



# **Camera sensoren**

## Realisatie verslag

**Bachelor Toegepaste Informatica  
Application Development**

**Arslan Bulent**

Academiejaar 2022-2023

## Inhoud

<b>1</b>	<b>INLEIDING.....</b>	<b>2</b>
<b>2</b>	<b>STAGEBEDRIJF .....</b>	<b>3</b>
2.1	Delta Technics .....	3
2.2	Delta Care.....	3
2.3	Extra info.....	3
<b>3</b>	<b>REALISATIES .....</b>	<b>4</b>
3.1	Bestaande tool.....	4
3.2	Opdracht.....	5
<b>4</b>	<b>KENNIS VERWERVEN .....</b>	<b>6</b>
4.1	Mobotix AI.....	6
4.2	Camera menu .....	6
4.3	Coördinaten x, y .....	7
<b>5</b>	<b>DESIGNS.....</b>	<b>8</b>
5.1	Fabric JS.....	8
5.2	React .....	8
5.3	Visual studio code .....	8
5.4	ATOMIC .....	8
<b>6</b>	<b>OPBOUW.....</b>	<b>9</b>
6.1	Korte uitleg .....	9
<b>7</b>	<b>API .....</b>	<b>11</b>
7.1	Authorisatie.....	11
7.2	Permanent opslaan.....	11
7.3	Detectie veld(en) aanmaken .....	12
7.4	Privacy mode .....	12
7.5	Detectiezone namen lijst opvragen .....	12
<b>8</b>	<b>POLYGON (CRUD) .....</b>	<b>13</b>
8.1	Cirkel .....	13
8.2	Polyline aanmaken .....	14
8.3	Slim polygon herkenning .....	14
8.4	Selecteer cirkel en polyline punt.....	15
8.5	Punt verwijderen .....	15
8.6	Canvas coördinaten opmaken .....	16
<b>9</b>	<b>FRONTEND.....</b>	<b>17</b>
9.1	Start scherm.....	17
9.2	Lijst opvullen .....	18
9.3	Detectiegebieden aanmaken.....	19
9.4	Camera view.....	20
9.5	Permanent opslaan.....	21
<b>10</b>	<b>AUTOMATISCHE BED HERKENNING .....</b>	<b>22</b>
10.1	Label me .....	22
10.2	Training data .....	23
10.3	Klassen .....	24
10.4	Txt file .....	24
10.5	Training .....	25
10.6	Test resultaat .....	25
<b>11</b>	<b>CONCLUSIE.....</b>	<b>26</b>

# **1 INLEIDING**

Een stageplaats is een belangrijk aspect voor een student, omdat ze hierdoor kennis kunnen maken met het bedrijfsleven. Daarnaast biedt een stageplaats ook meer mogelijkheden voor communicatie en de kans om te leren van andere collega's.

## **2 STAGEBEDRIJF**

### **2.1 Delta Technics**

Dit bedrijf is zeer ervaren in de elektro- en telecomsector. Ze hebben zich gespecialiseerd in databekabeling volgens de nieuwste technologieën en standaarden, waaronder gestructureerde bekabeling in glasvezeltechnologie en "Twisted Pair" koperbekabeling. Daarnaast hebben ze ook aandacht besteed aan actieve apparaten en bieden ze oplossingen aan voor klanten. Ze hebben ook IOT-toepassingen toegevoegd aan hun assortiment.

### **2.2 Delta Care**

Delta Care is een dochterbedrijf van Delta Technics en biedt oplossingen om het leven van senioren en zorgbehoevenden aangenamer te maken.

De bijdrage die Delta Care hierin kan leveren, is het creëren van een aangename en veilige woonomgeving door het installeren van slimme technologie. Er zijn verschillende toepassingen, waarvan een voorbeeld de camera's met gezichtsherkenning zijn. Deze camera's zorgen ervoor dat bewoners in het woonzorgcentrum niet zomaar naar buiten kunnen gaan, tenzij ze nog bij bewustzijn zijn.

Deze technologieën zorgen ervoor dat de leefomgeving meedenkt met het ritme van de bewoner. Bij afwijkingen in dit dagelijkse ritme zal het netwerk geïnformeerd worden. Bij gevaar zal er ingegrepen worden. Dit is een aantrekkelijke manier om zelfstandig wonen langer mogelijk te maken en zorginstellingen te ondersteunen.

### **2.3 Extra info**

Zoals eerder vermeld, maakt Delta Care gebruik van specifieke camera's, met name Mobotix camera's. De hoofdvestiging van Mobotix bevindt zich in Duitsland. Deze camera's kunnen worden ingezet in verschillende toepassingen, zoals gezichtsherkenning, AI en warmtecamera's.

## 3 REALISATIES

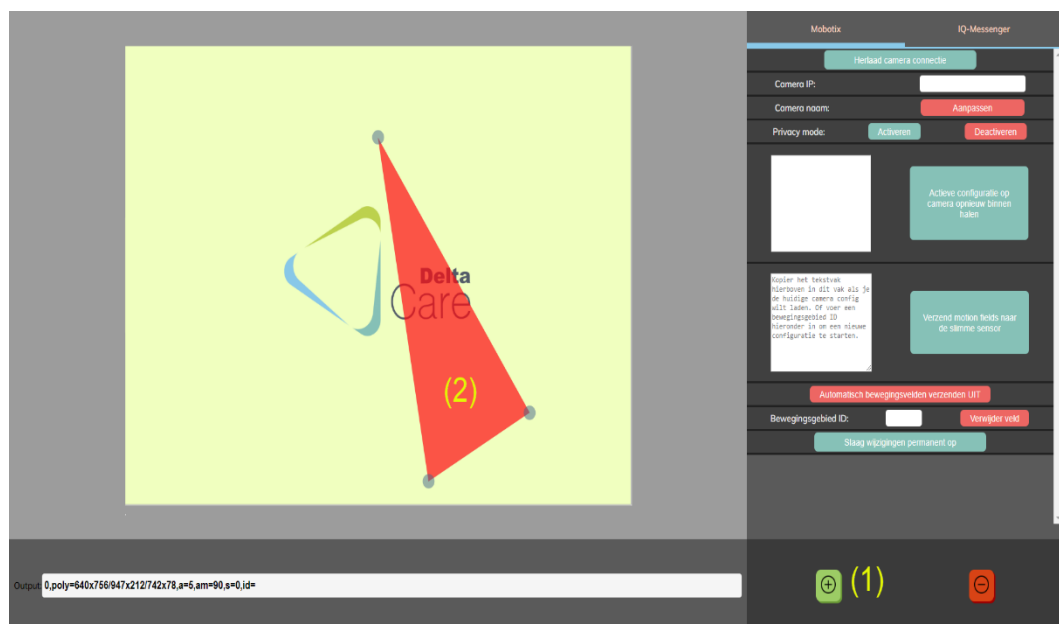
### 3.1 Bestaande tool

Het doel was om een bestaande applicatie, ontworpen door Delta Technics, te verbeteren en gebruiksvriendelijker te maken.

