



HAL
open science

Contrôle visuel de la qualité perçue par apprentissage automatique

Nassime Mountasir, Bruno Albert, Baptiste Lafabregue, Nicolas Lachiche

► To cite this version:

Nassime Mountasir, Bruno Albert, Baptiste Lafabregue, Nicolas Lachiche. Contrôle visuel de la qualité perçue par apprentissage automatique. ORASIS 2023, Laboratoire LIS, UMR 7020, May 2023, Carqueiranne, France. <hal-04219307>

HAL Id: hal-04219307

<https://hal.science/hal-04219307v1>

Submitted on 27 Sep 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Contrôle visuel de la qualité perçue par apprentissage automatique

Nassime Mountasir^{1,2}

Bruno Albert²

Baptiste Lafabregue¹

Nicolas Lachiche¹

¹Laboratoire ICube, Université de Strasbourg, Illkirch-Graffenstaden

²ENGLAB, Technology & Strategy Engineering, Schiltigheim

nassime.mountasir@icube.unistra.fr

Résumé

Afin de veiller à la qualité des produits développés dans un contexte industriel, il est nécessaire de recourir à des tâches de contrôle de la qualité perçue des pièces fabriquées. Cependant, cette tâche est complexe, car à la fois chronophage mais également variable en fonction des opérateurs en raison de la subjectivité dans la tâche d'inspection. Nous proposons ainsi une approche visant à détecter, classer et évaluer les anomalies de qualité perçue afin d'aider les opérateurs dans cette tâche. Dans le cadre de l'évaluation de notre méthode, nous avons créé un nouveau jeu de données. Notre approche est évaluée sur deux critères, la détection automatique des défauts et l'estimation de leur degré de gravité.

Mots clés

Inspection visuelle, contrôle qualité, apprentissage automatique, standardisation, identification de défauts

Abstract

In order to assure the quality of industrial products, it is necessary to implement perceived quality checks in the manufactured parts. However, this task is complex because of its time-consuming and hazardous aspects, due to the subjectivity in inspection of different operators. We propose an approach aiming to detect, classify and evaluate the defects in perceived quality to help the operators in this task. To evaluate our methodology, we have created a new use case. Our approach is evaluated based on two criteria, the automatic detection of defects and the estimation of their criticality.

Keywords

Visual inspection, quality control, machine learning, standardization, defects identification

1 Introduction

1.1 Contrôle de la qualité perçue

Dans le domaine industriel, les potentielles défaillances ou dérives des systèmes de production peuvent entraîner des anomalies, correspondant à des déviations par rapport à

l'état attendu, ou même des défauts, à savoir des anomalies non conformes, sur les pièces fabriquées[1].

Il est donc nécessaire de pouvoir s'assurer de la conformité d'un produit vis-à-vis des besoins et demandes d'un client, mais également eu égard aux contraintes et normes en vigueur. La vérification de la qualité des pièces produites correspond ainsi au contrôle qualité dans ce domaine.

Ce contrôle, dans beaucoup de domaines mais en particulier celui du contrôle de la qualité perçue, est encore assuré pour tout ou partie par des humains, appelés opérateurs de contrôle. La qualité perçue implique en effet la perception sensorielle de l'être humain qui observe une pièce.

Malgré les différentes méthodes existantes pour cadrer ce type de contrôle[2], ces tâches conservent un caractère variable en raison notamment de la subjectivité de chaque opérateur et des possibles variations dans les observations. Ainsi, deux opérateurs pourraient par exemple donner deux niveaux de gravité différents lors de l'évaluation d'une même anomalie.

Pour pouvoir accompagner les opérateurs dans leur tâche d'inspection, il est nécessaire de pouvoir détecter automatiquement des anomalies présentes sur des surfaces industrielles.

1.2 Inspection automatique

Afin de contrôler la qualité d'une pièce de manière automatisée, de nombreuses approches sont possibles[3] : l'utilisation de capteurs, de critères pour identifier les anomalies, ou encore l'inspection visuelle.

Pour procéder à l'inspection visuelle d'une pièce, un opérateur humain doit inspecter chacune des pièces, et déterminer si celle-ci présente un défaut ou non. Le cas échéant, il doit déterminer de quel type d'anomalie il s'agit et relever les propriétés mesurables associées. Enfin, selon un seuil prédéfini, il précise si la pièce est conforme ou non aux spécifications.

