Universitat Autònoma de Barcelona

Dipòsit digital de documents de la UAB

This is the **published version** of the bachelor thesis:

Matarín Ladero, Marc; Valveny Llobet, Ernest, dir. Aplicació i millora de model OCR per a reconeixement de codis de barres en imatges de pots cilíndrics. 2023. (Enginyeria de Dades)

This version is available at https://ddd.uab.cat/record/281555

under the terms of the BY-NC-ND license

Aplicació i millora de model OCR per a reconeixement de codis de barres en imatges de pots cilíndrics

Marc Matarín i Ladero

Resum— El treball consisteix a aplicar i millorar un model *OCR* per detectar etiquetes en codis de barres de productes cilíndrics, així com reconèixer i verificar els números corresponents. El reconeixement presenta dificultats a causa de la baixa resolució a la regió de l'etiqueta, distorsions d'il·luminació, deformacions ocasionades per la forma cilíndrica del producte i la possibilitat que les barres del codi interfereixin amb el text, sent detectades com a caràcters pel model *OCR* . L'objectiu principal és millorar l'eficiència i la precisió d'un model genèric existent. Aquest projecte es realitza en col·laboració amb l'empresa *Rosepetal SL* i és fonamental per al procés de detecció d'anomalies en els productes, com ara esgarrapades, taques o codis de barres incorrectes. Per dur a terme aquesta tasca, es compta amb les facilitats proporcionades per l'empresa, incloent-hi un entorn equipat amb una *GPU* d'alt rendiment per fer entrenaments i prediccions de xarxes neuronals, conjunts de dades específiques per fer proves i el suport i assessorament de l'equip de treball de *Rosepetal SL*. En resum, l'objectiu és millorar el reconeixement d'etiquetes a codis de barres de productes cilíndrics, abordant desafiaments com la baixa resolució, distorsions d'il·luminació i deformacions, en col·laboració amb *Rosepetal SL* per detectar possibles anomalies als productes.

Paraules clau— OCR, model, Entrenament, Avaluació, Detecció, Reconeixement, Segmentació, codi de barres, precisió, sensibilitat

Abstract— The work consists of applying and improving an *OCR* model to detect labels on cylindrical product barcodes, as well as recognizing and verifying the corresponding numbers. The recognition presents difficulties due to low resolution in the label region, illumination distortions, deformations caused by the cylindrical shape of the product and the possibility that the bars of the code interfere with the text, being detected as characters by the *OCR* model. The main objective is to improve the efficiency and accuracy of an existing generic model. This project is conducted in collaboration with the company *Rosepetal SL* and is essential for the process of detecting anomalies in products, such as scratches, stains, or incorrect barcodes. To conduct this work, the facilities provided by the company are available, including an environment equipped with a high-performance GPU for training and prediction of neural networks, specific data sets for testing and the support and advice of the *Rosepetal SL* team. In summary, the goal is to improve label recognition on cylindrical product barcodes, addressing challenges such as low resolution, illumination distortions and deformations, in collaboration with *Rosepetal SL* to detect potential product anomalies.

Index Terms— OCR, model, Training, Evaluation, Detection, Recognition, Segmentation, Scope Code, Precision, Recall

1 Introducció

CADA vegada més, s'utilitza la Intel·ligència Artificial per a tasques que fins ara han sigut realitzades directament per l'ésser humà. Un bon camp on utilitzar la *IA* és en detecció d'anomalies en un control de qualitat. En fàbriques de gran producció es necessita un mètode ràpid i precís per la gran quantitat de productes que es fabriquen. Cada producte serà detectat com a *OK* o *NOK* però pot ser per a diferents causes/anomalies: taques, bonys, mala disposició d'etiquetes/logotips, o en el cas d'aquest treball, que el codi no correspongui al producte. L'empresa *Rosepetal SL* [1] es dedica a implementar una interfícies per a empreses que els contracten per a obtenir aquest control de qualitat basat en intel·ligència artificial. En les cintes de producció els productes passen per un sector de quatre càmeres que fan una fotografia cada una des de les perspectives: davant, darrera, esquerra i dreta. Aquestes quatre fotografies es concatenen en una. Aquesta imatge serà la utilitzada per a l'entrenament i la detecció d'anomalies.

Fig. 1: Quatre fotos d'un esprai concatenades en una. *Rosepetal SL*

- *E-mail de contacte: Marc.Matarin@autonoma.cat*
- *Treball tutoritzat per: Ernest Valveny (Ciències de la Computació)*
- *Curs 2022/23*

- Juliol de 2023, Escola d'Enginyeria (UAB)

L'objectiu principal és detectar i reconèixer el text a les etiquetes, i per aconseguir-ho es considera la utilització de models pre-entrenats com *PaddleOCR* [2], *Tesseract* [3] i *MMOCR* [4]. No obstant això, a causa dels desafiaments de resolució, distorsió i deformació presents a les imatges de les etiquetes, cal fer un entrenament precís i aplicar diverses tècniques per millorar l'execució d'aquests sistemes ja existents. La idea és reentrenar els models existents per augmentar la seva eficiència i aconseguir un millor rendiment en la detecció i reconeixement de text a les etiquetes, superant les dificultats esmentades.

2 Estat de l'Art

El reconeixement òptic de caràcters (*OCR*) ha experimentat avenços significatius en les darreres dècades, impulsat pel desenvolupament de tècniques de visió per computador i aprenentatge automàtic. Dins l'ampli estat de l'art, ens centrem en les eines i arquitectures que l'empresa vol analitzar en aquest projecte. Aquestes arquitectures són: *PaddleOCR*, *MMOCR* i *Tesseract*, que han demostrat excel·lents capacitats de reconeixement de text en diverses aplicacions.

PaddleOCR és una biblioteca de codi obert *OCR* desenvolupada per *PaddlePaddle*, una plataforma d'aprenentatge profund. Utilitza xarxes neuronals convolucionals i recurrents per fer detecció i reconeixement de text en imatges. *PaddleOCR* ha destacat per la seva capacitat per reconèixer text en imatges amb diferents mides, orientacions i estils de font. A més, ofereix suport multilingüe i és molt eficient en termes de velocitat d'inferència.

Fig. 2: Exemple de predicció ***PaddleOCR*** (*PP-OCRv3 Chinese model*)

MMOCR, per altra banda, és una biblioteca *OCR* de codi obert basada en *PyTorch*[5] . Ofereix una àmplia gamma de funcionalitats, incloent detecció de text, reconeixement de text, segmentació de text i més. *MMOCR* s'ha destacat per la seva capacitat per abordar desafiaments específics a *OCR*, com la detecció i reconeixement de text en imatges amb diferents angles d'inclinació, distorsions i soroll. També proporciona una interfície fàcil de fer servir i una arquitectura modular que facilita la personalització i extensibilitat del sistema.

