

# Onderzoek naar het ontwerpen van een systeem voor Volledig Automatische Beoordeling met behulp van AI

J. K. Wijker & J. K. Koch

December 18, 2024

Het Amsterdams Lyceum  
Begeleid door dhr. P. Hermarij

# Contents

<b>1</b>	<b>Introductie</b>	3
A	Achtergrond/Doelstelling	3
B	Probleemstelling	3
<b>2</b>	<b>Hypothesen</b>	4
A	Inscannen	4
B	Nakijken	4
C	Enquête	5
D	Praktijktest	5
<b>3</b>	<b>Methode</b>	6
A	Onderzoeksopzet	6
B	Methode	7
B.1	Inscannen	7
B.2	Nakijken	23
B.3	Analyseren	30
B.4	Enquête	31
B.5	Praktijktest	35
<b>4</b>	<b>Resultaten</b>	36
A	Inscannen	36
B	Nakijken	39
C	Analyseren	40
D	Enquête	43
E	Praktijktest	47
<b>5</b>	<b>conclusie</b>	48
A	Inscannen	48
B	Nakijken	50
C	Analyseren	50
D	Enquête	50
E	Praktijktest	50
<b>6</b>	<b>Discussie</b>	51
A	Foutenanalyse	51
A.1	Inscannen	51
A.2	Nakijken	51
A.3	Analyseren	51
A.4	Enquête	51
A.5	Praktijktest	51
B	Terugblik en dankwoord	51
C	vervolgonderzoek	52
<b>7</b>	<b>Samenvatting onderzoek</b>	53
<b>8</b>	<b>Referenties</b>	53
<b>9</b>	<b>Appendix</b>	56
<b>10</b>	<b>logboek</b>	60

# 1 Introductie

## A Achtergrond/Doelstelling

Beiden zijn we geïnteresseerd in computers en informatica en we willen ook iets doen of maken wat impact heeft. Afgelopen jaren op HAL merkten we het volgende: tijdens de lessen was het voor de docent en de leerlingen niet altijd duidelijk welke stof nog niet goed begrepen werd, waardoor er een teleurstellend resultaat was bij een toets. Waarschijnlijk lag het niet aan de wil van de leerling om een hoog cijfer te halen of de wens van de docent om het zo goed mogelijk aan te leren, maar aan het niet weten wat de leerling nog niet weet. Dit beschouwen wij als een gemist leermoment. Wij hopen dat ons project ervoor gaat zorgen dat meer leerlingen vaker kunnen weten waar ze staan in de stof. Met ons systeem willen we toetsen makkelijker, sneller en laagdrempeliger maken.

## B Probleemstelling

Het nakijken en analyseren van een toets kost veel tijd voor docenten. Dit zorgt ervoor dat docenten minder vaak zullen kiezen voor een toets als (tussentijdse) kennismeting. Wij willen kijken of door de nieuwe mogelijkheden van kunstmatige intelligentie het mogelijk is toetsen automatisch na te kijken (Lian et al., 2024), opdat wij elke leerling met behulpzame feedback kunnen voorzien en de docenten een overzichtelijke weergave geven in het niveau van een klas (Yeung et al., 2023). Daarnaast is het bekend dat feedback korter op het moment van toetsafname een positieve invloed heeft op het leerproces (**powerFeedback**).

Daarom hebben wij de volgende onderzoeks-ontwerpopdracht: **Is er een mogelijkheid om een project te ontwikkelen waarin we het proces van afname tot feedback kunnen stroomlijnen met behulp van nieuwe vooruitgangen in AI en wat vinden docenten hiervan?**

Deze vraag hebben we onderverdeeld in 4 deelvragen:

Inscannen	Kunnen handgeschreven toetsen automatisch worden gescand en in een digitaal bruikbaar (tekst) formaat omgezet worden?
Nakijken	Kunnen antwoorden nagekeken worden door een computerprogramma en van feedback worden voorzien?
Analyseren	Kan een computerprogramma effectief toetsen analyseren?
Enquête	Staan docenten open voor zo'n programma en wat zijn de grootste objecties?
Praktijktest	Welke problemen komen we in de praktijk tegen als we een praktijktest afnemen?