De nombreux problèmes sont induits par ce type de contrôle. Dans un environnement industriel dans lequel des milliers de pièces sont produites chaque jour, il est extrêmement coûteux d'inspecter manuellement chacune des pièces pour s'assurer de sa conformité, que ce soit en terme de personnel disponible, ou de temps dédié à la seule tâche

d'inspection, représentant ainsi une charge mentale importante pour l'opérateur.

De plus, il est souvent difficile d'estimer, de manière fiable et cohérente, à quel point une anomalie est critique ou non. Il peut arriver qu'une pièce possède plusieurs critères de conformité qui ne sont pas toujours parfaitement déterminés dans le cahier des charges. Permettre d'estimer automatiquement le degré de gravité d'une anomalie pour un opérateur permet ainsi de faciliter son travail.

Nous avons ainsi travaillé sur une approche visant à détecter automatiquement les anomalies présentes sur des surfaces industrielles et à prédire leur degré de gravité tout en améliorant les performances de contrôle. Cette approche s'intègre dans un outil mobile destiné à guider et aider les opérateurs dans leurs tâches d'inspection.

La suite de l'article est décomposée comme suit : la section 2 décrit des approches similaires et les travaux connexes dans le contrôle automatisé de la qualité perçue, la section 3 présente notre approche permettant de détecter automatiquement les anomalies présentes sur des surfaces, la section 4 décrit les résultats de l'entraînement des modèles et des mesures de la fiabilité des prédictions réalisées, la section 5 présente une analyse des résultats obtenus et la section 6 apporte une conclusion à l'étude menée.

2 Travaux connexes

Dans le domaine du contrôle de la qualité perçue, différentes approches existent.

Une première famille d'approches consiste en l'implémentation de réseaux de neurones convolutifs dits classiques, visant à faire une classification sur des images.

Balzategui et al.[4] implémentent des réseaux de neurones convolutifs pour détecter la présence ou non de défauts sur des images de cellules photovoltaïques, afin de pouvoir segmenter sur celles-ci les différentes anomalies. Yang et al.[5] mettent en place un modèle entraîné à partir de VGG[6] pour l'inspection de pièces soudées à partir de lasers, processus très utilisé dans la production industrielle de batteries.

Cette catégorie d'approche ne permet cependant pas de détecter la présence de plusieurs classes d'objets sur l'image, et ne donnent aucune information sur la localisation des objets sur celles-ci. Ainsi, il n'est pas possible de connaître la position des anomalies, ni si celles-ci occupent une place importante ou non au sein de l'image.

D'autres approches ne se contentent pas uniquement de classer les images, mais également de détecter les différents objets présents sur une image. Ces approches apportent une information supplémentaire sur la classe des objets détectés, mais également sur leur localisation dans l'image.

Kardovskyi et al.[7], Gong et al.[8] et Kuric et al.[9] proposent par exemple la création d'un modèle basé sur R-CNN[10] pour inspecter respectivement la qualité de barres en acier sur des chantiers de construction, la présence de défauts sur des pièces de faible diamètre et des circuits im-

primés sur des cartes électroniques.

Enfin, certaines approches implémentent des modèles adaptés au contrôle de la qualité perçue. Shafiee et al.[11] développent une architecture de réseau de neurones convolutifs permettant l'inspection de la qualité des pièces à haut-débit, tandis que Liu et al.[12] proposent un modèle similaire à Faster R-CNN[13] pouvant détecter et localiser des défauts sur des surfaces en extrayant des régions de l'image.

Cependant, ces approches ne sont en général testées que sur des jeux de données spécifiques, en l'occurrence le jeu de données NEU-DET[14], comprenant des images en noir et blanc de différents défauts sur des bobines en métal, ne permettant pas de s'assurer de leur performance sur un cas réel et plus spécifique.

3 Expérimentations

3.1 Cas d'application

Pour développer et tester notre approche, peu de jeux de données sont disponibles en libre accès, et ce, notamment pour le problème rencontré : la plupart des ressources disponibles ne comportent qu'une seule anomalie, étant donné que celles-ci ont pour but uniquement de distinguer des pièces défectueuses de pièces non-défectueuses, et pas de classer les défauts[15].

Pour répondre à ce besoin, nous avons ainsi créé un cas d'application composé de plaques en PVC, cette matière étant une matière plastique très largement utilisée notamment dans le secteur du bâtiment (canalisations, profilés de fenêtre, revêtements de sol).

A partir de 6 plaques de coloris différents en PVC de 1000x1000mm, nous avons créé pour chaque couleur 100 plaques de 50x50mm, respectivement de couleur rouge, bleu, vert, jaune, noir et blanc. Un échantillon utilisé dans la suite de l'approche est présenté en figure 1.



