Technisch ontwerp

Inleiding

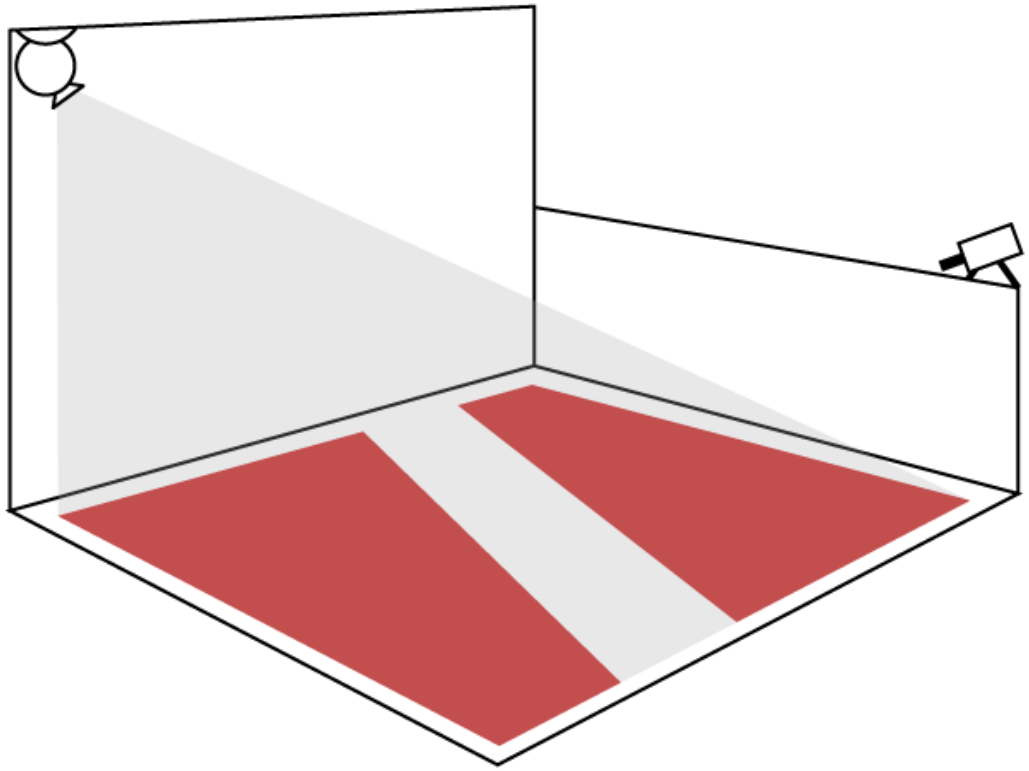
In dit document worden alle facetten van het product dat gemaakt gaat worden behandeld in verschillende stukken. Als eerste wordt de hardware besproken, hierin zal er verdiept worden op welke hardware er gebruikt gaat worden, welke opstelling ideaal is voor de beeldherkenning en wat voor belichting ideaal is. Als tweede worde de software besproken, dit zal dieper ingaan op het software design, de verschillende programma's en de communicatie daartussen.

Hardware

Onder hardware worden de verschillende fysieke benodigdheden voor het project bedoeld, dit gaat om de laptop die de software gaat draaien, de turret met de laser en de camera. Ook worden de te detecteren objecten hier beschreven.

Camera opstelling

De opstelling van de tuin is erg belangrijk. Dit zal namelijk deels de kwaliteit van de beelden die de camera zal bieden. Hieronder is schematisch een tekening te zien van het prototype van de Wet-Cat, met deze opstelling zijn we van plan te gaan presenteren op het moment dat we het project moeten opleveren. De camera zal in de opstelling maximaal 2 meter van de “tuin” verwijderd staan. Omdat anders de verschillende objecten niet herkend kan worden.



Objecten

Omdat we verschillende dingen gaan herkennen, zijn er verschillende markers gemaakt die het makkelijk maken om de verschillende objecten te herkennen. Er zijn vier verschillende markers die de hoeken van een "Danger Zone" aangeven en er zijn zes verschillende markers die de verschillende katten aangeven. Deze markers zullen op een dobbelsteem worden geplakt om zo aan te geven welke kat het is. Per kat zal de laser iets anders doen. de commando's staat in het hoofdstuk 'Turret' beschreven. Verder staan hieronder de verschillende markers die door het systeem moeten worden herkend.

