

Automatització d'esquemes per a la Indústria 4.0 amb Visió per Computador

Menció de Computació

Hernán Capilla Urbano

1462773

Índex

0.	Introducció	3
1.	Problemàtica	5
2.	Estat de l'art	8
3.	Pla de treball	12
4.	Progrés	13
4.1.	Punt 1 – Detecció de textos.....	13
5.	Planificació	21
6.	Bibliografia	22

0. Introducció

Aquest projecte consisteix a crear una eina que automatitzi la creació de la renovació d'un sinòptic ja existeix orientat a un sistema SCADA. Amb l'objectiu de simplificar i reduir la càrrega de treball de l'enginyer/a que estigui encarregat de fer aquesta tasca. De manera que, conservant la majoria de les propietats del sinòptic original es pugui crear una nova versió del ja existent sinòptic passant per alt les limitacions que implica el sistema anterior i format sobre el qual es va crear el sinòptic original en el seu moment. Tot això per tal de reduir la quantitat d'hores que l'enginyer ha de dedicar a tasques que no aporten valor al sinòptic final renovat.

Amb això dit, doncs, i abans d'entrar en matèria, hi ha certes terminologies que hem d'explicar per poder comprendre de quina manera actua aquesta eina i quin impacte té sobre l'estat de l'art actual de la matèria.

Comencem explicant què és un sistema SCADA i què és un sinòptic.

Un sistema SCADA (Supervisory Control and Data Acquisition) és un sistema complet emprat en allò que coneixem com a Indústria4.0 que s'utilitza per supervisar i controlar processos industrials de forma remota i que està compost per diverses parts:

- Supervisió: programari que permet la visualització i gestió de dades.
- Connexió amb controladors: dispositius que es connecten amb sensors i actuadors situats a la planta industrial.
- Bases de dades: emprades per emmagatzemar un històric de la informació recopilada.
- Interfícies d'usuari: pantalles en què els operadors poden observar les dades recopilades en temps real i interactuar-hi.

És sobre aquest darrer apartat sobre el qual ens fixarem, ja que un sinòptic està englobat en aquesta descripció. Se'n diu sinòptic a la pantalla que mostra l'estat dels processos de la planta, les dades i els elements en temps real [fig0.a]. D'aquesta manera facilita la comprensió del funcionament del procés industrial als operaris que hi treballen, els permet veure en temps real les dades recopilades del procés i, a més, disminueix el temps de resposta davant de qualsevol situació anòmla que pugui passar.

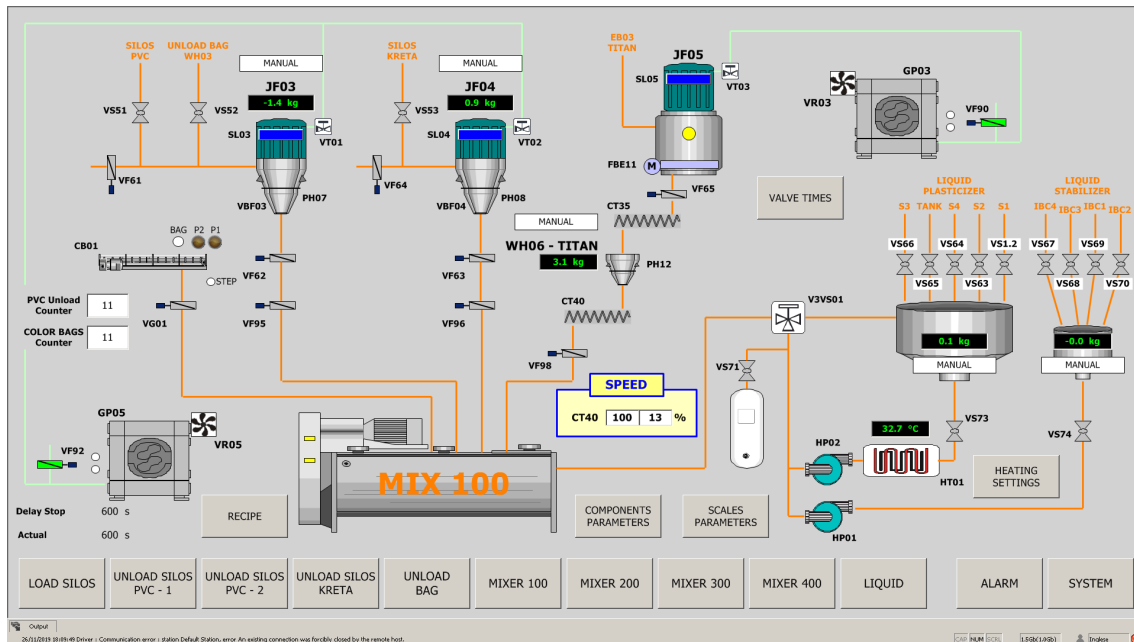


fig0.a – Exemple de sinòptic en un procés industrial

Tenint aquests conceptes clars, podem donar pas a endinsar-se en la pregunta de com es crea un sinòptic? Com passem de tenir un PLC, una base de dades i els diferents elements de la planta industrial que tots formen part d'un mateix sinòptic mostrant les dades en temps real?

Hi ha multitud d'eines i programes que ens ajuden en la tasca de crear el sinòptic. Alguns exemples destacables són TIAPortal, WinCC, PlantSCADA i SystemPlatform. Aquests són només alguns exemples, però hi ha més programes i entorns tant públics com propis de l'empresa per fer aquesta tasca.

El que és destacable és que ofereixen un entorn on manipular tant les connexions, com els elements i les seves característiques pròpies, les bases de dades i fins i tot editors de diagrames.

El paper de l'enginyer és el de, juntament amb tot el que hem esmentat abans, combinar-lo en la creació d'un diagrama que actuarà com a sinòptic. Per a aquest procés l'enginyer pot abordar des de la recollecció de senyals que van des de PLC fins a l'element en qüestió (bomba, motor, ventilador, etc.), com ara configurar les característiques pròpies que tindran aquests elements (variables, tipus de dades de les variables), quines variables utilitzarà dins de la base de dades, i quines fórmules empraran aquestes variables per calcular-ne els valors, quin tipus d'alarmes tindrà, quan s'activaran aquestes alarmes; fins al disseny i dibuix de tot aquest procés industrial.

Per a aquest primer treball de recollecció, un enginyer amb experiència pot arribar a dedicar-hi desenes d'hores per a una sola pantalla. Cal tenir en compte que cada element té les seves característiques i usos propis, i que una BombaA pot no actuar de la mateixa manera que una BombaB malgrat tenir tots dos el mateix element base. En el cas d'un enginyer sense gaire experiència, aquesta feina pot arribar a superar més de cent hores en els seus primers projectes.

Pel segon cas, el disseny de pantalla, ens trobem amb un treball també força complex. Si bé en el primer cas havíem de ser minuciosos en fer el treball de recollecció, en aquest segon s'ha de ser fins i tot més.

El principal motiu és que aquest serà la nostra “presentació” de cara al client. Un client que, en cas que la pantalla no mostri de forma correcta les dades anteriorment recopilades, o en cas que hi hagi algun error en com les mostra, el client perdrà moltíssim temps i diners; així com nosaltres; i que pot arribar a generar problemes tant de funcionament a la pròpia fàbrica com que doni possibilitat a accidents laborals. És per això que aquest apartat és tan important.

Fins ara hem vist com es creen els sinòptics des de zero. Ara que tenim una idea aproximada de què estem treballant, explicarem on actua aquest projecte i quines dificultats intenta resoldre.