FIGURE 1 – Échantillon de pièces utilisées correspondant respectivement, de gauche à droite et de haut en bas, à un trou, une tâche, une encoche, un trou de diamètre différent, une marque et une rayure

Nous avons ensuite réfléchi aux différents défauts couram-

ment présents dans l'industrie pouvant être présents sur des produits dans cette matière, et conçu un plan de réalisation de ces différentes anomalies, en créant pour chaque type, un niveau de gravité différent allant de 1 à 4. Dans le cas de défauts avec une seule propriété mesurable, chaque valeur de propriété correspond à un degré de gravité. Les autres défauts présentent une gravité graduelle en fonction de la combinaison des deux propriétés. La table 1 indique les différentes propriétés mesurables des défauts.

TABLE 1 – Propriétés mesurables associées aux différentes anomalies

Nom de l'anomalie	Propriété mesurable 1	Propriété mesurable 2
Rayure	Taille	Forme
Encoche	Largeur	Longueur
Trou	Diamètre	
Tâche	Diamètre	
Marque	Diamètre	

5 types de défauts ont ainsi été réalisés sur celles-ci, à savoir des rayures, des trous, des marques (correspondant à la pression d'un emporte-pièce sur la surface), des encoches et des tâches.

Nous avons pu ensuite prendre des photos avec un appareil mobile des différentes pièces dans deux conditions différentes : un éclairage naturel avec fond blanc, et un éclairage artificiel avec fond noir, afin de pouvoir distinguer certains défauts sur les pièces noires et blanches.

Le jeu de données ainsi constitué est composé de 2032 images, avec 448 images de trous et de marques, 454 images de tâches et d'encoches, ainsi que 228 images de rayures.

3.2 Détection de défauts

Pour pouvoir détecter automatiquement les défauts sur des surfaces, le choix s'est porté sur le modèle YOLOv5[16] pour sa facilité d'implémentation, mais également car il possède de très bonnes performances dans la réalisation d'inférences en temps réel.

Contrairement aux autres modèles convolutifs basés sur des régions tels que R-CNN[10], Fast R-CNN[17] ou Faster R-CNN[13], YOLO raisonne sur l'image dans sa globalité et non de manière indépendante sur différentes zones de l'image.

Pour ce faire, le modèle divise l'image en une grille de taille $S * S$ afin de générer un certain nombre de boîtes d'ancrage (*anchor boxes*), qui sont des boîtes symbolisées par des rectangles de hauteur et de largeur variable, ayant pour but de correspondre aux différents objets présents dans l'image.

Ces boîtes permettront ainsi de détecter des objets de taille différente, mais également des objets se chevauchant. Les boîtes ayant une probabilité de contenir un objet inférieure à un certain seuil sont ensuite supprimées, permettant un premier élagage dans la détection.

A la suite de cela, on dispose toujours de plusieurs dizaines voire centaines de boîtes d'ancrage pour un même objet malgré ce traitement. La suppression non-max (algorithme 1) permet de filtrer les boîtes afin d'associer seulement une boîte à un objet dans l'image selon un certain seuil.

Algorithme 1 Suppression non-max

```

seuil = 0.5
ordonner les boîtes par probabilité croissante
répéter
    sélectionner  $b \in boites$  avec  $P(b) = \max(P(b), boites)$ 
    si  $P(b) < seuil$  alors
        supprimer  $b$ 
    sinon
        pour chaque  $inter \in intersections(b)$  faire
            si  $IoU(b, inter) < seuil$  alors
                supprimer  $inter$ 
            fin si
        fin pour
    fin si
jusqu'à ce que aucune boîte n'est supprimée

```

Ce seuil est défini par l'Intersection over Union *abrégée IoU*). Dans la pratique, cela correspond à l'intersection entre la zone prédite correspondant à un objet et la zone correspondant réellement à l'objet (appelée zone de chevauchement), divisée par leur union.

Une fois ce processus effectué, il ne reste plus qu'à récupérer les boîtes d'ancrage restantes, qui correspondent aux boîtes englobantes (*bounding boxes*) des différents objets présents dans l'image. La figure 2 résume le modèle précédemment exposé.

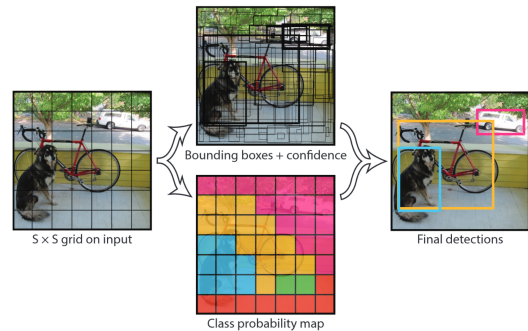


FIGURE 2 – Fonctionnement du modèle YOLO décrit dans l'article *You Only Look Once : Unified, Real-Time Object Detection*[18]

L'architecture du modèle YOLO est, elle, constituée de 6 blocs de convolution, ainsi que de 2 couches entièrement connectées.

