

**EMBARGO:**

Deze masterproefscriptie staat onder embargo tot 25-10-2015

# Mobile Deep Visual Detection and Recognition

Ondertitel (facultatief)

**Thijs VERCAMMEN**

Promotor(en): Prof. dr. ir. Toon Goedéme

Masterproef ingediend tot het behalen van  
de graad van master of Science in de

Co-promotor(en): Prof. dr. ir. Floris De Feyter

industriële wetenschappen: Elektronica-ICT  
ICT

Academiejaar 2021 - 2022

©Copyright KU Leuven

Zonder voorafgaande schriftelijke toestemming van zowel de promotor(en) als de auteur(s) is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, kan u zich richten tot KU Leuven Technologiecampus De Nayer, Jan De Nayerlaan 5, B-2860 Sint-Katelijne-Waver, +32 15 31 69 44 of via e-mail [iiw.denayer@kuleuven.be](mailto:iiw.denayer@kuleuven.be).

Voorafgaande schriftelijke toestemming van de promotor(en) is eveneens vereist voor het aanwenden van de in deze masterproef beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

# **Voorwoord**

Het voorwoord vul je persoonlijk in met een appreciatie of dankbetuiging aan de mensen die je hebben bijgestaan tijdens het verwezenlijken van je masterproef en je hebben gesteund tijdens je studie.

# Samenvatting

De (korte) samenvatting, toegankelijk voor een breed publiek, wordt in het Nederlands geschreven en bevat **maximum 3500 tekens**. Deze samenvatting moet ook verplicht opgeladen worden in KU Loket.

# Abstract

Het extended abstract of de wetenschappelijke samenvatting wordt in het Engels geschreven en bevat **500 tot 1.500 woorden**. Dit abstract moet **niet** in KU Loket opgeladen worden (vanwege de beperkte beschikbare ruimte daar).

**Keywords:** Voeg een vijftal keywords in (bv: Latex-template, thesis, ...)

# Inhoudsopgave

|  |      |
|--|------|
| <b>Voorwoord</b>   | iii  |
| <b>Samenvatting</b>  | iv   |
| <b>Abstract</b>  | v    |
| <b>Inhoud</b>  | vii  |
| <b>Figurenlijst</b>  | viii |
| <b>Tabellenlijst</b>   | ix   |
| <b>Symbolenlijst</b>   | x    |
| <b>Lijst met afkortingen</b>                                     | xi   |
| <b>1 Situering en doelstelling</b>                               | 1    |
| 1.1 Situering . . . . .  | 1    |
| 1.2 Probleemstelling . . . . .                                   | 1    |
| 1.3 Doelstellingen . . . . .                                     | 1    |
| <b>2 Herkenning en Detectie Algemeen</b>                         | 3    |
| 2.1 Deep learning-gebaseerde herkenningssystemen . . . . .       | 3    |
| 2.1.1 convolutioneel neuraal netwerk (CNN) . . . . .             | 3    |
| 2.1.2 Trainen van een CNN . . . . .                              | 5    |
| 2.1.3 Herkenning . . . . .                                       | 5    |
| 2.2 Deep learning-gebaseerde detector . . . . .                  | 6    |
| 2.2.1 Two-stage detector . . . . .                               | 6    |
| 2.2.2 One-stage detector . . . . .                               | 7    |
| <b>3 Herkenning en detectie implementatie op mobiel platform</b> | 9    |

|                                       |           |
|---------------------------------------|-----------|
| <b>4 Specificaties</b>                | <b>10</b> |
| 4.1 Algemene richtlijnen . . . . .    | 10        |
| <b>5 Richtlijnen voor formules</b>    | <b>12</b> |
| <b>6 Richtlijnen voor referenties</b> | <b>13</b> |
| <b>A Uitleg over de appendices</b>    | <b>15</b> |