In de huidige situatie is het mogelijk om bepaalde selectiepunten naar de camera te sturen, zodat deze velden kunnen dienen als detectiegebieden om een object te detecteren dat zich in dat gebied bevindt. Het is echter belangrijk op te merken dat niet alle rustkamers een bed hebben, sommige rustkamers hebben in plaats daarvan zetels.

Bij het aanmaken van de eerste selectiepunten wordt automatisch een sjabloon (2) met een paar punten door het systeem aangemaakt. Het onslachtige is dat je telkens aparte knoppen (1) moet indrukken om een punt toe te voegen of te verwijderen.

Het verplaatsen van selectiepunten gebeurt ook via een extra knop, terwijl dit veel gemakkelijker zou kunnen gebeuren met een muisklik.



## 3.2 Opdracht

Mijn opdracht was om een tool te ontwerpen met betere functionaliteiten, zoals eerder besproken. Hier zijn de specificaties:

- De klant moet het IP-adres kunnen invoeren.
- Wijzigingen moeten permanent kunnen worden opgeslagen.
- Het camerabeeld mag niet zichtbaar zijn vanwege privacy redenen.
- Bij elke opmaak van een detectiegebied moet er een melding verschijnen met de vraag "Weet je zeker?" (ja of nee).
- Er moeten API-oproepen zijn voor de benodigde functionaliteiten.
  - **API:** Application Programming Interface - een software-intermediair waarmee twee applicaties met elkaar kunnen communiceren. API 's zijn een toegankelijke manier om gegevens binnen en tussen organisaties te extraheren en te delen.
- Er moet een toastmelding zijn om acties aan de gebruiker te tonen.  
(<https://uxdesign.cc/the-different-types-of-messages-uxers-need-to-remember-to-design-d1d8e64f8207>)
  - **Toast:** een pop-upmelding die tijdelijk verschijnt om de gebruiker te informeren over een actie of een gebeurtenis. Het wordt vaak gebruikt om niet-storende meldingen weer te geven, zoals succesmeldingen, foutmeldingen of waarschuwingen.
- Het moet mogelijk zijn om de detectiezone namen lijst van de camera op te vragen en deze in een drop-down lijst te plaatsen.
- De coördinaten van elk selectiepunt moeten worden opgenomen.
- Het moet mogelijk zijn om polygonen, polyline en andere vormen op te maken.
  - **Polygon** : Een polygon is een gesloten vorm bestaande uit drie of meer rechte lijnsegmenten. Het wordt gekenmerkt door het hebben van een gesloten lus, waarbij het eindpunt van het laatste segment samenvalt met het beginpunt van het eerste segment. Polygons kunnen verschillende vormen hebben, zoals driehoeken, vierkanten, rechthoeken, veelhoeken, enz.
  - **Polyline:** Een polyline is een reeks van aan elkaar gekoppelde rechte lijnsegmenten, waarbij elk segment verbonden is met het vorige en het volgende segment. In tegenstelling tot een polygon, is een polyline open en heeft het geen gesloten lus.
- Als er tijd is, kan er gewerkt worden aan automatische bed herkenning.

## 4 KENNIS VERWERVEN

### 4.1 Mobotix AI

Het toestel waarmee ik heb gewerkt is een IP-camera. Het model heeft een horizontale hoekbereik van 103° en een verticaal bereik van 77°. Door deze kan men ervoor zorgen dat het camerabeeld effectief het gewenste gebied bestrijkt en belangrijke gebeurtenissen of objecten binnen dat gebied kan vastleggen. Dit is cruciaal voor beveiligings-, bewakings- en toezichttoepassingen, waarbij een goed zicht op het relevante gebied essentieel is voor de detectie, identificatie en analyse van gebeurtenissen.

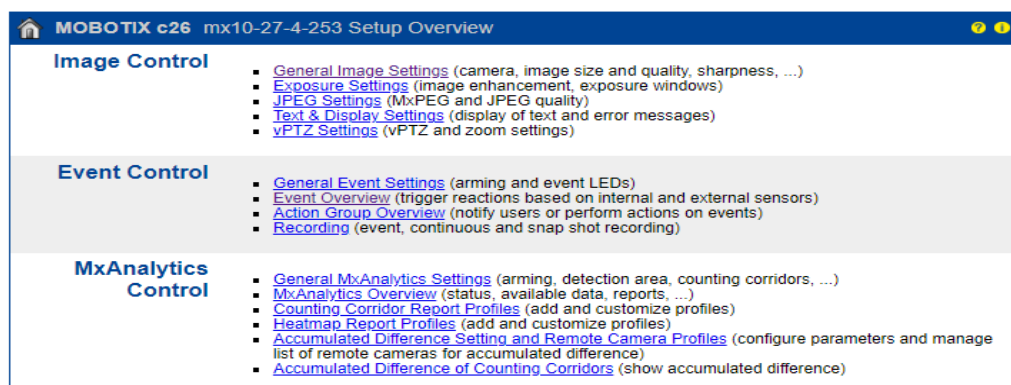
### 4.2 Camera menu

De eerste dag was voor mij het verkennen van de mogelijkheden van deze camera. Ik heb ontdekt dat deze twee belangrijke menu's heeft:

- **Admin menu:** In dit menu kan je de administratieve basisinstellingen wijzigen voor het gebruik van de camera, zoals gebruikersnamen en wachtwoorden, netwerkinstellingen, software-updates, enzovoort.



- **Setup menu:** Dit menu bevat de instellingen en configuraties voor zaken zoals beeldverwerking, tekstoverlay en bepaalde acties.

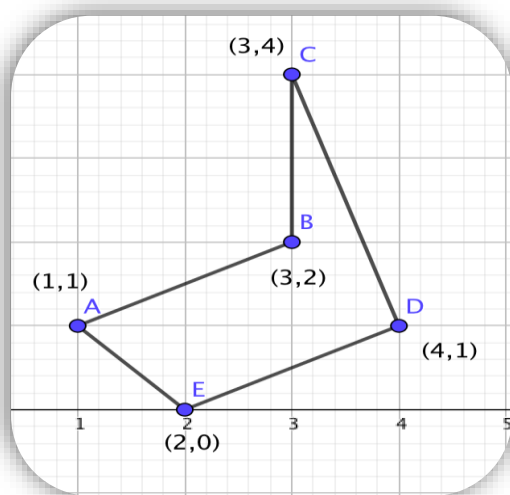


### 4.3 Coördinaten x, y

Een polygon bestaat uit een reeks van x- en y-coördinaten die de hoekpunten van de vorm definiëren. Met deze coördinaten kunnen we de vorm van de polygon bepalen door lijnsegmenten te verbinden tussen de opeenvolgende punten.