1. Problemàtica

Aquest projecte està enfocat a una problemàtica que afecta tant enginyers sènior com júnior, però, sobretot, els enginyers júnior, que generalment tenen menys experiència i estan menys preparats per a aquest tipus de tasques de creació de sinòptics i manipulació de les seves dades.

És en aquests primers projectes que es destinen a enginyers poc madurs en aquest tipus de feines, que s'ofereixen la creació de sinòptics a partir d'un model ja existent. És a dir, sinòptics que ja estan llestos i porten en funcionament des de fa molts anys, volen ser remodelats i actualitzats a altres versions o altres programes actuals.

Posem el següent exemple: *A la meua fàbrica de laminadures, porto 25 anys usant el mateix sinòptic per a la línia de creació de caramel. Aquest sinòptic es va fer amb la tecnologia més puntera de fa 25 anys, i durant aquest temps la meua cadena de muntatge ha anat canviant, abans tenia dues màquines de barreja de caramel i ara només en tinc una perquè fa 10 anys la vaig canviar a una de tecnologia puntera.*

Tinc tot aquest procés industrial en una planta ubicada a Villarobledo, Albacete, i està dividida en diverses seccions. Per a cada secció tinc un sinòptic que mostra la

seva respectiva línia de fabricació de cada laminadura. I cada sinòptic es va crear al seu moment a l'entorn per a SCADA's de WinCC, que era el programa més avançat al seu moment.

Però ara, el 2024, vull remodelar tot el sistema de sinòptics, perquè durant aquests anys hi ha hagut variacions en aquestes línies. I ara vull afegir més sensors que em permeten portar un millor control de la meva producció. És a dir, els elements ja existents tindran més característiques pròpies que els que tinc ara mateix.

Així que contractaré una empresa anomenada DummyCorp SL que s'encarregarà de renovar tot el sistema SCADA i els seus respectius sinòptics a uns de nous i afegir-hi aquestes noves funcionalitats. Prescindiré de WinCC perquè avui dia ja no és el millor programa que existeix al mercat i faré servir un nou entorn propi de DummyCorp anomenat DumDum.

I aquest n'és l'exemple. Això és el que està succeint avui dia al mercat de la Indústria4.0. Empreses que van fer les seves SCADA ara fa anys els volen remodelar i migrar de sistemes. Volen que les seves velles pantalles es vegin actualitzades i disposin de més i millor visibilitat, adaptat a les noves resolucions de pantalles i amb característiques noves per a cada element.

I és aquí on entra aquest projecte.

La dicotomia de l'assumpte és que una part important d'aquest treball de renovació de sinòptics és el temps i recursos emprats en únicament el disseny i nou dibuix de la pantalla.

Les pantalles velles no són reaprofitables, estan en formats concrets dins projectes concrets en entorns concrets en sistemes operatius concrets que tenen molts anys. Així doncs, el seu ús acaba sent relegat al de patró en què basar-se per poder realitzar la pantalla nova des de 0.

El problema que sorgeix també d'això és que aquestes pantalles es troben a les mateixes fàbriques, com cal esperar, i no estan pensades per ser manipulades de forma còmoda. Així doncs, a l'hora d'indagar l'enginyer en els diferents elements d'aquesta pantalla per prendre notes i fer-ne la renovació, s'han d'emprar màquines virtuals que emulen el sistema d'ús original a la fàbrica o escriptoris remots que hi connecten directament. Màquines virtuals i escriptoris remots que converteixen el simple fet d'obrir un arxiu ja pesat de per si en mida, en una tasca autènticament feixuga. Els recursos a les màquines virtuals o escriptoris remots, malgrat ser els indicats, moltes vegades no eviten el fet que el programa es pengi, o que vagi extremadament lent o l'input lag propi d'estar connectant-te a un servidor amb moltes capes de seguretat per sobre i que s'està utilitzant alhora en la producció de la fàbrica.

I això és només per començar, perquè un cop tinguis obert tot el que necessitis i identificats els elements, hauràs de fer el dibuix de la nova pantalla des de zero. Hauràs d'identificar cada element de la pantalla origen, crear-ne un de semblant per a la nova pantalla, prendre les seves dimensions, veure com encaixa a la nova pantalla, assegurar-te de les dimensions de la nova pantalla, veure que càpiga tot, i que el procés industrial és el mateix que el de la pantalla origen. Tot això tenint en compte que la màquina sobre la qual estàs treballant és lenta i incòmoda.

I, com bé hem indicat, només per començar a treballar a la nova pantalla. Com que diu l'exemple, volem migrar el sistema d'un antic a un de nou. Potser no volem que els elements, per exemple, les bombes, es vegin com al programa antic, i volem que siguin més grans o que tinguin una altra forma. Òbviament voldrem un resultat professional, en què estigui tot alineat, tot amb la seva nomenclatura respectiva i que s'aprofiti degudament l'espai.

Tot això és una feina que algú ha de fer. I com hem comentat anteriorment, tant els perfils júnior com els sènior s'han d'enfrontar a aquestes dificultats i incomoditats únicament per començar a treballar.

Aquí és on entra aquesta feina. En solucionar i alleugerir aquesta càrrega de treball que suposa desenes d'hores per al treballador en aquestes parts inicials de renovació de sinòptics en sistemes SCADA. Usant mitjançant:

- L'automatització del procés d'identificació d'elements per a cada pantalla de tota una planta.
- La senzilla implementació de nous elements seguint el diagrama de la pantalla original.
- El fàcil afegit de noves funcionalitats en forma de template modificable a aquests elements.
- L'adaptació ràpida tant a redissenys com a canvis en les proporcions de la pantalla final resultant mitjançant ús de diagrames basats en vectors que permeten un reescalfament ràpid a diferents mides i resolucions.
- La minimització de la necessitat de fer servir eines antigues i enutjoses que entorpeixen el procés de creació.

2. Estat de l'art

De tot allò esmentat anteriorment, ideem el següent plantejament sobre punts clau que devia tenir l'eina que desenvoluparem en aquest projecte. Aquests punts clau són els següents:

- El programa ha d'agafar una imatge com a input, i sense res més que això, ser capaç de realitzar les tasques següents:
 - El programa ha de ser capaç d'identificar els textos que corresponen a cada element. A diferència d'altres característiques dels elements, els textos són visibles a primera vista des del sinòptic. I han de ser detectats i pertànyer al seu element corresponent.
 - El programa ha d'identificar de forma autònoma els elements clau del diagrama original. Ha de ser capaç de detectar les formes dels elements i classificar-los al tipus d'elements que s'hi corresponen. Heu de distingir una bomba d'una vàlvula, una vàlvula d'un ventilador, un ventilador d'un tanc d'aigua, etc.
 - El programa ha de ser capaç de modificar o afegir les funcionalitats que volem als elements detectats. Un sistema basat en templates ha de ser emprat per a cada element diferent.
 - El programa ha de ser capaç de rastrejar les connexions (línies) que uneixen els diferents elements del diagrama. Aquestes línies indiquen l'ordre que segueix el procés industrial i, malgrat que no és un element com a tal, és important que mantinguin la coherència entre els elements i uneixin únicament els que veritablement estan connectats.
 - El programa ha de ser sensible i estar preparat per a un reescalat segons les necessitats de la pantalla final. Cal treballar amb un format basat en vectors perquè la fidelitat gràfica no es vegi afectada. Es treballarà amb SVG.

