

Automatiseret sortering i reverse logistics



Gruppe 4

| Studerende | Studienummer |
|-------------------------------|----------------|
| Anton Roll | S235679 |
| Hector Urioste | S235675 |
| Jonathan Rasborg | S223644 |
| Magnus Ditlev Sørensen | S235665 |

Repository: https://github.com/s223694/COLORSORT_2.git

Indhold

| | |
|--|----|
| <i>Problem</i> | 3 |
| <i>Løsning</i> | 4 |
| <i>Eksisterende løsninger</i> | 6 |
| <i>Design & implementering</i> | 8 |
| System flowchart: | 8 |
| Flowchart for robottens arbejdsgang: | 9 |
| Class boxes: de vigtigste funktioner og ansvar | 10 |
| Valg af designbeslutninger | 12 |
| Valg af farve-klassifikation (vision) | 12 |
| Sekventiel sortering (kontrol flow) | 13 |
| Ordrelogik og batch-sortering pr. Farve | 13 |
| Robusthed via “unknown” håndtering | 14 |
| <i>Test demonstration</i> | 15 |
| Testcases | 15 |
| Begrænsninger | 15 |
| <i>Erfaringer og læring: programmering & industriel automation</i> | 16 |
| <i>Diskussion & konklusion</i> | 17 |
| <i>Brug af LLM</i> | 19 |

Problem

En supply chain der kan håndtere uforudsete returnmodtagelser og udnytte værdien i returflows bliver stadig vigtigere i takt med stigende krav til cirkularitet, ressourceeffektivitet og dokumentation. I reverse logistics opstår der typisk et beslutningsproblem for hver enkelt return: Hvad er det? Hvilken tilstand er det i? Og hvilken rute skal det på? I praksis udføres en stor del af denne sortering manuelt¹, fordi returvarer ofte er uensartede, ankommer i blandede batches og ikke passer ind i standardiserede outbound-processer. Det betyder at returflowet ofte får en høj procesomkostning pr. enhed, lavere gennemløb og større variation i kvaliteten af beslutningerne.

Konsekvensen er at store mængder materialer, herunder plast, ender i blandede affaldsstrømme eller i for "lavværdi"-ruter, selv når en del af materialet potentielt kunne have været genanvendt eller udnyttet bedre. Når retur- og sorteringsprocesser bliver for dyre, er for langsomme eller for upræcise, bliver den optimale cirkulære behandling i praksis fravalgt, fordi det ikke kan betale sig. Små komponenter er særligt udfordrende, de er ofte billige i stykpriis, men dyre at håndtere manuelt, fordi tiden til identifikation og sortering ikke falder proportionalt med objektets størrelse. Derudover er små emner sværere at automatisere med "faste" løsninger, fordi variation i form, placering og overflade kan give usikker detektion og kræver fleksible systemer der kan håndtere både kendte og ukendte tilfælde.

Derfor fokuserer projektet på at reducere barrieren i returleddet gennem hurtig og fleksibel sortering af små emner i en kontrolleret celle, hvor processen kan standardiseres, testes og dokumenteres. Målet er at demonstrere kernen i et sådan system: et robust beslutningsflow fra detektion og klassifikation til fysisk sortering og efterfølgende registrering af resultatet, som kan danne grundlag for en mere skalerbar løsning i en industriel kontekst.

¹ <https://knappusa.com/contenthub/automating-reverse-logistics/>

Løsning

Vi vil demonstrere en sorteringsløsning, der kombinerer et fysisk OT (operational technology) setup med et softwarelag, som understøtter adgangskontrol og sporbarhed. Formålet er ikke alene at udvikle selve sorteringsprocessen, men også at vise hvordan en sådan løsning bør struktureres, så den er realistisk i en industriel kontekst hvor ansvar og dokumentation er centrale krav.

Løsningen organiseres omkring et brugerflow med rollebaseret adgang. Systemet starter på en loginside, hvor brugerens oplysninger valideres mod en database. Herefter tildeles brugeren adgang til enten en standard brugerflade eller en administrativ brugerflade afhængigt af rolle. Når en autoriseret bruger starter sorteringen, sendes instruktioner til robotstyringen/PLC, som aktiverer den fysiske proces. Kameraet indgår som vision modul, der understøtter klassifikation og dermed beslutningen om separation, således at information behandling kobles direkte til fysisk handling.

Et centralt element i løsningen er logging og historik, der skaber et spor af hændelser (fx login, opstart, stop og sorteringsresultater). Dette muliggør dokumentation af "hvem der gjorde hvad hvornår", og giver samtidig et datagrundlag for efterfølgende evaluering af performance, fejrate og driftstabilitet, hvilket er væsentligt i industrielle miljøer.