# Lijst van figuren

|     |  |    |
|-----|--|----|
| 2.1 | CNN met 2 convolutie lagen en 2 pooling lagen en een fully-connected layer . . . . . | 3  |
| 2.2 | Convolutie laag waarbij een filter wordt herleid tot een output feature. . . . .     | 4  |
| 2.3 | ReLU, waarbij het maximum wordt genomen van 0 en de input waarde. . . . .            | 5  |
| 2.4 | R-CNN . . . . .  | 6  |
| 2.5 | Faster R-CNN . . . . .   | 7  |
| 2.6 | One-stage detector met VGG net backbone . . . . .                                    | 7  |
| 4.1 | Dit is een voorbeeld van een figuur-float . . . . .                                  | 11 |

# **Lijst van tabellen**

|  |    |
|--|----|
| 4.1 Dit is een voorbeeld van een tabel . . . . . | 11 |
|--|----|

# Lijst van symbolen

Maak een lijst van de gebruikte symbolen. Geef het symbool, naam en eenheid. Gebruik steeds SI-eenheden en gebruik de symbolen en namen zoals deze voorkomen in de hedendaagse literatuur en normen. De symbolen worden alfabetisch gerangschikt in opeenvolgende lijsten: kleine letters, hoofdletters, Griekse kleine letters, Griekse hoofdletters. Onderstaande tabel geeft het format dat kan ingevuld en uitgebreid worden. Wanneer het symbool een eerste maal in de tekst of in een formule wordt gebruikt, moet het symbool verklaard worden. Verwijder deze tekst wanneer je je thesis maakt.

|          |                                   |                    |
|----------|-----------------------------------|--------------------|
| <i>b</i> | Breedte                           | [mm]               |
| <i>A</i> | Oppervlakte van de dwarsdoorsnede | [mm <sup>2</sup> ] |
| <i>c</i> | Lichtsnelheid                     | [m/s]              |

# **Lijst van afkortingen**

CNN Rols ReLu

# **Hoofdstuk 1**

## **Situering en doelstelling**

### **1.1 Situering**

Tegenwoordig wordt deep learning steeds meer en meer gebruikt om beeldverwerking problemen op te lossen. Via neurale netwerken kunnen we met meer en betere features werken om de afbeeldingen te analiseren. Maar veel van deze modellen hebben behoorlijk wat rekenkracht en geheugen nodig om te werken. Ook is er steeds meer interesse naar real-time toepassingen waarvan het resultaat zo snel mogelijk beschikbaar moet zijn. Dit wordt moeilijk bij veel hedendaagse systemen waarbij de foto eerst genomen moet worden en vervolgens door een computer geanalyseerd moet worden, omdat hedendaagse systemen veel rekenwerk en geheugen vragen. In deze masterproef wordt er onderzocht of de computer kan weggeleggen worden en de afbeelding meteen door het mobiel apparaat geanalyseerd kan worden. Dus er moet onderzocht worden hoe een bestaand model aangepast kan worden om efficiënt te werken op een mobiel apparaat. Hierbij moet vooral rekening gehouden worden met de rekenkracht en geheugen van het mobiele apparaat.

### **1.2 Probleemstelling**

Mobiele apparaten zijn kleine toestellen met beperkt geheugen en beperkte rekenkracht. In deze masterproef wordt er onderzocht hoe het rekenwerk beperkt kan worden zodat het resultaat real-time geleverd kan worden. Er gaat ook onderzocht worden hoe alle data efficiënt kan worden opgeslagen op het toestel.