Amb aquest recull dels requisits, busquem l'estat de l'art d'aquest projecte. Projectes realitzats amb anterioritat que compleixin aquesta funció que volem fer o siguin semblants.

El que vam trobar va ser el següent, i és que no hem trobat ni conegut l'existència de projectes emprats per a la Indústria4.0 ni l'elaboració automàtica de sinòptics industrials.

Assumim que és degut a la pròpia naturalesa del món de la indústria4.0. És un món en apogeu i els motors propis o eines d'empreses privades han de mantenir-se en secret i no ser accessibles al públic. A més, els sinòptics contenen informació crucial d'un sistema industrial complex. La publicació o el treball sobre aquests

podria suposar pistes o informació confidencial de processos industrials que no interessa que se sàpiguen a gran escala.

Tot i això, a la investigació inicial, trobem diversos projectes usats per a diferents propòsits que ens donarien una idea de cap a on encaminar-nos per realitzar l'eina que compleixi amb les idees anteriorment esmentades.

LIVE: Towards Layer-wise Image Vectorization

Aquest projecte de la Universitat del Nord-oest s'aplica a la problemàtica de la conversió d'imatges planes a SVG. Passar una imatge directament i passar-la a un format basat en vectors.

Això ho aconsegueix mitjançant la divisió per capes de la imatge original. I a poc a poc i capa mitjançant anar creant el SVG recursivament entorn d'això, de manera que a cada iteració es vagi semblant més i més a la imatge original.

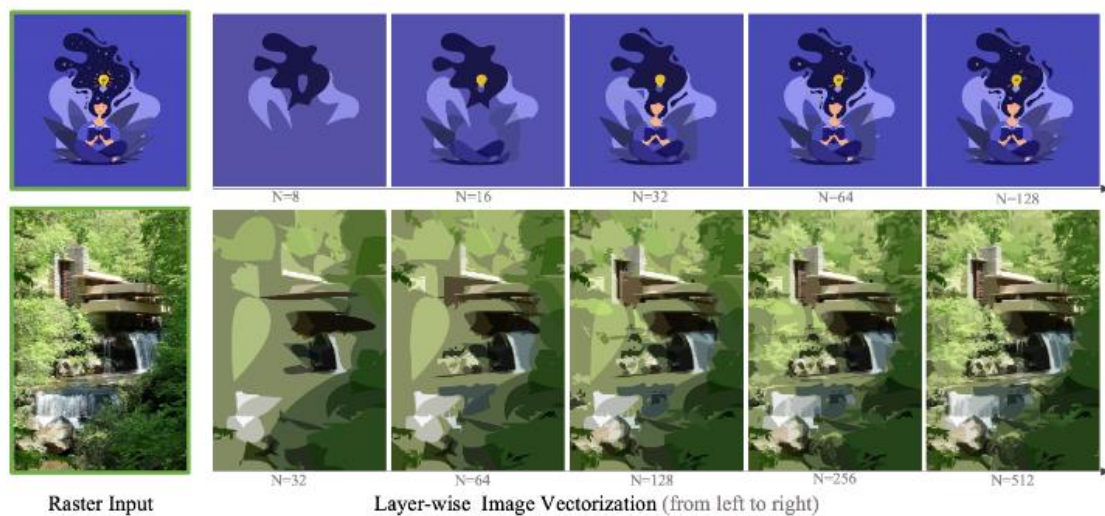


fig2.a – Procés de vectoritzat

Aquest projecte encaixava amb la idea amb la partíem al principi, i és que era la de processar la imatge input del sinòptic i convertir-la enterament a SVG. El problema amb això era que els diferents elements del sinòptic perdien les seves característiques, allò que els feia útils de cara a la indústria4.0. A més que l'enfocament de fer-ho capa per capa no ens semblava interessant des del punt de vista pràctic. Un sinòptic pot tenir desenes d'elements, línies de connexions entre

elements i també conté textos. No ens serveix de res vectoritzar-ho tot si perdiem les propietats i no classificàvem cada element per la seva classe.

A més, els sinòptics són pràcticament sempre imatges en 2D, no hi ha necessitat de fer una divisió per capes. Sinó que és més interessant detectar directament quins tipus d'elements clau sosté la imatge.

Raster-to-Vector: Revisiting Floorplan Transformation

Aquest Projecte de la Universitat de Washington tracta la problemàtica de convertir una imatge a un pla vectoritzat i després a un 3D en una casuística arquitectònica [fig2.b].

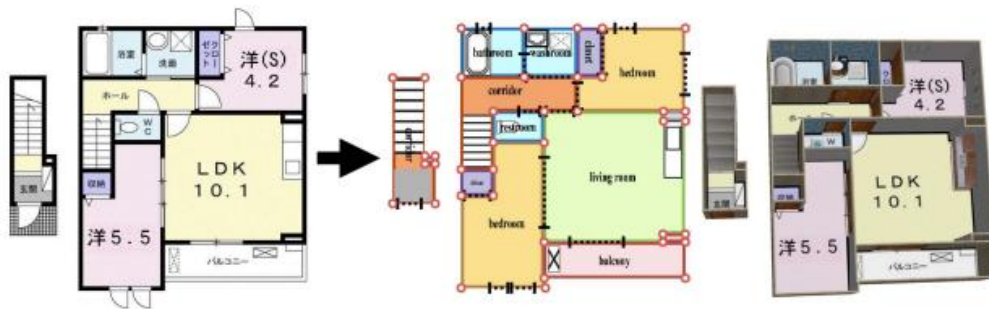


fig2.b – D'imatge a pla vectoritzat i model 3D

S'hi explica que fan servir un model per detectar els punts d'unió dels murs per identificar cada habitació, agafar-ne les dimensions i després passar-lo a un format basat en vectors [fig2.c].

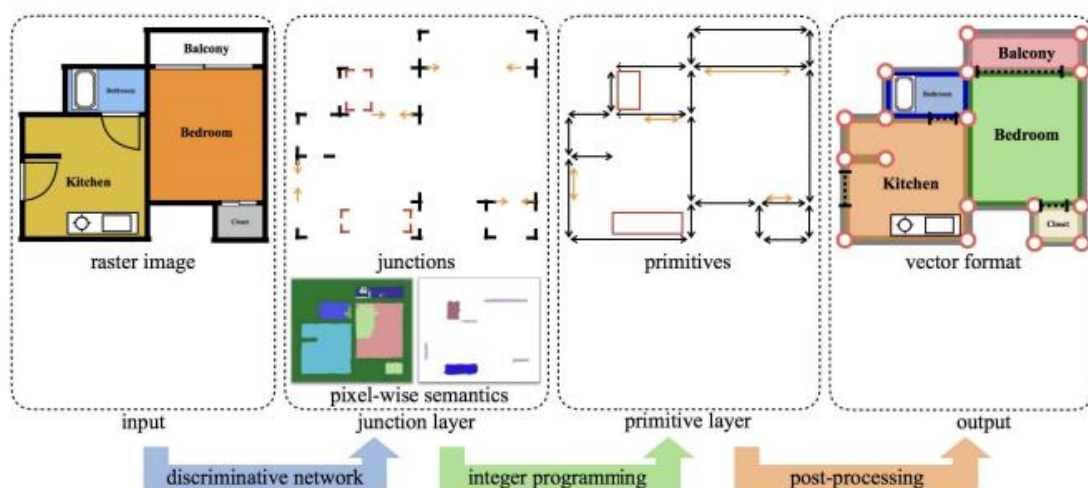


fig2.c – Procés de detecció de punts clau

Per realitzar aquest procés de detecció, creeu un conjunt de formes per a cadascun dels punts clau que es poden arribar a trobar [fig2.d].