De objecten zal worden geprint op een grote van 10x10 cm. Dit word gedaan om het goed zichtbaar te maken voor de camera.

De eerste 4 objecten zijn bedoeld voor het instellen van de “danger” zones, de object helemaal recht zal gebruikt worden om de desbetreffende “danger” zone te verwijderen, mist de object zich binnen de zone bevind.

Als laatste zijn er 3 verschillende soorten katten. Om zo te simuleren dat er verschillende manier zijn om verschillende soorten katten te schieten.

Laptop

Het programma zal op een laptop gedraait worden, deze laptop staat in verbinding met de camera via USB en met de turret via een Arduino. De laptop heeft een aantal specificaties die vereist zijn om het programma te kunnen draaien, hieronder staat een lijst met vereiste specificaties.

Specificatie	Minimaal	Aanbevolen
CPU	Dual Core 2.2 GHz	Quad Core 2.4 GHz
RAM	2 GB	4 GB
Videokaart	\AMD 3850/Nvidia 260	AMD 4870/Nvidia 370 of hoger
OS	Windows Vista	Windows 8.1

Turret

De turret is een zelf gemaakt apparaat dat met behulp van twee servo motoren op een punt kan richten met een laserpointer. Omdat deze zelf gemaakt is, is het nodig om verschillende materialen hiervoor te krijgen. Ook is toegang tot het fablab nodig om dit te maken.

Benodigdheden

Type	Hoeveelheid	Kosten
Hout	Plaat 4x4 m	7,75

Fablab

Op het moment dat de benodigdheden aanwezig zijn kan er gebouwd worden. Om de turret successvol te bouwen is er toegang tot het fablab nodig om daar het hout met een lasersnijder te snijden en alle in elkaar te zetten. Volgens de planning is het fablab pas in week 5 nodig.

Software

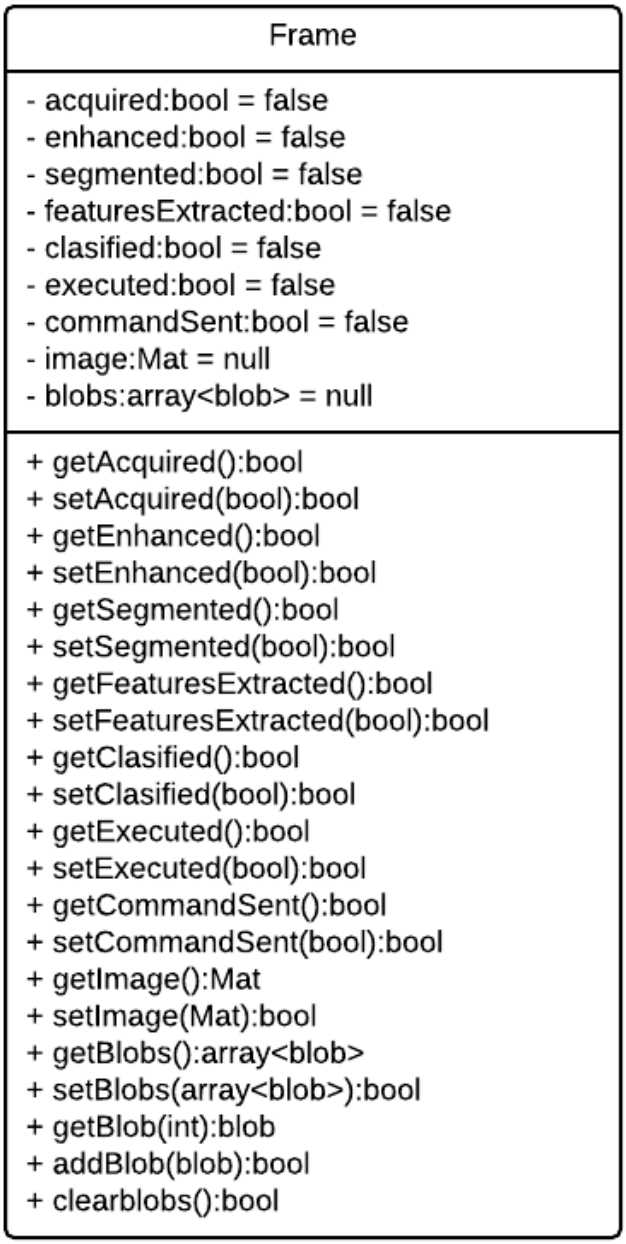
De software bestaat uit drie delen, het Image Processing programma, het Aiming programma en de communicatie daartussen. Hieronder wordt in detail behandeld hoe deze drie in elkaar steken en hoe de software is ontworpen.