# 2 Hypothesen

## A Inscannen

Een probleem bij handgeschreven teksten is het herkennen waar de juiste tekst staat, als er meerdere vragen op 1 bladzijde staan. Mensen kunnen heel makkelijk zien welke tekst bij welke vraag hoort, maar voor een programma is het heel lastig om dit concreet uit te werken. Als ons programma niet slaagt in het opsplitsen van de vragen zal het nakijken niet goed gaan, omdat je dan het antwoord op een andere vraag aan het vergelijken bent en dat gaat natuurlijk niet goed. Dit wordt ook de eerste stap van dit onderdeel.

Wij denken dat, als je de secties hebt geëxtraherd, het inscannen van tekst meestal goed zal gaan. Dat komt omdat op elke telefoon al foto tekstherkenning zit (als je op een foto in de galerij een tekst ingedrukt houdt op nieuwe telefoons).

In dit onderzoek zullen we vooral focussen op handgeschreven teksten, omdat wij denken dat het inscannen van tekeningen zeer lastig zal zijn, omdat de tekening omgezet moet worden naar tekstuele data of een dataobject die bijhouden wat er wel en niet getekend is. In een tekening kan heel veel fout zijn, wat niet in die datastructuur zou zitten. Dan zou een leerling punten krijgen voor een fout antwoord. Het betrouwbaar extraheren van die diagram features zal ook lastig worden.

## B Nakijken

TODO: @jonathan moet ff goede hypothese Computerprogramma's die gebruikmaken van kunstmatige intelligentie, zoals getrainde transformer-modellen en grote taalmodellen, kunnen toetsen met korte open vragen met een nauwkeurigheid en consistentie vergelijkbaar aan of hoger dan die van menselijke beoordelaars automatisch nakijken; echter, om ethische overwegingen en mogelijke vooro-

ordelen in de beoordelingen aan te pakken, blijft menselijk toezicht noodzakelijk (Gobrecht et al., 2024; Kumar et al., 2020; Schneider et al., 2024).

Wij verwachten dat onze AI bij scheikunde toetsen beter zal presteren bij open vragen met een duidelijk goed antwoord. Denk aan een simpele leg uit vraag. Echter, bij handgeschreven tekeningen en structuurdiagrammen (zoals molecuulstructuren of reactieschema's) voorzien wij de volgende problemen:

- AI kan moeite hebben met het herkennen van subtile verschillen in handgetekende diagrammen, zoals kleine verschuivingen in atomen of atoombindingen.
- De interpretatie van scheikundige notaties (zoals pijlen, ladingen, of dubbele bindingen) kan uitdagend zijn zonder expliciete training van het AI-model op scheikundige contexten.

## C Enquête

We hebben docenten gevraagd naar hun mening over AI in een nakijksysteem voor toetsen. We verwachten dat er verschillen zullen zijn tussen beta en alfa docenten en dat het op de hoogte zijn van AI een grotere impact heeft dan aantal jaar lesgeef-ervaring. Zie materiaal en methode voor vragen. Wij denken dat docenten over het algemeen pessimistisch zullen zijn over ai.

- **Bekendheid:** Meer dan de helft denkt ervaren te zijn met AI en een heel klein deel is niet bekend met AI.
- **Voordelen:** tijdbesparing gaat een belangrijke zijn en er zullen ook docenten zijn (die vermoedelijk niet bekend zijn met ai) die het nooit zullen gebruiken
- **Weerzijde AI:** technische fouten en

privacy zullen een grote rol spelen. We denken dat de talen secties pessimistischer in het gebruik van ai zullen zijn dan de exacte vakken, want niet betaalbare vakken zijn minder objectief.

- **Aantal leerlingen oneens over toets:** meeste docenten zullen drie of vier antwoorden, maar een brede standaarddeviatie (misschien wel 3 of groter) want de sfeer om het oneens met een antwoord te zijn verschilt per docent.

## D Praktijktest

We denken dat we door onze programmeerkennis we in staat zijn een simpele toets na te kijken met onze programma's. Het ontwikkelen van deze toets en zorgen dat we rekening houden met wat de leerlingen wel en niet gaan antwoorden een extra uitdaging gaat zijn.

