

se kode her: <https://editor.p5js.org/odfs2005/sketches/OOCrbMhVK>

## Design

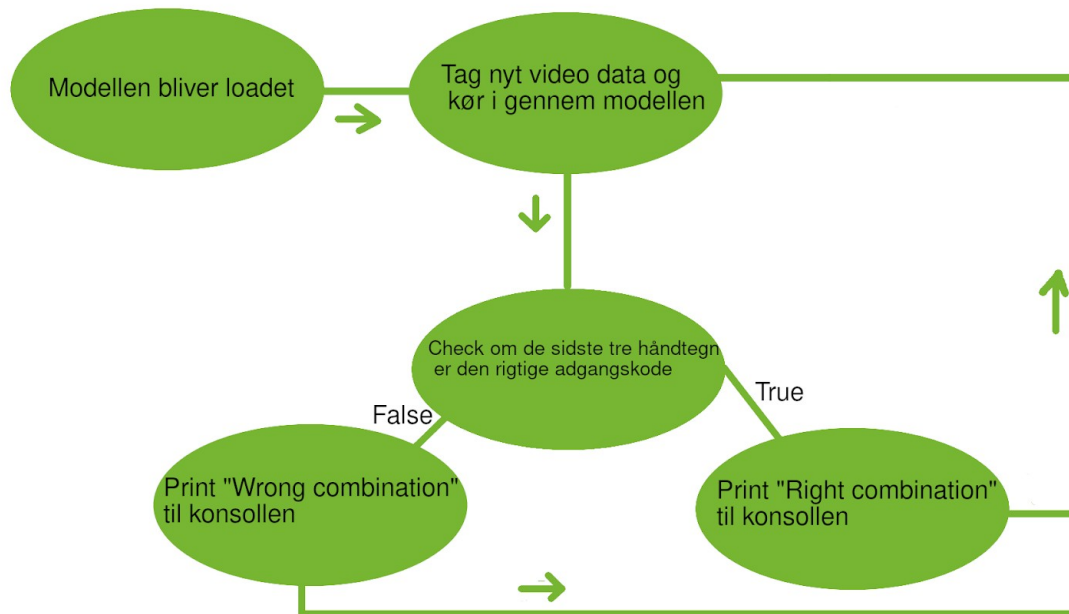
Machine learning er når en computer kan forudsige et eller andet, som kan forbedres gennem erfaring og ved at blive trænet. Maskine læring algoritmer bygger en model baseret på prøvedata, kendt som træningsdata, for at foretage forudsigelser eller beslutninger uden at være programmeret til at gøre det.

Matematiske regressionsmodeller er et eksempel på en meget simpel klassisk machine learning algoritme, som underbygger hvad machine learning går ud på. Du kan træne modellen med en masse data for at gøre modellen mere præcis, derfor hvis du indsætter et tal i modellen vil der komme et tal ud baseret på det trænede data.

Hvis en machine learning algoritme skal differentiere mellem en kat eller en hund bruger man ofte begrebet "Deep learning" fx hvis en algoritme skal gætte på om det billeddata man putter ind i modellen forestiller en hund eller en kat, vil den køre pixel dataen gennem et neuralt netværk, som er trænet til at differentiere mellem hunde og katte. Jo flere billeder man har af den samme ting (med små variationer), jo mere præcis bliver algoritmen, fordi at små vægte i netværket bliver ændret som har indflydelse på resultatet man får til sidst.

En god måde at træne AI'en på, kunne være at tage billeder af håndtegnene med forskellige baggrunde og omgivelser. På den måde kunne den fokusere mest på selve håndtegnet, frem for hvad den ser i baggrunden. Nogle af vores håndtegn fungerede f.eks. kun med en helt hvid baggrund, da vi kun havde taget billeder af håndtegnet med hvid baggrund. Det betød, at da vi viste håndtegnet med en anden baggrund, havde AI'en sværere ved at genkende det, da den var vant til kun at se det på en hvid baggrund. Derudover er det også en god idé at tage billeder fra lidt forskellige vinkler, da man ikke altid kommer til at sidde det helt samme sted ift. kameraet.

Vores implementering af vores model er et program som skal tjekke om man laver den rigtige kombination af håndtegn, som virker som en slags adgangskode, men i stedet for at have et keyboard bruger man et webcam og laver håndtegn.



Ifølge flow diagrammet starter programmet med at lade vores model og indsamle videodata fra vores webcam og køre dataen igennem modellen så vi får et resultat. Derefter tjekker vi om de sidste tre håndtegn der er blevet opfanget af programmet er i den rigtige kombination og hvis det er sandt printer vi "Right combination" til konsollen og hvis det er falsk printer vi "Wrong combination" til konsollen. Derefter gentager processen sig igen, hvor den indsamler nyt videodata osv.

### - Implementing

Det første der sker i koden er at globale variabler som "label", "video" og "imageModeURL" bliver initialiseret med deres værdier. Vi har uploadet vores machine learning model som en json fil på googles servere, som vi kan referer til med en URL i vores kode. Derefter ved hjælp af ml5 machine learning biblioteket "loader" vi modellen ind i en global variabel kaldet "classifier" i preload() funktionen. Derefter bliver setup() kaldt som begynder at filme gennem webcamet og bliver vist på skærmen.

Når classifyVideo() bliver kaldt bruger vi ml5's funktion som hedder classify(), som tager to parameter; video dataen og en callback function som bliver kaldt når modellen får et svar tilbage. Callback() funktionen tager imod to parameter; error og en array med resultaterne sorteret efter procentvis størst sandsynlighed.

```
61     arr.push(results[0].label);
62     console.log("Fangede: ", results[0].label);
63     console.log("Din nuværende kombination er: ")
64     for (let i = 0; i < arr.length; i++) {
65         console.log(arr[i]);
66     }
67     let error = 0;
68
69
70     if (arr.length >= 3){
71         for (let i = 0; i < pass.length; i++) {
72             if (arr[i] != pass[i]){
73                 error = 1;
74             }
75         }
76
77         if (error == 1) {
78             console.log("Wrong combination!");
79         }
80         else {
81             console.log("Right combination");
82         }
83     }
```

Når vi får resultaterne lagre vi håndtegnet i en array (Se linje 61) også tjekker vi om vi allerede har genkendt over 3 håndtegn tidligere (Se linje 70), og hvis det er sandt laver vi et for loop for at se om de genkendte håndtegn er i den rigtige rækkefølge i forhold til adgangskoden (Se linje 71). Hvis en af håndtegne ikke matcher den rigtige kombination printer vi "Wrong combination" til konsollen, og hvis de allesammen matcher printer vi "Right combination" til konsollen.

## - Test

Video link: <https://youtu.be/AcJm-WcUYBg>

I videon demonstrerer Oliver at vores program virker. Han starter med at vise håndtegnet "OK" og så F-fingeren og til sidst "OK" igen. Programmet printer "Right combination" til konsollen når han har lavet den rigtige kombination af håndtegn.

## - Konklusion

Vi har opfyldt alle kravene i kravspecifikationen, da vi bruger teachable machines, p5.js- og ml5.js bibliotekerne, vi genkender mindst 2 ting og anvender programmet i samarbejde med et andet, som er et program som tjekker om man har lavet de korrekte håndtegn i forhold til en adgangskode.

Mulige forbedringer kunne være at træne algoritmen til at genkende nogle flere håndtegn, så man kunne lave nogle lidt mere indviklede adgangskoder. Vi kunne helt sikkert også have trænet

algoritmen bedre med bare de 3 håndtegn vi har, da den ikke altid genkender rigtigt. Men i sidste ende virker vores program altså som det skal.