borescope**, helping to ensure that our model can effectively detect the different types of damages that may occur in aircraft engines. We selected the **yolov11s.pt** model due to its balance between **accuracy and computational efficiency**, making it highly suitable for real-time damage detection tasks in aircraft engines, especially in environments with **limited processing resources**. Table III.2 exposes the hyperparameters applied during the initial experiments.

Modele	Image dimensions	Batch	Epoch	Number of classes
yolov11s.pt	640×640	16	50	7

Table 5: Table III.2: Hyperparameters applied during the initial experiments

III.3.1 Model training results

The bar graphin the figure III.10, illustrates the detection result of a YOLO model on seven industrial defect categories. The vertical axis denotes the number of detected instances, while the horizontal axis includes the classes of defects. The most frequent defect that is detected is "Tip curl," having approximately 1,450 instances, followed by "Burned" and "Nick" with over 1,000 detections for each. This demonstrates that the model can successfully detect these types of defects. In comparison, the "Corrosion" category includes fewer than 120 detections, indicating the model struggles to detect this flaw perhaps due to insufficient training examples or its visual appearance being too fine-grained. This imbalance between classes replicates a more universal issue referred to as "class imbalance," which can end up reducing the accuracy of a model. To solve this, it is advisable to improve model performance with data augmentation for minority classes, using loss functions such as Focal Loss, or class weighting during training time.

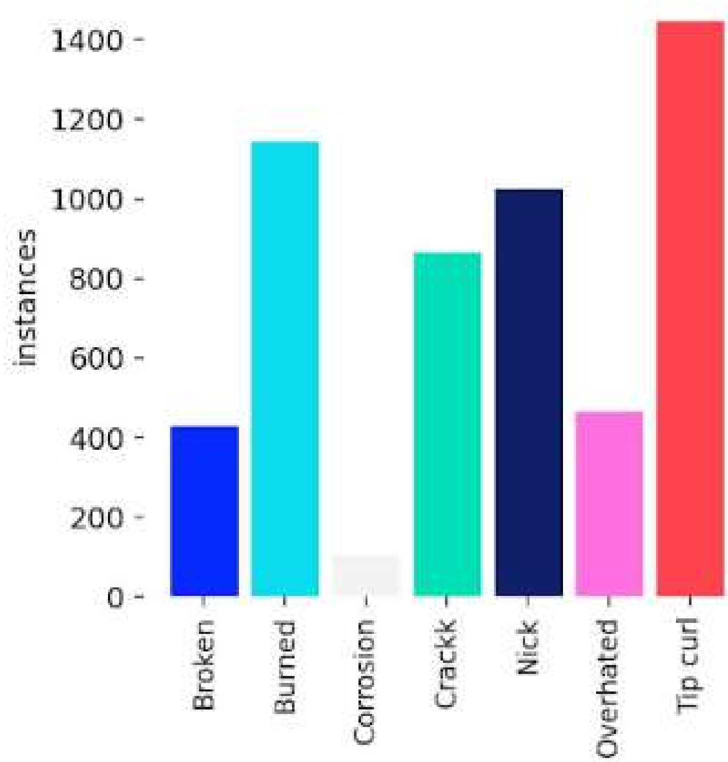


Figure III.10: Distribution of instances according to the seven classes

Figure III.11 depicts Performance Evaluation Curves : Precision, Recall, Confidence, and F1 Score for Multi-Class Classification.