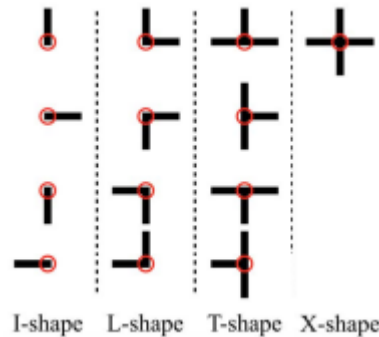


fig2.d – Conjunt de formes clau

Constaten que emprant aquestes formes clau i veient la diferència entre les coordenades entre elles, poden definir on són els murs que es connecten entre aquests punts clau.

La forma que tenen per definir quan s'ha acabat el procés és una vegada tots els punts clau quedin connectats entre si. Doncs aquesta serà la dimensió de la casa i tot allò que quedi fora serà exterior. Havent acabat així el procés.

Punts que cal destacar d'aquest projecte:

- Ús d'una imatge representant un plànol com a entrada.
- Detecció de punts clau i formes.
- Desintegració per parts del procés.
- Ús d'un format basat en vectors.

Aquest darrer projecte va ser extremadament interessant de llegir i és el que més pistes i indicacions em va donar sobre com avançar i com abordar la nostra problemàtica. Es pot observar que en si la problemàtica que sosté el projecte és molt semblant a la nostra.

Cal indicar que no han estat els únics projectes investigats, en total explico més de deu projectes que tracten el vectoritzat d'imatges de diverses maneres. És per aquest motiu que hem seleccionat aquests dos darrers com a millor representació de tota la recerca per començar el projecte.

Ha estat amb aquest darrer però, amb allò après dels altres, i amb reunions amb professionals del sector de l'Enginyeria Informàtica amb amplis coneixements de Visió per Computador que ideem el pla de treball per abordar els diferents requisits que havia de sostenir i solucionar el nostre projecte .

3. Pla de treball

Un cop recopilats els requisits i funcionalitats que esperem del projecte. I investigats també diversos projectes relacionats amb la matèria del que volem fer. Ideem un pla de treball.

Hem optat per dividir el projecte en diverses fases. D'aquesta manera, prioritzem la modularització del projecte darrere de la següent afirmació <<Divideix i venceràs>>. Filosofia que hem après al grau i que ens serà molt útil pel que volem abordar.

La modularització del projecte sosté la idea d'anar tractant a poc a poc els diferents problemes amb què ens trobem generalment en un sinòptic i anar-los solucionant darrere d'un tractament correcte dels mateixos. Així mateix, tenim en ment la idea futura de poder realitzar diferents millores en seccions ja abordades anteriorment amb l'objectiu de treure encara més partit al sistema.

El pla de treball és el següent:

1. **Detecció de textos:** Emprar un OCR (Optical Character Recognition) per realitzar un primer escombrat per tot el sinòptic. Obtenir els diferents textos que s'hi troben, les seves coordenades i la seva mida. La idea és basant-se en aquesta àrea on s'ha trobat el text, substituir-lo pel color predominant de la zona, que assumim serà fons, i pintar tota la zona d'aquest color, aquesta forma extraïem el text de la imatge original. La idea darrere d'això és evitar que més endavant el text pugui suposar una interferència o actuï com a soroll en els passos següents. El text extret serà col·locat des d'aquest pas, mantenint la seva mida i coordenades en un SVG que actuarà com a sinòptic final.
2. **Detecció d'elements:** Emprar YOLO per realitzar una primera detecció de cada element diferent dins de cada sinòptic, obtenint-ne també les coordenades. D'aquesta manera obtenim la classe de l'element, que serà introduïda per l'enginyer, i ens servirà per a la posterior detecció dins de tots els sinòptics d'una mateixa planta. La detecció agafarà aquest element i el compararà mitjançant pattern recognition amb aquest mateix element a diverses orientacions. Així, es podrà abordar totes les possibilitats d'aquest element dins de tots els sinòptics. En obtenir la classe de l'element, ens permet enginyeria un sistema de substitució basat en templates ja vectoritzats i amb característiques modificables segons les conveniències de l'usuari. Amb això esperem que, un cop detectat cada element diferent, ja es comptarà amb la seva classe i en un futur podrà ser fàcilment reemplaçable en cas de requerir-ho. Cal recalcar que els elements detectats i classificats, seran afegits usant el seu templat al sinòptic final abans

esmentat al punt 1. *[Aquesta idea està encara en desenvolupament i la seva explicació serà més precisa a l'Informe de Progrés II]*

3. **Detecció de connexions:** Emprar un sistema de reconeixement basat en píxels des de cada element per traçar els camins de les connexions que connecten cada element amb el següent. Cada element està connectat a un altre (o no) seguint un camí que equival a les línies del procés industrial. Detectarem això i els relacionarem entre si emprant algorismes com Manhattan, per exemple, per unir-los buscant la distància mínima entre tots dos i evitant alhora la cruïlla de camins que no estan pensats per creuar-se. *[Aquesta idea encara està en desenvolupament i la seva explicació serà més precisa a l'Informe de Progrés II]*

Aquest és el procés que esperem seguir per fer aquest projecte. Cal destacar que el punt 2 i 3 s'abordarà de millor manera al següent Informe de Progrés. Aquestes són les idees que tenim actualment, però amb l'avenç del projecte es poden veure alterades darrere d'una millor manera d'abordar la situació.

4. Progrés

4.1. Punt 1 – Detecció de textos

En aquest apartat descriuré el procés que vaig seguir en la cerca i aplicació d'un OCR per fer el Punt 1 descrit al pla de treball. Tot el projecte està sent realitzat a Python.

Primerament, vaig fer una cerca preliminar d'OCR's de codi lliure. Les característiques que tenia clares des del principi de la cerca era que no s'haurien d'enfrontar a molt de soroll, i que no calia que fossin extremadament potents. L'objectiu era fer una escombrada ràpida d'un diagrama en 2D en què en el pitjor dels casos la imatge es veuria una mica borrós, però no prou perquè la detecció s'hagués d'enfrontar a problemes que requerissin dilatacions o contraccions o molt preprocessament de la imatge. A més d'això, no es veuria enfrontada a gaire soroll. Això és perquè per norma general als sinòptics s'intenta evitar col·locar textos que intercedeixin amb altres elements del diagrama. Això es fa per seguretat perquè no puguin sorgir confusions entre els operaris que visualitzen aquestes pantalles en una passada ràpida i que pugui portar a una malinterpretació en el nom de l'element. Això és una dada que, si bé no està com a tal escrita enlloc, és una cosa que he anat veient i se m'ha recalcat en aquest darrer any que he estat treballant al sector de la Indústria 4.0.

La cerca va concloure en dues principals llibreries: EasyOCR, Pytesseract.

També es recomanava que en cas d'imatges on hi hagués una gran quantitat de textos, una OCR extensament entrenada seria una gran aliada a tenir en compte. Això serà important més endavant.

Al principi, per al que s'usaria i donades les condicions anteriorment expressades amb les dues llibreries trobades seria suficient.

A l'inici treballem amb un petit diagrama fet per nosaltres mateixos [fig4.1.a] a Draw.io (web per fer diagrames a SVG) per fer petites proves de l'OCR i decidir sobre quin decantar-nos.