Sorteringsbeslutningen baseres på tre relevante dimensioner:

1. **Variant** (størrelse/grundform)
2. **Kvalitet** (synligt snavs eller markante skader)
3. **Materiale** (simuleret): Da vi ikke kan identificere plasttyper (PP/PE/ABS) med de tilgængelige sensorer, simulerer vi materialefraktioner ved at anvende farve som indikator. Det gør det muligt at demonstrere logikken i materialeseparation i reverse logistics uden at påstå, at materialet måles direkte.

Output: de tre fraktioner

Outputtet struktureres i tre praktisk anvendelige fraktioner, der svarer til strategier i en cradle to cradle/cirkulær designlogik, hvor værdien søges fastholdt på det højest mulige niveau:

A) Reuse (direkte genbrug i systemet)

Emner der klassificeres som intakte og i tilstrækkelig stand, og som samtidig er funktionelt egnede til direkte genindsættelse, separeres til en reuse-fraktion. Denne fraktion repræsenterer den mest værdibevarende disposition, fordi emnet kan bringes tilbage i kredsløb uden rework eller procestrin. I reverse logistics svarer det til at undgå værdiforringelse og unødvendige omkostninger ved at prioritere direkte genbrug.

B) Resale (second-life/videresalg)

Emner der er funktionelle og uden kritiske skader, men som ikke matches optimalt til intern reuse (fx kosmetisk kvalitet, overskud eller krav til intern standard), separeres til en resale-fraktion. Her fastholdes værdien ved at bringe emnet tilbage i brug gennem sekundære markeder eller andre second-life flows, uden at gå via oparbejdning.

C) Remanufacturing (rework/refurbish)

Emner der ikke bør reuse eller resale pga. Synligt snavs, slid eller markante (ikke destruktive) skader, separeres til en remanufacturing-fraktion. Formålet i projektet er ikke at udføre selve remanufacturing-processen, men at demonstrere den afgørende forudsætning for den; korrekt og konsistent fraktionering samt sporbarhed, så emner kan sendes videre til downstream rework (fxrens, reparation, inspektion) og derefter bringes tilbage til en anvendelig specifikation.

Hierarki (value-retention)

De tre fraktioner afspejler et hierarki, hvor reuse og resale prioriteres over remanufacturing, fordi de typisk bevarer mest værdi og kræver mindst energi og bearbejdning. Remanufacturing anvendes, når værdien stadig kan fastholdes på produktniveau, men kræver et rework-trin.

Eksisterende løsninger

Løsning 1: BEUMER PostNord parcel centre (case):
<https://www.beumergroup.com/reference/courier-express-parcel/postnord-parcel-centre/>

Relevans for vores projekt:

Et industrielt eksempel på en optimeret sorteringsproces med singulation, scanning og chutes, som reducerer manuel håndtering. I casen kommer pakker ind i "bulk" via automatiske tipper-containere og transportbånd. Herefter omdannes bulk-flowet til én stabil strøm, hvor pakkerne bliver jævnt adskilt og alignede. Systemet detekterer overlap/dobbelte samt pakker der er "out-of-spec" (fx for høje, lange, tynde eller overvægtige) og sorterer dem fra til en separat linje, så hovedflowet forbliver stabilt. Derefter føres pakkerne ind i en cross-belt loop sorter, hvor identifikation sker via kamera-scanning og OCR, og hvor software/SCADA optimerer flowet på tværs af systemet. Til sidst afleveres pakkerne til dispatch via et stort antal chutes (fx spiral chutes, boom conveyors og direkte chutes), så manuel håndtering minimeres.

Pros:

Løsningen skaber et stabilt og forudsigeligt flow gennem singulation og automatisk håndtering af dobbelte/out-of-spec emner. Flowet optimeres på tværs af systemet (system-wide) via software/SCADA, hvor input kan tilpasses kapaciteten i downstream- processer. Derudover minimerer løsningen manuel håndtering og tunge løft og har mange dispatch-muligheder via forskellige chute-typer, hvilket understøtter høj kapacitet og effektiv drift.

Cons:

Løsningen er dyr, pladskrævende og skaleret til et stort pakkecenter. Den passer ikke 1:1 til vores setup, da den primært arbejder med pakker og labels (OCR), mens vores vision i MVP'en handler om klassifikation baseret på farve/form. Casen har også fokus på transport- og sorteringsflow frem for robot pick-and-place, som vi bruger i vores celle.