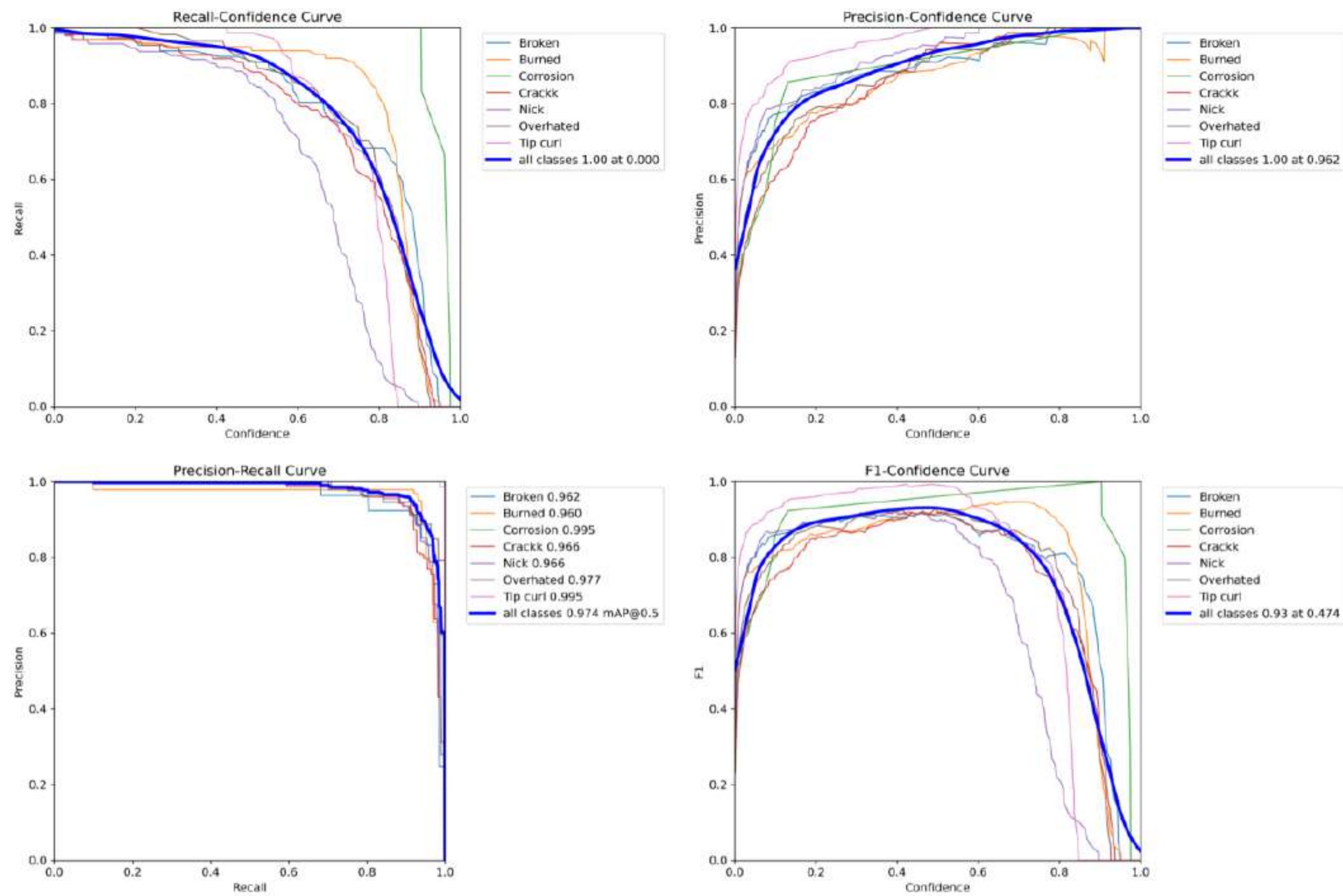


Figure III.11: Performance Curves for Different Classes

Figure III.12 presents the **Normalized Confusion Matrix** to evaluate the performance of the **proposed prediction approach**.

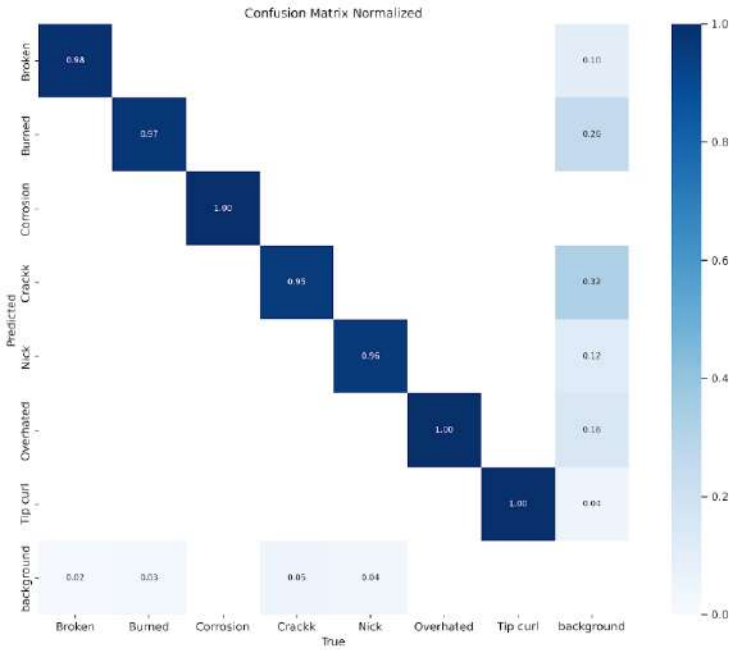


Figure III.12: The confusion matrix

We can analyse the Confusion Matrix as follows:
1-Broken: were correctly classified with 98% accuracy, while 2% were misclassified as background.
2-Burned.:were correctly classified with 97% accuracy, while 3% were misclassified as background
3-Corrosion:The model recognized them excellently, achieving 100% correct classification
4-Crackk:were correctly classified with 95% accuracy, while 5% were misclassified as background.
5-Nick: were correctly classified with 96% accuracy, while 4% were misclassified as background.
6-Overhated:The model recognized them excellently, achieving 100% correct classification
7-Tip curl : The model recognized them excellently, achieving 100% correct classification
The figure III.13 presents the training results of the yolov11s.pt model, including the performance progression across epochs using metrics such as mAP50, Precision, and Loss.