Wet-Cat Image Processing

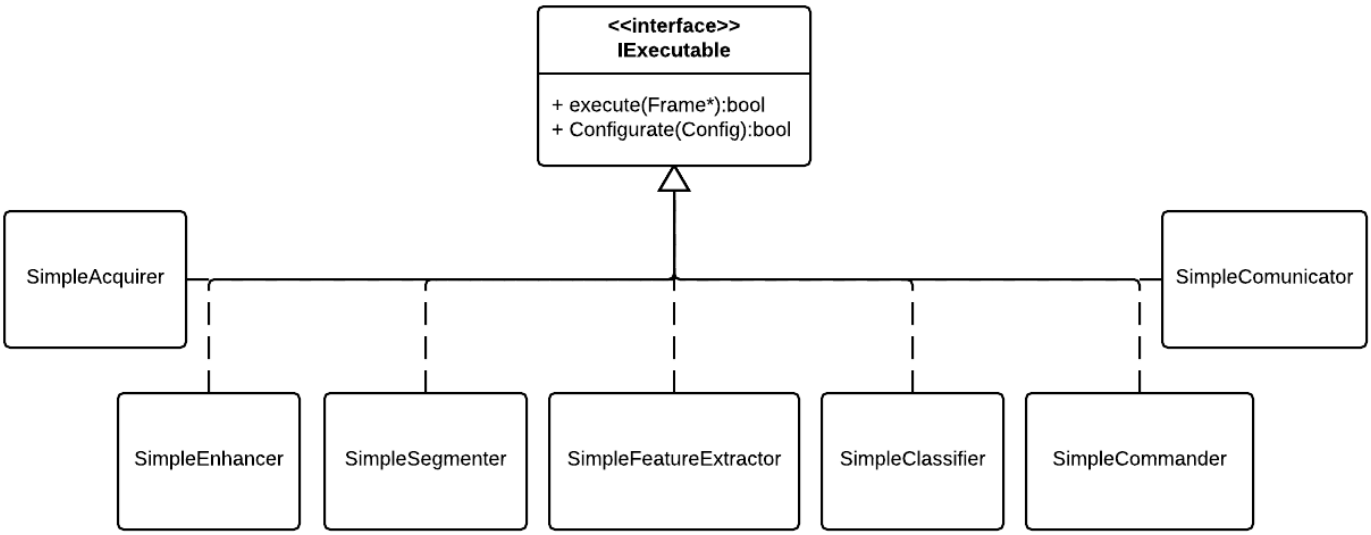
De Wet-Cat Image Processing software is de software die, met behulp van de OpenCV libraries, de beelden van de camera zal gaan behandelen en de relevante informatie hieruit haalt. Deze software kan opgedeelt worden in zeven gedeeltes.

- Acquisition
- Enhancement
- Segmentation
- Feature extraction
- Classification
- Command
- Communication

Elk gedeelte zal een interface krijgen waarmee gecommuniceerd kan worden, op deze manier zijn de delen onafhankelijk van elkaar te gebruiken. Deze structuur is gekozen voor maximale uitbreidbaarheid. Ook zal er een data klasse komen waarin alles word bijgehouden dat nodig is voor een frame, van acquisitie tot communicatie. Hierdoor is het makkelijk om data door te geven tussen de verschillende delen. De data klasse ziet er als volgd uit.



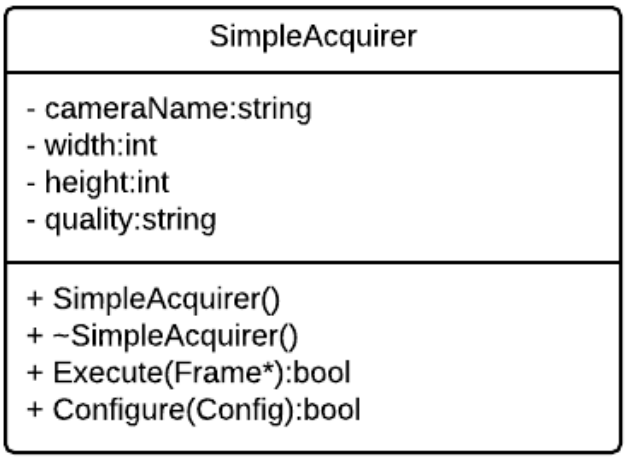
Per onderdeel word de volgende structuur gehanteerd.



