Automatització d'esquemes per a la Indústria 4.0 amb Visió per Computador  
Necessitat  
En l'àmbit creixent de l'automatització industrial, existeixen els anomenats SCADA. Un SCADA (Supervisory Control and Data Acquisition) és un esquema l'objectiu del qual és mostrar en temps real els estats pels quals estan passant les màquines que executen els processos d'una fàbrica o planta; i estan pensats perquè els operaris puguin observar en tot moment per quin punt estan passant tant els materials com les màquines que els processen dins del cicle. Aquests processos poden ser des d'una cadena de muntatge de peces per a diversos àmbits fins a cadenes de producció d'aliments, etc. Hi ha SCADA per a tot tipus de cadenes de producció. Cada màquina observada dins d'una pantalla SCADA és, en termes informàtics, un objecte. Cada objecte té diversos atributs com poden ser: Nom, Senyal d'alarma, Connexió a màquina A, connexió a màquina B, etc. Cada màquina està connectada a un PLC (Controlador lògic programable), que actua com el “cervell” dins del procés industrial i que conté el codi de funcionament de cada màquina, i actua segons les condicions que té cadascuna per assolir l'objectiu que s'espera d'ella.  
El cas és que molts d'aquests sistemes SCADA tenen molts anys, ja que es van crear fa dècades o fins i tot més. Aquestes pantalles es van crear en el seu moment amb l'objectiu abans esmentat. I moltes d'elles es van fer en sistemes o programes que avui dia, amb l'evolució constant en el sector de la Indústria 4.0, ja no són òptims o eficients. És per aquest motiu que en els últims anys, les empreses propietàries d'aquests SCADA busquen millorar les seves pantalles i esquemes amb nous programes que els puguin proporcionar nous avenços tecnològics. Ja sigui perquè els programes antics no tenen tantes opcions com els nous o perquè no ofereixen les funcionalitats que tenen els nous programes que van sortint.  
Quin és el problema? Que avui dia el format que s'ha estandarditzat en la majoria dels casos és el .XML.

És aquí on entra la necessitat d'un sistema que, a partir de diagrames creats originalment en formats diferents de l'.XML, pugui resoldre la necessitat d'utilitzar empleats per refer exactament les mateixes formes i patrons en un format .XML. Aquest és l'objectiu d'aquest treball. Crear una eina que, a partir d'imatges JPG o PNG, pugui recrear aquestes mateixes formes en .XML, estalviant així temps i esforç als empleats perquè puguin dedicar-se a altres tasques.

Objectiu  
Crear un programa basat en un model o xarxa neuronal que, donada una imatge en format JPG o PNG, sigui capaç d'identificar i replicar totes les formes en un fitxer .XML.

# Paraules clau:

* Computer Vision
* Computer Vision Model
* Pattern Recognition
* SCADA
* Automation
* Industry 4.0

Tasques  
Cal indicar alguns dels passos que s'han de seguir per poder realitzar aquest sistema.

* Creació d'un banc gegantí d'imatges de formes ja etiquetades amb la seva respectiva traducció a .XML.
* Creació d'un model que permeti detectar formes en una imatge d'entrada d'un diagrama i que sigui capaç de traduir-les a .XML. Aquest model seguirà els passos següents:
  + Detectarà les formes de l'input.
  + Buscarà les seves homònimes segons el seu entrenament.
  + Recrearà les formes detectades en .XML com a output.
  + Buscarà si hi ha formes que es poden agrupar com a formes més complexes. P. ex.: quatre segments rectes que tinguin punts en comú que facin de límits podran ser denominats quadrats. S'agruparan com a tals, estalviant així "espai" en el diagrama final. Aquest pas és important perquè, quan es trobi amb quadrats que connecten amb línies o fletxes, ha de ser capaç de distingir què és un quadrat i què és una fletxa que apunta a un altre lloc.
  + També ha de ser capaç de detectar text i indicar-ho com a tal.
* El model ha de ser capaç de treballar amb imatges en major o menor resolució. Algunes imatges es veuran malament a causa de la qualitat de la imatge i perquè poden abastar diagrames realment grans.