### **1.3 Doelstellingen**

Het uiteindelijke doel van deze masterproef is er voor zorgen dat een bestaand deep learning model aangepast kan worden zodat dit real-time resultaten kan geven op een mobiel apparaat. Dit gebeurt aan de hand van de volgende stappen:

- grondig begrijpen van een deep learning herkenningsysteem

- grondig begrijpen van een deep learning detectiesysteem
- Onderzoeken welke technieken er gebruikt kunnen worden om bestaande systemen op een mobiel apparaat te implementeren.
- onderzoeken voor optimalisaties voor een herkenningsysteem
- onderzoeken voor optimalisaties voor een detectiesysteem
- gevonden technieken testen en analiseren
- werkend prototype applicatie ontwerpen voor een mobiel apparaat

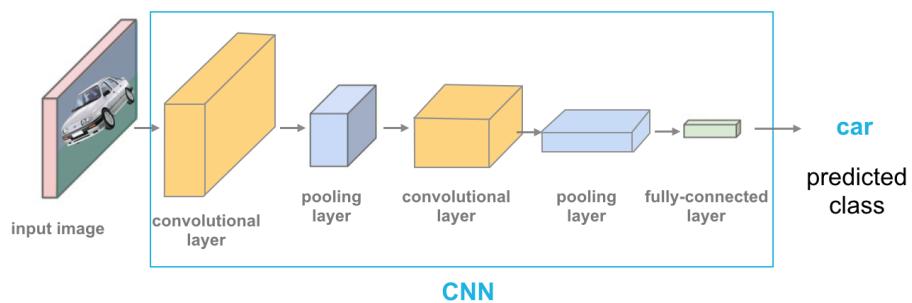
## Hoofdstuk 2

# Herkenning en Detectie Algemeen

## 2.1 Deep learning-gebaseerde herkenningssystemen

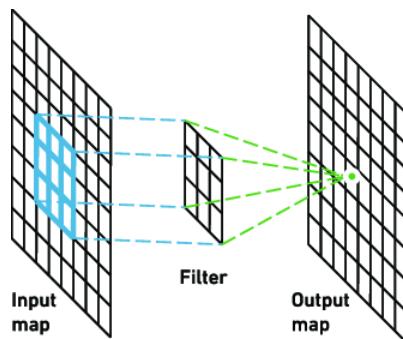
Herkenningssystemen of classificatie systemen voorspellen wat de klasse van een object is in een afbeelding. Dus het herkennen van objecten in digitale afbeeldingen zonder deze te localiseren of aan te duiden. Voor een herkenningssysteem is er een goed getraind netwerk nodig dat input afbeeldingen omzet in features. Er moet een database zijn met daarin de gegevens van de objecten die men wilt herkennen. Vervolgens hebben is er ook een methode nodig om features van het neurale netwerk te vergelijken met de gegevens in de database om het juiste object te herkennen.

### 2.1.1 convolutioneel neuraal netwerk (CNN)



Figuur 2.1: CNN met 2 convolutie lagen en 2 pooling lagen en een fully-connected layer

De belangrijkste bouwsteen van een herkenningssysteem is een goed getrainde CNN (figuur 2.1). In tegenstelling tot fully connected netwerken wordt bij een CNN de gewichten gedeeld over verschillende locaties om zo het aantal parameters te verminderen.



**Figuur 2.2:** Convolutie laag waarbij een filter wordt herleid tot een output feature.

### 2.1.1.1 Convolutie laag

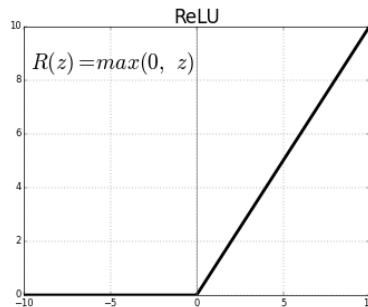
Het belangrijkste deel van een CNN zijn de convolutielagen (figuur 2.2) waarbij men een kernel/filter over de input laag gaan wat als output een feature map genereert. Een kernel bestaat uit set van gewichten die met de input worden vermenigvuldigd, deze kernel wordt over de input afbeelding geschoven. Al de pixels binnen het veld van de kernel worden gereduceert tot een enkele waarde. CNN leren verschillende features met verschillende kernels in parallel. Waardoor de matrices met feature mappen steeds kleiner worden maar ook dieper worden. Een andere factor van een convolutie laag is de stride, deze waarde geeft aan met hoeveel pixels de kernel telkens moet doorschuiven. Een CNN bestaat uit een opeenvolging van een aantal convolutie lagen die steeds meer high-level features extraheren. Hoe meer convolutielagen een netwerk telt hoe meer features er uit de input worden gehaald, maar hoe trager het netwerk is.

