

SOFTWARE COMPUTER VISION APPLICATION FOR THE LOCALISATION OF A PART

BÎRLOGEANU Vlad-Cristian

Facultatea: Inginerie Industrială și Robotică, Specializarea: Informatică Aplicată în Inginerie Industrială,
Anul de studii: IV, e-mail: vladbirlogeanu@yahoo.com

Conducător științific: Șl.dr.ing. **Ioan Cristian TARBĂ**

ABSTRACT: This software application is designed to facilitate the orientation of a part in the workspace of a CNC machine by means of Computer Vision. This problem is twofold as the part must first be validated and then reorientated in the CNC workspace. The validation section of the application is done by using the YOLOv5(You Only Look Once) Algorithm to check the part for possible defects on the metal surface (crazing, inclusion, patching, pitting, rolling, scratching). The localization section is achieved by using the OpenCV library to determinate the distance between the part and the target location.

KEY WORDS: Computer Vision, OpenCV, YOLOv5, Object-detection.

1. Introducere

Scopul acestei aplicații este de a valida un reper ce urmează a fi prelucrat și de a oferi instrucțiuni pentru re poziționarea acestuia într-o poziție precizată de către utilizator. Această primă parte a aplicației se realizează folosind algoritmul YOLOv5 care caută posibilele defecte ale reperului pe un semnal video primit de la o cameră orientată perpendicular pe masa de lucru. În cazul în care algoritmul găsește defecte pe suprafața metalică, acestea sunt evidențiate utilizatorului cu ajutorul unor căsuțe de delimitare aplicate pe semnalul video în timp real și de un scor de încredere al algoritmului în identificarea defectului corespunzător. Odată realizată etapa de validare a reperului, cu ajutorul unui algoritm de detectare a muchiilor, se va calcula distanța dintre reperul de prelucrat și un ghidaj folosit de utilizator pentru a orienta reperul pe masa de lucru în poziția dorită. În urma calculului, se vor afișa utilizatorului valorile pe axa X, respectiv Y cu care acesta trebuie să deplaseze reperul pentru a atinge poziția dorită.

2. Stadiul actual

În stadiul actual al aplicației avem un prototip al secțiunii de validare a reperului folosind algoritmul YOLOv5. În continuare va fi prezentată funcționarea algoritmului YOLOv5, respectiv rezultatele obținute folosind acest algoritm în aplicația în cauză.

În rezolvarea problemelor de detectare de obiecte, algoritmi folosiți până în prezent folosesc clasificatori cu care parcurg o imagine la intervale prestabilite și varii dimensiuni. Din acest punct de vedere, algoritmul YOLOv5 redimensionează imaginile pentru a obține un raport pătrat, apoi aplică o rețea convoluțională pe întreaga imagine simultan, nu gradual (astfel se obține viteza care permite aplicarea acestui algoritm pe semnale video ce furnizează până la 140 cadre pe secundă). Odată realizată detectarea obiectelor, se aplică non-max suppression pentru a elimina multiplele căsuțe de delimitare ce desemnează același obiect.

2.1. Funcționarea algoritmului

În continuare se va detalia funcționarea algoritmului YOLOv5. Odată redimensionată imaginea, aceasta se divizează într-un tablou de $S \times S$ celule, în fiecare celulă urmându-se a se aplica B casuțe de delimitare. Centrul unui obiect determină celula care se va ocupa de detectarea acestuia. Fiecare casuță de

delimitare conține 5 elemente: (x,y) – coordonatele față de marginile celulei; (w,h) – coordonatele față de marginile imaginii; un factor de încredere în clasificarea obiectului. Fiecare celulă prezice probabilități pentru apartenența unui obiect la fiecare clasă din cele C clase prezente în model, apoi alege clasa cu probabilitatea cea mai mare. Dimensiunea tensorului de predicții este după cum urmează: $S \times S \times (B*5 + C)$.