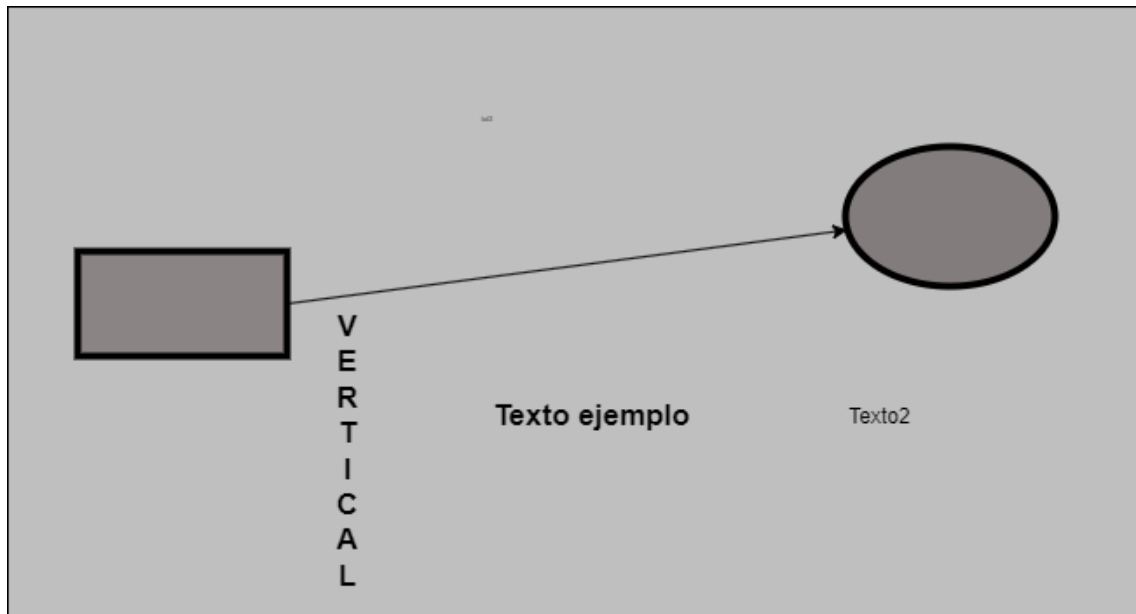


fig4.1.a – Diagrama d'exemple

Aquest diagrama, si bé no era gran cosa, ens servia per tastar una mica de cada cosa:

- Un text molt petit (està situat sobre la fletxa i sembla un esborrall)
- Un text en vertical
- Un text ben definit a una mida normal
- Un text ben definit una mica més petit que conté un número

Sobre això cal indicar diverses coses. I és que no estàvem acostumats a treballar amb OCR. Era la primera vegada que treballàvem amb aquesta tecnologia fora de classe i no sabíem exactament què esperar-ne, com ens arribarien les dades, o quina era la seva capacitat de detecció real, les seves fortaleces i debilitats.

Vam crear un petit script de test per comprovar quina llibreria havíem de triar. A la següent imatge [fig4.1.b] podem observar com va actuar EasyOCR amb el diagrama d'exemple esmentat:

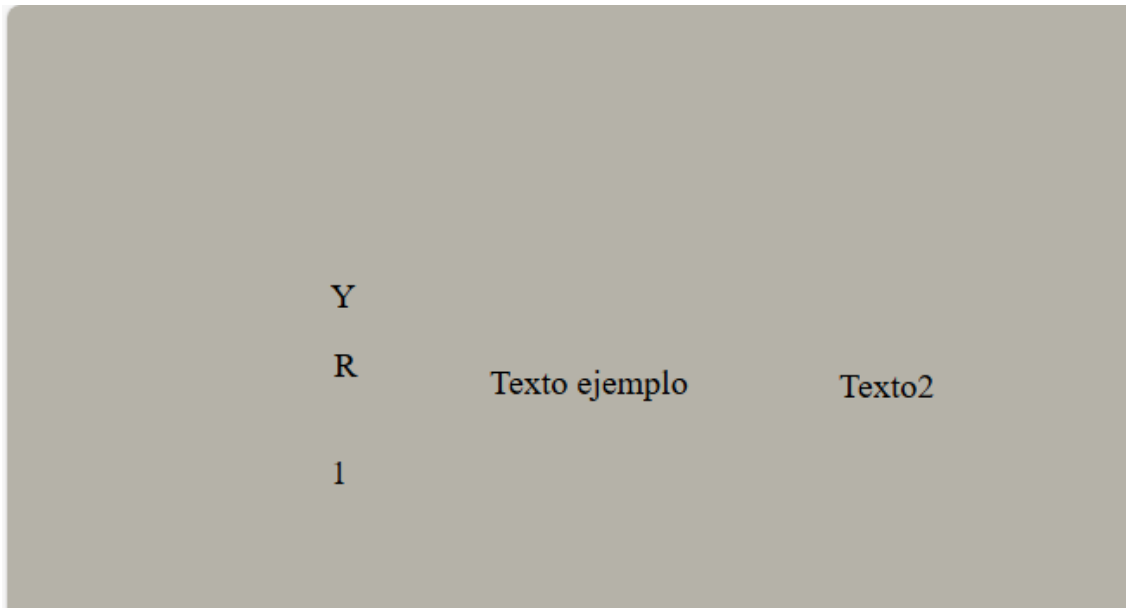


fig4.1.b – Resultat d'EasyOCR

Es pot observar com:

- El text molt petit ha desaparegut, no s'ha detectat.
- El text vertical ha estat detectat erròniament.
- El text definit ha estat detectat perfecte.
- El text amb número ha estat detectat perfecte.

Ara observem el resultat de Pytesseract davant la mateixa imatge:

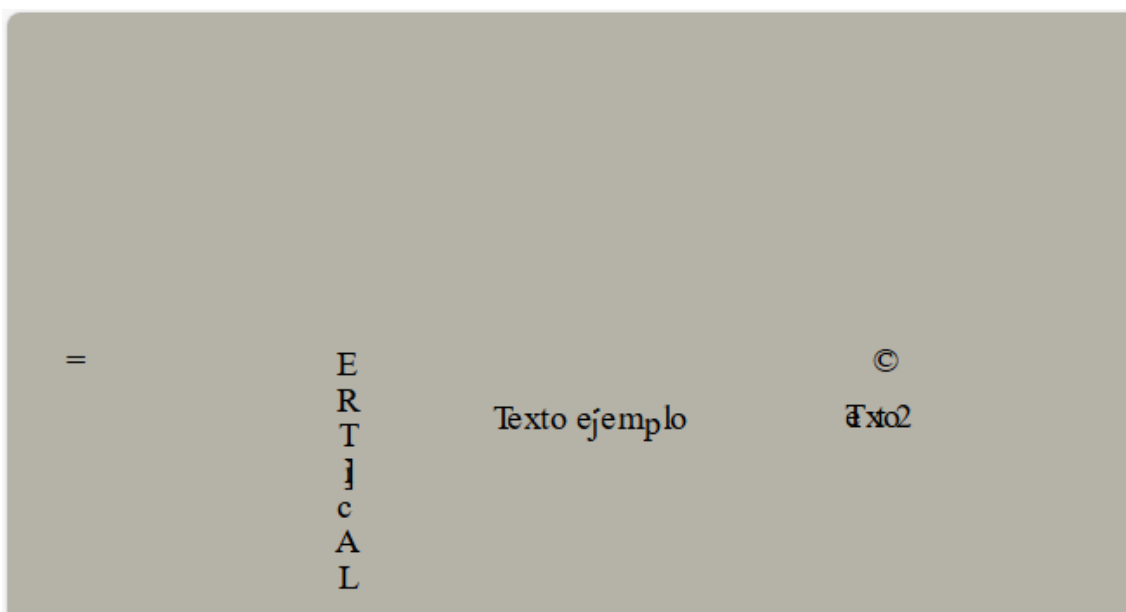


fig4.1.c – Resultat de Pytesseract

Es pot observar com:

- El text molt petit ha desaparegut, no s'ha detectat.
- El text vertical ha estat detectat gairebé perfecte, però la V no ha estat detectada, la I ha tingut alguna mena d'error i la C és minúscula.
- El text ben definit ha estat detectat correctament però ho ha detectat de forma individual cada caràcter.
- El text amb número ha passat pel mateix que l'anterior.
- Hi ha petits caràcters solts pel llenç, fallades de detecció i detectats com a caràcters inexistents originalment.