Tesseract és una de les biblioteques *OCR* més populars i àmpliament utilitzades. Desenvolupada per *Google*, ha estat utilitzada en nombrosos projectes i aplicacions comercials. *Tesseract* és coneguda per la precisió i capacitat per reconèixer text en una àmplia varietat d'idiomes i fonts. Utilitza un enfocament basat en el reconeixement de patrons i tècniques d'aprenentatge automàtic per fer el reconeixement de text. A més, *Tesseract* ofereix funcionalitats addicionals, com ara la segmentació de text i la capacitat de corregir errors ortogràfics.

Aquestes biblioteques de *OCR* estan recolzades per tècniques d'aprenentatge automàtic i visió per computador avançades. Els models de xarxes neuronals utilitzats en aquestes biblioteques s'entrenen en grans conjunts de dades etiquetades per capturar patrons i característiques rellevants del text. A més, s'apliquen tècniques de processament d'imatges, com ara la normalització de contrast, l'eliminació de soroll i la correcció de distorsions, per millorar la qualitat de les imatges i facilitar el reconeixement del text.

L'estat de l'art de *l'OCR* també inclou enfocaments basats en la combinació de múltiples models i tècniques. Per exemple, es poden utilitzar xarxes neuronals convolucionals per a la detecció de text i xarxes neuronals recurrents per al reconeixement de text, combinant així les fortaleses de tots dos enfocaments. A més, s'estan explorant enfocaments de transferència d'aprenentatge i aprenentatge per reforçament (*Reinforcement Learning*) per millorar encara més el rendiment de *l'OCR*.

A més, s'estan aplicant enfocaments *d'OCR* específics per a dominis i escenaris particulars. Per exemple, a *l'OCR* de documents mèdics, s'utilitzen tècniques especialitzades per reconèixer terminologia mèdica i estructures de dades específiques. A *l'OCR* de caràcters escrits a mà, s'utilitzen models d'aprenentatge profund entrenats en conjunts de dades d'escriptura a mà per aconseguir més precisió en el reconeixement.

Un altre aspecte important de l'estat de l'art de *l'OCR* és la integració amb tecnologies relacionades, com ara la traducció automàtica i l'extracció d'informació. En combinar l'OCR amb tècniques de processament del llenguatge natural, es pot fer una comprensió més profunda del contingut dels documents i fer tasques addicionals, com l'extracció d'entitats o la traducció de text.

Tenint en compte l'estat de l'art de *l'OCR*, s'evidencia la importància de millorar el rendiment i la precisió dels sistemes *d'OCR* aplicats a imatges amb dificultats específiques, especialment en el context de la fabricació de productes. Les imatges de productes poden presentar desafiaments com a baixa resolució, distorsions d'il·luminació, deformacions degut a la forma cilíndrica dels objectes i la possible obstrucció del text per les barres del codi de barres.

3 Objectius Inicials

En el camp de la visió per computadora i el processament d'imatges, un dels desafiaments més comuns és la detecció i reconeixement d'objectes i text en imatges o vídeos. En aquest context, el present treball té com a objectiu principal la generació d'un flux de *Node-RED* [6], una plataforma de programació visual, utilitzada per l'empresa, per a dur a terme un procés complet que abasti des de la detecció d'etiquetes fins a la verificació del text reconegut. Tot això ha de ser funcional i amb una precisió òptima a l'hora de detectar i reconèixer. A continuació, es descriuran els passos principals d'aquest procés.

3.1 Generació de flux de *Node-RED*:

Node-RED és una eina que permet la creació de fluxos de treball mitjançant la connexió de nodes predefinits. En aquest cas, s'utilitzarà *Node-RED* per a dissenyar un flux que permeti el processament d'imatges i l'extracció d'informació rellevant.

3.2 Detecció de text

El primer pas del flux consisteix en la detecció d'etiquetes en una imatge. Aquesta etapa utilitza algorismes de visió per computadora per a identificar i localitzar les etiquetes presents en la imatge. Aquesta detecció pot basar-se en tècniques com a detecció de vores, detecció de característiques o algorismes d'aprenentatge automàtic.

3.3 Segmentació de ROI

Una vegada que s'han detectat les etiquetes, el següent pas és la segmentació de les regions d'interès (*ROI*, per les seves sigles en anglès). Això implica aïllar les àrees de la imatge que contenen les etiquetes detectades, amb l'objectiu de reduir el soroll i millorar la precisió del procés de reconeixement de text.

3.4 Detecció de text en la *ROI*

En aquesta etapa, s'aplica un algorisme de detecció de text específic a les *ROI* identificades anteriorment. Aquest algorisme és capaç d'identificar les regions de la imatge que contenen text i delimitar-les per al seu posterior processament.

3.5 Reconeixement del text

Una vegada que s'han detectat les regions de la imatge que contenen text, s'utilitza un algorisme de *OCR* per a convertir les imatges de text en text comprensible per la màquina. Aquest procés implica la identificació dels caràcters individuals i la seva posterior agrupació per a formar paraules i frases.

3.6 Verificació del text reconegut

Finalment, es duu a terme la verificació del text reconegut mitjançant tècniques de comparació i validació. Això implica verificar si el text reconegut coincideix amb uns certs criteris predefinits, com una llista de codis de barres vàlids. Es poden utilitzar algorismes de comparació de cadenes de text i tècniques d'aprenentatge automàtic per a dur a terme aquesta verificació.

4 Metodologia

4.1 Metodologia de l'Empresa

En l'empresa *Rosepetal SL* s'ha complert un conveni, juntament amb la universitat. Aquest ha constat d'un horari de cinc hores diàries, amb una reunió de seguiment , també diària, de tipus *SCRUM*. En aquesta reunió diària s'exposa el treball fet durant el mateix dia, les dificultats i inconvenients i la tasca que es duria a terme el dia següent. La feina ha estat organitzada en *Sprints* de dues setmanes on es fa una revisió de tota la matèria feta durant els 14 dies anteriors i es proposen noves tasques per l'*Sprint* següent.

4.2 Metodologia en el projecte

Per a dur a terme el meu treball i que sigui útil per l'empresa, s'ha de fer amb un cert mètode. Aquest projecte, a part de fer un model de detecció de l'etiqueta i posteriorment la detecció i reconeixement del text, ha de poder ser implementat i executat en la seva interfície.

Aquesta interfície utilitza l'eina de *Node-RED* per a connectar les càmeres amb tot el procés de validesa de cada producte. Per a les execucions, s'utilitza un servidor amb una targeta gràfica d'última generació per a fer execucions de manera òptima.