Hieronder word per onderdeel de structuur verduidelijkt.

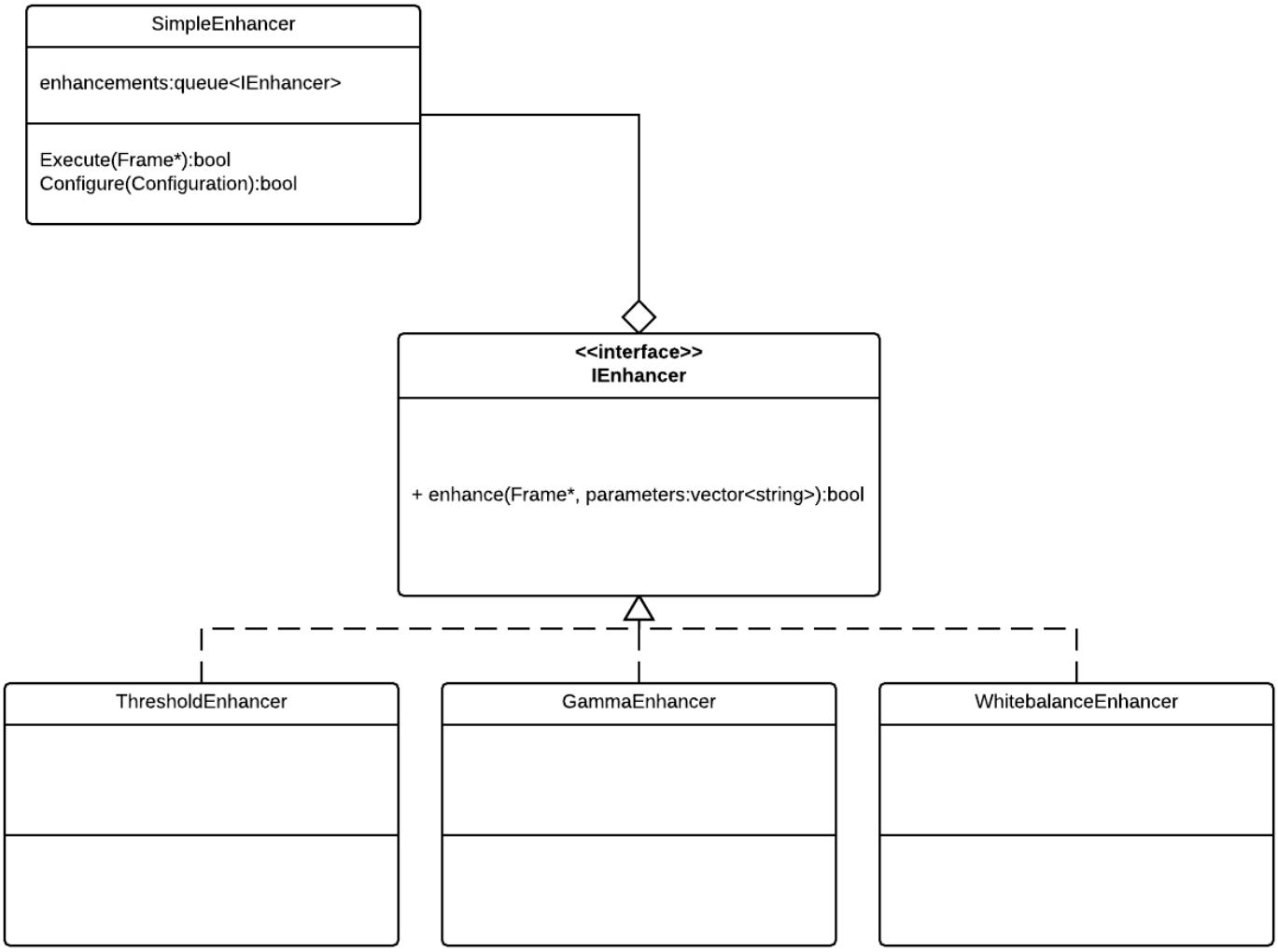
Acquisition

Een acquisitie deel zal vanuit een camera het beeld ophalen en teruggeven. Afhankelijk van wat voor camera gebruikt word en wat voor instellingen gedaan moeten worden kunnen er verschillende acquisitie delen worden gebruikt. In eerste instantie zal er een "Simple Acquisition" deel gemaakt worden dat vanuit de standaard windows camera het beeld pakt dat gebruikt gaat worden. Hieronder staat de klassediagram van de "Simple Acquisitor".