Laten we een eenvoudig voorbeeld nemen van een polygon met de volgende x- en y-coördinaten:

x: [1, 3, 3, 4, 2]   y: [1, 2, 4, 1, 0]



In dit voorbeeld hebben we vijf punten met de volgende coördinaten: (1, 1), (3, 2), (3, 4), (4, 1) en (2, 0). Door deze punten te verbinden, krijgen een polygon met zijden tussen de punten (1, 1) - (3, 2) - (3, 4) - (4, 1) - (2, 0).

De x- en y-coördinaten worden gebruikt om de vorm van de polygon te definiëren en kunnen worden geïnterpreteerd als punten op een cartesisch coördinatensysteem. Het aantal punten en de volgorde waarin ze worden opgegeven, bepalen de vorm van de polygon.

Dit is slechts een eenvoudig voorbeeld om het concept van polygonen en hun x- en y-coördinaten te illustreren. In werkelijke toepassingen kunnen polygonen veel complexer zijn en kunnen ze worden gebruikt om meer gedetailleerde vormen en regio's te definiëren.



## 5 DESIGNS

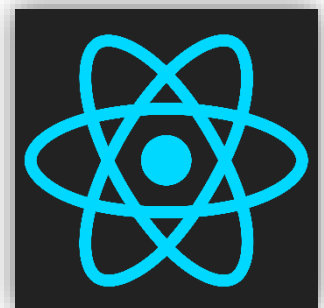
### 5.1 Fabric JS

Fabric.js is een krachtige JavaScript-bibliotheek voor het maken en beheren van interactieve grafische objecten op het web. Als voorbeeld: je kunt met Fabric.js verschillende grafische objecten maken, zoals rechthoeken, cirkels, driehoeken, lijnen, tekst en afbeeldingen. Deze objecten kunnen worden aangepast en beheerd, inclusief het wijzigen van hun positie, grootte, rotatie, kleur, transparantie en andere eigenschappen.



### 5.2 React

Voor het project heb ik React gebruikt als frontend. Ik koos voor React omdat uit onderzoek bleek dat het een zeer populaire keuze is. Daarnaast had ik ook ervaring met React vanuit mijn opleiding.



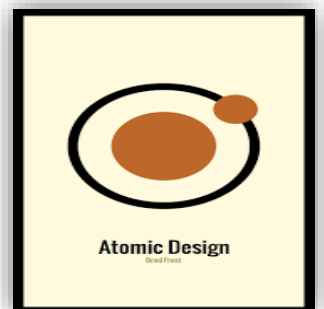
### 5.3 Visual studio code

Visual studio heb ik gebruikt als code platform omdat deze ten eerste zeer stabiel is en gebruiksvriendelijk.



### 5.4 ATOMIC

Naarmate het ambacht van webdesign blijft evalueren, heb ik toch gekozen voor ATOMIC Design zodat we naarmate ons eigen huisstijl hadden.



## 6 OPBOUW

Na voldoende kennis te hebben opgedaan, ben ik begonnen met het maken van de eerste API-calls naar de camera. Op deze manier kon ik de juiste verzoeken versturen en de benodigde informatie ophalen. Dit gebeurde in een aparte service-laag.

Zoals eerder vermeld, had de camera verschillende submenu's die we konden benaderen. Voor mijn toepassing was het belangrijk dat ik informatie kon ophalen en verzenden vanuit event control. Dit kon worden gedaan via het setupmenu. Hier kon ik alles doen wat betreft het tekenen van detectiezones.

### 6.1 Korte uitleg

In de afbeelding hieronder ziet u parameter velden die te maken hebben met detectie zones.

- 1- In het veld met parameters kunnen we waarden invoeren waarmee we detectiezones kunnen maken. Dit stelt ons in staat om specifieke eigenschappen en criteria in te stellen voor het detecteren van objecten.
- 2- In het invoerveld wordt automatisch een ID-nummer toegewezen dat afkomstig is van de parameter box. Dit ID-nummer dient als een unieke identificatie voor de detectiezone..
- 3- Dit zijn de namen van onze objecten die we later via de API kunnen ophalen. Deze namen kunnen worden gebruikt om specifieke detectiezones te identificeren en ermee te werken.

MOBOTIX c26 mx10-27-4-253 Image Analysis Events

Attribute

Value

Explanation

Video Motion Settings

☐ Hide Unused Windows

On and highlight on event

☒ Show Video Motion Window ID

1

Off

1

General MxActivitySensor Settings

☐ Detect Fast Motion
☐ Show Direction Indicator
☐ Show Live Position Indicator

Off

MxActivitySensor 2.1 Settings

☐ MxActivitySensor 2.1

Behavioral Detection

12

☐ Enable alarm visualization
☐ Show Object Speed
☐

Edit shared configuration of MxAnalytics

Events

VM

5

☐ Behavioral Detection
☐ MxActivitySensor
☒ Video Motion

Motion detection using simple digital image analysis.

2

VM2

AS

Tim Zeno

3

Add new profile

Event Sensor Type

☐ Behavioral Detection
☐ MxActivitySensor
☒ Video Motion

Event Dead Time:

Time to wait [0..3600 s] before the event can trigger anew.

Event Sensor Type:

Choose the image analysis sensor.

Window List:

List VM windows (by their ID). If one of the listed windows detects motion, this event is triggered. Entries are separated by commas, and can consist of single windows or ranges (e.g. "1,2,4-7"). The ranges can be open ("3-" or "-4"). Leaving this field empty selects all windows.