Løsning 2: Sorteringssystem apotek

<https://www.sn.dk/art4352357/naestved-kommune/apotekerens-nye-robot-holder-styr-paa-30000-aesker/>

Apoteksrobotten fungerer som et automatiseret lager- og plukkesystem, der opbevarer medicin kompakt og struktureret. Når en medarbejder bestiller en vare, finder robotten automatisk den korrekte pakke, scanner den og leverer den til udleveringspunktet. Det reducerer både ventetid og fejl og frigør tid for personalet til rådgivning og kundeservice.

Relevans for vores projekt:

Løsningen er relevant, fordi den demonstrerer automatiseret håndtering og sortering af mange små enheder med høj præcision og sporbarhed. Den viser, hvordan robotter kan overtage monotone lager- og plukkeopgaver og integreres tæt med informationssystemer. Det minder om vores projekt ved fokus på korrekt klassifikation, stabil drift og reduktion af manuel håndtering.

Pros:

Det automatiserede sorteringsssystem øger effektiviteten i apotekets interne logistik ved at sikre hurtig og præcis håndtering af medicinpakninger. Systemet kan lagre og sortere op mod 30.000 æsker på begrænset plads, hvilket optimerer lagerudnyttelsen. Reeltidsregistrering af lagerbeholdning reducerer risikoen for udsolgte varer og forbedrer genbestillings processen. Samtidig mindskes manuelle plukfejl, hvilket øger patientsikkerheden. Automatiseringen reducerer den fysiske belastning for medarbejderne og forbedrer ergonomien. Endelig bidrager standardiserede arbejdsgange til en mere stabil og forudsigelig drift.

Cons:

Implementeringen af et automatiseret sorteringsssystem kræver en betydelig initial investering, hvilket kan være en økonomisk udfordring. Systemet medfører desuden øgede krav til teknisk vedligeholdelse og IT-support. Ved systemnedbrud kan apotekets drift blive sårbar, da arbejdsgangene i høj grad er afhængige af teknologien.

Design & implementering

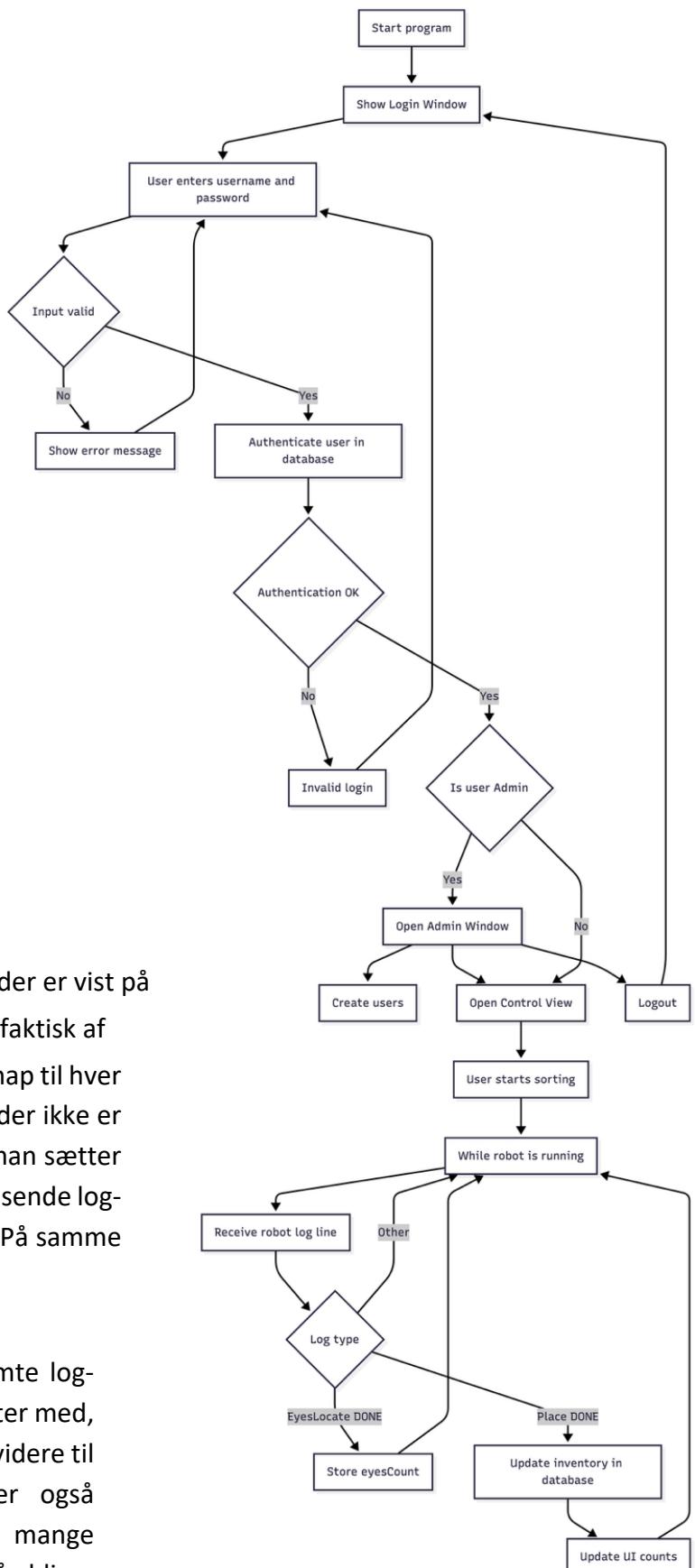
System flowchart:

På figuren til højre, vises et flowdiagram for hele systemet, som består af en GUI lavet med Avalonia, en tilhørende database, og en Universalt Robot, med et monteret kamera.

Det er designet således, at der er 2 roller, som man kan logge ind med, "Admin" og "User". Hvis man bliver genkendt som "User", vises kontrolpanelet med det samme. Hvis man derimod bliver genkendt som "Admin", tilgår man Admin-panelet, hvor man kan oprette andre Admins, og Users, men også tilgå kontrolpanelet.

I kontrolpanelet, er der flere funktioner, end der er vist på figuren. Blokken "User starts sorting", består faktisk af 4 forskellige funktioner. Vi har én sorteringsknap til hver farve, og så en der sorterer alle farver, indtil der ikke er flere. Systemet er automatisk fra øjeblikket man sætter det i gang, da vi har justeret URScriptsne til at sende logbeskeder tilbage til computerens IP-adresse. På samme måde opdaterer vi automatisk og databasen.

Vi har lavet funktioner, som lytter til bestemte logbeskeder, såsom "END", som hvert script slutter med, hvilket bestemmer at robotten enten skal gå videre til næste farve eller stoppe helt. Derudover også "EyesWorkpCount =", som fortæller hvor mange enheder af den bestemte farve, som så bliver opdateret i databasen.

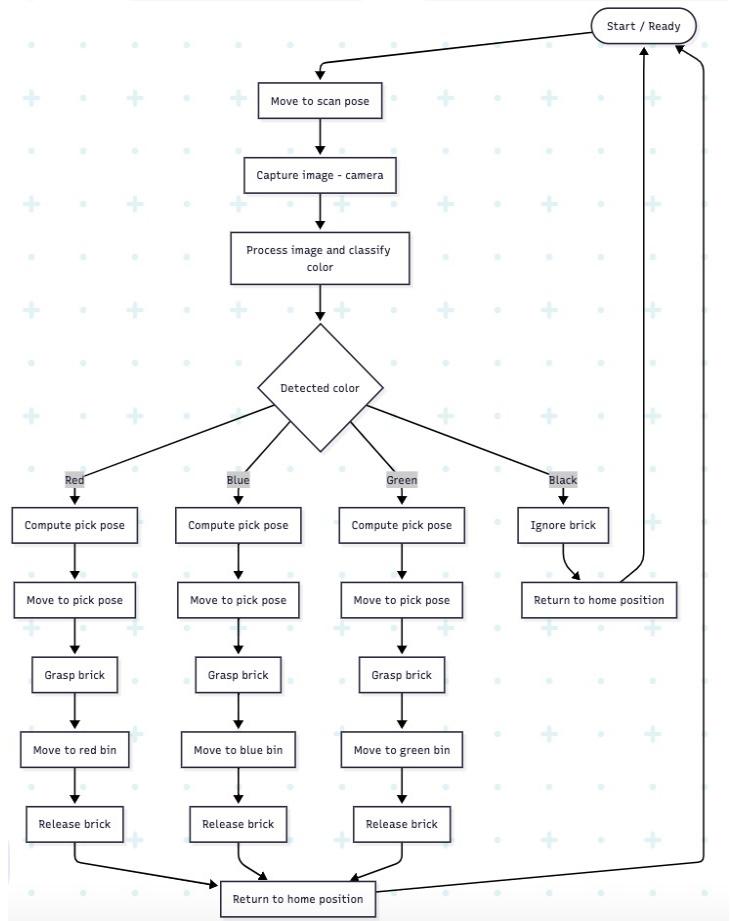


Flowchart for robottens arbejdsgang:

Brugeroplysninger, adgangsniveauer og logfiler håndteres og lagres i en central database. Efter login præsenteres brugeren for en grafisk brugergrænseflade (GUI), hvor systemstatus, aktuelle processtrin og eventuelle fejlmeldelser vises.

GUI'en fungerer som det primære interface mellem operatøren og robotcellen og anvendes til at starte og overvåge sorteringsprocessen.