5 Planificació

En el present projecte s'ha establert una planificació per a dur a terme durant l'estada a l'empresa com es mostra en la següent taula.

TASCA	Hores	Inici	Final
Provar models per defecte	25	1	2
Crear imatges sintètiques	25	2	3
Entrenar models OCR	80	3	6
Provar models entrenats	10	6	7
Servidors gRPC	55	7	9
Verificació	15	9	9
Flux de Node-RED	50	9	11

Fig. 3: Planificació per setmanes y hores dedicades en cada tasca

5.1 Provar models OCR per defecte

El primer pas consisteix a provar diversos models OCR per defecte, com *PaddleOCR, MMOCR i Tesseract*, sobre etiquetes reals. Això permetrà avaluar el seu rendiment en la

detecció i reconeixement de text en imatges d'etiquetes existents. S'analitzaran mètriques com la precisió, el temps de processament i la capacitat per a reconèixer diferents tipus d'etiquetes.

5.2 Generar imatges sintètiques d'etiquetes de codis de barres

Amb la finalitat d'ampliar i diversificar el conjunt de dades, es generaran imatges sintètiques d'etiquetes de codis de barres. Aquestes imatges s'utilitzaran tant per a l'entrenament dels models *OCR* com per a les proves posteriors. Es generaran etiquetes amb diferents fonts, grandàries, orientacions i nivells de soroll per a simular diferents escenaris.

5.3 Entrenar models OCR

A continuació, es procedirà a l'entrenament dels models *OCR* prèviament seleccionats utilitzant el conjunt de dades compost per etiquetes reals i sintètiques. Es realitzaran ajustos en els hiperparàmetres i es realitzaran múltiples iteracions d'entrenament per a optimitzar el rendiment dels models. S'avaluaran les mètriques d'acompliment i se seleccionaran els models que obtinguin els millors resultats.

5.4 Provar nous models entrenats

Una vegada entrenats els models, es realitzaran proves exhaustives utilitzant imatges d'etiquetes reals i sintètiques. Es mesurarà la precisió, la velocitat de processament i la capacitat per a reconèixer diferents tipus de text en les etiquetes. Es compararan els resultats obtinguts amb els models per defecte per a determinar la millora reeixida mitjançant l'entrenament personalitzat.

Amb base en els resultats obtinguts en les proves, se seleccionaran els models de detecció i reconeixement que millor s'exerceixin en termes de precisió i velocitat. Aquests models seran utilitzats en les etapes posteriors del procés.

5.5 Creació de servidors gRPC

Es procedirà a la creació de servidors *gRPC* [7] per a la detecció d'etiquetes i el reconeixement de text. Aquests servidors permetran rebre sol·licituds d'imatges i proporcionar respostes amb la detecció i el text reconegut.

5.6 Prova de mètodes de verificació amb *Zbar*

En aquesta etapa, es provaran diferents mètodes de verificació utilitzant la biblioteca *Zbar* [8]. Aquesta és una eina que permet la lectura de codis de barres i això permetrà la verificació del text reconegut. S'exploraran diferents algorismes i tècniques per a verificar la informació extreta dels codis de barres en les etiquetes.

5.7 Creació del flux de *Node-RED*

Finalment, es crearà un flux de *Node-RED* que cridi als servidors *gRPC* per a la detecció d'etiquetes i el reconeixement de text, així com per a realitzar la verificació utilitzant *Zbar*. Aquest flux permetrà una integració completa dels diferents components del procés i automatitzarà l'execució de les tasques. El flux de *Node-RED* es dissenyarà de manera visual, connectant els nodes corresponents a cada pas del procés, des de la recepció de la imatge fins a la verificació final del text reconegut.

És important tenir en compte que durant tot el procés es realitzaran proves i validacions per a garantir la qualitat dels resultats. Es podran ajustar paràmetres, incorporar nous models o realitzar millores addicionals segons sigui necessari.

6 Eines utilitzades

6.1 Barcode4js

Barcode4js és una biblioteca *JavaScript* que permet generar i descodificar codis de barres en aplicacions web. Proporciona funcions per a la creació d'imatges de codis de barres de diferents tipus, com a codis de barres lineals o codis *QR*, i també ofereix la capacitat de descodificar la informació continguda en els codis de barres. És una eina útil per a aplicacions que requereixen la generació o lectura de codis de barres venda.

6.2 *gRPC*

gRPC és un marc de treball de comunicació de codi obert desenvolupat per *Google*. Proporciona un mecanisme eficient i d'alta velocitat per a la comunicació entre diferents sistemes mitjançant l'ús de crides de procediment remot (*RPC*). *gRPC* permet definir serveis i missatges utilitzant el llenguatge de definició d'interfície (*IDL*). És àmpliament utilitzat en arquitectures de *microserveis* i aplicacions distribuïdes per a facilitar la comunicació entre components i sistemes.

6.3 ZBAR

ZBAR és una biblioteca de codi obert que proporciona capacitats de lectura i descodificació de codis de barres. És compatible amb diversos tipus de codis de barres, com a codis de barres lineals, codis *QR*, etc. *ZBAR* permet la detecció i descodificació de codis de barres en diferents formats d'imatges. És una eina útil per a la verificació i validació de codis de barres en aplicacions com a control d'inventari, logística i processament de pagaments.

6.4 *Node-RED*

Node-RED és una plataforma de programació visual basada en nodes que permet crear fluxos de treball de manera intuïtiva. Utilitzant una interfície gràfica, els usuaris poden connectar diferents nodes per a definir la lògica i el comportament d'una aplicació. *Node-RED* és àmpliament utilitzat en el desenvolupament d'aplicacions *IoT* (Internet de les coses) i la integració de sistemes. Proporciona una

àmplia gamma de nodes predefinits que permeten la connexió amb diferents serveis i dispositius, com a bases de dades, serveis web, sensors i actuadors. Els usuaris poden arrossegar i deixar anar aquests nodes en una interfície visual i configurar les seves propietats per a definir la lògica de l'aplicació. *Node-RED* és altament personalitzable i extensible, ja que permet la creació de nodes personalitzats i la integració de biblioteques externes. A més, ofereix una interfície d'usuari amigable per a monitorar i depurar els fluxos de treball en temps real. És una eina poderosa i flexible per al desenvolupament ràpid d'aplicacions i l'automatització de processos.