Ces blocs sont ainsi constitués soit de couches de convolution et de *max pooling*, soit uniquement de couches de convolution. On retrouve ainsi 24 couches de convolution,

suivies de 2 couches entièrement connectées, comme visible sur la figure 3.

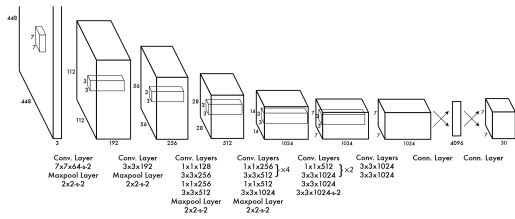


FIGURE 3 – Architecture du modèle YOLO par Redmon et al.[18]

3.3 Estimation du degré de gravité

Chaque défaut détecté par le modèle est associé à un niveau de gravité, qui permet de définir le degré critique de l'anomalie. Dans un cas industriel, le degré doit être représentatif de la conformité du produit, définie par un expert ou extraites des spécifications de contrôle.

Lors de l'inspection, l'opérateur renseigne les valeurs des propriétés mesurables associées (un diamètre pour un trou ou une marque, une largeur et une longueur pour une encoche, etc.), qui servent de base à la prédiction du degré de gravité.

Afin de réaliser cette prédiction, nous avons choisi d'utiliser la méthode XGBoost[19] (pour *eXtreme Gradient Boosting*), une implémentation open-source des modèles par boosting. Ces modèles font partie des approches par ensemble, utilisant les prédictions de plusieurs modèles pour réaliser la prédiction finale.

Le *boosting*, ou *gradient boosting* consiste à combiner plusieurs modèles dits faibles, généralement des arbres de décision, qui sont entraînés consécutivement. Cette technique permet de réduire l'erreur des modèles au fur et à mesure en fournissant la sortie et l'erreur. La figure 4 montre un exemple de représentation de cette méthode.

Ce modèle peut être entraîné sur des données numériques comme des données catégorielles, mais permet également de gérer des données manquantes. Enfin, contrairement aux réseaux de neurones, XGBoost est basé sur des arbres de décision, permettant de fonctionner de manière efficace sur des jeux de données qui ne sont pas volumineux.

3.4 Métriques

Pour mesurer la capacité de généralisation du modèle, on utilise principalement la mAP (*mean Average Precision*), correspondant à la surface sous la courbe précision - rappel pour chacune des classes selon un seuil d'IoU.

Cette métrique, couramment utilisée dans des problèmes de détection d'objets, permet de connaître la performance du modèle dans la réalisation des *bounding boxes* car plus la mAP est élevée, notamment avec un seuil d'IoU élevé (signifiant que la boîte doit être très précisément placée), et plus le placement des boîtes est juste.

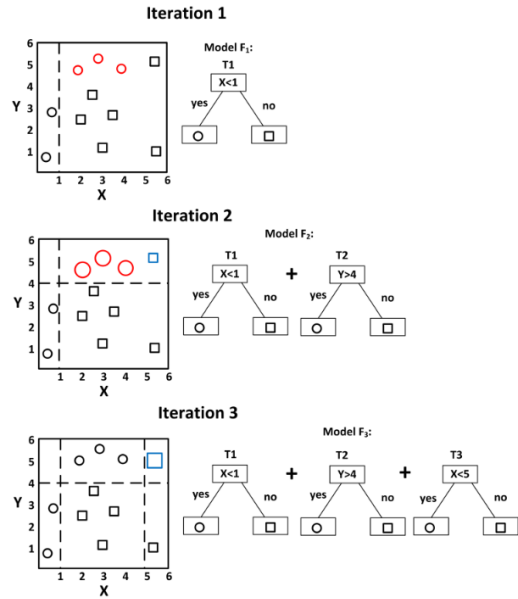


FIGURE 4 – Modélisation du processus de *gradient boosting* par Zhang et al.[20]

A des fins de comparaison, on utilise notamment la mAP avec un seuil d'IoU entre 0.5 et 0.95, notée $mAP@[0.5 : 0.95]$ et la mAP avec un seuil de 0.5 d'IoU, notée $mAP@[0.5]$.