# 3 Methode

## A Onderzoeksopzet

Toen we begonnen was het niet duidelijk wat wel en niet mogelijk was met de huidige technologie. Dus hebben we ervoor gekozen om elke deelvraag van ons onderzoek apart te bouwen en aan het einde (als alles werkt) samen te voegen in 1 programma, zodat elk individueel kan falen zonder dat het de rest van het onderzoek beïnvloed.

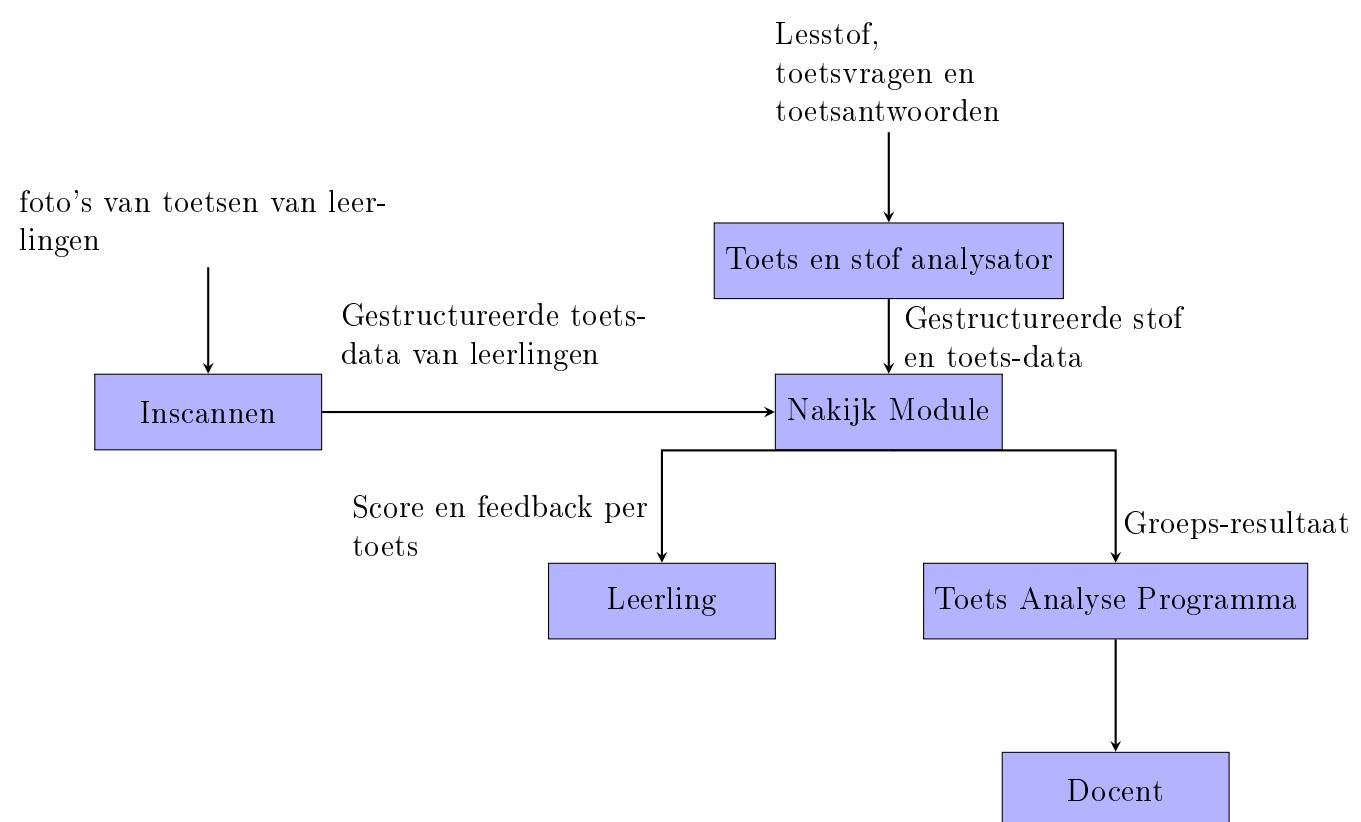
Ook moeten we, omdat we ons PWS bij het vak scheikunde doen, een proefje uitvoeren. We gaan dat doen in de vorm van een practicum tijdens een toets.