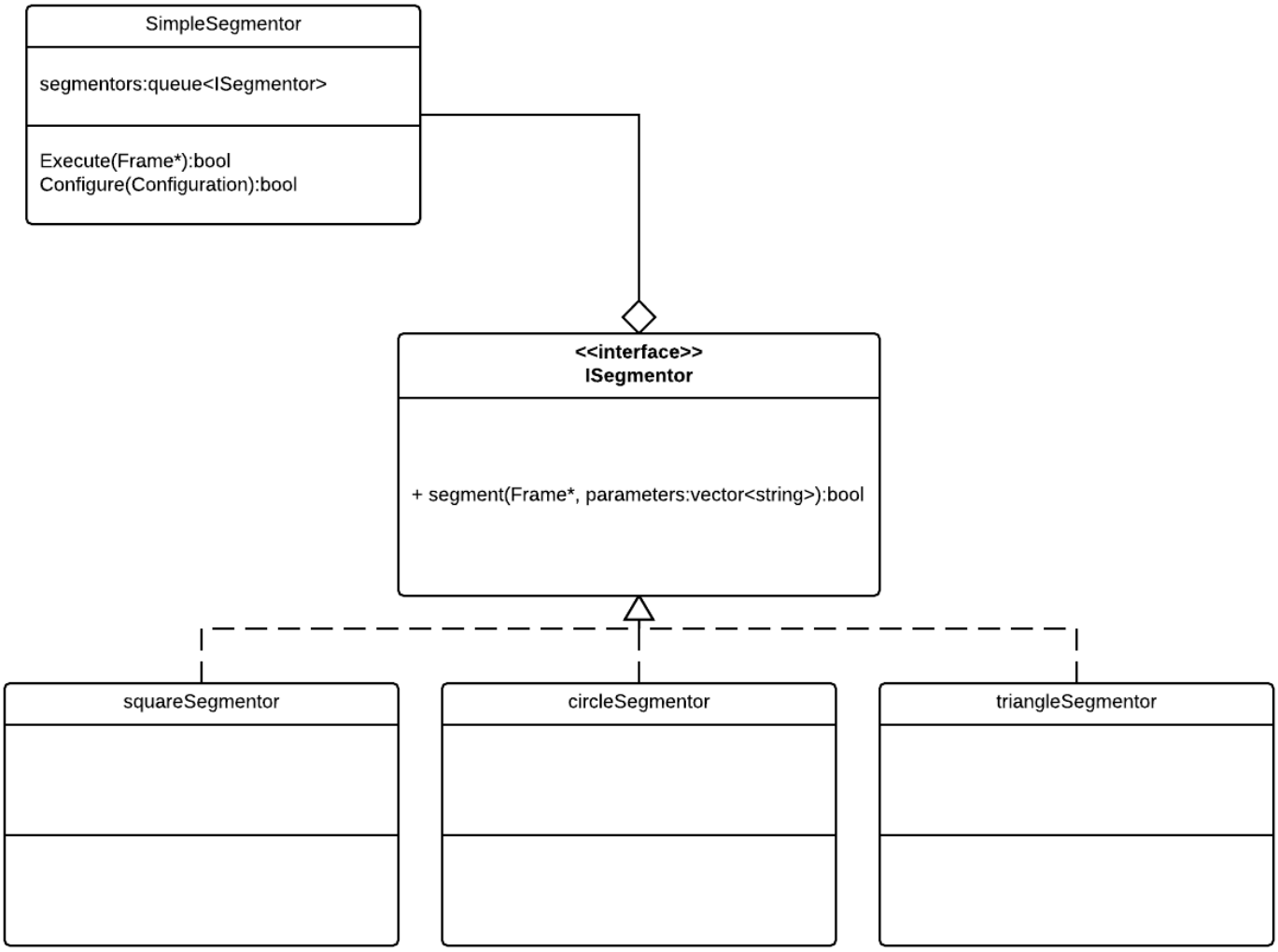
Enhancement

Een Enhancement deel is het deel dat een frame, dat uit de Acquisition stap komt, gaat verbeteren door met filters het beeld te bewerken. Omdat het enhancement is opgebouwd in delen is het dus mogelijk om meerdere enhancements achter elkaar uit te voeren, op deze manier is het makkelijk om verschillende filters te gebruiken in deze stap. Hieronder staat de klassediagram van een enhancement deel.