Når processen startes via GUI'en, bevæger robotten sig til en foruddefineret scan pose, hvor kameraet aktiveres, og et billede af arbejdsmrådet bliver taget. Billedet behandles af systemet, som identificerer objektets position og klassificerer dets farve.



Resultatet af farveklassifikationen evalueres i beslutningsblokken "Detected color", hvorefter systemet vælger den passende handlingssekvens. For farverne rød, blå og grøn beregnes en pick pose, hvorefter robotten udfører en pick-and-place-operation og placerer kloden i den tilsvarende sorteringsbeholder.

Når kloden er frigivet, returnerer robotten til sin hjemmeposition. Hvis den detekterede farve er sort, udføres der ingen manipulation. Kloden ignoreres bevidst, da den ikke indgår i sorteringskriterierne, og robotten returnerer direkte til hjemmepositionen.

Efter hver gennemført cyklus opdateres systemets tilstand i databasen, hvor relevante procesdata såsom detekteret farve og bruger-ID kan logges. Disse data kan efterfølgende anvendes til statistik, sporbarhed og systemevaluering via GUI'en. Herefter vender systemet tilbage til Start / Ready, hvor det afventer næste kommando fra brugeren.

Class boxes: de vigtigste funktioner og ansvar

| |
|--|
| LoginWindow |
| <p>Ansvar:</p> <ul style="list-style-type: none">- Vise login UI- Autentificere bruger- Navigere til Admin eller Control |
| <p>Metoder:</p> <ul style="list-style-type: none">- Login_Click()- EnsureDbAndAdminAsync() |
| |

| |
|--|
| AdminWindow |
| <p>Ansvar:</p> <ul style="list-style-type: none">- Oprette nye brugere- Vælge roller (Admin/User)- Navigere til ControlView eller Logout |
| <p>Metoder:</p> <ul style="list-style-type: none">- Create_Click()- OpenControl_Click()- Logout_Click() |
| |

| |
|---|
| ControlView |
| <p>Ansvar:</p> <ul style="list-style-type: none">- Styre robotten via scripts- Modtage og fortolke robot-log- Opdatere inventory automatisk- Håndtere Sort All-sekvens |
| <p>Metoder:</p> <ul style="list-style-type: none">- SendScriptAsync()- TryHandleEyesLocateCount()- TryHandlePlaceDone()- TryHandleRunEnd() |
| |

| |
|--|
| RobotService |
| <p>Ansvar:</p> <ul style="list-style-type: none">- Sende URScript til robotten- Modtage robot-log via TCP |
| |

- Udsende log-events til UI

Metoder:

- StartLogListener()
- StopLogListener()
- SendScriptAsync()

InventoryRepository

Ansvar:

- Læse inventory counts fra database

Opdatere inventory i transaktioner

Metoder:

- GetCountsAsync()
- ChangeCountAsync()

UserRepository

Ansvar:

- Oprette brugere
- Autentificere login
- Sikre default admin-bruger

Metoder:

- AuthenticateAsync()
- CreateUserAsync()
- EnsureDefaultAdminAsync()

PasswordHasher

Ansvar:

- Hashe passwords med salt
- Verificere passwords sikkert

Metoder:

- HashPassword()
- VerifyPassword()

DatabaseService

Ansvar:

- Bestemme database placering
- Initialisere database og tabeller

Metoder:

- InitializeAsync()
- GetConnectionString()

ComponentColor (enum)**Ansvar:**

- Definere tilladte farver
- Sikre typesikkerhed i systemet

Værdier:

- Red
- Green
- Blue

Valg af designbeslutninger

Dette afsnit forklarer de vigtigste designvalg der er blevet lavet i løsningen, samt hvorfor vi har prioriteret specifikke tekniske løsninger i forhold til projektets mål og rammer.

Fokus er på, hvordan vi har gjort MVP'en stabil, demonstrerbar og realistisk i en industriel kontekst, hvor driftsikkerhed, sporbarhed og et kontrolleret flow er afgørende. Klassifikationsmetode, kontrolstrategi og hvordan vi har tænkt robusthed og ordrelogik ind i systemet, vil også blive forklaret og beskrevet.

Valg af farve-klassifikation (vision)

Vi har valgt farve-klassifikation med brug af OnEyes som vores primære klassifikationsmetode, da det er den mest robuste og realistiske løsning inden for projektets rammer, altså tid, hardware og scope. Målet med projektet er at demonstrere selve sorteringsprocessen (detektion -> klassifikation -> separation), og her fungerer farver i praksis som en proxy for materialefraktioner. OnRobot EYES er understøttet af farve- og kontur-detection og er designet til hurtig kalibrering, hvilket passer direkte til vores MVP-krav om en løsning, der kan sættes op og demonstreres stabilt.