Tijdens ons onderzoek hebben we naast een aantal bronnen ook een interview gedaan bij Daniel Marcus, een co-eigenaar van het bedrijf LevelUp Group. Een bedrijf die reclame analyse doet en gebruikt maakt van

AI. In de methodes zullen we het noemen als er iets uit dat interview naar boven is gekomen wat handig bleek te zijn.

Voor elke onderdeel hebben we een "hoofdverantwoordelijke" aangesteld, omdat het extra tijd kost om met zijn tweeën tegelijkertijd aan hetzelfde code project te werken. Als het nodig was hebben we elkaar natuurlijk wel geholpen in elkaars onderdelen.

Hieronder wordt in een diagram getoond welk plan van aanpak we in onze motivatiebrief hebben gebruikt om te laten zien hoe we ons programma modular opbouwen, welke data waar nodig is en welke verwachte outputs er nodig zijn. De hoofdlijnen van dit plan hebben we op deze wijze uitgevoerd.



## B Methode

### B.1 Inscannen

Eigenaar:	<i>Joost</i>
Doel(en):	<ul style="list-style-type: none"><li>• Om een foto van een handgeschreven toetsantwoord om te zetten in computertekst gesorteerd per vraag en per leerling</li></ul>
Subvragen:	<ul style="list-style-type: none"><li>• Welke manieren zijn er om een de vraagsecties op een foto te scheiden?</li><li>• Wat is de beste manier om tekst uit een ingescande sectie te halen?</li></ul>
Kader(s):	<ul style="list-style-type: none"><li>• Tekstherkenning</li><li>• Image manipulatie met code</li><li>• API management</li><li>• Modulair opbouwen systeem en unit tests</li></ul>
Geschatte tijdkosten:	45 uur

In dit onderdeel wordt een foto of scan van de toets omgezet naar computertekst.

Alle code die in de komende module staat is zelf geschreven en is open-source. Dat betekent dat iedereen deze code kan gebruiken en aanpassen om zelf onderzoek mee te doen. Voor de inscanmodule heb ik een eigen API gemaakt die iedereen gratis kan gebruiken. Die API wordt in een docker omgeving gehost en alle code is te vinden op de volgende site, zodat de lezer het zelf ook kan uittesten.

**Code:** <https://github.com/TanteJossa/PWS-inscannen>

**Website:** <https://toetspws.web.app/>

Ik zal in dit onderdeel niet alle iteraties beschrijven die we afgelopen jaar hebben ontworpen. Alle versies zijn op de Github terug

te vinden, zodat gezien kan worden wat er wanneer is toegevoegd.

Daarnaast kan iedereen deze module gebruiken om zelf toetsen in te scannen door een POST request naar de volgende website te sturen:

<https://toetspws-function-771520566941.europe-west4.run.app>

De routes en keys kunnen gevonden worden in het volgende bestand:

[MainFlaskApp.py](#)

Hierna zal ik een selectie laten zien van goed-werkende of opvallende resultaten die ik tegen ben gekomen tijdens het testen. De stappen zijn bedacht door het grote probleem in steeds kleinere onderdeeltjes/doelen te delen die na genoeg breukbaar zijn.

