Het verbeteren van image pre-processing voor OCR met artificiële intelligentie: een vergelijkende studie

Onderzoeksvoorstel Bachelorproef 2021-2022

Niels Van den Eynde[1]

Samenvatting
We vertrekken van veel voorkomende problemen die we kunnen hebben als we foto's van gebruikers verwerken in onze software. In deze bachelorproef behandelen we deze problematiek uit het standpunt van Docbyte, een bedrijf dat alles dat met documenten te maken heeft tracht te digitaliseren. Zo kan het bijvoorbeeld zijn dat er een identiteitskaart moet worden ingescand. Vaak wordt een eindgebruiker ook de keuze gegeven om met een smartphone een foto te nemen van dit document. Het probleem hiermee is dat bepaalde gegevens op deze foto's onleesbaar kunnen worden voor OCR software. De foto kan bijvoorbeeld overbelicht geraken als gevolg van de flits van de camera, ook kunnen er vlekken aanwezig zijn of kan de afbeelding gekanteld zijn. In deze bachelorproef stel ik een classificatiemodel op. Zo'n model heeft normaal als output een vector die de lengte heeft van het aantal mogelijke categorieen waartoe de input kan behoren. Elke waarde van die vector stelt dan de waarschijnlijkheid dat ons gegeven tot die categorie behoort voor. Hier kunnen we dan de waarden die boven een bepaalde grens aannemen als fouten die onze foto bevat. Normaal loopt zo'n foto een vast proces door om de foto te verbeteren. Echter kunnen we nu op basis van onze classificatie dit proces gaan aanpassen zodat er enkel gecorrigeerd wordt op basis van de geclassificeerde fouten. Nadien zal ik een vergelijkende studie opstellen met enerzijds mijn model, en anderzijds bestaande oplossingen voor pre-processing. Deze bachelorproef heeft hopelijk als resultaat dat het inderdaad economisch en technisch zinvol is om zo'n oplossing zelf te maken. Verder zou zoiets nog kunnen gebruikt worden in bedrijven die gelijkaardige noden hebben.

Sleutelwoorden
Machineleertechnieken en kunstmatige intelligentie. afbeeldingsclassificatie — beelverbetering — vergelijkende studie

Co-promotor
Co-promotor[2] (Bedrijfsnaam)

Contact: [1] niels.vandeneynde@student.hogent.be; [2] ;

Inhoudsopgave

1. Introductie

Tegenwoordig zie je het meer en meer. Denk maar aan de hele crypto gekte. Wanneer je een account aanmaakt bij een 'exchange' (een plaats waar je cryptomunten kan kopen), zal je verplicht worden om je account te verifiëren alvorens je effectief aan de slag kan gaan. Deze verificatie gebeurt vaak aan de hand van een identiteitsbewijs dat je moet uploaden. Vervolgens gaan allerlei algoritmen hier de relevante gegevens uithalen, controleren op vervalsing, en kijken of de naam waarop het account staat effectief overeen komt met de naam die op je document staat.

Net hier komt het probleem waarvoor in deze bachelorproef naar oplossingen wordt gezocht naar boven. Namelijk de menselijke, zwakke schakel. Een foto nemen van een document lijkt een eenvoudige opgave, echter wanneer een computer deze foto analyseert, is er weinig ruimte voor imperfecties. Een foto met vlekken, overbelichting, of een die 'skewed' is al snel bijna compleet onleesbaar door OCR software.

Zo komen we bij docbyte, dit bedrijf wilt alles dat met documenten te maken heeft automatiseren. Hier is uiteraard van het grootste belang dat onze afbeelding effectief als tekst kan worden opgeslagen. Een vereiste hiervoor is dat de afbeelding zo goed mogelijk wordt gepre-processed zodat de OCR het grootste deel van alle woorden kan interpreteren. De rest kan dan eventueel voorspeld worden door iets als GPT-3.

Momenteel worden er enkele tools van derde partijen gebruikt, met verschillende maten van succes. Wat in deze bachelorproef bereikt wil worden is kijken of zoiets niet 'in-house' gedaan kan worden. Bovendien zou het ook erg nuttig zijn om te weten of dit economisch zin heeft voor een

bedrijf.

- Hoe goed presteren deze zelfgemaakte oplossingen?
- Kan er op rekenkracht of executietijd bespaard worden door deze classificatie uit te voeren?
- Hoe complex is zo'n oplossing en wat is de kost?

Deze bachelorproef zal zich bezig houden met het zoeken naar antwoorden op bovenstaande onderzoeksvragen.

2. State-of-the-art

Eerst wil ik het hebben over hoe image pre-processing ten behoeve van OCR in de praktijk vaak wordt gedaan. Als eerste zal de afbeelding naar greyscale worden omgezet. Nadien zal er op basis van een threshold een conversie gemaakt worden zodat elke pixel een 0 of een 1 is (0 voor zwart, 1 voor wit). Alle waarden onder deze drempel worden als 0 beschouwd en het omgekeerd voor 1. Vaak is dit een adaptieve threshold die zich aanpast aan het stukje van de afbeelding. Verder worden er correcties uitgevoerd voor 'skedwedness', nadien blur correction enzovoort. Hierna wordt er een score gegeven op de accuracy die de OCR software behaald. Deze hele pipeline wordt dan herhaald totdat de accuracy een acceptabele waarde heeft. Een implementatie van dit proces is beschreven in een talk op pycon enkele jaren geleden. (Chastagnol, 2013)

Dit is meestal waar men stopt. Tijdens mijn literatuurstudie ben ik niet veel onderzoeken tegengekomen die een gelijkaardige invalshoek als deze bachelorproef hadden. Zo was er bijvoorbeeld wel een artikel over een binaire classifier die simpelweg oordeelde of het een goede of slechte afbeelding was.