6.5 *YOLO*

YOLO [9] (*You Only Look Once*) és un algorisme de detecció d'objectes en imatges i vídeos en temps real. En lloc de processar la imatge diverses vegades, *YOLO* divideix la imatge en una quadrícula i prediu els quadres delimitadors i classes d'objectes en cada quadre. Això ho fa ràpid i eficient. Durant l'entrenament, s'alimenta a la xarxa neuronal amb imatges etiquetades perquè aprengui a reconèixer objectes. En l'etapa d'inferència, es passa una imatge a través de la xarxa per a detectar objectes. *YOLO* és àmpliament utilitzat en aplicacions com a seguretat, conducció autònoma i realitat augmentada.

7 Desenvolupament

7.1 Provar models preentrenats

Durant el desenvolupament del projecte, es van dur a terme proves exhaustives utilitzant diferents models de *MMOCR* i *PaddleOCR* per a predir sobre imatges de codis de barres reals. Aquestes proves es van realitzar amb un conjunt de 36 imatges que representaven una varietat d'etiquetes amb codis de barres. No obstant això, els resultats obtinguts no van ser satisfactoris en tots dos casos. Els models de *OCR* utilitzats no van aconseguir un percentatge de precisió acceptable en el reconeixement de caràcters. En la majoria de les imatges, el text no era detectat correctament i, quan es detectava, el reconeixement era incorrecte en molts caràcters.

Aquests problemes es van deure als inconvenients discutits anteriorment. Com el soroll degut a la baixa resolució, desenfocament, perspectiva o les barres interferint el text. Els models de *OCR* tenen limitacions en la seva capacitat per a manejar diferents tipus d'etiquetes, fonts, grandàries, orientacions i nivells de soroll en les imatges. A més, és possible que els models per defecte no estiguin entrenats específicament per a reconèixer els caràcters presents en els codis de barres de les etiquetes utilitzades en el projecte, ja que s'entrenen amb tot tipus de caràcters i frases i no només nombres decimals.

La baixa precisió en la detecció i reconeixement de text en les imatges de codis de barres reals representa un desafiament important per al projecte. La correcta extracció i reconeixement dels caràcters en els codis de barres és fonamental per al funcionament adequat del sistema.

7.2 Generació d'Imatges sintètiques

Per a millorar l'execució i el rendiment dels models esmentats anteriorment, es necessitaria fer entrenaments exhaustius dels mateixos models amb, mínim, milers d'imatges d'etiquetes que tinguin diferents codis, sorolls, distorsions, esgarrapades, mides, etc. Aquest, no és el cas ja que en el moment no es tenen masses imatges dels productes i a més s'hauria de fer tot l'etiquetatge, ja sigui de la detecció com del reconeixement. Per això s'arriba a la conclusió de que el mètode més efectiu es crear imatges sintètiques de codis de barres que compleixin les característiques esmentades, amb certa aleatorietat.

Per això s'ha fet servir la llibreria *barcode4J* [10] de *JavaScript*. En concret es generen codis de tipus EAN13 [11] (13 dígits). En un codi d'aquest llenguatge, primer es genera un nombre de 12 dígits aleatori i es calcula el *cheksum* que és l'últim caràcter (dígit de control), després és generen certs nombres aleatoris dintre d'un rang que determinen aquestes diferents característiques: mida de la imatge, mida de la font, índex per a diferents fonts, separació entre el nombre i les barres. Aquest codi acaba generant la imatge i guardant el nombre per al procés posterior.

Fig. 4: Exemple d'imatge sintètica generada aleatòriament

Un cop es té la imatge generada, es modifica. Fent servir el llenguatge de *Python*, s'obté la imatge i se li aplica un filtre de Gaussiana, soroll i possiblement alguna esgarrapada que pugui interferir en el nombre; totes aquestes modificacions de la imatge es generen, també, segons uns generadors aleatoris.

Aquestes imatges serviran per l'entrenament de detecció i per a fer una avaluació de tot el model conjunt de detecció i reconeixement. Ara calen unes imatges per a entrenar el model de reconeixement. Aquestes son imatges de dígits de diferents fonts, mides, inclinacions, etc.

Fig. 5: Exemple d'imatge sintètica després d'aplicar-li els filtres

7.3 Creació de Datasets i entrenament de models

Un cop es tenen les imatges sintètiques dels codis de barres, s'han de crear els *datasets* que continguin aquestes

imatges i les seves respectives etiquetes per a la detecció i el reconeixement, aquestes són la *bounding box* i el nombre respectivament.

Com es comenta anteriorment, les etiquetes de reconeixement ja estan guardades en un fitxer de text on es guarda cada imatge amb el seu nombre corresponent en aquest format: *Code_43.png 6845400811023*. El repte, ara, és obtenir la *bounding box* del text ja que cada imatge tindrà unes mides i posicions diferents, ja sigui de la font, de la imatge, etc. I a l'hora de generar els codis no es pot obtenir aquesta *bounding box*, ja que només pots configurar uns valors per dir la mida, la separació, la font, etc, però no la posició exacte respecte la imatge.

Per a fer això es fa un detector de contorns, i a partir d'uns filtres es poden obtenir els contorns més petits que seran els caràcters. Des del primer caràcter fins a l'últim es fa un rectangle i aquest serà la *bounding box* que es guarda en aquest format: *Code_43.png [{"transcription": "684540081102", "points": [[62.0, 105.0], [491.0, 105.0], [491.0, 120.0], [62.0, 120.0]]}]*.

Fig. 6: Exemple d'imatge sintètica amb *bounding box* corresponent

7.4 Entrenar Models de detecció i reconeixement de text

Amb aquestes dades es duu a terme un entrenament amb diferents models de detecció tant de *PaddleOCR* con de *MMOCR*. Un cop entrenat, es prova la detecció i el reconeixement però aquesta detecció, a l'hora de predir, falla bastant ja que en molts casos es deixa els nombres dels extrems, blocs mes grans o fins i tot no detecta res. A part d'això, el reconeixement acaba reconeixent nombres "1" o lletres "l" en les barres que interfereixen en el text, a més, al ser models de reconeixement per defecte, preentrenats amb lletres, nombres, paraules, frases, etc, pot arribar a fer un mal reconeixement.

Degut a aquest inconvenient es passa a fer 3 *bounding boxs* que determinen el primer nombre, els següents 6 i els últims 6. D'aquesta manera, s'intenta evitar tant el reconeixement de les barres com a caràcters com la no detecció d'alguns nombres.

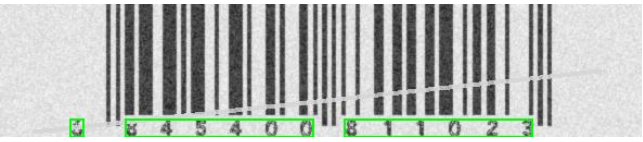


Fig. 7: Exemple d'imatge sintètica amb 3 *bounding boxs* corresponents

Amb aquest canvi, es veu que l'acompliment dels models entrenats millora considerablement, però en moltes prediccions, aquests segueixen deixant-se de detectar nombres, en aquest cas, també blocs sencers.