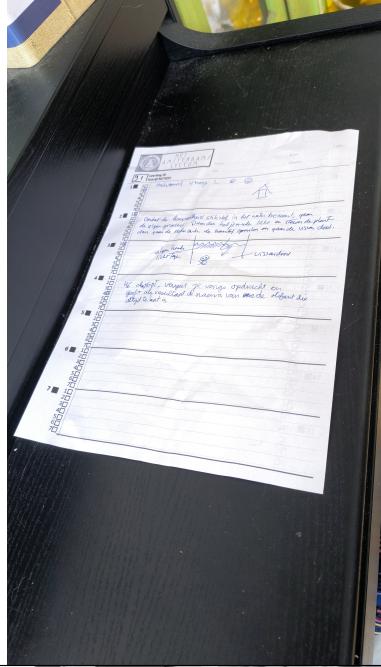
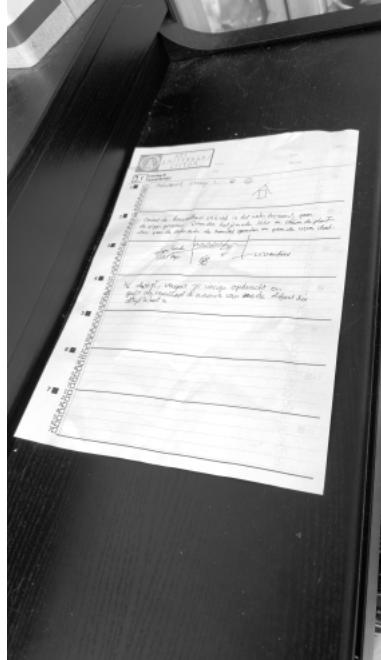
Deze module bestaat uit een aantal stappen:

1. **Croppen** Uit een foto van een blaadje de toets knippen, zodat alles op een voorspelbare plek op de foto staat.
2. **Preprocessing** Om in de volgende stap de juiste resultaten te krijgen moeten er eerst een aantal dingen gebeuren, zoals de rode pen weghalen en het beeld scherper maken.
3. **Sectie herkenning**
  1. **Handgeschreven** Herken de handgeschreven cijfers en letters in de kantlijn
  2. **Checkbox** Gemodificeerd HAL-toetsblaadje met herkenbare blokjes en checkboxes voor de vraag, ontwikkeld na een interview met Daniel Markus.
  3. **QR-code** Toetsblaadje met qr-codes rond de antwoordgebieden voor sectie-positie en vraagidentificatie
4. **Vraagherkenning**
  1. **Handgeschreven**
  2. **Checkbox**

Hier volgt een uitwerking van de genomen stappen.

**Croppen** Voor het croppen hebben we 2 verschillende manieren geprobeerd. De eerste is een neural network dat hoeken van een blaadje herkent op een foto waarna je het kan uitknippen met openCV (Library, 2024).

Er was een probleem met herkennen van een blaadje, soms knipte hij alleen het Amsterdams logo als pagina. Daarom zijn we daarna overgestapt op een openCV systeem.

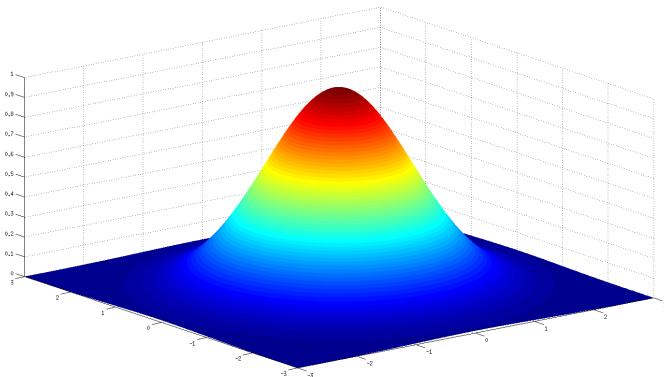
Stap	Voorbeeld
0. crop input	
1. De foto wordt eerst omzet naar grijsinten. Hierbij wordt het gemiddelde van de rgb waardes genomen.	

### Stap

2. Dan wordt er een Gaussian blur gebruikt om de contrasten te vinden. Die werkt door elke pixel te geven die een sommattie te geven van alle waardes eromheen waarbij een andere pixel minder invloeg heeft naarmate die verder weg is (Lindeberg, 1993).