Es van fer moltes proves canviant diferents paràmetres dels dos models. Però a la conclusió que s'hi va arribar és que EasyOCR era millor per a la detecció de textos horitzontals en gairebé tots els casos. Mentre Pytesseract era millor per a la detecció dels textos verticals, però per als horitzontals era un despropòsit.

El que és bo de tots dos OCR i que sembla que és una cosa estandarditzada en diversos OCR que també van ser considerats (com Keras), només que no tan exhaustivament, és que et tornaven les dades amb: el text detectat, els quatre punts de coordenades en què s'ha trobat i una puntuació de confiança del resultat.

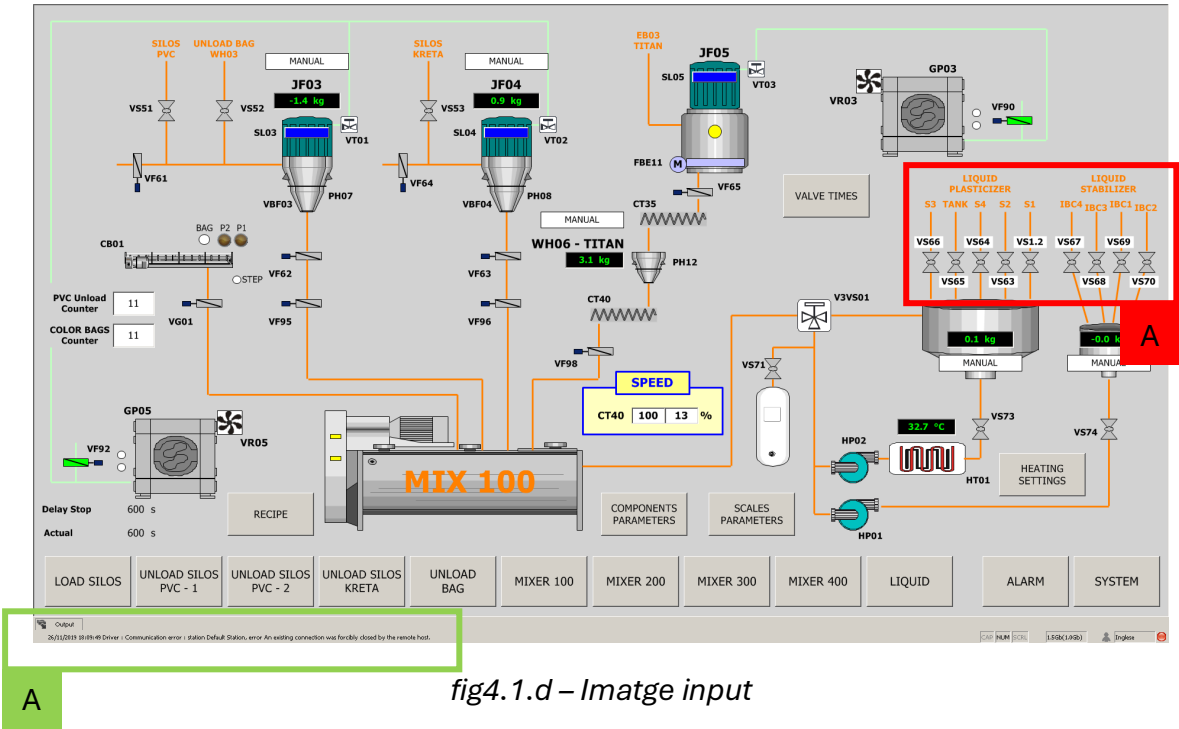
Això és molt interessant ja que, amb el text i els punts de coordenades, podem emprar la llibreria Svgwrite per col·locar directament en un SVG amb les dimensions originals de la imatge el text detectat.

La confiança és una altra dada molt important, però en aquest estat del procés l'estem desestimant.

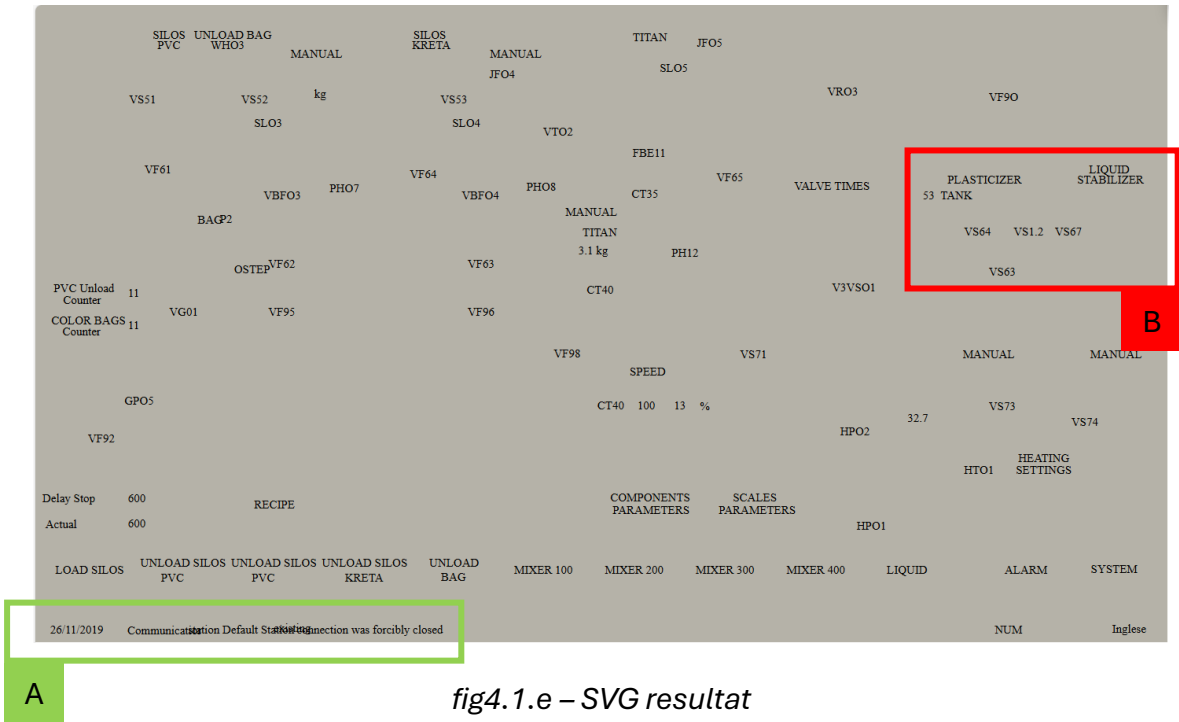
A més d'això, l'EasyOCR inclou un paràmetre per indicar l'idioma del text que es vol detectar. Això és molt útil també.