Fig. 8: Exemple de detecció de model OCR entrenat amb 3 *bounding boxs*.

Després d'una ampla recerca , es decideix canviar el model detecció de text a un model de detecció de *Yolo* on hi ha una sola classe que és el text. Es mantenen les mateixes etiqu etes per la detecció del text i només veient l'execució de l'entrenament del model de *yolov8*, segons el que indiquen els valors *P/R* sembla que té una gran millora.

S'entrena també el model de reconeixement amb les imatges sintètiques dels dígits i ja es té completat el procés de detecció i reconeixement. Ara falta el model de detecció d'etiquetes que continguin el codi de barres del producte.

Fig. 9: Exemple de detecció de model de *YOLO* entrenat amb 3 *bounding boxs*.

7.5 Entrenar model de detecció d'etiquetes

En aquest context, s'ha seleccionat el model de detecció *YOLOv8* com una eina eficaç per a la detecció d'etiquetes etiquetes en la imatge completa (*Fig 1*). No obstant això, s'ha identificat un desafiament: la preparació manual del conjunt de dades requeriria un temps considerable, ja que seria necessari escriure manualment les coordenades de les etiquetes per a cada imatge. Per a superar aquest obstacle, s'ha adoptat un enfocament alternatiu per a obtenir les ubicacions de les etiquetes de manera més eficient.

En lloc de fer una tasca manual intensiva, s'ha desenvolupat un detector de contorns per a localitzar automàticament les etiquetes en les imatges. Aquest procés implica aplicar filtres i tècniques de processament d'imatges binàries per a identificar i segmentar els contorns de les etiquetes presents en cada imatge. En utilitzar aquest enfocament, s'ha aconseguit extreure les ubicacions precises de les etiquetes sense necessitat d'intervenció manual.

Fig. 10: Exemple de detecció de contorns.

Una vegada que s'han obtingut les imatges i les seves corresponents etiquetes generades pel detector de contorns, es procedeix a l'entrenament del model *YOLOv8*. Aquest procés d'entrenament es basa en entrenar al model amb el conjunt de dades preparat, que consisteix en les imatges juntament amb les etiquetes trobades automàticament. És important destacar que s'ha observat un èxit notable en aquest procés d'entrenament.

El model *YOLOv8* aprèn a reconèixer patrons i característiques específiques en les imatges que li permeten detectar i localitzar les etiquetes de manera precisa. A mesura que el model s'entrena amb més imatges i etiquetes, millora la seva capacitat per a generalitzar i detectar amb major precisió les etiquetes en noves imatges no vistes prèviament.

Gràcies a aquest enfocament combinat de detecció automàtica de contorns i entrenament amb *YOLOv8*, s'ha aconseguit obtenir resultats prometedors en termes de detecció d'etiquetes. L'automatització del procés d'ubicació d'etiquetes ha permès reduir significativament el temps i els recursos requerits per a preparar el conjunt de dades. A més, el model ha demostrat un rendiment elevat en la detecció precisa de les etiquetes en diverses imatges.

7.5 Creació de serveis *gRPC*

En el context actual, una vegada que s'han establert tots els models necessaris, es procedeix a generar els serveis *gRPC* per a facilitar la interacció i el processament d'imatges. Aquests serveis ofereixen dues funcionalitats principals.

El primer servei *gRPC* rep com a paràmetre la imatge completa i utilitza el model de detecció d'etiquetes *YOLOv8* per a identificar les etiquetes presents en la imatge. Utilitzant aquest model, es genera una *bounding box* al voltant de cada etiqueta detectada, i es proporciona com a resposta la informació d'aquesta, incloent-hi les coordenades i l'índex de confiança corresponent. Això permet localitzar i delimitar amb precisió les etiquetes en la imatge completa.

El segon servei *gRPC* s'encarrega de rebre la Regió d'Interès (*ROI*) d'una etiqueta específica, que correspon a una imatge retallada que conté aquesta etiqueta. Si la *ROI* es troba en posició vertical, es realitza una transformació per a girar-la i col·locar-la en posició horitzontal. Després, utilitzant el model de detecció de text *YOLO*, es detecten els textos presents en la *ROI*. Aquestes *ROI*s identificades com a text es passen al model de reconeixement de caràcters *PaddleOCR*. Aquest model s'especialitza únicament en el reconeixement de text i és capaç d'extreure els caràcters continguts en cada *ROI*.

El servei de reconeixement de text retorna com a resposta les *bounding boxs* detectades juntament amb la seva confiança respectiva, és a dir, la certesa que cada *bounding box* conté text vàlid. A més, també proporciona el text reconegut en cada *ROI* i la confiança associada a la precisió del reconeixement.

En resum, el sistema implementa dos serveis *gRPC* per a facilitar el processament d'imatges. El primer utilitza el model *YOLOv8* per a detectar etiquetes en la imatge completa i retorna la informació de les *bounding boxs* juntament amb la seva confiança. El segon servei rep una *ROI* d'una etiqueta específica, l'ajusta si és necessari i després utilitza els models *YOLO* per a detectar els textos en la *ROI* i *PaddleOCR* per a reconèixer els caràcters.

Aquest enfocament permet una detecció i reconeixement precís d'etiquetes i text en imatges. En utilitzar models especialitzats i tècniques de processament avançades, s'aconsegueix una major eficiència i exactitud en l'extracció d'informació rellevant de les imatges. Els serveis *gRPC* proporcionen una interfície senzilla i eficient per a interactuar amb aquests models i obtenir els resultats desitjats.

7.6 Creació de *Docker* i imatge

El següent pas en el desenvolupament del sistema és la creació d'un arxiu *Dockerfile* que permeti la instal·lació de totes les llibreries i dependències necessàries, així com la configuració adequada per a executar els serveis *gRPC* i utilitzar els pesos dels models entrenats. Aquest enfocament és fonamental per a garantir la portabilitat del sistema i evitar problemes de dependències en executar-lo en diferents màquines.

El *Dockerfile* comença amb la selecció d'una imatge base, que servirà com a punt de partida per a la construcció del contenidor. Aquesta imatge base generalment inclou un sistema operatiu i les eines bàsiques necessàries. A continuació, s'instal·len les llibreries i dependències específiques requerides per al funcionament dels serveis *gRPC* i els models *YOLO* i *PaddleOCR*.