epoch	time	train/box_l	train/cls_lo	train/dfl_lo	metrics/pre	metrics/rec	metrics/mA	metrics/mA	val/box_los	val/cls_loss	val/dfl_loss	lr/pg0	lr/pg1	lr/pg2
1	889.114	1.84041	3.47205	1.59319	0.40475	0.31164	0.22785	0.08797	2.18034	2.59384	2.08907	0.0003018	0.0003018	0.00030
2	994.929	1.94347	2.33387	1.64524	0.45011	0.22914	0.2467	0.10317	2.25221	2.80847	2.02756	0.0005928	0.0005928	0.00059
3	1099.39	1.91055	2.23332	1.64822	0.47039	0.29336	0.24117	0.10872	2.13066	2.54514	1.95074	0.0008718	0.0008718	0.00087
4	1203.85	1.89085	2.14691	1.64064	0.48727	0.29238	0.35597	0.17616	2.06393	2.31731	1.91358	0.000855	0.000855	0.0008
5	1306.91	1.80578	1.95346	1.57253	0.55879	0.38472	0.44309	0.22825	1.90372	1.91299	1.72841	0.000837	0.000837	0.0008
6	1411.2	1.7603	1.86482	1.52124	0.50198	0.45542	0.44986	0.22298	1.78805	1.89496	1.69719	0.000819	0.000819	0.0008
7	1513.16	1.72041	1.78033	1.49864	0.62589	0.43692	0.44647	0.211	1.85392	1.72932	1.6786	0.000801	0.000801	0.0008
8	1616.18	1.68849	1.68173	1.47598	0.58239	0.56336	0.61299	0.3233	1.74569	1.63259	1.63597	0.000783	0.000783	0.0007
9	1720.25	1.64806	1.62599	1.44643	0.51861	0.61847	0.58627	0.30891	1.71955	1.60021	1.54618	0.000765	0.000765	0.0007
10	1822.81	1.60701	1.53212	1.41223	0.6226	0.59814	0.64717	0.33877	1.71445	1.49734	1.54791	0.000747	0.000747	0.0007
11	1927.31	1.58111	1.4794	1.39379	0.55586	0.6457	0.61118	0.31098	1.65419	1.49208	1.51801	0.000729	0.000729	0.0007
12	2031.08	1.54445	1.40848	1.36932	0.63344	0.71164	0.67227	0.3712	1.57626	1.38061	1.44124	0.000711	0.000711	0.0007
13	2136.82	1.54569	1.38112	1.36109	0.66768	0.66916	0.65177	0.34594	1.62045	1.30212	1.47762	0.000693	0.000693	0.0006
14	2240.18	1.52294	1.34895	1.35161	0.61122	0.70358	0.65487	0.36647	1.57016	1.33716	1.45136	0.000675	0.000675	0.0006
15	2344.74	1.49702	1.30264	1.33893	0.6957	0.69741	0.75178	0.43477	1.52939	1.23186	1.40711	0.000657	0.000657	0.0006
16	2448.4	1.48441	1.26742	1.32441	0.65952	0.70423	0.71688	0.39513	1.54402	1.19551	1.38763	0.000639	0.000639	0.0006
17	2553.44	1.43675	1.2133	1.29309	0.66368	0.72364	0.75898	0.45061	1.46726	1.18457	1.34389	0.000621	0.000621	0.0006
18	2656.57	1.43186	1.2028	1.28971	0.72701	0.71755	0.77191	0.45409	1.51787	1.25066	1.41826	0.000603	0.000603	0.0006
19	2760.33	1.42146	1.1727	1.27459	0.63164	0.6658	0.69751	0.40623	1.4698	1.17874	1.3404	0.000585	0.000585	0.0005
20	2864.26	1.3956	1.17974	1.26136	0.77621	0.70449	0.78007	0.47213	1.48226	1.15648	1.36889	0.000567	0.000567	0.0005
21	2967.49	1.39312	1.12261	1.27166	0.78341	0.72446	0.81504	0.5031	1.45027	1.07287	1.35586	0.000549	0.000549	0.0005
22	3071.22	1.37797	1.09929	1.25913	0.75876	0.75122	0.77343	0.45467	1.40444	0.99696	1.29259	0.000531	0.000531	0.0005
23	3172.9	1.36488	1.06701	1.24224	0.74203	0.76334	0.84243	0.54315	1.38849	0.99139	1.29837	0.000513	0.000513	0.0005
24	3276.79	1.34323	1.05266	1.23045	0.77271	0.76688	0.82435	0.51279	1.37319	0.9998	1.26914	0.000495	0.000495	0.0004
25	3378.84	1.3207	1.02083	1.21916	0.7937	0.81191	0.84522	0.5371	1.35978	0.98257	1.28345	0.000477	0.000477	0.0004
26	3480.34	1.3098	1.01248	1.21756	0.83347	0.7772	0.85923	0.55543	1.3292	0.90702	1.26591	0.000459	0.000459	0.0004
27	3584.14	1.284	0.96952	1.19979	0.78292	0.77692	0.82297	0.53116	1.39537	0.92994	1.33915	0.000441	0.000441	0.0004
28	3685.98	1.2721	0.96838	1.19156	0.78428	0.80188	0.84657	0.5461	1.31531	0.92484	1.24037	0.000423	0.000423	0.0004
29	3788.34	1.25434	0.95026	1.17797	0.84465	0.80734	0.88644	0.57492	1.29553	0.87212	1.2509	0.0004051	0.0004051	0.00040
30	3890.07	1.22191	0.88633	1.17059	0.79517	0.83579	0.87642	0.58511	1.28819	0.83778	1.212	0.0003871	0.0003871	0.00038
31	3992.25	1.23428	0.90428	1.16967	0.84854	0.8561	0.89563	0.59218	1.31375	0.82484	1.24059	0.0003691	0.0003691	0.00036
32	4096.2	1.20346	0.87775	1.15195	0.85926	0.85071	0.89626	0.58855	1.28011	0.80861	1.20547	0.0003511	0.0003511	0.00035
33	4197.74	1.19767	0.8618	1.1526	0.80429	0.83407	0.89489	0.60387	1.20377	0.80115	1.16444	0.0003331	0.0003331	0.00033
34	4300.16	1.18052	0.83806	1.13113	0.84168	0.8495	0.90207	0.61375	1.21088	0.81305	1.17281	0.0003151	0.0003151	0.00031
35	4403.46	1.16359	0.81717	1.12893	0.81016	0.8814	0.90584	0.6253	1.22354	0.77395	1.19239	0.0002971	0.0002971	0.00029
36	4506.01	1.14592	0.79727	1.12319	0.83001	0.87981	0.91988	0.63538	1.21855	0.75009	1.18448	0.0002791	0.0002791	0.00027
37	4609.97	1.13693	0.79072	1.11795	0.86927	0.87669	0.93072	0.64154	1.19298	0.7217	1.16536	0.0002611	0.0002611	0.00026
38	4713.61	1.13547	0.77467	1.11341	0.88107	0.89109	0.93241	0.64609	1.1802	0.72768	1.16047	0.0002431	0.0002431	0.00024
39	4816.63	1.10353	0.75785	1.10024	0.8854	0.8805	0.93211	0.65296	1.15847	0.70366	1.14551	0.0002251	0.0002251	0.00022
40	4921.45	1.09645	0.74568	1.09411	0.89053	0.8958	0.93882	0.66327	1.16693	0.71202	1.1442	0.0002071	0.0002071	0.00020
41	5021.28	1.07101	0.68984	1.1124	0.88252	0.89223	0.93977	0.6631	1.1542	0.69874	1.13994	0.0001891	0.0001891	0.00018
42	5119.54	1.03537	0.64583	1.09654	0.89138	0.91803	0.95322	0.68787	1.11435	0.66114	1.1044	0.0001711	0.0001711	0.00017
43	5217.85	1.02325	0.62163	1.08773	0.89444	0.9095	0.95838	0.68828	1.0862	0.63014	1.09879	0.0001531	0.0001531	0.00015
44	5316.95	1.01631	0.60917	1.07932	0.88245	0.92034	0.95862	0.69204	1.08929	0.63228	1.10022	0.0001351	0.0001351	0.00013
45	5415.55	0.98923	0.58877	1.06649	0.89684	0.92182	0.95725	0.70338	1.07861	0.62327	1.09688	0.0001171	0.0001171	0.00011
46	5513.79	0.97567	0.56785	1.056	0.90566	0.91639	0.95825	0.70607	1.05393	0.60571	1.09221	9.9081e-0	9.9081e-0	9.9081
47	5612.02	0.95285	0.56222	1.0491	0.91198	0.91659	0.9619	0.71143	1.06646	0.59963	1.08875	8.10828e-	8.10828e-	8.1082
48	5710.93	0.93512	0.54787	1.03996	0.92017	0.9146	0.97029	0.71504	1.03943	0.57883	1.07084	6.30846e-	6.30846e-	6.3084
49	5810.03	0.92533	0.53578	1.03257	0.92773	0.92334	0.97033	0.7236	1.03011	0.57328	1.06347	4.50864e-	4.50864e-	4.5086
50	5907.44	0.91853	0.53103	1.03604	0.9301	0.93324	0.97426	0.72358	1.02043	0.56094	1.06421	2.70882e-	2.70882e-	2.7088

