**Proof Of Concept: Image Recognition**

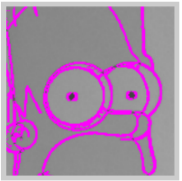
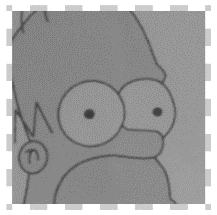
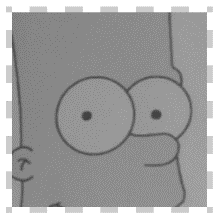
**Doel:**

Een image recognition web app maken die karakters van de Simpsons kan herkennen.

**Eerste idee:**

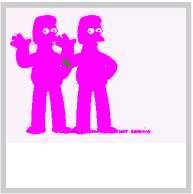
Na wat googelen vond ik een goede API die me kon helpen om het verschil tussen 2 afbeeldingen weer te geven. Resemble.js:

<https://huddle.github.io/Resemble.js/>



Met deze library kan je het verschil tussen 2 foto’s weergeven. De eerst 2 zijn de foto’s die je hebt vergeleken. De laatste foto is het resultaat. Er wordt ook een procent mee gegeven in welke mate de foto’s verschillend zijn. Dit is het eerste waarmee ik aan de slag ben gegaan.

Probleem 1:



De gefotografeerde afbeelding kan exact hetzelfde zijn maar toch een 100% verschil aangeven.

Zoals hierboven te zien worden de 2 foto’s met elkaar vergeleken en is het personage van de 2 afbeeldingen hetzelfde maar toch zal resemble.js zeggen dat de foto’s totaal niet gelijkend zijn.

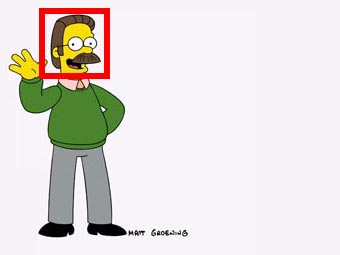
Dit is zeer logisch, resemble.js gaat pixel per pixel vergelijken en berekent zo hoe groot het verschil is. Deze manier is dus niet bruikbaar wanneer je de gebruiker zelf foto’s gaat laten nemen. De foto’s gaan te hard van elkaar verschillen.

Mogelijke oplossing:

<https://trackingjs.com/>

Trackingjs is een javascript bibliotheek die aan fotoanalyse doet. Zo kan je locaties van bepaalde kleuren in een foto opvragen.

Met Trakingjs kan je ook een hoofd zoeken in een foto en enkel de data Uri van het hoofd opslaan. Met deze methode zou je 2 verschillende foto’s van hetzelfde personage kunnen nemen en enkel het verschil tussen de 2 hoofden meten. Zoals je kan zien aan de foto’s hieronder zijn de 2 personages een beetje verplaatst in locatie maar hun hoofd blijft hetzelfde.



Deze methode werkt gedeeltelijk, als je ze uitvoert met 2 mensen. Dit voorbeeld kan je zien in FirstIdea.html. Hierbij lees ik 9 foto’s in waarbij ik enkel de data Uri van het hoofd opsla.

Als je dan een foto wil vergelijken, zoek je eerst naar het hoofd van de foto en dan vergelijk je het hoofd met elke data Uri die je eerder hebt opgeslagen. De foto met het kleinste verschil toon ik om aan te geven wie de persoon in kwestie is.

Problemen:

Het werkt slecht. Eerst en vooral wordt het hoofd vaak niet gevonden en ten tweede is de vergelijking tussen 2 foto’s nog steeds te letterlijk. Vaak als 2 verschillende personen hun hoofd op dezelfde manier houden, worden de 2 als gelijk gezien.

Het werkt ook enkel met foto’s van mensen. Als er een cartoonpersonage wordt geanalyseerd wordt het hoofd niet herkend en dus zal het niet meer werken.

**Tweede Idee:**

<https://kairos.com/>

Kairos is een Face analysis API waarmee je ook gezichtsherkenning kan doen. Hiermee heb ik een kleine testapplicatie gemaakt. Deze vind je terug in kairosResemble.html

Je kan met deze API een galerij opslaan van foto’s en deze vergelijken met een nieuwe zelf gekozen foto. Dit werkt zeer goed, de gezichtsherkenning gaat uitstekend. Het probleem is weer dat de API gemaakt is voor foto’s van mensen. Als je een cartoon gaat analyseren wordt het gezicht niet herkend en kan er dus geen herkenning plaatsvinden.