I en industriel reverse logistics-kontekts kan materialer (forskellige plasttyper) kræve forskellige downstream-behandlinger, hvor direkte materialegenkendelse typisk kræver mere specialiserede sensorer fx. En spektral måling², som ikke er tilgængelige i vores setup. Derfor bruger vi farver til at demonstrere logikken i fraktioner og beslutningsflow, uden at påstå at vi mäter materialerne direkte.

² <https://www.cimbria.com/en/products/optical-sorting/industrial-applications>

Sekventiel sortering (kontrol flow)

Robotten er implementeret som en sekventiel proces styret af et simpelt og deterministisk kontrol flow. Det valg er taget for at sikre en simpel adfærd, samt en stabil drift. Når robotten arbejder sekventielt (én handling ad gangen: detect -> classify -> pick -> place), reducerer risikoen for at systemet kommer ud af sync mellem vision, robot og brugerkommandoer. Dette gør systemet lettere at teste, demonstrere og fejlfinde, hvilket er afgørende i en MVP. Den sekventielle strategi understøtter også robust fejlhåndtering. Hvis der ikke findes flere emner i den ønskede klasse, kan processen afsluttes og hvis en handling fejler, kan systemet logge hændelsen og enten retry eller stoppe sikkert. Dette giver en realistisk OT-tankegang, hvordan man får en mere stabil proces med tydelige stoptilstande end en hurtig proces der fejler uforudsigtigt

Ordrelogik og batch-sortering pr. Farve

I demonstrationen udfører systemet en ordre bestående af 2 blå, 1 grøn og 1 rød, hvor vi også har valgt at lægge en prop ind mellem, for at vise, at det kun er farverne der bliver detektere.

Det centrale design valg her er, at ordren afvikles som en kø af ordrelinjer, hvor hver ordrelinje (fx. 2 x Blue) behandles færdig, før systemet går videre til næste farve. Dette betyder dog ikke, at robotten tager ét billede og handler ud fra det billede. Tværtimod tager OnRobot EYES et nyt billede mellem hvert pick, men vi holder stadig logikken simpel ved at lade systemet blive ved med den ønskede farve, indtil det givne antal er opnået. I en industriel kontakt med bedre robotter, ville dette oftere være ud fra et enkelt billede og evt. Flere robotarme der håndterer sorteringen.

Dette giver tre klare fordele i MVP'en:

1. **Deterministisk adfærd:** Vi ved præcist, hvilken farve systemet leder efter på hvert tidspunkt. Dette gør det hurtigt og klart at identificere fejl; "bør vi kalibrere den farve bedre?", "er der dårligt lys her".
2. **Let validering:** Det er tydeligt i vores MVP, at systemet først er færdiggjort når begge de blå klodser er samlet op og derefter går videre til grøn og rød. Derudover bliver dette også registreret i databasen og GUI'en, hvor det er muligt for os, at overvåge processen i realtid og efter.
3. **Mindre kompleksitet i styringen:** Vi undgår at skulle optimere rækkefølgen dynamisk eller håndtere blandede skift i logik for hvert enkelt emne. Dette gør at vi reducerer fejlmuligheder mellem processen (GUI -> vision -> robot).

Robusthed via “unknown” håndtering

I et fysisk sorteringssetup vil der næsten altid opstå situationer, hvor der ligger objekter i scene, som ikke er en del af sorteringslogikken. For at gøre MVP'en mere robust, har vi derfor bevidst valgt et “unknown” - princip: systemet udfører kun pick-and-place når der detekteres en af de definerede farver/klasser. Objekter udenfor disse klasser ignoreres og bliver liggende, hvilket reducerer for fejl-pick og uforudsigelig adfærd.

Dette demonstreres bevidst i vores video, hvor vi har en prop i scene. Proppen fungerer som et kontrolleret “unknown”-objekt og viser at du systemet ikke forsøger at håndtere alt, men blot de farver vi har klassificeret. Dette er et klassisk princip i OT; “hellere undlade at agere end at agere forkert, når klassifikationen ikke matcher inputtet”.

Test demonstration

Dette afsnit beskriver hvordan systemet er blevet testet, samt hvordan den endelige løsning demonstreres i praksis. Formålet med afsnittet er at verificere at løsningen fungerer stabilt end-to-end fra login og rollebaseret adgang til robot-sortering og logging/historik, samt at systemet håndterer ukendte emner på en sikker måde (fail-safe).