Figure III.13: Table of Training Results for the Deep Learning Model

At the end of the training process, specifically at epoch 50, the model demonstrated its best performance across various metrics. **The precision** reached a peak value of **93.34%**, while **the recall** achieved **93.01%**, indicating the model's strong ability to distinguish between classes and accurately detect different types of damage. **The mAP@0.5** reached **97.42%**, the highest among all epochs, reflecting a high level of accuracy in object localization using bounding boxes. Additionally, the **mAP@0.5:0.95** achieved a value of **72.35%**, which is a more comprehensive metric that captures performance across varying IoU thresh-

olds, confirming a well-balanced trade-off between precision and recall. These results suggest that the model reached a stable and optimally tuned state by the end of the training phase, with no signs of overfitting, making it well-suited for practical deployment in aircraft engine damage detection tasks.
Figures III.14 and III.15 , III.16 represent the performance curves corresponding to the results presented in the table

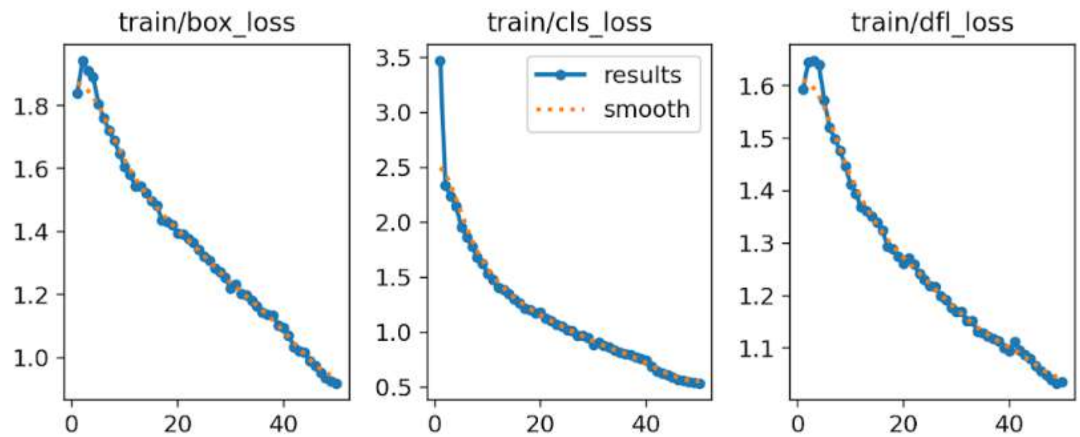


Figure III.14: Loss curves of the training phase.

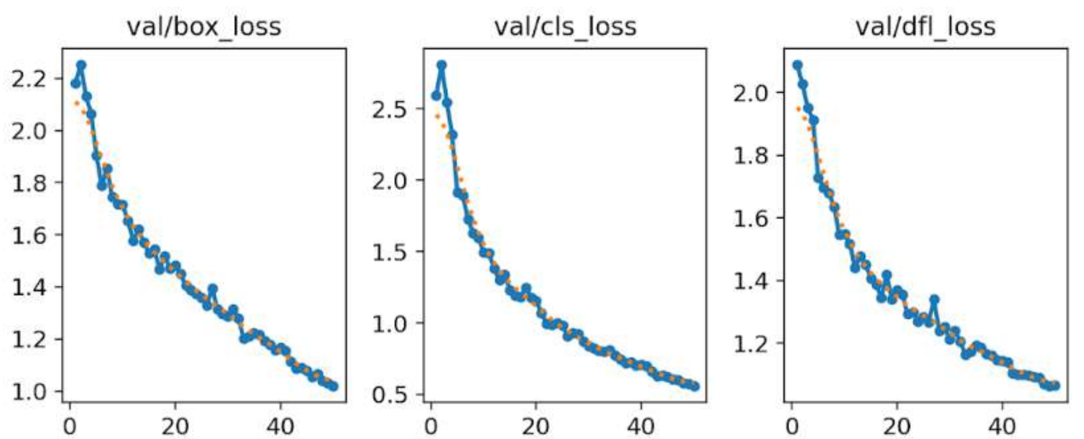


Figure III.15: Loss curves of the validation phase.