4 Résultats

4.1 Détection de défauts

Le modèle a été entraîné durant 200 époques sur les images des différentes anomalies, à partir des poids d'un modèle pré-entraîné sur le jeu de données COCO[21].

La figure 5 montre les différentes courbes d'apprentissage. Les trois premières courbes correspondent à l'évolution des différentes fonctions de pertes (*loss*) du modèle.

La *box loss* est la courbe de *loss* pour la prédiction de la position du centre d'un objet. La *obj loss*, pour *objectivness loss*, indique l'évolution de la capacité du modèle à définir les bonnes régions d'intérêt pour le modèle : plus cette métrique est élevée, plus un objet a une probabilité élevée de se trouver dans les zones détectées par le modèle. Enfin, la *cls loss* (pour *classification loss*) représente l'évolution du *loss* dans la prédiction de la classe pour un objet donné.

Suite à l'entraînement, on obtient une *box loss* de 0.013, une *obj loss* de 0.004 et une *cls loss* de 0.0002.

On a également une précision de 0.99 et un rappel de 0.94. Enfin, au niveau des mAP, on constate une $mAP@[0.5]$ de 0.99 et une $mAP@[0.5 : 0.95]$ de 0.73.

Toutes les courbes descendent ou montent progressivement et convergent vers un extremum, montrant que le modèle s'est correctement entraîné.

La figure 6 présente la matrice de confusion des différentes classes suite à l'entraînement du modèle.

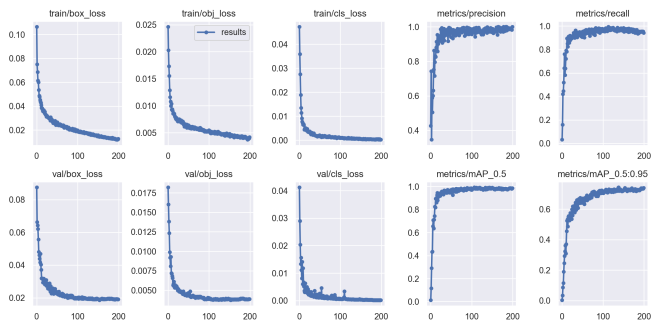


FIGURE 5 – Courbes d'apprentissage du modèle

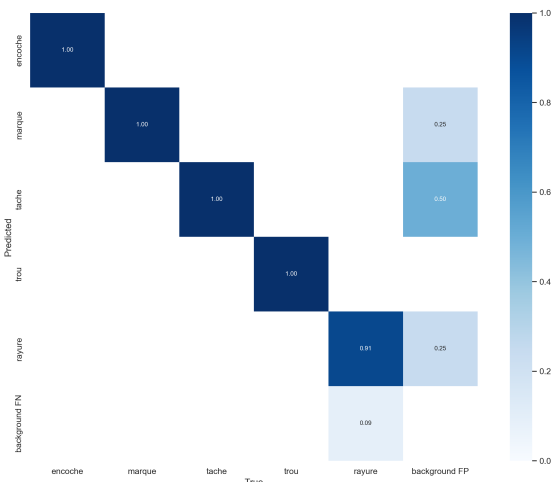


FIGURE 6 – Matrice de confusion de l'entraînement du modèle

4.2 Estimation du degré de gravité

En entraînant XGBoost sur les données des propriétés mesurables, on obtient une exactitude (*accuracy*) de près de 100%.

Outre sa capacité de classification, celui-ci permet d'afficher l'importance des différents paramètres dans les prédictions du modèle. On peut ainsi afficher l'importance des paramètres selon 3 paramètres différents :

- la couverture (*cover*), indiquant à quel point les paramètres sont présents dans les prédictions réalisées, correspondant au nombre relatif d'observations lié à un paramètre,
- le poids (*weight*), pour lequel plus cette valeur pour un paramètre est élevée, plus celui-ci influe dans la prédiction, calculé à partir du nombre de fois relatif que le paramètre apparaît dans les arbres du modèle.
- le gain calculé par la contribution relative du paramètre en prenant chaque contribution pour chaque arbre du modèle ; plus celui-ci est élevé, plus le paramètre est important pour la prédiction et apporte de l'information.

Les figures 7, 8 et 9 indiquent les différents scores d'im-

portance des paramètres dans les prédictions en fonction respectivement de la couverture, du poids et du gain apportés.