Per a instal·lar aquestes llibreries i dependències, s'utilitzen instruccions en el *Dockerfile* per a executar les instruccions corresponents. Això pot incloure la instal·lació de paquets específics utilitzant administradors de paquets com *pip* o *apt-get*, la descàrrega d'arxius addicionals necessaris i la configuració de variables d'entorn per al correcte funcionament dels serveis.

Una vegada que totes les llibreries i dependències s'han instal·lat correctament, es copien els serveis *gRPC* i els pesos dels models entrenats al contenidor. Això s'aconsegueix utilitzant comandos **COPY* en el *Dockerfile*, que copien els arxius i directoris rellevants des de la màquina amfitriona al contenidor.

A més de la instal·lació i còpia d'arxius, el fitxer també pot incloure instruccions per a configurar l'exposició de ports necessaris per a la comunicació amb els serveis *gRPC* i establir qualsevol altra configuració específica requerida per al sistema.

Una vegada que s'ha completat la configuració de l'arxiu, es procedeix a construir la imatge del contenidor utilitzant la instrucció "*docker build*". Aquest procés implica la compilació de tots els passos definits en el *Dockerfile* per a generar una imatge coherent i llista per a ser executada.

Una vegada construïda la imatge, es pot executar el contenidor utilitzant el comando "*docker run*". Això crea una instància del contenidor a partir de la imatge i s'assegura que totes les dependències i configuracions necessàries estiguin en el seu lloc. El contenidor s'executa en un entorn aïllat i es pot accedir als serveis *gRPC* proporcionats.

La utilització de *Docker* proporciona nombrosos beneficis. En primer lloc, garanteix que el sistema es pugui executar de manera consistent en diferents màquines, independentment de les diferències en els sistemes operatius i les configuracions locals. A més, evita problemes de dependències, ja que totes les llibreries i dependències necessàries s'instal·len dins del contenidor i no interfereixen amb l'entorn del sistema local. També proporciona una forma fàcil d'escalar i distribuir el sistema, ja que es pot crear i desplegar fàcilment múltiples instàncies del contenidor en diferents màquines.

7.7 Flux de *Node-RED*

El flux de *Node-RED* creat consta de diverses etapes que es connecten entre si per a realitzar el processament d'imatges i la detecció i reconeixement d'etiquetes i text. A continuació, es descriurà el flux detalladament

Injecció de dades:
Aquest node d'injecció s'utilitza per a iniciar el flux i activar l'execució.

Funció "Inicial":
Aquest node de funció s'encarrega d'inicialitzar totes les variables globals necessàries per al flux. Aquí es poden definir i configurar les variables que s'utilitzaran en les etapes següents.

Funció "*get_image_paths*":
En aquest node de funció, es proporciona un directori específic i es guarda una llista amb les rutes de totes les imatges presents en aquest directori. Aquesta funció permet obtenir la ubicació de les imatges que es processaran.

Funció "*get_image*":
En aquest node de funció, s'accedeix a la imatge actual segons l'iterador i es carrega en el flux per al seu processament posterior. Aquest pas permet treballar amb una imatge alhora dins del bucle d'iteració.

Funció "*f_tag_detection*":
Aquest node de funció és una de les etapes més importants del flux. Aquí es realitza una anomenada al servei *gRPC* de detecció d'etiquetes. Utilitzant el model *YOLOv8* de detecció d'etiquetes, s'envia la imatge al servei i s'obté la Regió d'Interès (*ROI*) que conté l'etiqueta. La *ROI* es passa al següent pas del flux.

Funció "*f_text_detection*":
En aquest node de funció, es realitza una anomenada al segon servidor *gRPC*, on es detecta i reconeix el text del codi de barres present en la *ROI* obtinguda anteriorment. S'utilitza el model *YOLO* per a la detecció de text i el model *PaddleOCR* per al reconeixement de caràcters. La funció retorna les *bounding boxs* detectades, la confiança associada i el text reconegut.

Funció de càlcul del codi de barres:
En aquesta etapa, s'utilitzen les llibreries de *ZBar* per a calcular el codi que hauria de tenir el codi de barres en la imatge. Després, es compara amb el text reconegut del codi de barres obtingut en el pas anterior.

Funció de càlcul de *distància de Levenshtein*:
En aquest node de funció, s'utilitza l'algorisme de *distància de Levenshtein* [12] per a calcular la diferència entre el codi de barres esperat i el text reconegut. Això proporciona una mesura de la similitud entre tots dos codis i permet avaluar la seva precisió.

Iteració i bucle:
El flux es repeteix per a cada imatge en el directori especificat. Això s'aconsegueix mitjançant l'ús de nodes de control de bucle o mitjançant la configuració adequada dels nodes de funció.

Aquest flux de *Node-RED* permet processar imatges en un directori específic, utilitzant serveis *gRPC* per a la detecció d'etiquetes i el reconeixement de text en els codis de barres presents. Els passos posteriors de comparació i càlcul de similitud brinden informació sobre la precisió del reconeixement.

8 Resultats i Conclusions

El projecte se centra en l'avaluació de diferents resultats utilitzant un conjunt de 300 imatges, que poden ser imatges d'esprais de pintura o ampolles de vi i cava.

Com es comenta anteriorment, per a la detecció d'etiquetes, s'utilitza el model *YOLOv8*. Els resultats mostren que aquest model té una precisió del 87,4% i una sensibilitat (*recall*) del 93,2% en mitjana. La precisió es refereix al percentatge d'àrea detectada correctament respecte l'àrea detectada total. . Mentre que la sensibilitat és el percentatge d'àrea correcta detectada respecte la correcta total.

En uant a la detecció de text en les imatges, també s'utilitza el model *YOLOv8*. Els resultats mostren una precisió del 86,2% i una sensibilitat del 89,5%. Això indica que el model és capaç d'identificar correctament el text present en les imatges en un alt percentatge de casos.

En l'àmbit del reconeixement de text, s'han avaluat quatre models diferents: *PaddleOCR PPOCR_v3* en anglès (sense entrenar i entrenat) i *MMOCR* (*SATRN*) (sense entrenar i entrenat). Aquests models s'utilitzen per a reconèixer i comprendre el text present en les imatges. Els resultats obtinguts són els següents:

- ***PaddleOCR* per defecte**: 82,8% de precisió. Això significa que el model reconeix correctament el 82,8% del text present en les imatges avaluades.
- ***PaddleOCR* entrenat**: 94,9% de precisió. En entrenar el model amb dades addicionals, s'aconsegueix millorar significativament la precisió del reconeixement de text.
- ***MMOCR* per defecte**: 71,1% de precisió. Aquest model mostra una precisió més baixa en comparació amb PaddleOCR per defecte.
- ***MMOCR* entrenat**: 96,2% de precisió. Igual que amb *PaddleOCR*, l'entrenament millora dràsticament la precisió del model, aconseguint un valor molt alt.