Dificultats  
La creació del conjunt de dades train/test pot ser molt difícil. Tinc accés a nombroses pantalles SCADA, però he de fer un etiquetatge manual per distingir quines parts són importants i quines no. S'haurà de mantenir un estàndard en les formes que etiquetaré, i aquest estàndard podria ser que totes mantinguin el rang de colors en escala de grisos. A més a més, he d'indicar les formes més comunes que es poden trobar en els diagrames: quadrats, rectangles, cercles, fletxes, bombes, etc.

És important que el model pugui discernir quines formes pertanyen a una cosa i quines a una altra. En els models, moltes d'aquestes formes estaran interconnectades entre si. P. ex.: una bomba seguida d'una fletxa que surti de la mateixa bomba fins a un tanc d'aigua. Totes les formes estaran connectades, però el model ha de saber quina forma té cada cosa i agrupar-les en el que correspongui a cada forma. Al final, en el diagrama .XML resultant no poden haver-hi milions de línies, sinó formes acotades. Si alguna forma conté una línia al seu interior, el model ha de ser prou intel·ligent per saber que dins d'aquesta forma hi ha una línia de més i afegir-la com a part d'aquest grup.

L'entrenament del model pot arribar a ser molt difícil. Caldrà veure si existeixen models previs de detecció de formes. He de crear una eina que tradueixi del format en què es facin els esquemes de formes a .XML.

Planificació:  
Octubre:

* 7 – 13 buscar models de detecció de formes
* 14 – 20 buscar un dataset adequat per a l'algorisme
* 21 – 27 comprovar el funcionament del model i buscar algorismes i mètodes que millorin el seu funcionament
* 28 – 3/11 Ajustaments de paràmetres i entrenament

Novembre:

* 4 – 10 crear informe de progrés 1
* 10/11 Informe de progrés 1
* 11 – 17 buscar millores de l'algorisme
* 18 – 24 implementar millores de l'algorisme
* 25 – 1/12 comprovar amb altres datasets

Desembre:

* 2 – 8 crear o buscar un traductor .csv (o el format que sigui) a .XML
* 9 – 15 crear informe de progrés 2
* 15/12 Informe de progrés 2
* 16 – 22 encapsular tot en un sol programa
* 23 – 29 revisió d'errors
* 30 – 5/1 vacances

Gener:

* 6 – 19 crear informe final. Passar-lo a LaTeX
* 19/01 informe final
* 20 – 26 crear presentació

Febrer:

* Presentació.

# Bibliografia:

Maria Dziuba, Ivan Jarsky, Valeria Efimova, Andrey Filchenkov. (2023, 10 Juny). *Image Vectorization: a Review.* Arxiv.org, Cornell University.   
https://arxiv.org/abs/2306.06441

Albert Obradors Caro. (2024, 15 Maig). *Implementació de la vectorització de capes ràster per la visualització amb la tecnologia "vector tiles" a la plataforma ARGOS*. Universitat Politècnica de Barcelona.   
https://upcommons.upc.edu/bitstream/handle/2117/414399/183491.pdf

Vage Egiazarian, Oleg Voynov, Alexey Artemov, Denis Volkhonskiy, Aleksandr Safin, Maria Taktasheva, Denis Zorin, Evgeny Burnaev. *Deep Vectorization of Technical Drawings*. Skolkovo Institute of Science and Technology & New York University.   
https://www.ecva.net/papers/eccv\_2020/papers\_ECCV/papers/123580579.pdf

Xu Ma, Yuqian Zhou, Xingqian Xu, Bin Sun, Valerii Filev, Nikita Orlov, Yun Fu, Humphrey Shi. (2022). *LIVE: Towards Layer-wise Image Vectorization. Git Hub. Northeastern University.*   
https://ma-xu.github.io/LIVE/