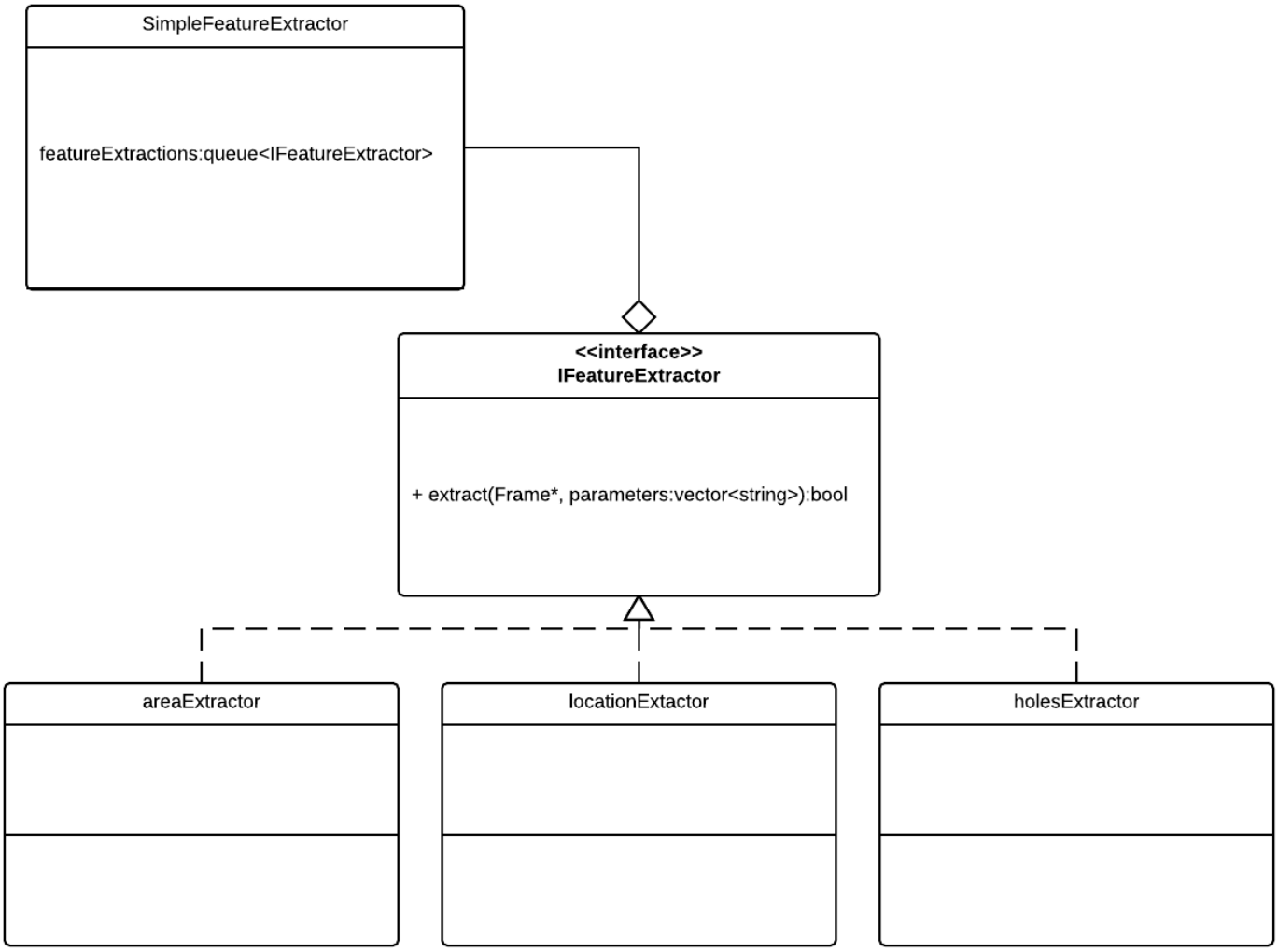
Segmentation

Bij het segmenteren zal er uit het frame, dat optioneel door de enhancement stap gegaan is, belangrijke segmenten gehaald worden. Denk hieraan aan het herkennen van alle ronde voorwerpen om hierna te kijken of een van deze voorwerpen het voorwerp is dat we nodig hebben. Hieronder staat de klassediagram van een segmentator.