Aquests resultats demostren que els models entrenats tenen un rendiment molt millor en comparació amb els models per defecte. L'entrenament permet ajustar els models a les dades específiques utilitzades en el projecte, la qual cosa millora la seva capacitat per a reconèixer i comprendre el text en les imatges.

En resum, el projecte utilitza tècniques de detecció d'objectes i reconeixement de text per a analitzar imatges d'esprais de pintura i ampolles de vi i cava. Els resultats mostren alts nivells de precisió i recuperació en la detecció d'etiquetes i text, i l'entrenament dels models millora encara més el rendiment, aconseguint alts percentatges de precisió en el reconeixement de text.

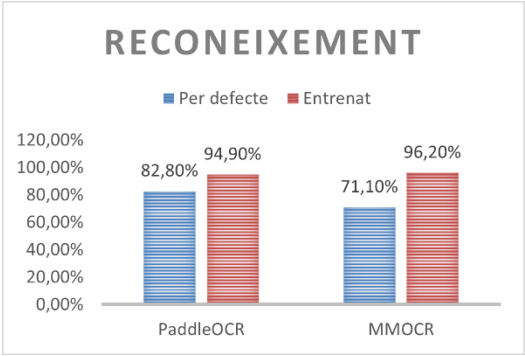


Fig. 11. Resultats de models de Reconeixement

9 Agraïments

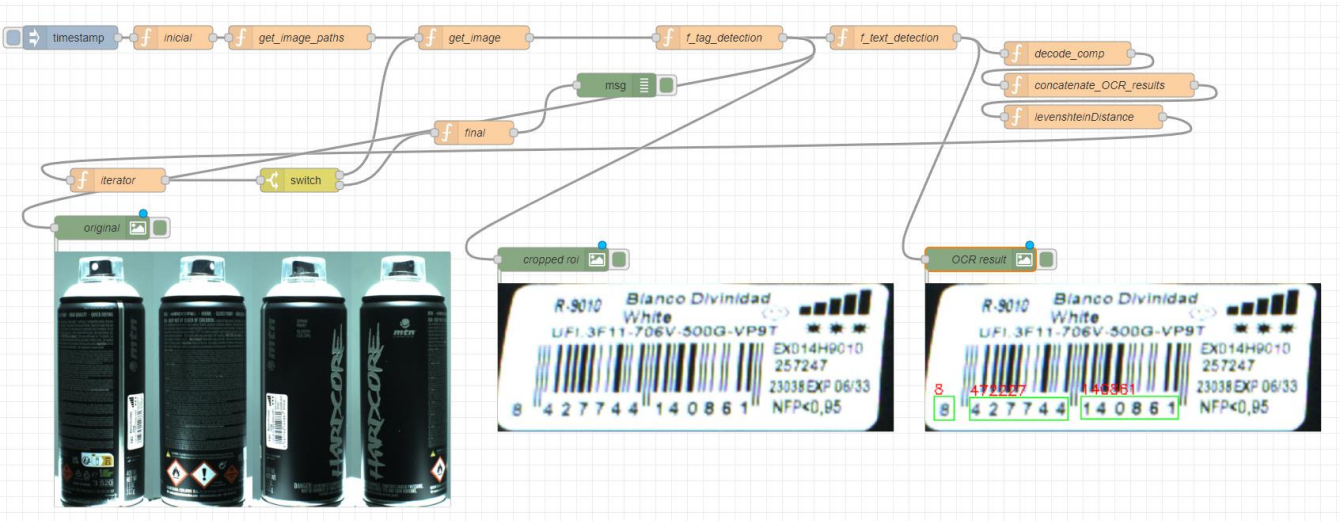
M'agradaria expressar el meu més sincer agraïment al meu tutor del treball, Ernest Valveny i a tots els integrants de l'empresa *Rosepetal SL*. El seu valuós assessorament diari i el material proporcionat han estat fonamentals per a l'èxit d'aquest projecte.

10 Bibliografia

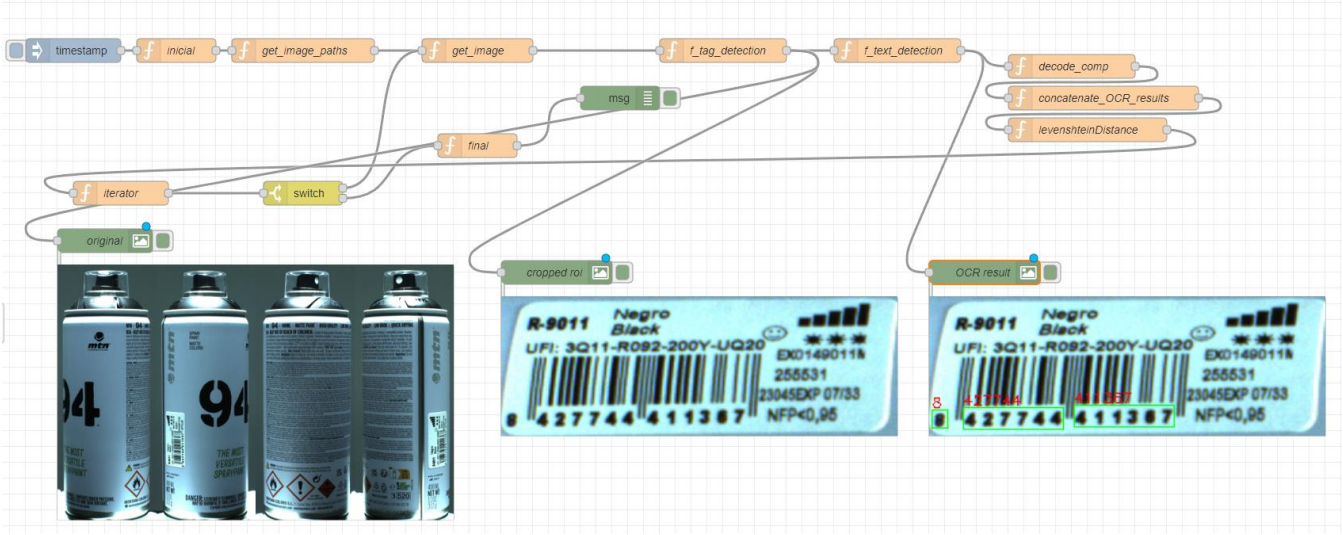
[1] Rosepetal SL, (n.d.), https://www.rosepetal.ai/
[2] PaddleOCR, (n.d.), https://github.com/PaddlePaddle/PaddleOCR
[3] Tesseract, (n.d.), https://github.com/tesseract-ocr/tesseract
[4] MMOCR, (n.d.), https://github.com/open-mmlab/mmocr
[5] Pytorch, (n.d.), https://pytorch.org/
[6] Node-RED, (n.d.), https://nodered.org/
[7] gRPC, (n.d.), https://grpc.io/
[8] Zbar, (n.d.), https://zbar.sourceforge.net/
[9] YOLO, (n.d.), https://github.com/ultralytics/ultralytics
[10] barcode4J, (n.d.), https://barcode4j.sourceforge.net/
[11] EAN13, (n.d.), https://en.wikipedia.org/wiki/International_Article_Number
[12] Distància de Levenshtein, (n.d.), https://en.wikipedia.org/wiki/Levenshtein_distance