În Figura 1 se prezintă aplicarea algoritmului YOLOv5 pe o imagine.

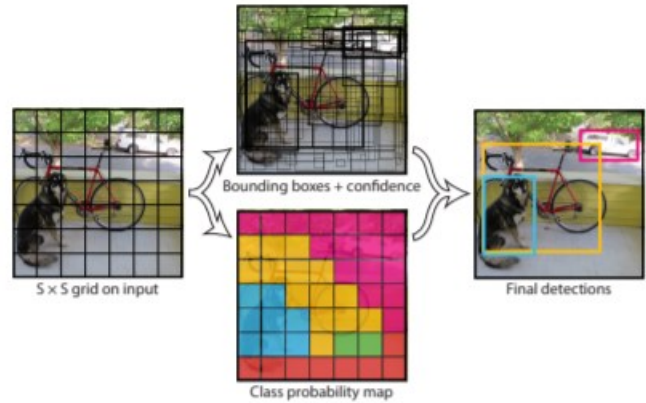


Fig.1. Aplicarea algoritmului YOLOv5 pe o imagine [1]

2.2. Arhitectura rețelei convoluționale

În Figura 2 se prezintă arhitectura rețelei YOLOv5. Aceasta este formată din 24 de straturi convoluționale urmate de 2 straturi interconectate care realizează precizarea claselor, respectiv a locațiilor casetelor de delimitare. Cazul ilustrat în figura 2 este extras din lucrarea științifică a autorilor algoritmului, în demonstrația prezentă folosindu-se un tablou de celule de 7 x 7 și un număr de 20 de clase.

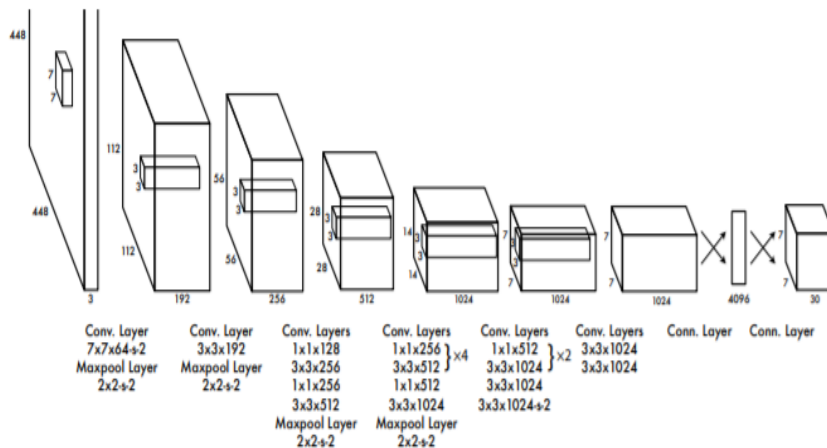


Fig.2. Arhitectura rețelei convoluționale din spatele algoritmului YOLOv5 [1]

2.3. Loss Function

Loss Function-ul reprezintă funcția ce îi oferă modelului posibilitatea de a se verifica pe parcursul procesului de antrenare. Aceasta calculează diferența dintre precizarea modelului și valoarea reală a resurselor folosite la antrenare ținând cont de mulți factori. Ecuația (1) folosită la calcularea Loss Function-ului este prezentată mai jos.

$$\begin{aligned}
 \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 \\
 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] + \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{obj} [(C_i - \hat{C}_i)^2 \\
 + \lambda_{nonj} \sum_{i=0}^{S^2} \sum_{j=0}^B 1_{ij}^{noobj} [(C_i - \hat{C}_i)^2 + \sum_{c \in classes} 1_i^{obj} \sum (p_i(c) - \hat{p}_i(c))^2]
 \end{aligned} \tag{1}$$

2.4. Limitările algoritmului

Întrucât numărul de căsuțe de delimitare per celulă este limitat, se induce limitarea de detectare a grupurilor de obiecte, în special cele de mici dimensiuni [1].