Així doncs, es va avançar en el projecte havent escollit EasyOCR com a model OCR amb què actuar. Però aquesta vegada s'havia de provar en un sinòptic de veritat i veure com responia.

La imatge escollida per a la prova va ser la següent [fig4.1.c] i els motius van ser que té multitud de textos, tots d'una mida semblant, i per als que tenen mides dispars no tenen diferències gaire exagerades. També hi ha textos que contenen números. A més d'això, la imatge és en color, de manera que hi ha textos de diferents colors i textos pintats sobre colors diferents.



Els resultats van ser els següents [fig4.1.e]:



Si bé a simple vista pot semblar un bon resultat, i ho és, hi ha parts que falten com a la zona B marcada, i altres que queden solapades com per exemple la zona A.

EasyOCR estava bé i complia les expectatives, però necessitàvem més, per més proves que vam fer, per més preprocessament que instéssim realitzar a la imatge, la detecció no millorava. Intentem binaritzar la imatge segons diferents thresholds per tal de ressaltar més els textos i minimitzar l'impacte de les formes [fig4.1.f], així com passar la imatge a escala de grisos per observar els resultats [fig4.1.g] .

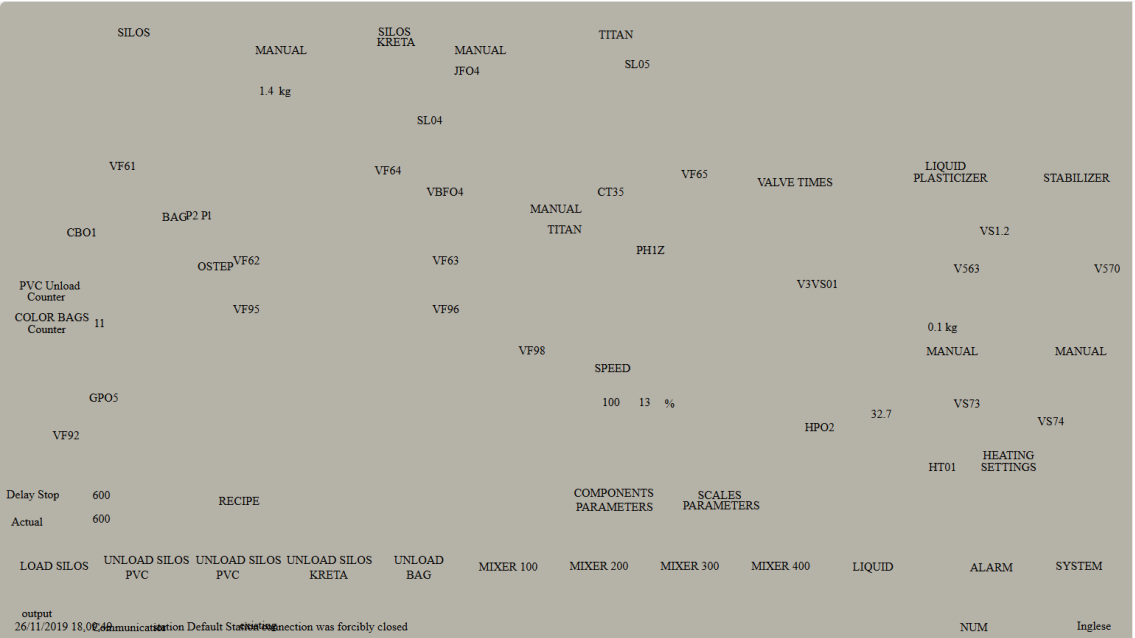


fig4.1.f – SVG resultat de la imatge binaritzada

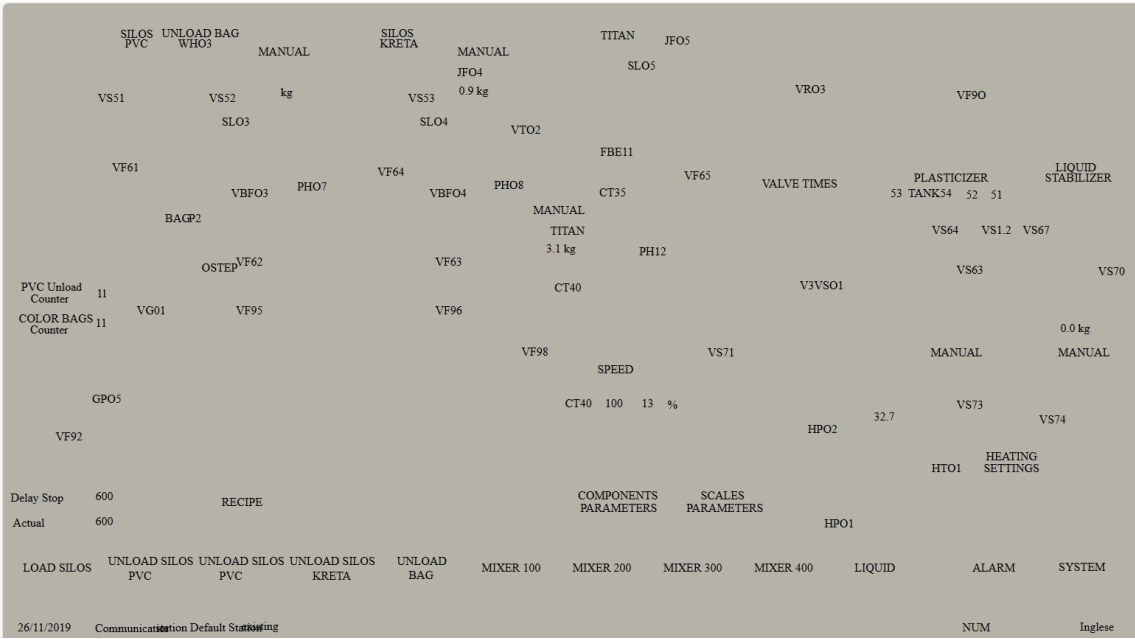


fig4.1.g – SVG resultat de la imatge en escala de grisos

Com es pot observar, sí que és cert que hi ha una certa millora respecte a l'original. Però únicament en el cas de la imatge en escala de grisos. .

La conclusió a què vam arribar és que necessitàvem més, un OCR preentrenat exclusivament en la detecció de textos en diagrames 2d.

Després de molta recerca trobem un OCR especialitzat en la lectura d'arxius d'estil PDF i en llibres escanejats. El nom d'aquest OCR és Surya-OCR.

Així que fem proves i analitzem els resultats [fig4.1.h].