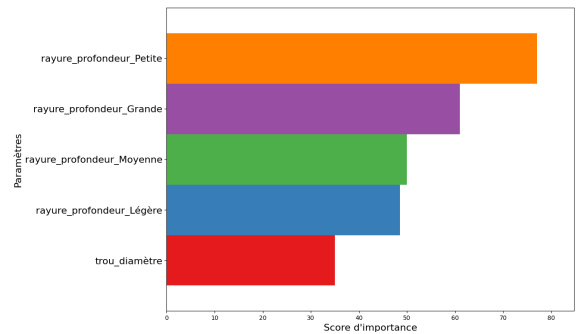


FIGURE 7 – Score d'importance en fonction de la couverture

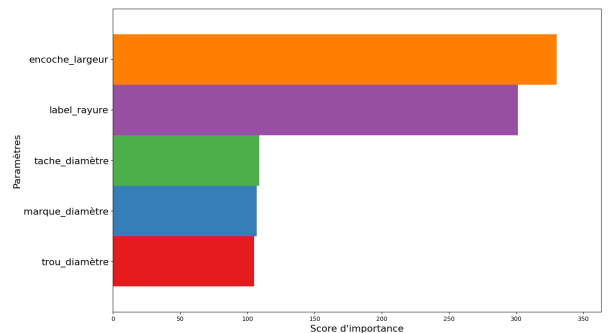


FIGURE 8 – Score d'importance en fonction du poids

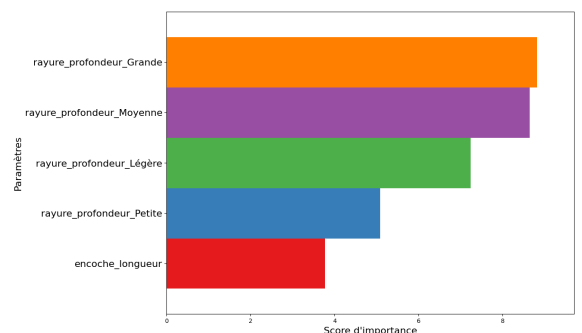


FIGURE 9 – Score d'importance en fonction du gain

4.3 Robustesse du modèle

Afin de mesurer la fiabilité de la plateforme, et notamment du module d'apprentissage automatique, 5 opérateurs ont procédé à l'inspection des mêmes pièces (figure 1) extraites du cas d'application.

L'objectif est de vérifier la fiabilité et la cohérence du modèle dans un cas d'utilisation pratique.

Ces mesures ont été réalisées dans des conditions réalistes mais maîtrisées d'inspection à l'aide d'une tablette, avec

une lumière naturelle et sur un bureau de travail blanc, sans complètement figer la position de la caméra, ajoutant de la flexibilité pour l'opérateur mais complexifiant le contrôle automatisé.

Les différentes pièces inspectées correspondent à :

- une pièce jaune avec un trou de 3mm (avec un degré de gravité de 1),
- une pièce noire avec une tâche de 4mm (avec un degré de gravité de 3),
- une pièce rouge avec une encoche de 1mm x 5mm (avec un degré de gravité de 1),
- une pièce blanche avec un trou de 10mm (avec un degré de gravité de 3),
- une pièce verte avec une marque de 12mm (avec un degré de gravité de 3),
- une pièce bleue avec une rayure de forme droite et de taille légère (avec un degré de gravité de 2).

Lors de l'inspection, l'anomalie sur la deuxième, troisième, cinquième et sixième pièces ont été correctement détectées et leur degré de gravité correctement estimé, ceci pour tous les opérateurs.

Pour la première pièce, le trou a été confondu avec une tâche et pour la quatrième pièce, le trou a été confondu avec une marque.

Sur les 24 anomalies analysées par les opérateurs, 22 ont été correctement prédites par le modèle, soit un taux de réussite de 91.67%. On peut ainsi considérer que la plateforme et le modèle sont suffisamment fiables en tenant compte des conditions d'essais maîtrisés, particulièrement dans le cas d'applications non critiques.

5 Discussion

5.1 Entraînement des modèles

On constate que l'entraînement nous donne un modèle efficace : on peut considérer qu'une $mAP@[0.5 : 0.95]$ de 73% est une valeur élevée, car à titre de comparaison, YOLOv5 entraîné sur les données COCO apporte une $mAP@[0.5 : 0.95]$ de 50.7%.

Le modèle réalise cependant des faux positifs et négatifs sur les rayures, même si les autres classes sont correctement détectées.

Ceci peut s'expliquer par le faible nombre d'images caractérisant cette anomalie comparé aux autres anomalies, car de telles images sont deux fois moins présentes dans le jeu de données : le problème pourrait donc être pallié en ajoutant des prises de vue de pièces avec des rayures.