Dit onderzoek tracht echter nog een stap verder te gaan. We kiezen om door middel van de classificatie van fouten de pipeline aan te passen zodat deze enkel de correcties uitvoert die nodig zijn. Hierna kijken we naar de snelheidswinst die we boeken.

Wel hebben we nog het volgende artikel dat geschreven is door iemand die werkt voor een bedrijf dat zich specialiseert in 'intelligent data processing'. Dit is iets wat OCR eigenlijk 'vervangt' volgens de auteur. Volgens dit artikel is het niet zo nuttig om deep learning te gebruiken omdat er betere tools zijn om deze fouten uit de afbeeldingen te halen. (Clark, 2020). Zo kan je bijvoorbeeld perfect de 'skew' berekenen met een traditioneel algoritme. (Hull, 1998).

3. Methodologie

Het plan is om een model te trainen dat afbeeldingen classificeert op basis van hun meest voorkomende imperfecties.

Als eerste moeten we een pipeline bouwen met alle mogelijke correcties die je maar kan bedenken. Dus dat will zeggen correctie voor skewedness, noise, overbelichting en zo verder. Veel van deze correcties kunnen met niet veel code in python me behulp van openCV worden uitgeschreven. Nadien kunnen we deze pipeline 'at runtime' aanpassen met het resultaat van een classificatie. Dat brengt ons bij het volgende aspect van dit onderzoek, het neural network.

Een neural network geeft als eindresultaat een vector terug met daarin de verschillende kansen. Vaak wordt er gewoon gekeken naar welke optie de hoogste kans heeft, dit wordt dan aanzien als de klasse waartoe het object behoort. In ons gevallen kan het heel goed zijn dat er meerdere fouten zijn op een afbeelding. Hiervoor gaan we dus gewoon kijken naar de kans per imperfectie. Hierna kunnen we enkel de fouten beschouwen waarbij de probaliteit boven een bepaalde waarde ligt.

Om zo'n model te trainen heb je natuurlijk heel wat data nodig. Hiervoor zal ik online kijken. Om deze data aan te vullen is er ook de mogelijkheid om zelf samples te gaan genereren door bepaalde afbeeldingen te gaan bewerken met python en openCV. Zo kunnen we gemakkelijk deze 'skewedness' introduceren en tegelijk de afbeelding programmatisch labelen. Hetzelfde voor overbelichting, noise en blur. Op het eerste zicht zijn er genoeg datasets met tekst. Het model zelf zal zeer waarschijnlijk een convolutional neural network worden.

Een belangrijk component van deze bachelorproef is kijken naar de kwaliteit van zo'n zelfgemaakte oplossingen en de tijd, kost en complexiteit die hierbij komt kijkt. Vervolgens zal ik een vergelijkende studie maken met andere tools die reeds op de markt zijn om aan image pre-processing doen. Zo zit deze functionaliteit reeds ingebakken bij sommige OCR software. Het mogelijke struikelblok van deze studie zou zijn dat een neural network simpelweg te veel rekenkracht zou vereisen. Het is namelijk ook mogelijk om sommige van deze andere imperfecties zonder een neural network op te sporen.

4. Verwachte resultaten

Een oplossingen maken voor het correct voorspellen welke fouten er in een afbeeldingen zitten, hierna op basis van deze voorspellingen bepaalde verbetering uitvoeren is dus de opgave. Als mijn model de fouten nauwkeuriger kan herkennen dan andere tools is het onderzoek een succes. Ook zal ik onder andere noteren hoeveel rekenkracht, mankracht enzovoort er nodig was om tot mijn oplossing te komen. Dit om een idee te geven van het kostenplaatje van zo'n oplossing. Dit samen met enkele grafieken die de accuracy van zo'n model plot tegenover de tijd en geld die er in kruipen. Ook zullen er enkele figures zijn om deze oplossingen te vergelijken met de concurrenten. Zo kunnen we het aantal woorden dat een OCR uit de afbeelding kan halen vergelijken tussen mijn oplossing en andere oplossingen die geen neural networks gebruiken.

5. Verwachte conclusies

De verwachte conclusie hier is dat er wel degelijk een voordeel is aan het gebruiken van neural networks bij het preprocessen van een afbeelding voor OCR. Ook hopen we dat onze oplossing toch competitief is met de state of the art. Verder verwachten we op het einde van de dag dat we een goede pipeline kunnen bouwen die de accuracy van een OCR systeem aanzienlijk omhoog duwt.

Referenties

Chastagnol, F. (2013, maart 23). Building an image processing pipeline with Python. https://www.youtube.com/watch?v=B1d9dpqBDVA

Clark, M. (2020). Can Deep Learning and AI Help in Preprocessing Images for OCR? https://www.infrrd.ai/blog/can-deep-learning-and-ai-help-in-preprocessing-images-for-ocr

Hull, J. J. (1998). Document image skew detection: Survey and annotated bibliography.