APÈNDIX

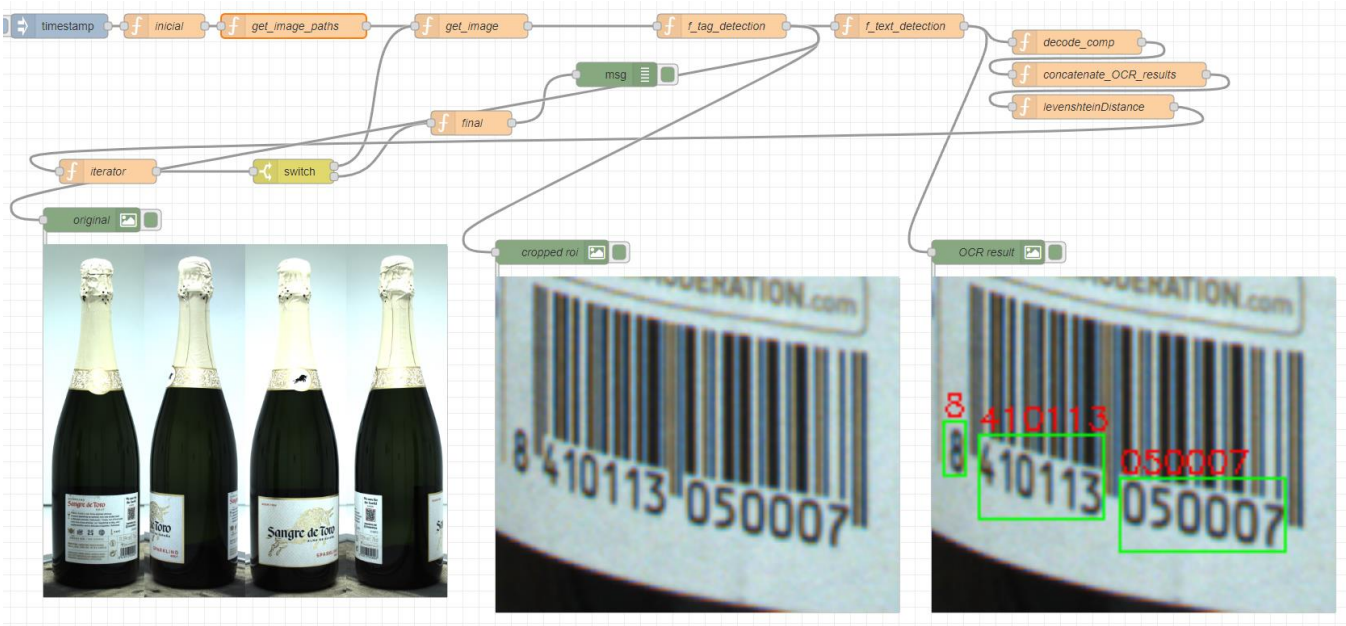
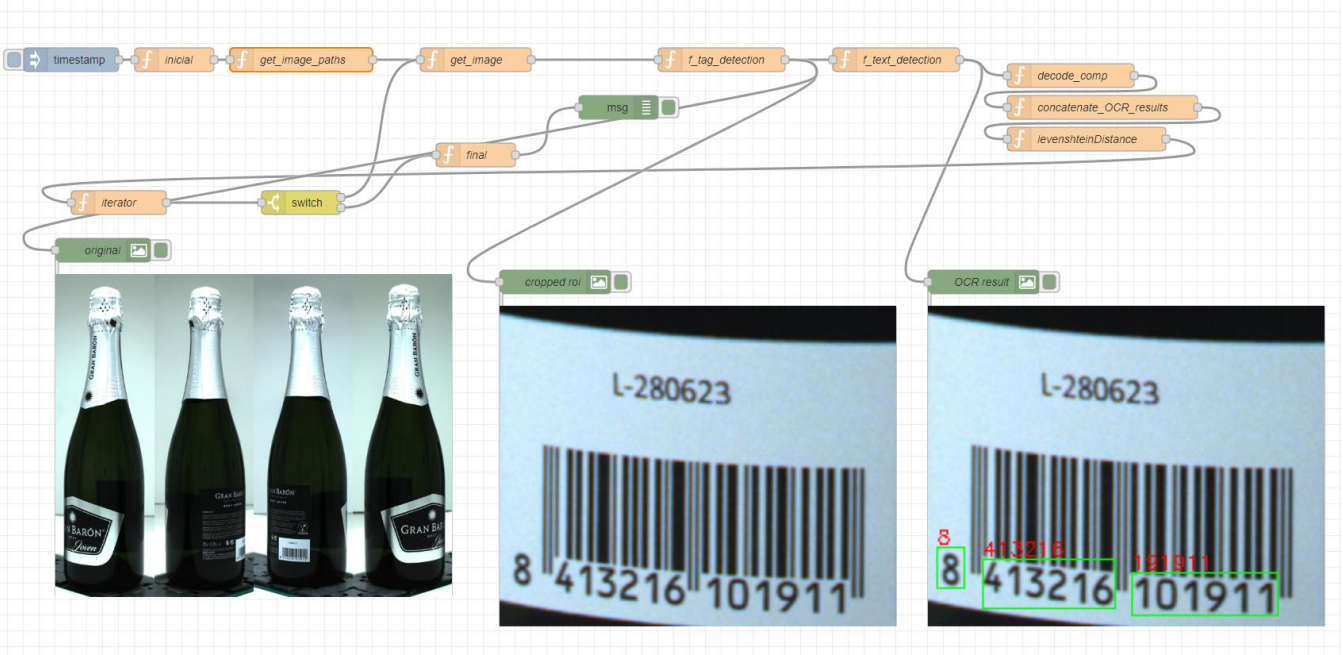
A1. Exemple 1 Node-RED



A2. Exemple 2 Node-RED



A3. Exemple 3 Node-RED



A4. Exemple 4 Node-RED