Add Rectangle

Highlight selected

Inactive

Delete

Inactive

Delete

Inactive

Delete

Set

Factory

Restore

Close

Less

10

## 7 API

### 7.1 Authorisatie

Zoals te zien is in de afbeelding, hebben we een autorisatieobject gemaakt waarmee de gebruikersnaam en het wachtwoord kunnen worden meegegeven. Deze handeling is noodzakelijk om toegang te krijgen tot het camerasysteem. Het wordt gebruikt om te controleren of een entiteit geautoriseerd is om bepaalde acties uit te voeren, zoals het openen van gegevens, het uitvoeren van een bewerking of het verkrijgen van toegang tot bepaalde bronnen.

```
const authorization = {  
  username: 'admin',  
  password: 'meinsm123',  
};
```

### 7.2 Permanent opslaan

Permanent opslaan wordt gebruikt om ervoor te zorgen dat gegevens behouden blijven, zelfs als er een stroomstoring of spanningsuitval optreedt. Hierdoor kunnen de laatste instellingen en parameters worden opgeslagen en behouden, zodat ze niet verloren gaan bij een onverwachte onderbreking. Dit zorgt voor een betrouwbare werking van het systeem en voorkomt dat gegevens verloren gaan.

```
//Configuration store  
getStore(ip, username, password) {  
  return axios  
    .get(  
      `http://${username}:${password}@${ip}/admin/rcontrol?action=storeconfig`  
    )  
    .catch((error) => {  
      console.log(error);  
    });  
}
```

## 7.3 Detectie veld(en) aanmaken

Deze call wordt gebruikt om detectievelden naar de camera te sturen, zoals te zien is in afbeelding 1 pagina 10 met box nummer 1.

```
//Create motions line
createMotionLine(ip, username, password, motion) {
  return axios
    .get(
      `http://${username}:${password}@${ip}/control/control?set&section=eventcontrol&motion_area=${motion}`
    )
    .catch((error) => {
      console.log(error);
    });
}
```

## 7.4 Privacy mode

Elke bewoner in het woonzorgcentrum heeft recht op privacy en we houden daar rekening mee. Geen enkele medewerker mag zomaar naar de kamers kijken, tenzij er sprake is van noodgevallen, zoals wanneer een bewoner een zone betreedt waar hij/zij niet zou moeten zijn.

Via een ondersteunende API-oproep sturen we een commando naar de camera om de privacy modus in te schakelen (enable) of uit te schakelen (disable).

```
// Display mode
getDisplayBlock(ip, username, password, display) {
  return axios
    .get(
      `http://${username}:${password}@${ip}/admin/control?set&section=logo&display=${display}`
    )
    .catch((error) => {
      console.log(error);
    });
}
```

## 7.5 Detectiezone namen lijst opvragen

In de motionlijst zijn namen toegekend die kunnen worden gebruikt om objecten of detectiezones te benoemen. Deze lijst bevindt zich in de configuratie van de camera, onder het menu 'Setup', submenu 'Event Control', zoals eerder vermeld in afbeelding nr.1, kader 3. met behulp van deze API-oproep kunnen we de volledige lijst met namen ophalen die ze kunnen gebruiken bij het instellen van detectiezones.

```
//Create motions line
createMotionLine(ip, username, password, motion) {
  return axios
    .get(
      `http://${username}:${password}@${ip}/control/control?set&section=eventcontrol&motion_area=${motion}`
    )
    .catch((error) => {
      console.log(error);
    });
}
```

Zie screenshot event control waar we ons object namen kunnen ophalen via camera configuratie.

<b>Image Analysis Events</b>	<input type="text" value="VM"/>	Video Motion
	<input type="text" value="VM2"/>	Video Motion
	<input type="text" value="AS"/>	Video Motion
	<input type="text" value="Sevgi"/>	Video Motion

## 8 POLYGON (CRUD)

### 8.1 Cirkel

De knooppunten in cirkelvorm worden aangemaakt om gemakkelijk correcties te kunnen uitvoeren op een object. Wanneer er aanpassingen moeten worden gedaan aan een object, kan iemand eenvoudig een knooppunt vastpakken en wijzigingen aanbrengen.

Deze functie wordt eerst opgeroepen bij het aanmaken van een detectiezone, waarmee cirkels worden gemaakt. Deze cirkels vormen de knooppunten tussen de polylines. Telkens wanneer er op het scherm van het canvas wordt geklikt op een positie waarvan de coördinaten niet bekend zijn, wordt er een nieuw knooppunt aangemaakt. Elk knooppunt krijgt een unieke ID en wordt gekoppeld aan de polygon waar het bij hoort, die ook zijn eigen ID heeft.

```
dataPoint.polygon.map((point, b) => {  
  circle[b] = new fabric.Circle({  
    radius: 5,  
    fill: 'white',  
    stroke: 'red',  
    left: point.x,  
    top: point.y,  
    originX: 'center',  
    originY: 'center',  
    hasControls: false,  
    parentId: a,  
    id: b,  
    fireRightClick: true,  
    fireMiddleClick: true,  
    stopContextMenu: true,  
  });  
  canvas.add(circle[b]);  
});
```

## 8.2 Polyline aanmaken

De polyline wordt gebruikt om de randen van de polygon zichtbaar te maken. Deze lijnen worden gekoppeld aan de knooppunten die eerder zijn aangemaakt. Deze functie wordt opgeroepen samen met het aanmaken van de knooppunten voor een detectiezone. Hierdoor worden de randen van de detectiezone zichtbaar gemaakt. De lijnen vormen de verbinding tussen de knooppunten, waardoor de omtrek van de detectiezone wordt weergegeven.

```
dataPoint.polygon.map((point, b) => {
  circle[b] = new fabric.Circle({
    radius: 5,
    fill: 'white',
    stroke: 'red',
    left: point.x,
    top: point.y,
    originX: 'center',
    originY: 'center',
    hasControls: false,
    parentId: a,
    id: b,
    fireRightClick: true,
    fireMiddleClick: true,
    stopContextMenu: true,
  });
  canvas.add(circle[b]);
});
```

## 8.3 Slim polygon herkenning

De slimme polygon selectie is ontworpen om ervoor te zorgen dat wanneer er wijzigingen moeten worden aangebracht in een van de detectiezones, automatisch de juiste polygon wordt geselecteerd wanneer je er met de muis overheen gaat. Daarom is er ook een ID gekoppeld aan elke polygon, zodat deze correct kan worden geïdentificeerd en geselecteerd.

```
canvas.on('mouse:over', (e) => {
  if (e.target !== null) {
    if (e.target.id === undefined) {
      refy.current = e.target.parentId + 1;
    }
  }
});
```



## 8.4 Selecteer cirkel en polyline punt

Om aanpassingen te kunnen doen aan een bestaande detectiezone, moet je op een specifieke knop klikken om het gewenste knooppunt te selecteren waar je wijzigingen wilt aanbrengen.