Voor foto’s van mensen werkt het zeer goed en snel.

**Derde idee:**

Aangezien face tracking het probleem van wijziging in grootte of wijziging van locatie niet oplost, heb ik naar een andere oplossing gezocht: de user verplichten om de foto op dezelfde locatie te nemen.



Dit heb ik gedaan door op het beeld van de webcam een vierkant te plaatsen met 2 bolletjes om de pupillen van de cartoon te zetten. Hierdoor zal voor elk karakter de locatie miniem veranderen en de grootte ongeveer gelijk zijn.

Mijn idee was om deze methode te gebruiken en dan met behulp van resemble.js de foto’s te vergelijken met een databank vol foto’s en hun bijhorende namen.

Ik ben hieraan begonnen door een laravel project te starten en een databank aan te maken met 1 table: Characters. Aangezien ik voor resemble.js enkel de data Uri van de foto nodig heb, heeft de table 2 rows: “name” en “dataUri”. Bij het laden van de website zou ik al de data Uri’s opvragen en dan kan ik gemakkelijk een gemaakte foto vergelijken met al de foto’s van de databank.

Probleem:

Ik kan de fotoherkenning niet in de frontend uitvoeren. Als mijn databank bijvoorbeeld 400 personages bevat, zal ik bij het laden van de website 400 data Uri’s moeten opvragen voor ik begin met de herkenning. Deze manier is zeker niet performant.

Oplossing:

De data analyse en vergelijking moet in de backend gebeuren. Ik heb dit gedaan met behulp van Python.

Versie 1:

Ik heb met Python een wiskundige methode gevonden die 2 foto’s met elkaar vergelijkt en een waarde returnt van 1 tot 1000.



Hoe kleiner het resultaat hoe groter de gelijkenis tussen de 2 foto’s. Ik heb met behulp van PHP een For lus gemaakt die per personage in de databank dit programma uitvoert en het resultaat met het kleinste verschil geef ik terug als uitkomst.

Deze methode is te zien in het laravel project achter /recognitionEquation

Zo is het resultaat de foto waarbij de “rms” het kleinst is.

Deze methode werkt al beter en het gebeurt allemaal in de backend. Met deze methode heb ik er ook een manier bij geschreven om een foto op te slaan in de databank als het een fout resultaat geeft. Zo zou de databank dus kunnen groeien als ze meer gebruikt wordt en een beter resultaat geven.

Probleem: De berekening duurt zeer lang. Als er in de databank een 40 tal karakters zitten, duurt de berekening al enkele seconden.

De vergelijk is nog steeds te letterlijk, er geeft nog vaak een fout resultaat.

Versie 2:

Hierbij maak ik gebruik van opencv2. Dit is een foto analyse & manipulatie bibliotheek voor Python. Ik maak gebruik van de “templatematching” methode.

Ik neem bij deze methode een stuk van de foto. En vergelijk dat stuk met al de foto’s in de databank.



Dit zorgt ervoor dat de locatie van het personage weinig uitmaakt. De “templatematcher” gaat elk punt van de foto af en kijkt of de template “past” in de controlefoto.

We krijgen een resultaat terug van 0 tot 1. Als er geen enkele overeenkomst is, wordt er 0 terug gegeven, 1 is een perfecte overeenkomst. Met deze functie toon ik weer de foto met de grootste overeenkomst.

De laatste versie heeft geeft het beste resultaat en is ook het performantste. Ik doe in deze Python commando ook de SQL query in de Python zelf waardoor de For lust veel sneller werkt.

Conclusie:

De laatste versie werkt het beste maar het is nog steeds een redelijk letterlijke vergelijking tussen 2 foto’s, het bevat geen AI. Dit is normaal geen probleem als het gaat over foto’s van cartoon personages aangezien hun uiterlijk redelijk onveranderd blijft. Het enige probleem zijn de verschillende camerahoeken en de verschillende gelaatsuitdrukkingen.



Dit probleem word opgelost door de user de optie te geven om bij een fout resultaat zijn foto mee op te slaan in de databank.

Benodigdheden:

Python:

PIL: <http://www.pythonware.com/products/pil/>

MySQLdb: <http://mysql-python.sourceforge.net/MySQLdb.html>

cv2: <http://docs.opencv.org/3.2.0/d5/de5/tutorial_py_setup_in_windows.html>

numpy: <http://www.numpy.org/>

Javascript:

Tracking: <https://trackingjs.com/>

Resemblejs: <https://huddle.github.io/Resemble.js/>

Kairos: <https://www.kairos.com/>