2.5. Utilizarea algoritmului YOLOv5 în aplicație

Pentru a pregăti rețeaua convoluțională de folosire în aplicația noastră, aceasta a trebuit antrenată pentru a detecta defecte ce pot apărea pe suprafața metalică a reperului nostru. Pentru a o antrena am folosit un set de date realizat de Northeastern University (NEU) [2] compus din 1800 de poze în format grayscale cu 6 defecte: fisurări, zgârieturi, defecte la laminare, pitting, incluziuni, petice. Acestea au fost nevoite să le aplicăm etichete folosind software-ul Labelimg [3] pentru ca algoritmul să determine locația obiectelor în imaginile folosite la antrenament. Odată finalizată antrenarea rețelei se poate observa în Figura 3 și Figura 4 un prim test al prototipului.

Din punct de vedere al cercetării, un set de date mai voluminos și variat ar crește atât precizia cât și gama de defecte acoperite de această aplicație, putând furniza astfel o soluție tangibilă și competitivă pentru problemele de detectare de defecte pe suprafețe metalice.

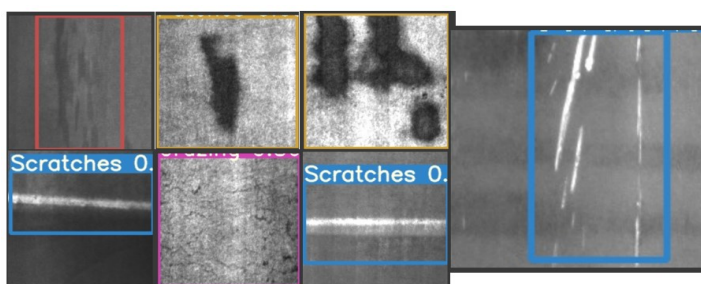


Fig.3. Imagini prelucrate folosind algoritmul YOLOv5

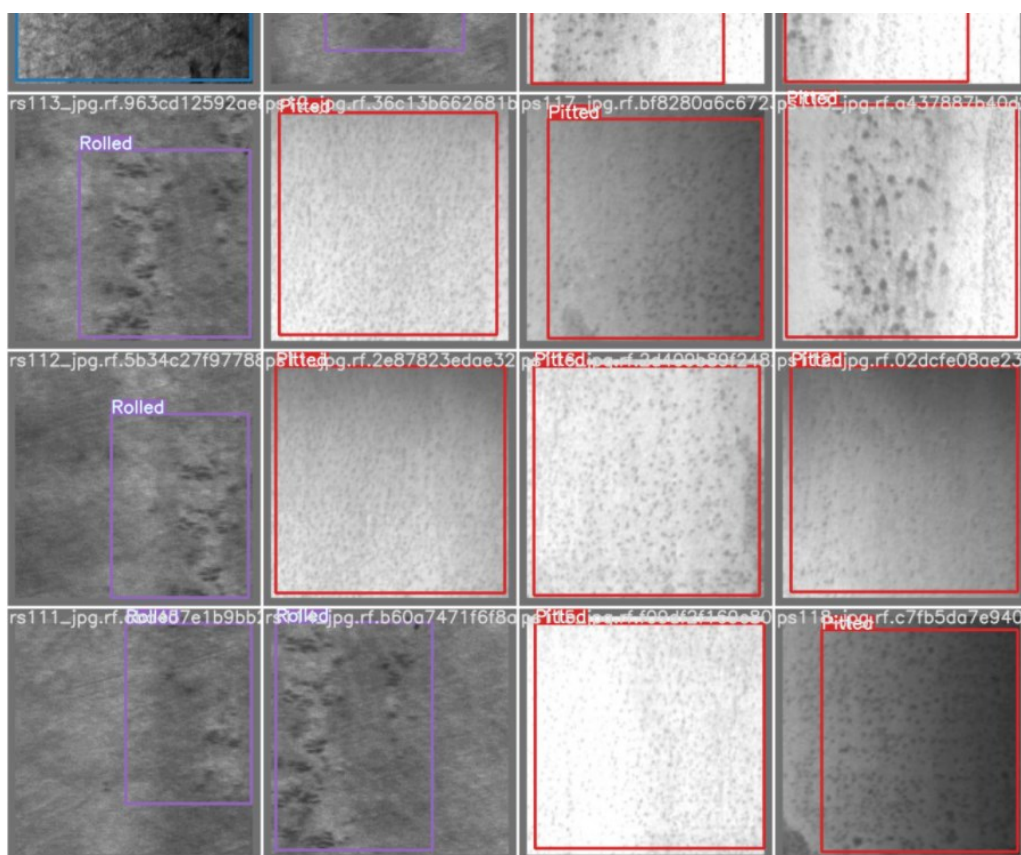


Fig.4. Imagini prelucrate folosind algoritmul YOLOv5

2.6. Viitoare îmbunătățiri ale aplicației

În continuare urmează testarea prototipului pe repere în mediul de lucru și aducerea de îmbunătățiri acestuia pentru a asigura fiabilitate în condițiile de lucru. De asemenea urmează să realizăm secțiunea de localizare a reperului și de afișare a instrucțiunilor pentru re poziționarea acestuia în poziția specificată de utilizator. Pentru aceasta vom dezvolta un algoritm care va folosi dimensiunile unui obiect de referință, obiectul pe care utilizatorul îl va amplasa în spațiul de lucru pentru a alege poziția în care va fi relocat reperul, respectiv distanța dintre cameră și spațiul de lucru pentru a determina distanța pe axele X și Y a reperului față de obiectul de referință care reprezintă destinația acestuia. Odată calculate acestea vor fi afișate utilizatorului.

3. Concluzii