Er is ook rekening gehouden met de situatie waarin je op een bestaand knooppunt klikt, zodat er geen nieuw knooppunt wordt aangemaakt. Dit wordt bereikt door de waarde van "data[0]" op één te zetten, waardoor het systeem begrijpt dat er geen nieuw knooppunt moet worden toegevoegd, maar dat je wijzigingen wilt aanbrengen aan het geselecteerde knooppunt.

```
canvas.on('selection:created', (e) => {  
  data[0] = 1;  
  selectedItem = {  
    parentId: e.selected[0].parentId,  
    nodeId: e.selected[0].id,  
  };  
});
```

## 8.5 Punt verwijderen

De handeling is vergelijkbaar met wat eerder is besproken, maar er wordt ook rekening gehouden met welke knop van de muis er wordt geklikt. Als je de rechterknop indrukt, begrijpt het systeem dat je het geselecteerde knooppunt wilt verwijderen. Dit stelt je in staat om specifieke knooppunten te verwijderen door met de rechterknop van de muis te klikken.

```
fabric.util.addListener(  
  document.getElementsByClassName('upper-canvas')[0],  
  'contextmenu',  
  function (e) {  
    e.preventDefault();  
    if (e.button >= 0) {  
      if (selectedItem.nodeId !== undefined) {  
        objList.current[selectedItem.parentId].polygon.splice(  
          selectedItem.nodeId,  
          1  
        );  
      } else {  
        objList.current.splice(selectedItem.parentId, 1);  
      }  
      canvas.clear();  
      node();  
      data[0] = 0;  
    }  
  }  
);
```

## 8.6 Canvas coördinaten opmaken

Tijdens de testfase heb ik ontdekt dat de coördinaten van het HTML-canvas niet overeenkomen met die van de camera. Het is erg belangrijk dat de detectiezone die in HTML is gemaakt overeenkomt met het canvas van de camera.

Bij het HTML-canvas beginnen de x- en y-coördinaten in de linkerbovenhoek met een waarde van nul. Bij de camera begint dit echter aan de linkerkant, maar onderaan het beeld, en de x- en y-coördinaten beginnen met een waarde van 34.

**HTML canvas**



**Mobitix canvas**



Om ervoor te zorgen dat de verhouding tussen de applicatie en het cameracanvas klopt met betrekking tot de detectiezone, is er een berekeningsfactor toegepast. Deze factor zorgt ervoor dat de coördinaten correct worden omgezet en de detectiezone nauwkeurig wordt weergegeven op het cameracanvas.

```
const send = (object) => {
  let polyList;
  let offsetX = 86;
  let offsetX1 = 1152 / 785;
  let offsetY = 941;
  let offsetY1 = 941 / 565;

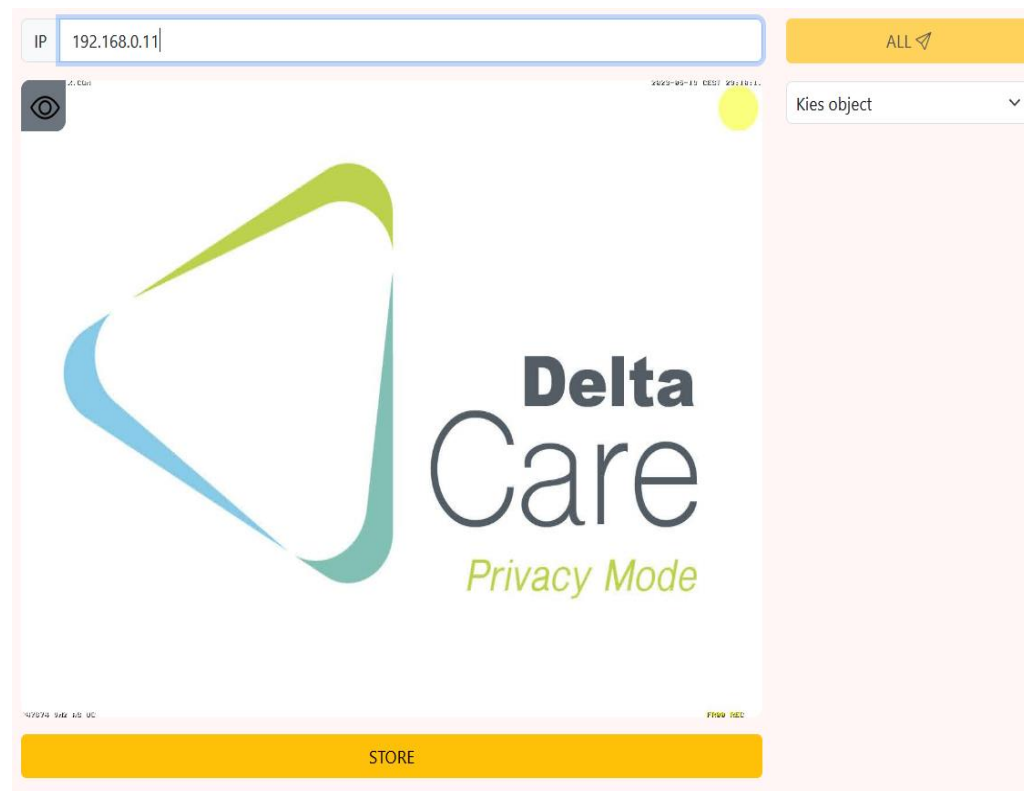
  object.polygon.map((subData, a) => {
    let vx = parseInt(subData.x * offsetX1 + offsetX);
    let vy = parseInt(offsetY - parseInt(subData.y) * offsetY1);
    if (polyList === undefined) {
      polyList = vx + 'x' + vy;
    } else {
      polyList = polyList + (vx + 'x' + vy);
    }
    if (a !== object.polygon.length - 1) {
      polyList = polyList + '/';
    }
  });

  return polyList;
};
```

## 9 FRONTEND

### 9.1 Start scherm

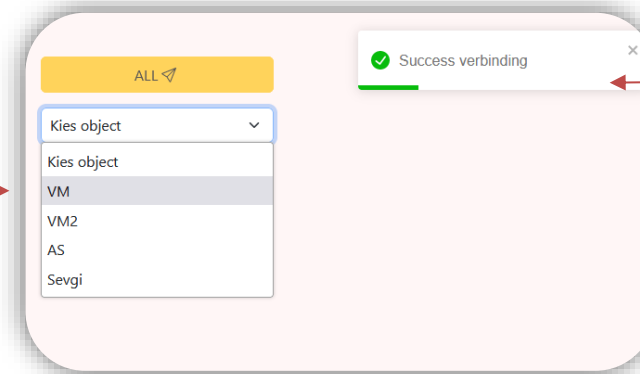
Om verbinding te maken met de camera, geven we het IP-adres in dat bij de betreffende camera hoort. Op deze manier kunnen we verschillende handelingen uitvoeren. Bovenaan links is er een knop die bedoeld is om het camerabeeld in- of uit te schakelen in de privacy modus. Uit privacyoverwegingen worden er geen beelden getoond op het scherm wanneer het



camerabeeld wordt opgevraagd. Deze privacy modus zorgt ervoor dat het camerabeeld niet zichtbaar is om de privacy van de bewoners te waarborgen.