Concernant les résultats obtenus par XGBoost (100% d'*accuracy*), ceux-ci peuvent s'expliquer par le fait que la relation entre les valeurs des propriétés mesurables et le niveau de gravité est très simple.

Ce constat se poursuit également en regardant les différentes valeurs de scores d'importance. Leur plage de valeurs n'est pas très élevée, notamment au niveau du gain, avec une plage entre 0 et 10. Étant donné la simplicité des données, peu d'information est apporté par les différents paramètres.

Au niveau de la couverture, les paramètres liés aux rayures sont les plus présents dans les prédictions, suivi des paramètres sur les anomalies à une seule propriété mesurable (trou, tâche et marque). Ceci peut s'expliquer par le fait que plus de paramètres définissent une rayure qu'une autre anomalie (on a une combinaison de 8 valeurs différentes), mais également car seules les rayures possèdent un degré de gravité allant jusqu'à 4, les autres allant uniquement jusqu'à 3.

5.2 Robustesse du modèle

Les deux anomalies qui ont été mal détectées, à savoir une tâche à la place d'un trou sur une pièce jaune et une marque à la place d'un trou sur une pièce blanche, peuvent s'expliquer par un manque de luminosité pour la prédiction.

Dans les deux cas, le manque de luminosité ne permet pas de distinguer efficacement les ombres et la profondeur, pouvant créer de la confusion pour le modèle.

En outre, les trois anomalies en question ne sont pas nécessairement parfaitement identifiables, même pour un opérateur humain qui effectuerait l'inspection des dites pièces.

Cependant, les degrés de gravité des anomalies ont été correctement prédits dans la totalité des cas, et ce même si pour une des pièces, la valeur de propriété mesurable renseignée n'était pas exactement la même.

6 Conclusion

L'approche développée dans cette étude montre comment des modèles d'apprentissage automatique peuvent aider des opérateurs à réaliser le contrôle de la qualité perçue de pièces industrielles.

En développant une approche combinant réseaux de neurones convolutifs basés sur des régions et arbres de décision, nous montrons comment il est possible de prédire et classer des anomalies de qualité perçue, et d'estimer leur degré de gravité, et ce dans un environnement où les conditions de contrôle sont moins maîtrisées, laissant une plus grande liberté à l'opérateur.

Nous montrons que le modèle est capable de détecter de manière fiable les différents défauts présents sur les pièces industrielles, et de prédire de manière consistante la gravité de ceux-ci dans le contexte du cas d'application considéré, qui représente un cas d'application typique du domaine industriel. Cela nous amène ainsi à penser que les méthodes seraient pertinentes pour d'autres cas d'application.

Cependant, celui-ci pourrait être amélioré en rééquilibrant le nombre de rayures par rapport aux autres anomalies et en ajoutant de la variabilité dans les valeurs de propriétés mesurables. De plus, afin de valider notre approche, celle-ci devrait être évaluée sur d'autres jeux de données de pièces industrielles.

De manière générale, il existe de nombreux cas industriels de contrôle de ce type, le cas d'application présenté permettrait d'observer les résultats d'inspection dans un contexte particulier.

Nos actuels et futurs travaux ont pour but de pouvoir anticiper, ou du moins identifier, lorsque des défaillances sont en train de survenir sur les machines produisant les pièces afin de prévenir leurs pannes avant qu'elles ne surviennent, et pouvoir ainsi déterminer leur état. Ainsi, on pourrait se servir de l'inspection des pièces pour faire le lien avec l'état des machines.