Testen er gennemført som en praktisk accepttest i den færdige opsætning. Testen bliver vurderet ud fra følgende kriterier:

- **Least privilege:** En standardbruger kan køre sortering, mens administrative funktioner (fx brugerstyring) er begrænset til en admin.
- **Stabilt flow:** Et helt sorteringsjob kan gennemføres uden at systemet låser eller mister tilstand.
- **Ingen selv-escalering:** Ved almindelig brugeroprettelse er det ikke muligt at opnå admin-rettigheder. Admin kan kun tildels via adminfunktion af en allerede autentificeret admin.
- **Korrekt sortering:** Kendte emner sorteres til korrekt fraktion/placering.
- **Fail-safe:** Genstande der ikke kan klassificeres sikkert, sorteres ikke forkert (heller ingen handling end fejlsortering)
- **Sporbarhed:** Centrale hændelser registreres, så forløbet kan efterkontrolleres (fx start/stop og resultater i historik)

Testcases

Vi har gennemført følgende scenarier og dokumenteret dem via video:

1. [Video 1:](#) End-to-end demonstration af GUI↔robot (sorteringsjob, unknown-håndtering, historik/log).
2. [Video 2:](#) Demonstration af adgangskontrol og brugeroprettelse, herunder at almindelig oprettelse ikke kan give admin-rettigheder.

Begrænsninger

Testen dokumenterer korrekt funktion i den valgte opsætning, men resultater kan påvirkes af miljøforhold (lys/kalibrering). En næste iteration bør inkludere mere systematisk test under varierende betingelser og kvantificering af fejlrater.

Erfaringer og læring: programmering & industriel automation

Vi har i projektet arbejdet med programmering, industriel automatisering og samarbejdet mellem menneske og robot. Dette har givet os en dybere forståelse for de processer, der indgår ved opsætning af teknologi inden for Industry 4.0 og 5.0. Projektet har blandt andet omfattet kalibrering, programmering samt integration af database, GUI og robot.

Derudover har projektforløbet bidraget til en øget forståelse for forskellene mellem traditionelle IT-systemer og OT-systemer. Her stilles der særligt høje krav til driftssikkerhed, cybersikkerhed samt robottens evne til at operere i realtid. Human-robot collaboration har været et centralt element i projektet, hvor vi er blevet bekendt med både de fordele og de udfordringer, der følger med implementeringen af ny teknologi. En af de vigtigste erfaringer, vi tager med os, er forståelsen for, hvordan teknologi kan anvendes af medarbejdere i en produktionslinje til at optimere workflows og dermed øge både produktivitet og fleksibilitet. Dette er afgørende i arbejdet med robotter, hvis primære styrke ligger i deres evne til at gentage præcise bevægelser og processer pålideligt gennem hele produktionslinjen.

Diskussion & konklusion

Opfyldte løsningen vores mål?

Overordnet set, synes vi at have opnået vores mål. Vi har en demonstrerbar MVP, der viser den centrale proces *detektion -> klassifikation -> pick -> place* og som kan køre en ordre på flere farver. I vores video demonstrerer vi en ordre med 2 blå, 1 grøn og 1 rød, samt en prop som "unkown"-objekt. Systemet sorterer farverne i korrekt rækkefølge og registrerer efterfølgende resultatet i databasen, hvilket gør løsningen "industriel realistisk" ift. Sporbarhed og dokumentation og robust overfor objekter udenfor sorteringslogikken.

Et vigtigt succes-kriterie for os, var at demoen var reproducerbar, altså at vores MVP kunne udføre handlinger uden manuelle indgreb undervejs (udover at lave handlinger i GUI'en, selvfølgelig). Derudover bliver resultaterne registreret i databasen og GUI'en, efter endt sortering. Dette skaber sporbarhed, som også var et af vores mål.

Samtidig må vi være ærlige om MVP-niveauet: vores løsning demonstrerer proceslogikken og integrationen mellem GUI, vision, robot og logging, men den er ikke optimeret til høj throughput eller drift i stærkt varierende omgivelser. Det er dog et bevidst designvalg, hvor vi har prioriteret stabilitet og tydelig demonstration fremfor maksimal kapacitet.

Hvad er begrænsningerne ved vores endelige produkt?