## 9.2 Lijst opvullen

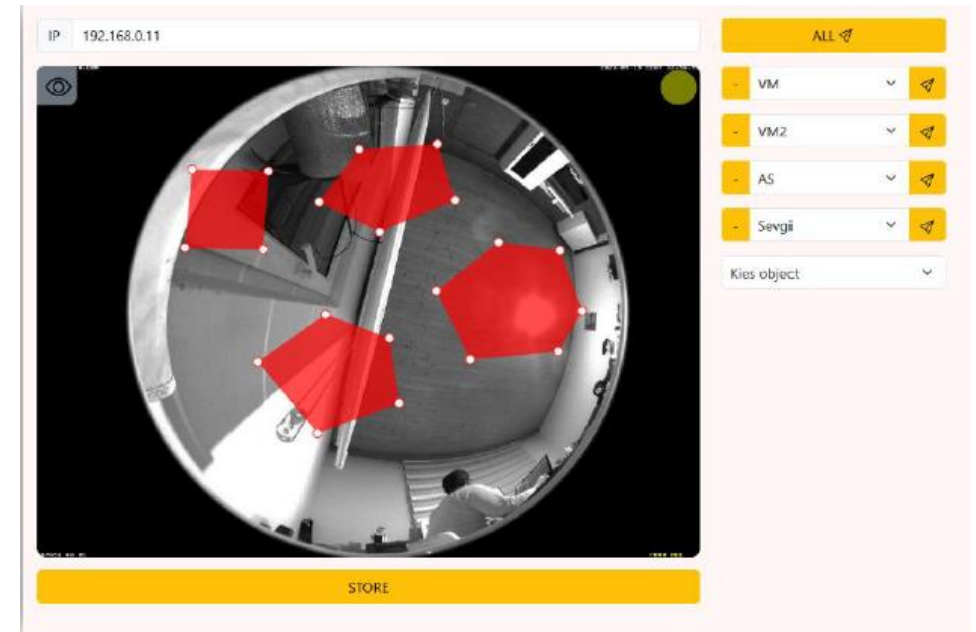
Na een succesvolle verbinding wordt er een melding weergegeven en halen we een lijst op van interne events via de camera API. Vervolgens presenteren we deze lijst in een dropdown-menu.



Toast (melding)

### 9.3 Detectiegebieden aanmaken

In onze applicatie kunnen we detectiegebieden definiëren door middel van muisklikken, waarbij we gebruikmaken van de eerder opgehaalde lijst van events. We kunnen detectiegebieden maken door ten minste twee selectiepunten (eilanden) te kiezen. Hierdoor wordt een polylijn gecreëerd. Bij het selecteren van een derde punt wordt automatisch een polygoonobject gemaakt. Elk eiland (detectiegebied) kan worden verplaatst, verwijderd of opnieuw aangemaakt. Als er meerdere polygoonobjecten zijn, kunnen we wijzigingen aanbrengen door er eenvoudigweg overheen te gaan.



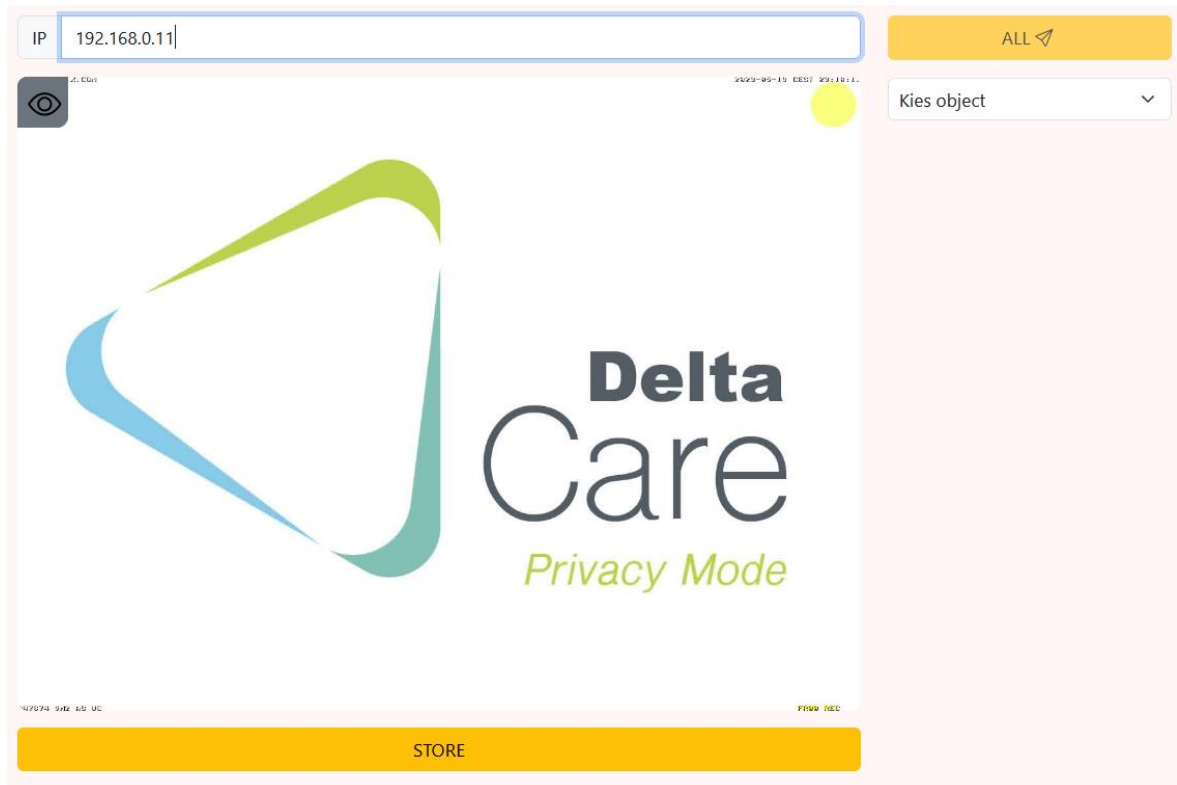
## 9.4 Camera view

De coördinaten (x- en y-coördinaten) van de polygoonselectiepunten kunnen naar de camera worden verzonden om het detectiegebied in te stellen. Deze coördinaten kunnen individueel of als groep worden verzonden. Je kan dit eenvoudig aanpassen door er met de muis overheen te gaan. Op die manier kun je de detectiegebieden en hun vorm gemakkelijk beheren en aanpassen.



## 9.5 Permanent opslaan

Het is van groot belang om de gegevens permanent op te slaan, zelfs na een stroomuitval, om ervoor te zorgen dat deze behouden blijven. Wanneer er op de "Store" knop onder het canvas scherm wordt gedrukt, worden de gegevens permanent opgeslagen. Daarna wordt de privacy-modus opnieuw ingeschakeld, zodat het scherm weer teruggaat naar de privacy-modus.