### 2.1.1.2 Lineare activatie fucntie

Elke convolutie laag wordt gevolgd door een niet-lineaire activatie functie, de meest gebruikt functie hiervoor is de rectified linear unit (ReLU) (figuur 2.3). De ReLU wordt vaak gebruikt omdat deze eenvoudig is, kan exact 0 weergeven en ziet er linear uit.  $\text{Max}(0,x)$  is de ReLU bewerking, dus er wordt verdergegaan met 0 of de input waarde. Zonder een niet-lineaire activatie functie kan het CNN herleid worden tot 1 convolutie laag die geen high-level features kan extraheren. Andere mogelijkheden voor Lineare activatie functies zijn: Sigmoid en Tangens hyperbolicus maar deze functies vragen meer rekenwerk.

### 2.1.1.3 Pooling laag

Een volgende bouwsteen is de pooling laag waarbij het aantal samples in de feature map wordt verlaagt. De meest voorkomende methode is max-pooling waarbij er verder wordt gegaan met de maximum waarde in een bepaalde regio. Het doel van een pooling laag is om het aantal parameters te verminderen en zo ook het rekenwerk te verminderen. Er kan ook gebruik gemaakt worden van avarage pooling waarbij er verder wordt gegaan met de gemiddelde waarde van een regio. Er is ook minimal pooling waarbij er verder wordt gegaan met de minimum waarde.



**Figuur 2.3:** ReLu, waarbij het maximum wordt genomen van 0 en de input waarde.

#### 2.1.1.4 Fully-connected lagen

Op het einde van elk CNN volgen er meestal 1 of meerdere fully connected lagen. Deze lagen connecteren elke input van één laag met elke activatie eenheid van de volgende laag. Dit zorgt voor meer parameters en meer rekenwerk waardoor deze lagen een vertragende factor vormen. De fully connected lagen gaan niet-lineaire combinaties leren van de features van de convolutie lagen. De fully connected lagen zorgen voor een classificatie op basis van de features van de convolutie lagen.

### 2.1.2 Trainen van een CNN

Het trainen van een CNN bestaat uit het leveren van veel voorbeelden aan het netwerk. Op basis van het resultaat van deze voorbeelden worden telkens de gewichten van de kernels aangepast, zodat er steeds een beter resultaat wordt geleverd.

#### 2.1.2.1 Back propagation

De loss functie geeft de error van de voorspelling weer tijdens het trainen van een neuraal netwerk. Op basis van de loss functie worden er gradienten berekend die gebruikt worden om de gewichten van het netwerken aan te passen. De gradienten worden berekend door de loss af te leiden via de ketting regel. De gewichten worden aangepast zodat de loss geminimaliseerd wordt dit process noemt men back propagation

### 2.1.3 Herkenning

Eens dat er een getraind CNN is kunnen we verdergaan met de effectieve herkenning. Als men bepaald objecten in een afbeelding wil ontdekken gaat men met behulp van het CNN de afbeelding omzetten in een embedding. Embeddings zijn vector representaties die kunnen worden vergeleken in een embedding space, waar gelijkaardige objecten dichter bij elkaar liggen. De embedding van de input afbeelding wordt vergeleken met de embeddings die zich in een gallerij bevinden. De gallerij is een database die alle mogelijke objecten die men wilt herkennen bevat. Met behulp van