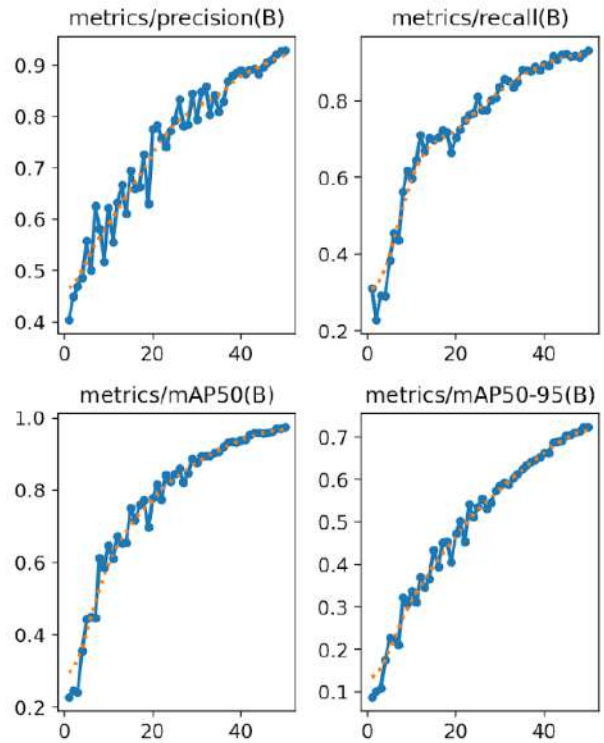


Figure III.16: Progression of performance evaluation metrics throughout the training.

III.3.1.2 Model performance results on validation images

The quantitative results of damage detection using the proposed approach are presented in figure III.17.

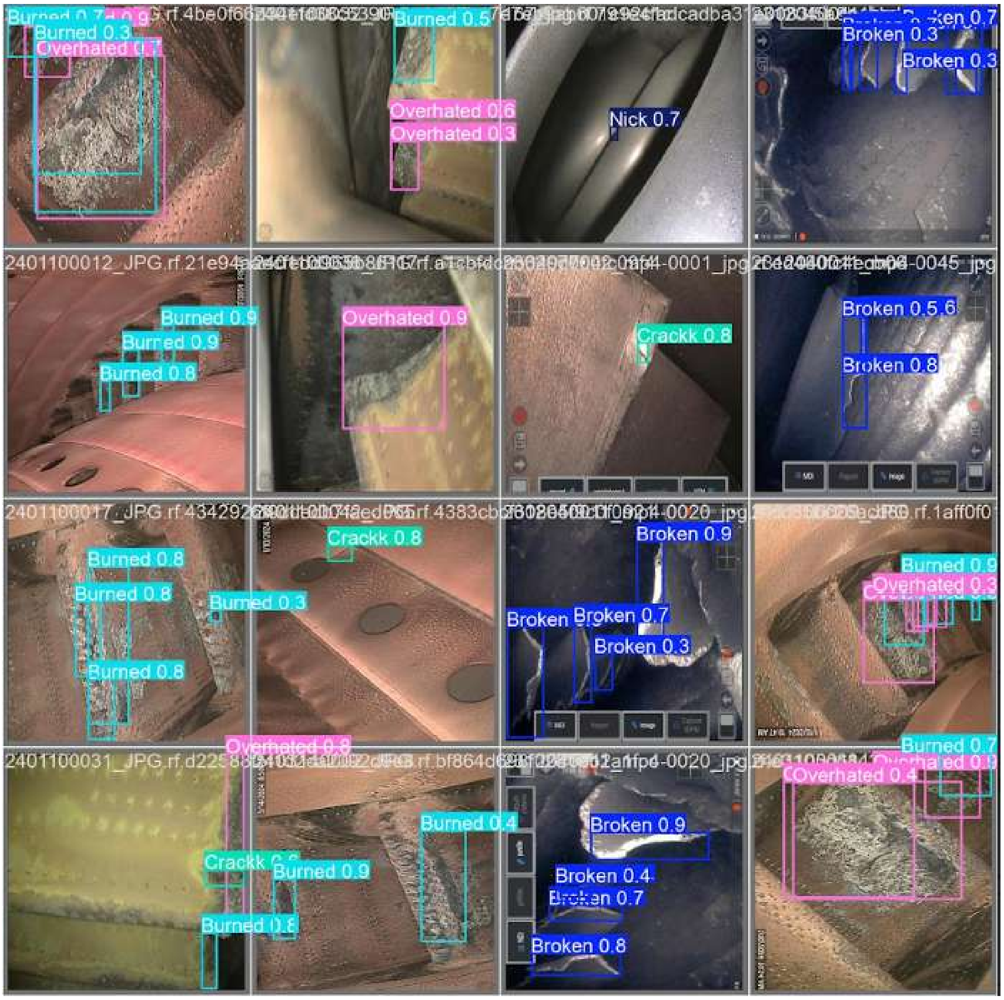


Figure III.17: Obtained results of damages detection by the proposed approach: Yol11 model

III.3.1.3 Performance evaluation of the proposed approach on test data to confirm its generalizability

We tested the model on a borescopic video to assess its effectiveness in detecting aircraft engine damage. We present the results in the following table:

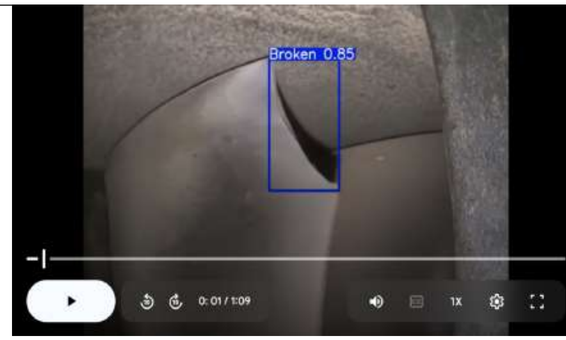
Class	Time	Detection Results
No damage was detected.	0:00	
broken	**0:01 TO 0:07**	
Nick	**0:08**	
Tip curl	**0:08**	
Tip curl	**0:09**	