**Vanaf deze pagina zij we nog in de ontwikkelingsfase. Dit werd met andere woorden nog niet geïmplementeerd in de applicatie.**

## 10 AUTOMATISCHE BED HERKENNING

### 10.1 Label me

LabelMe is een populaire annotatietool die wordt gebruikt voor het handmatig labelen en segmenteren van objecten in afbeeldingen. Door handmatig objecten in afbeeldingen te labelen, kan LabelMe helpen bij het maken van datasets met gelabelde gegevens, die vervolgens kunnen worden gebruikt om algoritmen te trainen om automatisch objecten te detecteren, segmenteren of classificeren.

Hier worden alle afbeeldingen geladen waaruit we gegevens willen extraheren voor onze dataset, en er wordt ook een label aan elke afbeelding gekoppeld. In tool Labelme kunnen we met behulp van een recbox selectiepunten maken en een label, zoals "bed", toewijzen. Zodra de selectie met de recbox is voltooid, wordt er een txt-bestand aangemaakt. Zie txt afbeelding afb 4



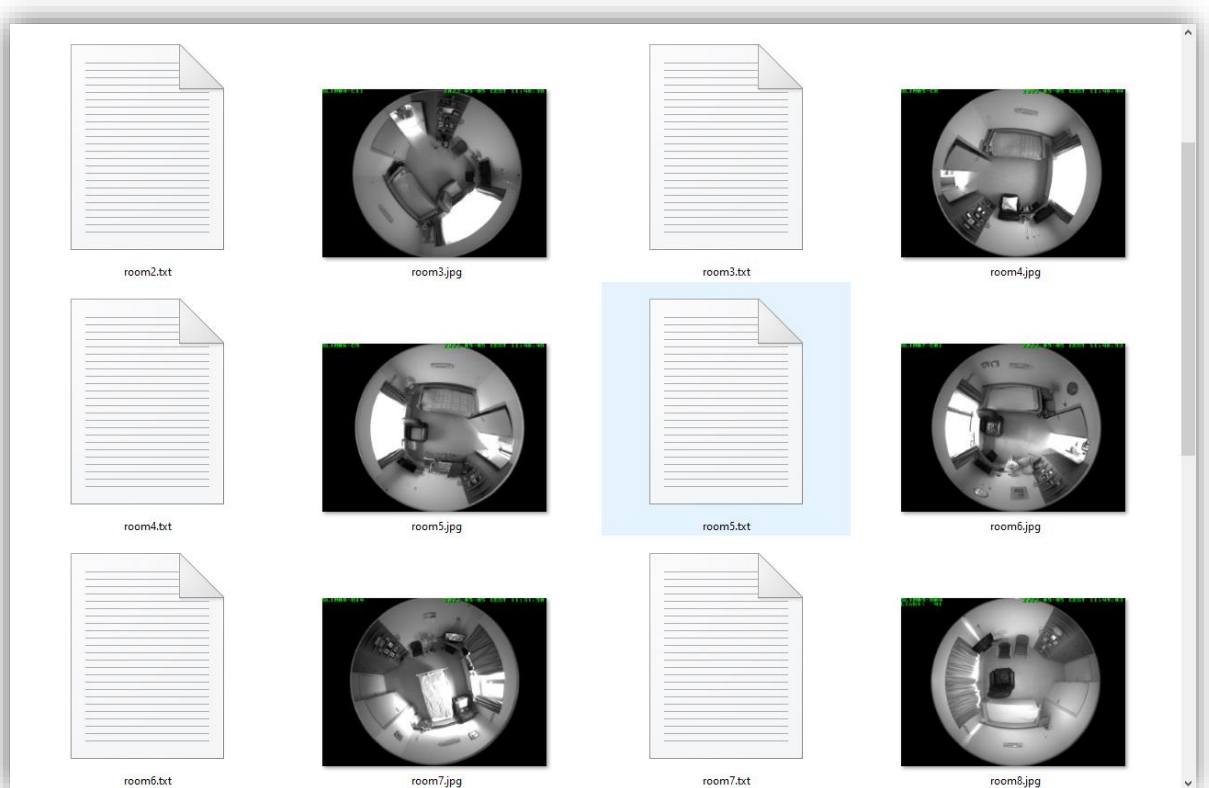
Afb1



## 10.2 Training data

Trainingsgegevens zijn van essentieel belang om AI-modellen te trainen en hen in staat te stellen patronen te herkennen, voorspellingen te doen en beslissingen te nemen op basis van nieuwe gegevens. Het is ook belangrijk dat de afbeeldingsformaten hetzelfde zijn. Om effectief te zijn, moeten AI-modellen kunnen werken met afbeeldingen die dezelfde dimensies en formaten hebben. Dit betekent dat alle afbeeldingen in de training set dezelfde breedte, hoogte en kleurdiepte moeten hebben. Als de afbeeldingen verschillende formaten hebben, kan dit problemen veroorzaken tijdens het trainingsproces, omdat het model mogelijk niet in staat is om de afbeeldingen op een consistente en betrouwbare manier te verwerken.

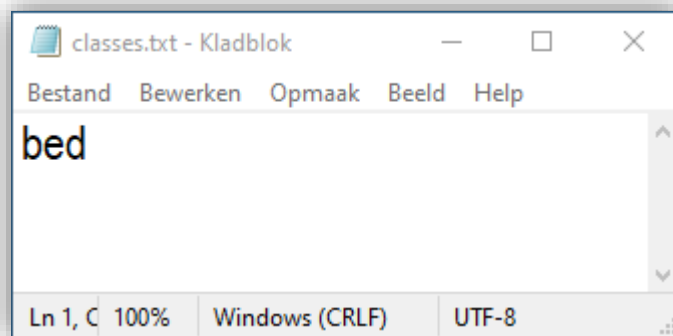
Afb 2



### 10.3 Klassen

Na het aanmaken van het txt-bestand (dataset) wordt er ook een classetxt bestand gegenereerd en hierin word alle labelnamen opgenomen (zie afb.3) die je zou willen weergeven als objectnaam.