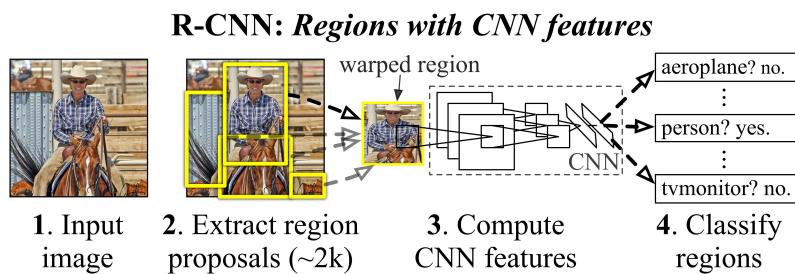
een query kunnen we gelijkaardige objecten uit de gallerij halen om deze te gaan vergelijken in een embedding space. Gelijkaardige embeddings kunnen gezocht worden via de nearest neighbour techniek, waar we naar de klasse van de dichtsbijzijnde buur gaan kijken.

## 2.2 Deep learning-gebaseerde detector

Object detectie is het localiseren en classificeren van objecten in een afbeelding, waarbij de objecten aangeduid worden met een Bounding box. Door gebruik te maken van CNN kunnen er vrij nauwkeurige object detectoren ontworpen worden. Object detectie maakt voornamelijk gebruik van twee methodes: de single-stage detector en de two-stage detector.

### 2.2.1 Two-stage detector

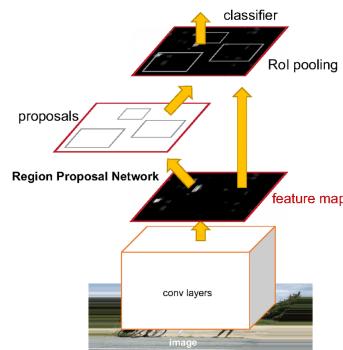
Zoals de naam zegt bestaat deze methode uit 2 niveaus. Het eerste deel worden er Regions of Interest (Rols) gecreëerd, dit is het filteren van regio's waarbij de kans groot is dat deze een object bevatten. Het tweede deel classificeert en verfijnt de localisatie van de Rols die in het eerste deel gecreëerd werden. Dit gebeurt door elk van de Rols door een CNN te voeren. Region-based Convolutional Neural Network (R-CNN) is het basis principe van de two-stage detectoren weergegeven in figuur 2.4. Hierbij wordt met een region proposal algoritme regios uit de afbeelding gefilterd waar de kans groot is dat er objecten op staan.



Figuur 2.4: R-CNN

R-CNN is een trage detector vermits elke Rols door een CNN moet gaan. Deze methode is geëvolueerd tot de veel snellere methode Faster R-CNN (figuur 2.5). Hierbij wordt de afbeelding door een CNN behandelt en vervolgens maakt men gebruik van een Region Proposal Network (RPN). Het RPN gaat zoals bij R-CNN regios uit de afbeelding filteren waar de kans groot is dat er objecten opstaan, maar het RPN werkt sneller en levert betere resultaten.

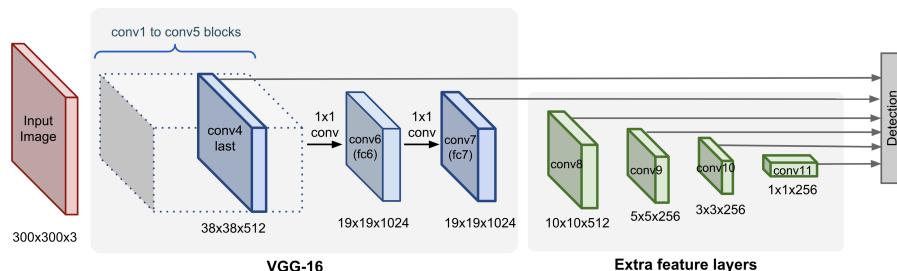
1. CNN waarbij men eerst features uit de afbeelding haalt
2. RPN waarbij een netwerk op basis van de features Rols genereert
3. Rol pooling
4. Classificatie en regresie laag: verfijnt de posities van de Rols



Figuur 2.5: Faster R-CNN

## 2.2.2 One-stage detector

Bij one-stage detectoren gebeurt object detectie in één keer met één neuraal netwerk. Dus er is geen region proposal niveau meer zoals bij de two-stage detector (figuur 2.6). One-stage detectoren zijn sneller dan two-stage detectoren omdat ze alles in één keer doen, maar kunnen wat in nauwkeurigheid verliezen t.o.v. two-stage detectoren.



