

## Applikationsspecifik datasetkonstruktion för AI/ML på edge-enheter

Vi är intresserade av att undersöka vad som krävs för att ta fram kvalitativa dataset lämpade för träning och verifikation av ML/AI modeller. Syftet bakom det här projektet är att undersöka vilka krav, processer och tillvägagångssätt som bör tillämpas under skapandet av dataset som ska användas för att träna ML/AI modeller till lösa specifika uppgifter. Vi vill få en bättre inblick i vilka datamängder som krävs, vilken "datakvalité" som behövs samt hur datasets ska konstrueras för att lösa specifika uppgifter. Vi vill också få en bättre förståelse för hur ML/AI modeller presterar på MCUs jämfört med modeller som körs i molnet eller lokalt på en dator.

Projektet innefattar följande:

1. Utveckla datainsamlingsmjukvara för en MCU och tillhörande sensor.
2. Definiera metoder för att skapa datasets ämnade för specifika ML/AI uppgifter.
3. Utför datainsamlingsarbete samt efterföljande formatering och hantering.
4. Analysera och välj lämpliga ML/AI modeller baserat på uppgift.
5. Träna utvalda modeller i pytorch/keras/tensorflow och utvärdera prestanda över vanliga ML/AI parametrar.
6. Anpassa den mest effektiva modellen till en ST MCU med STM32Cube.AI flödet.
7. Anpassa datainsamlingsmjukvaran för att mata ML/AI modellen med sensordata i realtid.
8. Konstruera tester för att utvärdera applikationens förmåga att lösa uppgiften.

Utgångspunkten kan vara en enklare uppgift, exempelvis klassificera enhetens position i realtid baserat på sensordata. De tre möjliga positionerna kan definieras som "stående", "liggande" och i "rörelse/vibrerar". Projektet kan sedan utvidgas för att lösa en mer komplex klassificeringsuppgift. Alltså att man kan gå igenom samma steg som ovan men anpassar applikationen och stegen för att lösa en uppgift som en människa inte klarar av. En annan möjlig expansion av projektet kan vara att lägga till trådlös kommunikation. Alltså att man skickar vidare resultaten av klassificeringsuppgifterna till en extern enhet via typ BT.