Table III.3: Temporal Detection of Aero Engine Damage in Borescopic Video Sequences Using YOLOv11 (part1)

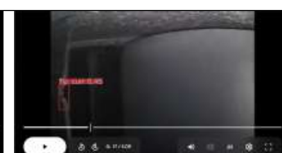
Class	Time	Detection Results
nick	**0:10**	
Tip curl	**0:11 TO 0:15**	
No damage was detected.	**0:11**	
Tip curl and Broken and Nick	**0:17**	 
Nick	**0:18**	
Nick And Tip curl	**0:19**	 

Table III.3: Temporal Detection of Aero Engine Damage in Borescopic Video Sequences Using YOLOv11 (part2)

Class	Time	Detection Results
NICK.	**0:20**	
Tip cur	**0:21**	
Nick And Tip curl	**0:22 TO 0:24**	 
No damage was detected	**0:25 TO 0:26**	
NICK	**0:27**	
No damage was detected	**0:28 TO 0:30**	

Table III.3: Temporal Detection of Aero Engine Damage in Borescopic Video Sequences Using YOLOv11 (part3)

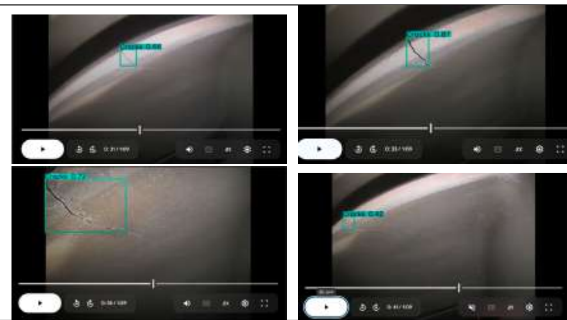
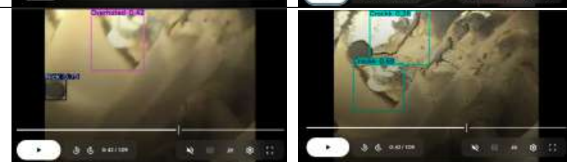
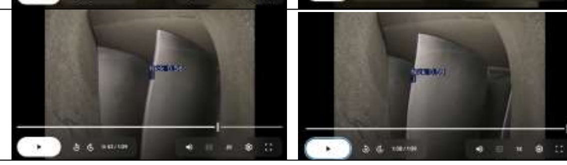
Class	Time	Detection Results
crackk	**0:31 TO 0:41**	
Overrated And Nick And crackk	**0:42 TO 0:47**	
Nick	**0:52 TO 1:09**	

Table III.3: Temporal Detection of Aero Engine Damage in Borescopic Video Sequences Using YOLOv11 (part4)

III.3.1.4 Examples of aircraft engine damage detection results by the model on test images.

Through testing the model training on the video, we found that it was able to recognize many different types of damage. We tested the approach on several images.

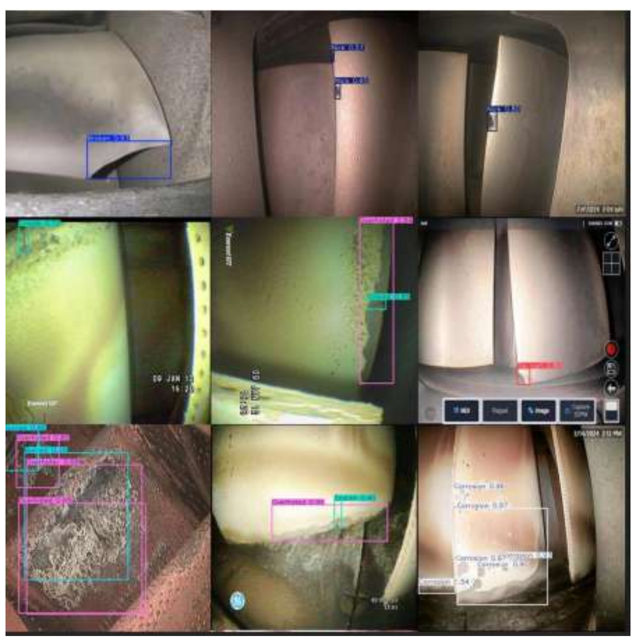


Figure III.18: Obtained results of damages detection by the proposed approach: Yol11 model on testing images

III.4 Presentation of a Single Interface from Our Approach

In this project, we designed an interface using the Streamlit library to display the results of a pre-trained YOLO model on images uploaded by the user. The application detects objects within the image and presents the output visually. To make the application accessible online, we used the Ngrok service, which creates a secure tunnel between the local application and the user's device, allowing the interface to be accessed through a public link without the need for external hosting. This approach is effective for testing and easily sharing models in local development environments. As demonstrated in the figure III.19