Figuur 2.6: One-stage detector met VGG net backbone

One-stage detectie netwerken bestaat uit 2 delen: een backbone en 1 of meerder convolutie lagen. De backbone van het netwerk is een getraind CNN classificatie netwerk dat de als feature extractor dient. Dit is een CNN waarvan de fully-connected lagen zijn weggelaten in figuur 2.6 vormt VGG de backbone van de single-stage detector. De extra convolutie lagen gaan de objecten detecteren op basis van de feature mappen die de output zijn van de backbone.

De input afbeelding wordt opgedeeld in een rooster, voor elke cel van het rooster gaat men de klasse en locatie van het object voorspellen. Als er geen object is wordt de cel gezien als een achtergrond klasse. Voor elke locatie van een object wordt er een vaste set van anchor/bounding boxen geëvalueert. Vervolgens is er nog een methode nodig voor de overbodige bounding boxen te verwijderen. Een eerste mogelijkheid is door enkel bounding boxen te tekenen waarvan de voorspelling boven een threshold ligt. Een andere methode is non-maxima suppression deze methode zorgt ervoor dat elk object maar één bounding box heeft. Deze techniek houdt enkel de bounding box over met de beste voorspelling en onderdrukt de rest van de bounding boxen. De twee bekendste technieken van one-stage detectie zijn: You Only Look Once (YOLO) en Single Shot Detection

(SSD).

## **Hoofdstuk 3**

# **Herkenning en detectie implementatie op mobiel platform**

# **Hoofdstuk 4**

## **Specificaties**

### **4.1 Algemene richtlijnen**

Alle figuren en tabellen worden genummerd en binnen een float omgeving geplaatst  
(`\begin{figure} figuurcontent \end{figure}`)

Foto's, grafieken, schema's,... worden alle onder de benaming 'Figuur' gecatalogeerd.

Het is belangrijk dat tabellen en figuren duidelijk zijn en dat ze alle informatie bevatten die nodig is om ze te begrijpen.

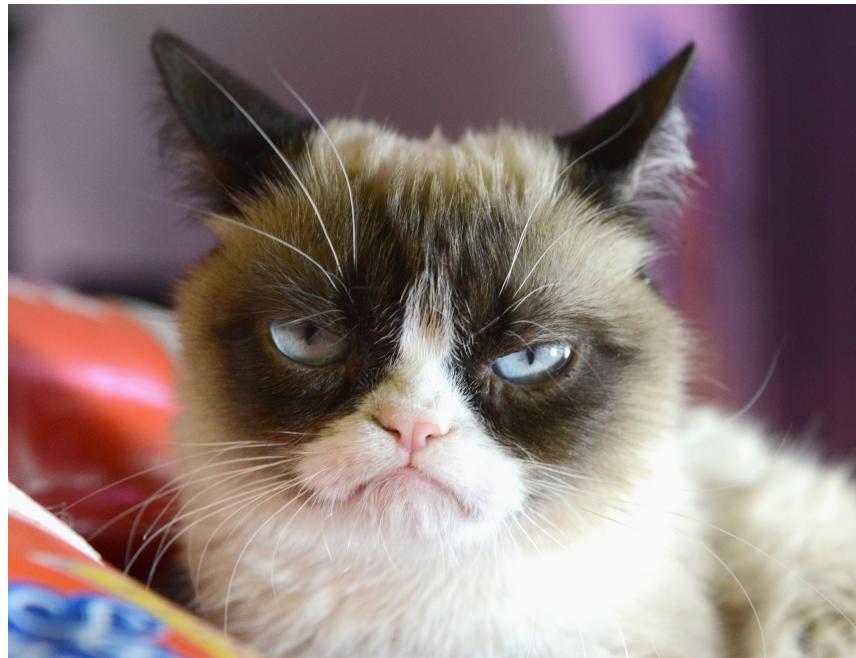
Tabellen worden bij voorkeur niet gesplitst over twee bladzijden. Indien een tabel niet op één bladzijde past, wordt het bijschrift op de volgende bladzijde hernomen en aangevuld met (vervolg). Ook de kolomkoppen van de tabel worden hernomen.