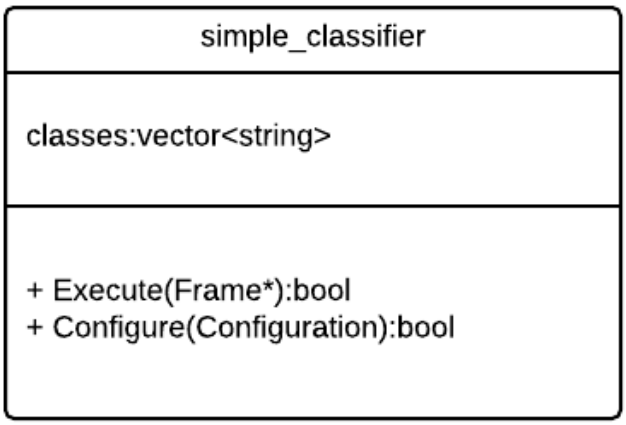
Feature extraction

Om erachter te komen welke segmenten de segementen zijn die nodig zijn voor de applicatie worden de eigenschappen van alle segmenten bepaald. Dit gebeurd door een feature extractor. Doorgaans geeft dit deel een array met features terug. Hironder staat de klassediagram van een feature extractor.



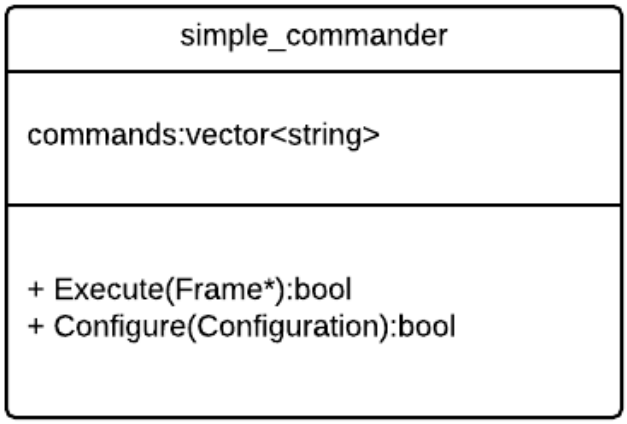
Classification

Als alle eigenschappen van ieder segment bepaald zijn kunnen de segmenten worden geklassificeerd met een classifier. Dit deel zal een naam geven aan elk belangrijk object. Alle segmenten die niet aan de eisen voldoen om geklassificeerd te worden worden op non-actief gezet. Hieronde staat de klassediagram van een classifier.



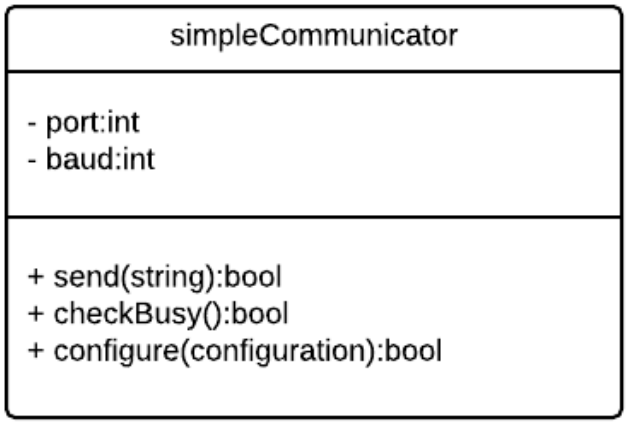
Command

Het commando blok is het eerste blok dat niet meer bij het vision gedeelte hoord. Dit blok zal bepalen welk object het belangrijkste is op dit moment (als er meerdere objecten zijn) en zal een commando aan het object binden. Dit commando zal dan moeten worden uitgevoegd, als het uitvoeren van een commando klaar is zal er een feedback moeten zijn zodat het command deel weer een nieuw commando mag geven. Hieronder staat de klassediagram van een commander.



Communication

Als laatse deel van de Wet-Cat Image Processing Softwate is er een communicatie deel, dit deel zal het commando vertalen naar een commando voor de turret. Ook zal dit blok worden gebruikt voor directe communicatie met de turret op het moment dat er gecalibreerd word. Hieronder staat een klassediagram van een communicator.