Références

- [1] B. Albert, C. Zanni-Merk, F. d. B. de Beuvron, M. Pillet, J.-L. Maire, C. Knecht, and J. Charrier, "A Smart System to Standardize the Specifications of Haptic Quality Control," *Procedia Computer Science*, vol. 112, pp. 723–730, Jan. 2017.
- [2] Y. Chen, Y. Ding, F. Zhao, E. Zhang, Z. Wu, and L. Shao, "Surface Defect Detection Methods for Industrial Products : A Review," *Applied Sciences*, vol. 11, p. 7657, Jan. 2021. Number : 16 Publisher : Multidisciplinary Digital Publishing Institute.
- [3] Q. Luo, X. Fang, J. Su, J. Zhou, B. Zhou, C. Yang, L. Liu, W. Gui, and L. Tian, "Automated Visual Defect Classification for Flat Steel Surface : A Survey," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, pp. 9329–9349, Dec. 2020. Conference Name : IEEE Transactions on Instrumentation and Measurement.
- [4] J. Balzategui, L. Eciolaza, N. Arana-Arexolaleiba, J. Altube, J.-P. Aguerre, I. Legarda-Ereño, and A. Apraiz, "Semi-automatic quality inspection of solar cell based on Convolutional Neural Networks," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 529–535, Sept. 2019. ISSN : 1946-0759.
- [5] Y. Yang, L. Pan, J. Ma, R. Yang, Y. Zhu, Y. Yang, and L. Zhang, "A High-Performance Deep Learning Algorithm for the Automated Optical Inspection of Laser Welding," *Applied Sciences*, vol. 10, p. 933, Jan. 2020. Number : 3 Publisher : Multidisciplinary Digital Publishing Institute.
- [6] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Apr. 2015. arXiv :1409.1556 [cs].
- [7] Y. Kardovskyi and S. Moon, "Artificial intelligence quality inspection of steel bars installation by integrating mask R-CNN and stereo vision," *Automation in Construction*, vol. 130, p. 103850, Oct. 2021.
- [8] X. Gong, H. Su, D. Xu, J. Zhang, L. Zhang, and Z. Zhang, "Visual Defect Inspection for Deep Aperture Components With Coarse-to-Fine Contour Extraction," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, pp. 3262–3274, June 2020. Conference Name : IEEE Transactions on Instrumentation and Measurement.
- [9] I. Kuric, M. Kandra, J. Klarák, V. Ivanov, and D. Więcek, "Visual Product Inspection Based on Deep Learning Methods," in *Advanced Manufacturing Processes* (V. Tonkonogyi, V. Ivanov, J. Trojanowska, G. Oborskyi, M. Edl, I. Kuric, I. Pavlenko, and P. Dasic, eds.), Lecture Notes in Mechanical Engineering, (Cham), pp. 148–156, Springer International Publishing, 2020.
- [10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," Oct. 2014. arXiv :1311.2524 [cs].
- [11] M. J. Shafiee, M. Famouri, G. Bathla, F. Li, and A. Wong, "TinyDefectNet : Highly Compact Deep Neural Network Architecture for High-Throughput Manufacturing Visual Quality Inspection," Nov. 2021. arXiv :2111.14319 [cs].
- [12] Z. Liu, R. Tang, G. Duan, and J. Tan, "TruingDet : Towards high-quality visual automatic defect inspection for mental surface," *Optics and Lasers in Engineering*, vol. 138, p. 106423, Mar. 2021. ADS Bibcode : 2021OptLE.13806423L.
- [13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems*, vol. 28, Curran Associates, Inc., 2015.
- [14] Y. He, K. Song, Q. Meng, and Y. Yan, "An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, pp. 1493–1504, Apr. 2020. Conference Name : IEEE Transactions on Instrumentation and Measurement.
- [15] P. Napoletano, F. Piccoli, and R. Schettini, "Semi-supervised anomaly detection for visual quality inspection," *Expert Systems with Applications*, vol. 183, p. 115275, Nov. 2021.
- [16] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, imyhxy, Lorna, Z. Yifu, C. Wong, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, tkianai, yxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, "ultralytics/yolov5 : v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," Nov. 2022.
- [17] R. Girshick, "Fast R-CNN," Sept. 2015. arXiv :1504.08083 [cs].
- [18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once : Unified, real-time object detection," 2015.
- [19] T. Chen and C. Guestrin, "Xgboost : A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, (New York, NY, USA), p. 785–794, Association for Computing Machinery, 2016.

- [20] Z. Zhang, G. Mayer, Y. Dauvilliers, G. Plazzi, F. Pizza, R. Fronczek, J. Santamaria, M. Partinen, S. Overeem, R. Peraita-Adrados, A. M. D. Silva, K. Sonka, R. D. Rio-Villegas, R. Heinzer, A. Wierzbicka, P. Young, B. Högl, C. L. Bassetti, M. Manconi, E. Feketeova, J. Mathis, T. Paiva, F. Canellas, M. Lécendreau, C. R. Baumann, L. Barateau, C. Pesenti, E. Antelmi, C. Gaig, A. Iranzo, L. Lillo-Triguero, P. Medrano-Martínez, J. Haba-Rubio, C. Gorban, G. Luca, G. J. Lammers, and R. Khatami, “Exploring the clinical features of narcolepsy type 1 versus narcolepsy type 2 from European Narcolepsy Network database with machine learning,” *Scientific Reports*, vol. 8, p. 10628, July 2018.
- [21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO : Common Objects in Context,” in *Computer Vision – ECCV 2014* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), Lecture Notes in Computer Science, (Cham), pp. 740–755, Springer International Publishing, 2014.