Figure III.19: Main Interface of the Application

The figure III.20 below shows the image upload feature from the device

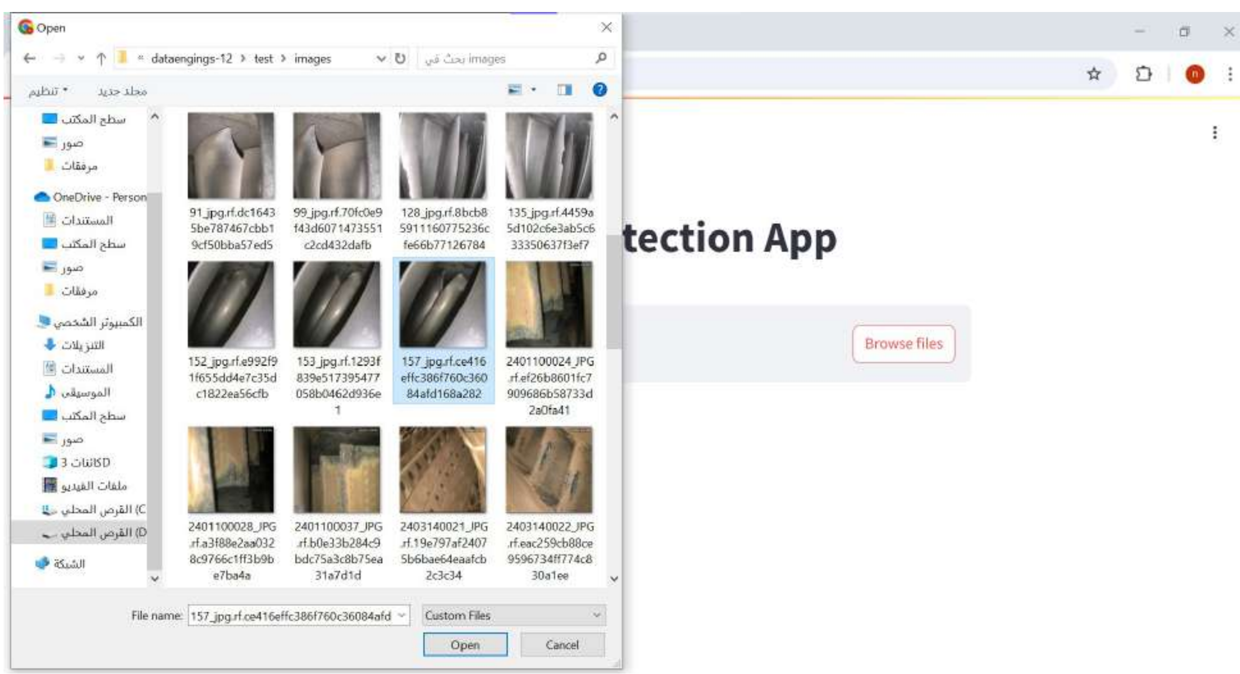


Figure III.20: Selecting an inspection image from the dataset folder to use in the defect detection application

The figure shows the image being displayed before the model testing process

Figure III.21: The image on which the model was tested

The figure shows the image being displayed after the model testing process

Figure III.22: The image after the damage was detected

III.5 Conclusion

Throughout this chapter, the performance of the new aircraft engine damage detection model using YOLOv11 was comprehensively assessed using an extensive experiment set on a balanced damage dataset for seven types of damage. The results indicated that the model has high performance with an mAP@0.5 of approximately 97.42%, high precision, and recall, indicating that the model is perfectly capable for classification as well as precise localization. Furthermore, experiments conducted on video streams and images confirmed the stability of the model in real-world applications, highlighting its application in computer vision-based intelligent inspection systems. The findings validate the feasibility and effectiveness of the proposed method as a viable solution to automating aircraft engine inspection operations.

GENERAL CONCLUSION

Borescope inspection is a useful non-destructive visual examination used to assess the internal components of equipment without disassembling or destroying it, especially in hard-to-reach areas. The majority of equipment can be assisted by this inspection, such as aircraft engines, turbines, and other rotating machinery. There are specialized firms such as Applied Technical Services that offer borescope inspection services for various applications, such as aircraft engines. Disassembling an airplane engine is a tough and expensive operation, and hence borescope inspection emerges as a viable and safe technique for ascertaining the condition of internal parts. It allows crack detection, corrosion, malfunctioning, leaks, and other visible faults without halting operations or jeopardizing equipment. Inspectors apply advanced equipment such as borescope cameras and Snake Eye cameras to document the condition with images and video footage, presenting technical reports to customers. In the event that an alleged flaw is identified, additional Non-Destructive Inspection (NDI) can be done to accurately define the size of the defect. Such procedures secure flight safety via defect identification in the early stages prior to them causing additional complications or failure calamities.26

Video borescope is a useful non-destructive testing tool used in the examination of aircraft engines and other equipment. It is capable of capturing images and videos to document the inspection and is equipped with advanced lighting to show impurities and defects. It is mainly used for the detection of cracks, corrosion, and debris in areas that are inaccessible. Inspectors face the difficulty of needing to discover minute cracks and debris accumulation, but the borescope helps by quickly and accurately diagnosing such issues, which contributes to the aircraft's safety.27

Although the **borescope** is a key technology for detecting aircraft engine damage, it is not entirely safe, as some damages may be missed during visual inspection or due to unclear images. With advancements in artificial intelligence, machine learning, and deep learning, we have experimented with computer vision models to detect damages in aircraft engines. This approach enhances the ability to detect early damages that may be invisible using traditional techniques.We combined deep learning techniques with traditional methods to detect various types of damage, and as a result, we obtained a large dataset consisting of images captured from the airline company **Air Algérie** We used the **YOLOv11** model on seven damage categories, which helped improve the accuracy and efficiency of detecting damages in aircraft engines.

Difficulty in Detecting Small Defects: Due to loss of detail in the image caused by processing within CNNs, detection of minute defects—such as tiny scratches or cracks—is not possible. **Variability in Images Captured Using Different Cameras** Borescope images are nonhomogeneous in quality, illumination, and orientations induced due to use of different cameras and inspection situations, hence limiting the overall generalizability of the model while also affecting detection performance negatively.