Folosirea algoritmului YOLOv5 conferă viteză aplicațiilor ce îl înglobează, astfel încât acesta poate fi aplicat pe un număr de până la 140 de cadre pe secundă, deci poate rula pe un semnal video transmis în timp real. Integrarea acestuia în cadrul unei aplicații de detectare a defectelor unor suprafețe prezintă un punct de pornire în direcția realizării unui sistem de procesare de imagini ce poate fi integrat în procesul de control al calității și ce poate reduce din costurile acestuia și crește viteza cu care acesta se desfășoară.

De asemenea, un sistem de localizare și re poziționare bazat pe procesarea de imagini are potențialul de a crește ritmul proceselor de producție și de a înainta eforturile de automatizare.

4. Bibliografie

- [1]. Redmon, J, Divvala, S, Girshick R, Farhadi A (2015). “You Only Look Once: Unified, Real-Time Object Detection”, University of Washington, Allen Institute for AI
- [2]. Song, K, Yan, Y. “NEU surface defect database”, Northeastern University, http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html
- [3]. Tzutalin. Labellmg. Git Code (2015). <https://github.com/tzutalin/labelImg>

5. Notății

Următoarele simboluri sunt utilizate în cadrul lucrării:

S - dimensiunea tabloului de celule în care este împărțită imaginea

B - numărul de căsuțe de delimitare folosite în fiecare celulă

I_i^{obj} – are valoarea 1 dacă există un obiect în celula I și 0 în restul cazurilor

I_{ij}^{obj} – are valoarea 1 dacă există un obiect în celula I și predictorul j are cel mai mare nivel de încredere

I_{ij}^{noobj} – reversul lui I_{ij}^{obj}

C – încrederea modelului în clasificarea unui obiect

x, y – coordonatele prezise de model ale centrului obiectului față de marginile celulei

\hat{X}, \hat{Y} – coordonatele centrului obiectului față de marginile celulei citite de model din setul de antrenament

w, h – coordonatele prezise de model ale centrului obiectului față de marginile imaginii

\hat{w}, \hat{h} - coordonatele centrului obiectului față de marginile imaginii citite de model din setul de antrenament