- Farve som proxy:** Farveklassifikation demonstrerer fraktioneringen, men ikke "rigtig" materialegenkendelse. Når man har at gøre med plasttyper som PVC, PP osv. Kræver det typisk andre sensorer og mere avancerede kameraer. I praksis i vores case, ville man også inkluderer variabler som form, vægt og overflade.
- Afhængighed af lys/kalibrering:** Vision løsningen er meget følsom overfor belysning, baggrund og kontrast, hvilket sker grundet skygger og reflekser, hvilket gør systemet mindre "plug-and-play". Drift i nye omgivelser ville derfor typisk kræve re-kalibrering.
- Sekventiel afvikling:** Vi kører én handling ad gangen: *detect -> classify -> pick -> place*. Denne proces er stabil og demonstrerbart, men begrænser kapacitet. I en industriel opsætning ville man typisk reducere ventetid ved at adskille flowet og bruge conveyor belts, fx ved at tilføje en buffer/staging-zone, så robotten altid har emner klar og conveyor ikke behøver at stoppe ved hvert pick.
- Begrænset håndtering af uforudsete scenarier:** Hvis Lego overlapper eller delvist er udenfor kameraets zone, kan egenskaberne fra OnRobot EYES blive usikre. Det kan ende i at klodserne ikke bliver detekteret eller giver en forkert farve, grundet genskær/dårligt lys. Dette går ikke i en industriel proces. I praksis kunne man få kameraet til at detekttere en "unknown" og evt. Sortere denne klasse til et bestemt segment/plads.

5. **“Unknown”-håndtering er ikke fuldt industrialiseret:** Vi demonstrerer robusthed i den forstand, at vores robot ikke samler vores prop op. Men hvad gør systemet i praksis, hvis der er et “unknown” produkt? Her kunne man, som beskrevet ovenover, kigge på at lave en process hvor systemet starter med at sortere de identificerbare produkter, som er klassificeret og så bliver de ikke identificerbare sorteret i en bunke.

Hvad ville vi ændre, hvis vi havde mere tid og andre ressourcer?

1. **Skal ikke være muligt at oprette sig som admin ved brugeroprettelse:** Det skal ikke være muligt at oprette sig som admin ved almindelig brugeroprettelse. Standard-registrering skal altid oprette en “user” og admin-rettigheder skal kun kunne tildeles af en allerede autentificeret admin.
2. **Mere robust materialeklassifikation (ud over farve):** Farve er en god proxy til at demonstrere fraktionering i MVP, men næste skridt ville være at udvide med flere features (form/størrelse) og på sigt sensorer, der kan sige noget om materiale (afhængigt af case). Målet ville være færre fejlklassifikationer og bedre generaliserbarhed. Derudover kunne der også tilføjes vægte, som kunne fungere som en del af sorteringsprocessen.
3. **Aktiv unknown-håndtering:** Dette er et klar forbedringspunkt, hvor vi i vores MVP demonstrerer hvordan systemet ikke detekterer de “unkown” men mere bliver der ikke gjort. Havde vi haft mere tid, ville vi have lavet en process, hvor vi starter med at køre de valgte farver, processen som vi viser i vores demo. Herefter ville vi køre et program, som vil blive kalibreret med “outline” i stedet for farvegenkendelse. Dette script ville vi køre til sidst og sortere de “unknown”-genstande i en separat bin. Dette bør selvfølgelig flagges tydeligt i databasen.
4. **Conveyer belt før og efter vores process (MVP):** Conveyer belts kunne være med til at separerer de forskellige klodser efter processen, samt hjælpe med at gøre forarbejde før vores process. Før vores process, kunne den fx. Afveje og sortere i størrelse, da der i en industriel kontekst vil være mange forskellige størrelser af plast. Der ville være små og store objekter, som ikke ville kunne blive samlet op af robotarmen, hvilket vil skabe nødstop.
5. **Bedre status og feedback (OT-venlig):** Vi ville også forbedre vores status- og feedbacklag i GUI’en, så det bliver mere OT-venligt og lettere at overvåge. I stedet for generelle statusbeskeder, kunne sytstemet vise tydelige proces-trin, eksempelvis: “Scanning”, “Found: Blue”, “Picking”, “Placed” og “Done / Failed (med fejstype)”. Dette gør både demonstration og drift mere gennemsigtigt, og det er en klar fordel når man skal fejlfinde, dokumentere og forbedre stabilitet.
6. **Videreudvikling af systemet med tilføjelse af robotarm mere:** En udvidelse af systemet kunne være at tilføje endnu en robotarm, som havde til opgave at tage objekter fra “lagret” baseret på ordre, så systemet blev helt cirkulært i Supplychain termer.

Brug af LLM

I projektet har gruppen anvendt brug af AI som støtteværktøj til udvalgte dele af arbejdsprocessen. LLM'en er brugt til research og sparring i forbindelse med identificering og opsummering af eksisterende løsninger. Derudover forslag til struktur og formuleringer i rapport og præsentation, samt hjælp til fejlfinding og udvikling af kode, herunder forslag til implementeringsmønstre og forklaring af fejlmeddelelser. Alt indhold fra LLM'en er vurderet kritisk og tilpasset ag gruppen, og vi har selv implementeret, testet og valideret den endelig løsning.