References

1 https://www.ultralytics.com/ar/blog/exploring-how-the-applications-of-computer-visio

2 https://www.ultralytics.com/ar/blog/exploring-image-processing-computer-vision-and-m

3 https://www.uoanbar.edu.iq/ComputerCollege//catalog/CS_depart/lectures/CS_4_1_ip.pdf

4 https://www.ultralytics.com/glossary/machine-learning-ml

5 https://www.ultralytics.com/glossary/supervised-learning

6 https://www.geeksforgeeks.org/getting-started-with-classification/

7 https://www.geeksforgeeks.org/regression-in-machine-learning/

8 https://www.ultralytics.com/glossary/unsupervised-learning

9 https://medium.com/data-science-collective/semi-supervised-learning-smarter-models-w

10 https://www.ultralytics.com/glossary/reinforcement-learning

11 https://www.linkedin.com/pulse/reinforcement-learning-concepts-applications-rocio-su

12 https://infinitesights.com/deep-learning/#what-is-deep-learning

13 https://www.zendesk.com.mx/blog/machine-learning-and-deep-learning

14 https://phoenixnap.com/blog/what-is-deep-learning

15 DifferentTypesofNeuralNetworksandApplications:EvidencefromFeedforward,Convolutionala PDF

16 https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-m-hdstat-rnn-deep-learning pdf

17 https://easyodm.tech/object-detection-with-cnns

18 https://www.ultralytics.com/glossary/recurrent-neural-network-rnn

19 A_Review_of_YOLO_Object_Detection_Algorithms_based.pdf

20 https://arxiv.org/html/2410.17725v1

21 https://airplanesworld.com/

22 https://bytexd.com/what-is-google-colab-a-beginner-guide/

23 https://www.teradata.com/insights/data-platform/what-is-python-programming-language

24 https://medium.com/we-talk-data/what-is-tensorflow-in-python-tensorflow-getting-sta

25 https://medium.com/data-science/what-is-pytorch-a84e4559f0e3

26 https://docs.ultralytics.com/help/FAQ/

27 https://atslab.com/inspection/aerospace-ndi/borescope-inspection-aircraft-engine/?utm_source=

28 https://pia.edu/blogs/borescope-use-in-aviation/?utm_source=chatgpt.com

29 I.Uzun,œDamagedetectioninaircraftengineborescopeinspectionusingdeeplearning, 2023.

30 X.Huang,L.Li,J.Zhang,D.Yin,X.Hu,andP.Du,œAnaero-enginedamagedetectionmethodwithlow-c Electronics,vol.11,no.13,p.2093,2022.

31 C.auxprojetsWikimedia,œAccidentduniliouchineil-76algrienen2018,URL,January252024, accessed:2024-05-20.

32 C.Y.Wong,P.Seshadri,andG.T.Parks,œAutomaticborescopedamageassessmentsforgasturbinebl inAIAAScitech2021Forum,2021,p.1488.

33 Y.Abdulrahman,M.M.Eltoum,A.Ayyad,B.Moyo,andY.Zweiri,œAero-enginebladedefectdetection Asystematicreviewofdeeplearningmodels,IEEEAccess,2023.

34 H.Shang,J.Wu,C.Sun,J.Liu,X.Chen,andR.Yan,œGlobalpriortransformernetworkinintelligent IEEETransactionsonIndustrialInformatics,2022.

35 https://e3arabi.com/%D8%A7%D9%84%D9%87%D9%86%D8%AF%D8%B3%D8%A9/%D8%A7%D9%84%D9%81%D8%AD%D8%B5-%D8%A3%D9%88-%D8%A7%D9%84%D8%A7%D8%AE%D8%AA%D8%A8%D8%A7%D8%B1-%D8%BA%D9%8A%D8%B1%D8%A7%D9%84%D8%AA%D8%AF%D9%85%D9%8A%D8%B1%D9%8A-%D8%B9%D9%84%D9%89-%D8%A7%D9%84%D8%B7/

ref SelmiNadine"Contributiondesparadigmesdel'intelligenceartificiellepourladtectiond' ventuelsdommagesauniveaudesmoteursd'avion"MASTERACADEMIQUE(2024)UniversitEchahidHamm

References for Figure

1 https://cdn.prod.website-files.com/6479eab6eb2ed5e597810e9e/67ed58d6033e50097fb8d633
678636efd68e8b6c1dd37c9f_66ad1c7d22ac043f3861394b_66ad1b83c84282b3b92a721b_
General%25252520Vision%25252520Model_fig1.png

2 https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQlnO4Wow_Vwo021NOOvn8tV_
en62UyHPvqNQ&s

3 https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSDgiKTy9c2UO-5caOLnSgtLpNhNVA-
2RoKw&s

4 https://framerusercontent.com/images/fnk54mdpjaJ4gGkIJVForm8z76U.png

5 https://framerusercontent.com/images/1oemDyPGPWGzC8Cpq3EHxfwno.png

6 https://easyodm.tech/object-detection-with-cnns

7 https://iq.opengenus.org/when-to-use-recurrent-neural-networks-rnn/

8 https://www.mdpi.com/2073-431X/13/12/336

9 https://docs.ultralytics.com/ar/models/yolo11/

10 https://www.analyticsvidhya.com/blog/2024/10/yolov11-object-detection/

11 https://aviationgoln.com/reciprocating-engines/

12 https://aviationgoln.com/reciprocating-engines/

13 https://insighttoengineering.wordpress.com/gas-turbine-engine/

14 https://scopeheal.com/borescope/

15 https://www.avnndt.com/boroscopic-inspection

16 https://www.google.com/imgres?q=python&imgurl=https%3A%2F%2F149860134.v2.
pressablecdn.com%2Fwp-content%2Fuploads%2Fpythoned.png&imgrefurl=https%3A%
2F%2Fquantumzeitgeist.com%2Fwhy-is-the-python-programming-language-so-popular%
2F&docid=n-uxTA8qPtkeVM&tbnid=j6FbFpGHW7sfiM&vet=12ahUKEwi9nuzt94KNAxX29bsIHUSJMmsQM
.i&w=900&h=900&hcb=2&ved=2ahUKEwi9nuzt9