Wet-Cat Aiming

De Wet-Cat Aiming Software is de software die op de microcontroller voor de turret gaat draaien. Deze software zal worden geschreven in C++ en word op een STM32F4Discovery gedraaid. Hieronder staat een overzicht van alle hardware die nodig is om de turret te kunnen maken.

Onderdeel	Artikel nr.	Aantal	Prijs (EUR)
Servo - HiTec HS-422 (Standard Size)	ROB-11884	2	15,36
Magnet Square	COM-08643	4	4,63
Laser Card Module	COM-00594	1	5,36
Machine Screw - Socket Head (6-32 ; 7/16"; 25 pack)	ROB-12485	1	1,61
Ball Bearing - Flanged (1/4" Bore, 1/2" OD, 2-Pack)	ROB-13012	2	3,38
Shaft - Solid (Stainless; 1/2"D x 2"L)	ROB-12212	1	1,54
Set Screw Hub - 1/2" Bore	ROB-12494	4	15,41
Servo Shaft Attachment - Hitec Standard	ROB-12464	2	12,33
		Totaal:	59,62

Een aantal onderdelen hiervan zullen we proberen om op een CNC machine te laten maken, deze onderdelen zijn`Shaft`, `Set Screw Hub` en `Servo Shaft Attachment` Als deze onderdelen op de CNC machine gemaakt kunnen worden zal dit erg veel kosten schelen.

Protocollen

Image Processing <-> Aiming

Om de Wet-Cat Image Processing (WCIP) en de Wet-Cat Aiming (WCA) software met elkaar te laten communiceren is er een protocol opgesteld waaraan voldaan moet worden. Om het protocol inzichtelijk te houden hebben we per functionaliteit een commando gedefineerd.

STOP

Dit is een commando dat aangeeft dat de turret per direct moet stoppen waar hij mee bezig is en naar zijn standaard positie moet terugkeren. Dit commando word, bijvoorbeeld, gebruikt op het moment dat de kat die gedetecteerd werd weg springt. Als de turret op dat moment nog aan het schieten is zal deze moeten stoppen.

Het commando ziet er als volgd uit:

```
stop;
```

AIM & FIRE

Dit commando geeft de coordinaten van het doel door en geeft het commando om te schieten door. Het commando om te schieten hangt af van de verschillende soorten doelen, de coordinaten zullen tussen 0 en 800 vallen.

Het commando ziet er als volgd uit, met X als X coordinaat en C als schiet commando:

```
C,X,Y;
```

DONE

Het `Done` commando is het commando dat de WCA terug geeft aan de WCIP op het moment dat hij klaar is met

schieten, na dit commando weet de WCIP dat de WCA weer vrij is om op een ander doel te schieten.

Het commando ziet er als volgd uit:

```
done;
```

MOVEMENT

De turret heeft de mogelijk om vrij rond te bewegen, dit is nodig om deze te callibreren. De commando's voor het instellen zijn de pijl toetsen op een toetsenbord. Via de seriële communicatie zal de volgende charakters worden meegestuurd: 65 - 68.

SET

Als laatste is er een commando om de hoeken van het bereik van de WCA te bepalen. Ook dit is om de WCA te calibreren. Op het moment dat er een hoek word ingesteld zal het huidige punt het maximale punt worden in die richting. Er zal 4 punten worden gecalibreerd. De waarde 51 - 54, word gebruikt om de verschillende hoeken vast te stellen.

Vanuit de computer word dan de pixel waarde tenopzichte van x en y gestuurd. De opbouw van de commande ziet er als volgd uit:

```
C, X, Y;
```

Code Guidelines

Binnen dit project gaan we C++ code schrijven, om te zorgen dat alle code gelijk blijft zijn er een aantal richtlijnen waaraan de code moet voldoen. Dit zorgd voor een overzichtelijke en nette code. Hieronder staat een lijst met richtlijnen

Richtlijn	Toepassing	Voorbeeld
Klasse namen met hoofdletter	Binnen een Klasse	`class Test`
Functies in camelcase	Alle code	`public void getAllVariablesFromClass()`
Variablen namen in camelcase	Alle code	`public thisIsAVariable;`
Parameters met hoofdletters en in camelcase	Alle code	`Public void function(int AnInteger, string ABigStringAsParameter)`