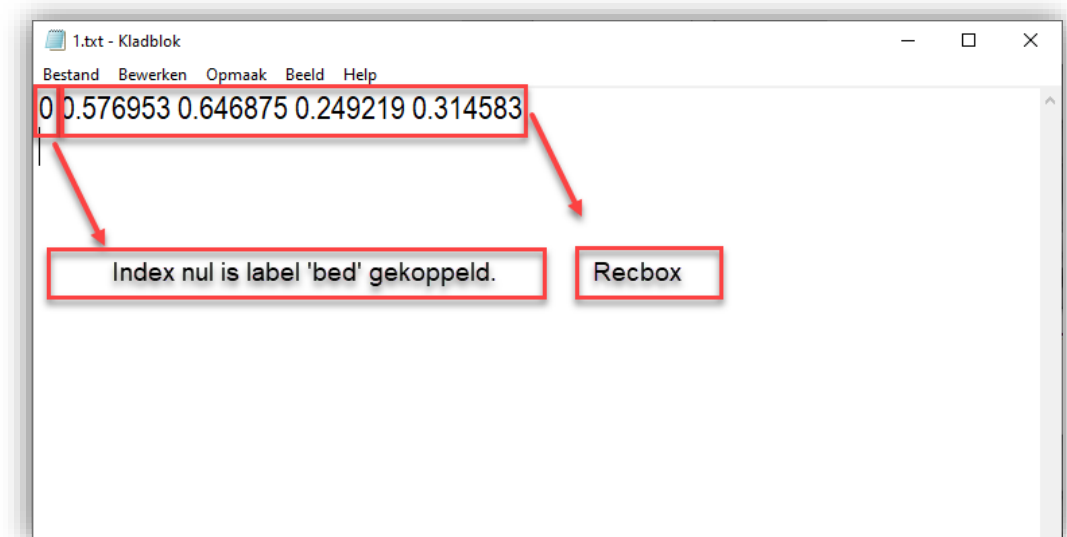
Afb 3



### 10.4 Txt file

De txt- bestanden worden verdeeld, waarbij het eerste getal de index is waaraan de labelnaam is gekoppeld, en de rest van de gegevens zijn de coördinaten van de recbox.

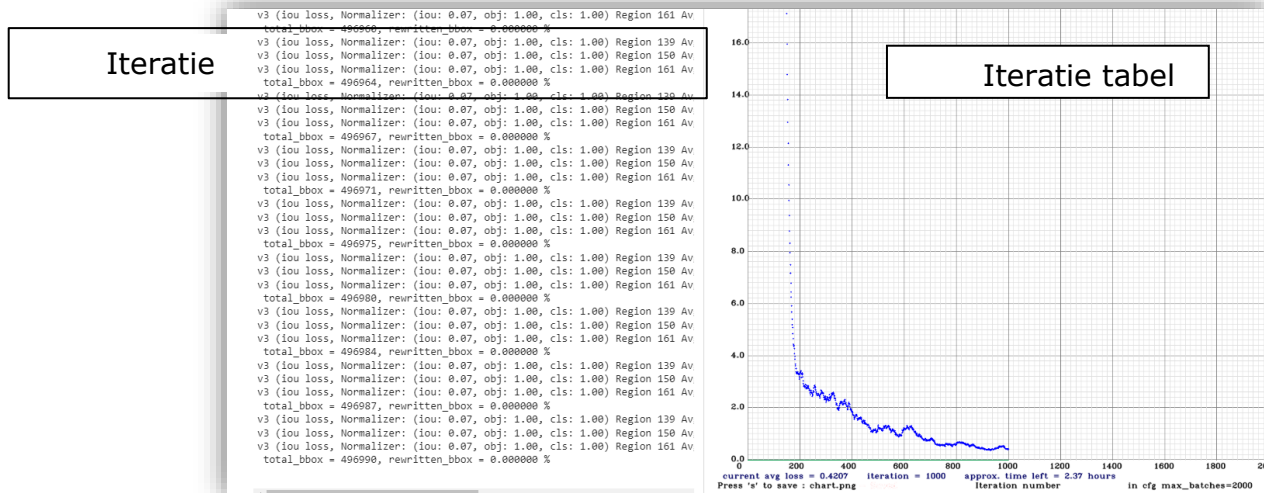
Afb 4



## 10.5 Training

AI-training is van cruciaal belang voor de ontwikkeling en prestaties van AI-systemen. Het trainen van een AI-model omvat het voeden van het model met grote hoeveelheden gegevens en het aanpassen van de parameters van het model om het te laten leren en patronen te ontdekken. Het doel van de training is om het AI-model te voorzien van voldoende informatie, zodat het nauwkeurige voorspellingen en beslissingen kan nemen op nieuwe, onbekende gegevens.

Afb 5



Iteratie verwijst naar het proces van het herhaaldelijk verfijnen en verbeteren van een AI-model door middel van iteratieve stappen. Het omvat het analyseren van de prestaties van het model, het identificeren van zwakke punten of gebieden die verbetering behoeven, en het aanbrengen van aanpassingen om de prestaties te optimaliseren.

## 10.6 Test resultaat

- 1- Allocate additional workspace:** Verwijst naar het toewijzen van extra ruimte of middelen voor het uitvoeren mijn project.
- 2- Weights:** Gewichten dienen als een manier om de kennis en patronen vast te leggen die het model heeft geleerd tijdens het trainingsproces.
- 3- Resultaat:** In de afbeelding "room1.jpg" is het bed met 98% zekerheid herkend.

Afb 6

```
total Batches: 59.304
avg_outputs = 489778
1 Allocate additional workspace_size = 134.22 MB
2 Loading weights from /content/gdrive/MyDrive/yolo_weights/backup/yolov4_delta_last.weights...
  seen 64, trained: 115 K-images (1 Kilo-batches_64)
  Done! Loaded 162 layers from weights-file
  Detection layer: 139 - type = 28
  Detection layer: 150 - type = 28
  Detection layer: 161 - type = 28
3 /content/darknet/data/data_images/room1.jpg: Predicted in 667.039000 milli-seconds.
  bed: 98%
```

## **11 CONCLUSIE**

Stage is belangrijk om te ervaren hoe het er in de echte programmeerwereld aan toegaat. Net zoals in het dagelijks leven leer je tijdens een stage hoe je met bepaalde onderwerpen moet omgaan. Dit leidt tot meer leerervaring en je wordt geconfronteerd met nieuwe technieken en ontwikkelt ook waardevolle soft skills. Deze leerervaring heeft ervoor gezorgd dat ik verder wil gaan in deze wereld en een bijdrage wil leveren, omdat dit de enige plek is waar ik me echt goed bij voel.

Gedurende drie maanden kan ik zeggen dat de tijd voorbij is gevlogen en dat ik zeker een zeer positief gevoel heb. Zoals in het leven zijn er goede en minder goede dagen, maar dat was ook te verwachten tijdens deze stageperiode.

Ik ben zeer dankbaar dat Delta Technics mij de kans heeft gegeven en dat ze het vertrouwen hadden om mij als stagiair aan te nemen.