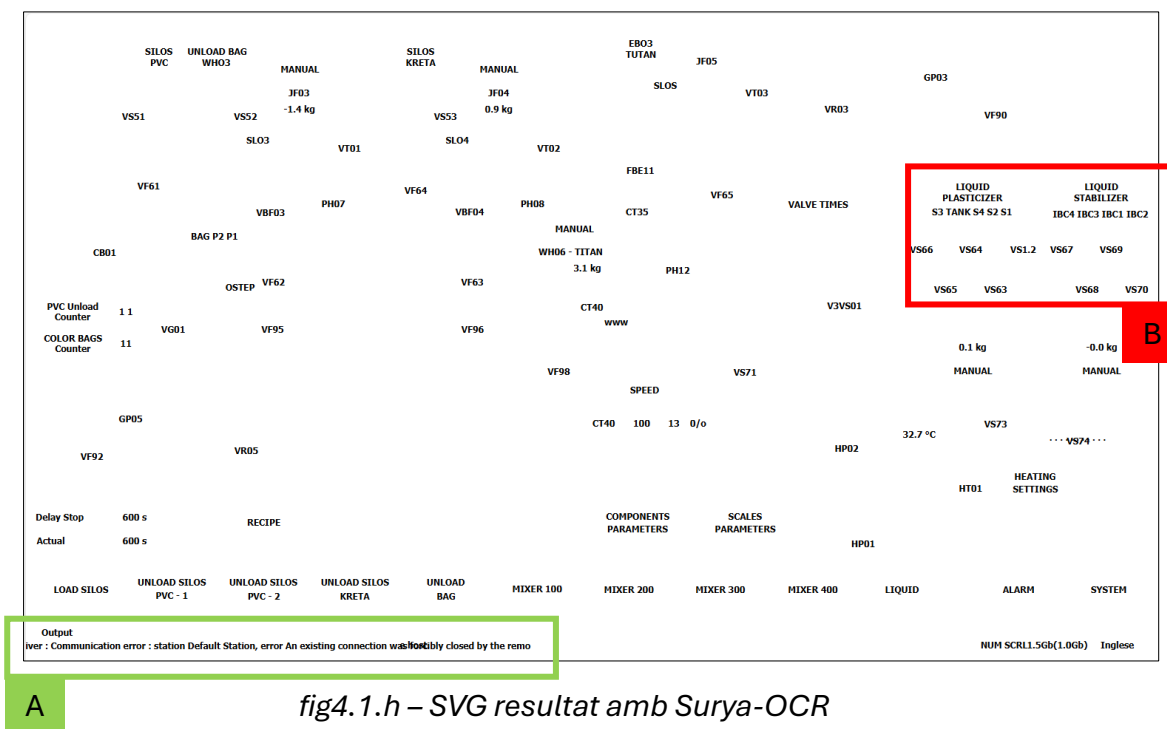


fig4.1.h – SVG resultat amb Surya-OCR

Es pot observar que aquestes zones on anteriorment no es detectava res, ara sí que ho feien. Surya-OCR s'adaptava molt millor a l'estil de diagrama que fèiem servir i millorava per molt la detecció dels textos a la Zona B.

Es poden veure encara algunes petites errades a la Zona A i en alguna altra. Però la millora era tan gran que vam prendre la decisió de quedar-nos amb aquest darrer model.

A partir d'aquí, vam fer una anàlisi de les confiança de cada text, marcant amb un color blau la Bounding Box d'aquelles que tingueren una confiança superior al 0.8 i en groc aquelles que superessin el 0.72, que és on vam detectar després de diverses proves que havia soroll. Tot el que no arribés a 0,72 va ser considerat vermell i, per tant, text no fiable. A la imatge següent es pot veure el resultat d'aquest procés [fig4.1.i].

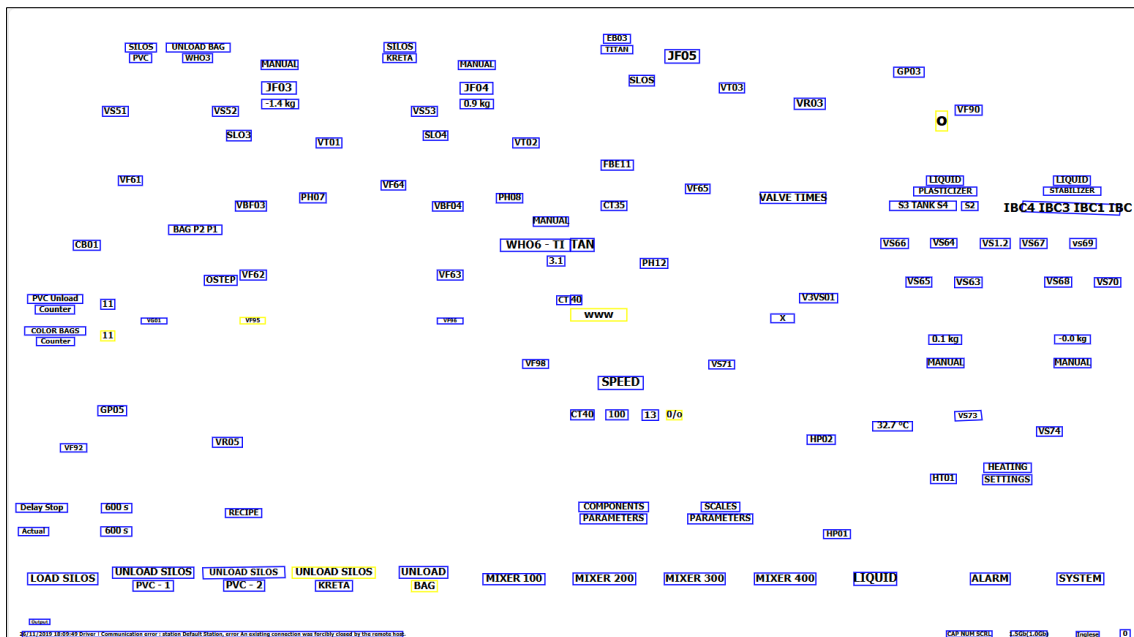


fig4.1.i – SVG resultat amb Surya-OCR, Bounding Boxes ressaltades

A partir d'aquí. Hem observat també que tot i que l'OCR sigui el mateix, millora la seva detecció en executar-se a trossos d'aquesta mateixa imatge més petits, malgrat que el contingut sigui el mateix.

Això és una millora que segueix en procés i que serà explicada detalladament al següent informe de progrés.

-- Fins aquí, de moment, el progrés. --

Actualment estic treballant en aquesta millora de detecció del text, però amb el temps que tinc, probablement decideixi seguir amb la planificació i no treballar-hi de moment. A l'espera de les festes de Nadal o algun pont on tingui més temps per treballar en aquells punts no prioritaris.

5. Planificació

La planificació s'ha vist altament alterada a causa del canvi de model de treball i degut també a una major comprensió i acotament en els requisits del projecte.

Així doncs, sense ànim de ser tan exacta quant a treball setmanal fa referència a l'informe inicial, la nova planificació és la següent:

Octubre:

- Creació de l'informe inicial.
- Investigació i creació del model de treball.
- Cerca d'OCR, implantació i creació d'eina d'exportació a SVG.

Novembre:

- Setmana 1:
 - Creació i implantació de millores de Surya-OCR.
 - Creació de l'informe de progrés I.
- Investigació sobre YOLO.
- Implantació de YOLO al projecte.
- Implantació de possibles millores de YOLO.

Desembre:

- Investigació per aplicar mètodes de reconeixement de camins.
- Setmana 2:
 - Aplicació dels mètodes trobats de reconeixement de camins.
 - Creació de l'Informe de Progrés II.
- Millores al sistema de reconeixement de camins.

Gener:

- [En cas de donar temps] Aplicar totes les millores que s'han apuntat durant el procés de creació del projecte.
- Encapsular-ho tot en un mateix programa.
- Creació del dossier final.

Febrer:

- Setmana 1:
 - Creació de la presentació del TFG.

6. Bibliografia

La bibliografia serà afegida de la manera més professional possible a l'Informe de Progrés II.