

Finn sex fel

Introduktion

Automatiserad visuell inspektion är viktigt inom allt från kvalitetskontroll i tillverkningsindustrin till förarlösa fordon och medicinsk bildanalys. Förmågan att upptäcka skillnader mellan ett "referensobjekt" och ett nytt objekt sparar tid och minskar risken för mänskliga fel.

Syftet med denna uppgift är att implementera, utvärdera och resonera kring en datorseende-lösning som automatiskt identifierar visuella avvikelseger mellan två snarlika bilder. Fokus ligger på problemlösning, val av metoder och resonemang kring resultatet.

Vi förväntar oss att ambitionsnivån på din lösning ska ligga runt 2-6 timmars nedlagd tid. Detta inkluderar implementering, utvärdering och dokumentering.

Uppgift

Skriv ett skript för att identifiera och markera de **6 skillnader** som finns mellan de två bifogade bilderna ([img1.png](#) och [img2.png](#)).

Du ska:

1. **Läsa in** bilderna.
2. **Bearbeta bilderna** för att isolera skillnaderna. Vi föreslår att du använder Python och biblioteket OpenCV, men andra verktyg (t.ex. scikit-image eller PIL) är självklart också tillåtna.
3. **Detektera skillnaderna:** Använd lämplig metod för att hitta var bilderna skiljer sig åt.
4. **Visualisera resultatet:** Markera de funna skillnaderna tydligt, förslagsvis genom att rita bounding boxes eller konturer runt avvikelseerna på en av originalbilderna.
5. **Resonera** kring din lösnings robusthet och begränsningar.

Krav

- **Kodkvalitet:** Din kod ska vara modulär, tydligt strukturerad och kommenterad.
- **Reproducerbarhet:** Det ska gå att köra din kod och få fram samma resultat som du presenterar.
- **Analys:** En kort reflektion (i README eller Notebook) där du beskriver:

- **Ditt tillvägagångssätt:** Hur attackerade du problemet?
- **Metod:** Vilken algoritm/logik använde du?
- **Resultat:** Lyckades du hitta alla 6 felet? Fick du några falska positiva?
- **Framtida förbättringar:** Hur skulle din lösning hantera om bilderna var tagna i olika ljus, vinklar eller upplösningar? Här räcker det att skriva någon rad, du behöver **inte** implementera någon kod.

Inlämning

- Källkodsfiler (**.py** eller **.ipynb**)
- Bifoga gärna den resulterande bilden där felet är markerade.
- README (inkl. körinstruktioner, beroenden/requirements.txt, samt din analys).
- Det går utmärkt att presentera hela lösningen och analysen i en Jupyter Notebook/Google Colab.

Bedömningskriterier

- Funktionalitet och korrekthet
- Struktur, läsbarhet och kodkvalitet
- Förståelse för ML-koncept
- Analys och reflektion