In de tekst wordt naar alle tabellen en figuren verwezen met het itemnummer. Schrijf dus niet 'onderstaande figuur toont....', maar wel 'Figuur 3.1 toont....'. Doe dit door gebruik te maken van de commando's `\label{}` en `\ref{}`. Geef figuren ook zinvolle captions (`\caption{Caption}`). Figuren worden gezentreerd op de bladzijde. Ook het bijschrift wordt gezentreerd en onder de figuur geplaatst. Na de figuurnummer volgt een de beschrijving van de figuur.

Figuur 4.1 toont een voorbeeld gegeven van een float omgeving voor een figuur. Hieronder wordt de syntax weergegeven.

```
\begin{figure} [!ht]
\centering
\includegraphics[width=0.75\linewidth]{image.jpg}
\caption{Dit is een voorbeeld van een figuur-float}
\label{fig:VoorbeeldFigFloat}
\end{figure}
```

Tabellen worden links uitgelijnd op de bladzijde. Ook het bijschrift wordt links uitgelijnd en boven de tabel geplaatst. Na de tabelnummer volgt de beschrijving van de tabel. Tabel 4.1 toont een voorbeeld van een eigen tabel. Vermijd om tabellen te kopieëren van andere werken, maar herwerk ze en plaats de nodige bronvermelding. De nodige syntax om tabel 4.1 te generen wordt hieronder



**Figuur 4.1:** Dit is een voorbeeld van een figuur-float

weergegeven:

```
\begin{table} [!ht]
\caption{Dit is een voorbeeld van een tabel}
\begin{tabular}{ccc}
\hline
Kolom 1 & Kolom 2 & Kolom 3 \\
\hline
1 & 2 & 3 \\
4 & 5 & 6 \\
\hline
\end{tabular}
\label{tab:VoorbeeldTableFloat}
\end{table}
```

Tot slot, let er op dat er expliciet naar elke tabel en figuur verwezen wordt vanuit de tekst.

**Tabel 4.1** Dit is een voorbeeld van een tabel

| Kolom 1 | Kolom 2 | Kolom 3 |
|---------|---------|---------|
| 1       | 2       | 3       |
| 4       | 5       | 6       |

## **Hoofdstuk 5**

# **Richtlijnen voor formules**

## **Hoofdstuk 6**

### **Richtlijnen voor referenties**

# **Bibliografie**

## **Bijlage A**

# **Uitleg over de appendices**

Bijlagen worden bij voorkeur enkel elektronisch ter beschikking gesteld. Indien essentieel kunnen in overleg met de promotor bijlagen in de scriptie opgenomen worden of als apart boekdeel voorzien worden.

Er wordt wel steeds een lijst met vermelding van alle bijlagen opgenomen in de scriptie. Bijlagen worden genummerd het een drukletter A, B, C,...

Voorbeelden van bijlagen:

Bijlage A: Detailtekeningen van de proefopstelling

Bijlage B: Meetgegevens (op USB)

FACULTEIT INDUSTRIËLE INGENIEURSWETENSCHAPPEN  
CAMPUS DE NAYER SINT-KATELIJNE-WAVER  
J. De Nayerlaan 5  
2860 SINT-KATELIJNE-WAVER, België  
tel. + 32 15 31 69 44  
[iw.denayer@kuleuven.be](mailto:iw.denayer@kuleuven.be)  
[www.iw.kuleuven.be](http://www.iw.kuleuven.be)