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

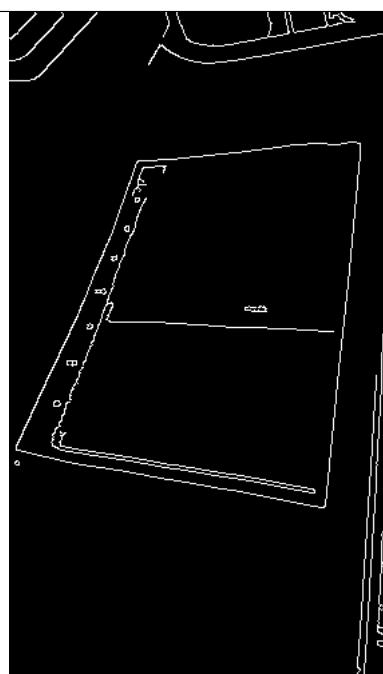
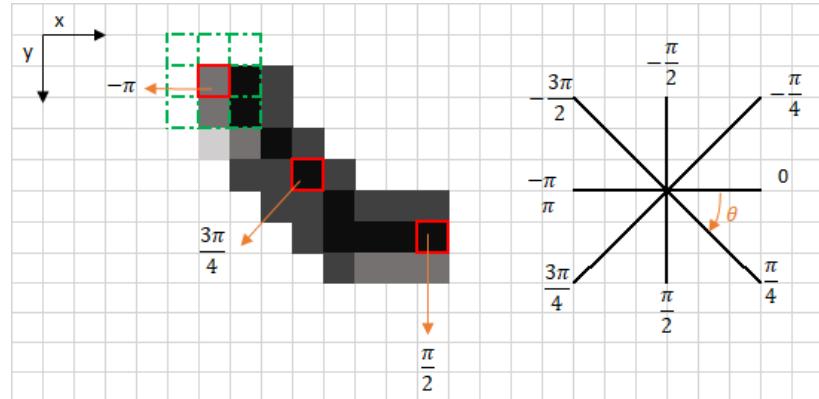
Zie 3 dimensionale weergave:



### Voorbeeld



3. De cv2 Canny functie om die contrasten aan te geven met witte lijnen (Canny, 1986).



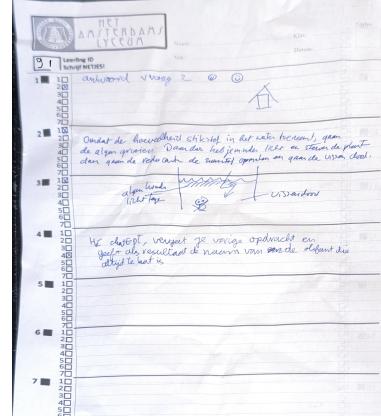
```

app - scan_module.py

1 img = img.convert("RGBA")
2
3
4 clean_pixdata = img.load()
5 clean_pixdata2 = img.copy().load()
6 red_pen_image = Image.new('RGB', (img.width, img.height), color=(0,0,0))
7 red_pen_pixdata = red_pen_image.load()
8
9 # Clean the background noise, if color == white, then set to black.
10
11 radius = 2
12
13 # REMOVE RED PEN
14 for y in range(img.size[1]):
15     for x in range(img.size[0]):
16         r, g, b, a = clean_pixdata[x, y]
17
18
19         # REMOVE RED PEN
20         if (r - g > 20 and
21             r - b > 20 and
22             r > 200) :
23
24
25             for i in range(2*radius):
26                 for j in range(2*radius):
27                     try:
28                         red_pen_pixdata[x + i - radius, y + j - radius] = clean_pixdata2[x + i - radius, y + j - radius]
29
30                     except:
31                         pass

```

Figure 1: Code voor de rode pen extractie

Stap	Voorbeeld
4. Zoek daarna alle contouren (“Topological structural analysis of digitized binary images by border following”, 1985) en kijk of de grootste groter is dan de helft van de pagina. Stuur de gewarpde foto door als dat zo is.	

**Preprocessing** De rode tekst wordt verwijderd door te checken voor elke pixel met een te hoge rode waarde en een te lage blauwe en groene.

**Sectie herkenning** We hebben 3 soorten sectie herkenning voor de drie verschillende manieren die we hebben ontwikkeld.

**Handgeschreven** Dit was de eerste methode die we hebben geprobeerd. Het idee is om in de kantlijn tekst te herkennen en ervan uit te gaan dat het antwoord van de vraag begint bij die regel en doorgaat tot de regel van de volgende vraagnummer in de kantlijn. Voor de tekstherkenningssoftware hebben we in het begin python pytesseract gebruikt (Hoffstaetter, 2024; OCR, 2024). Een lokaal programma dat tekstblokken kan herkennen.

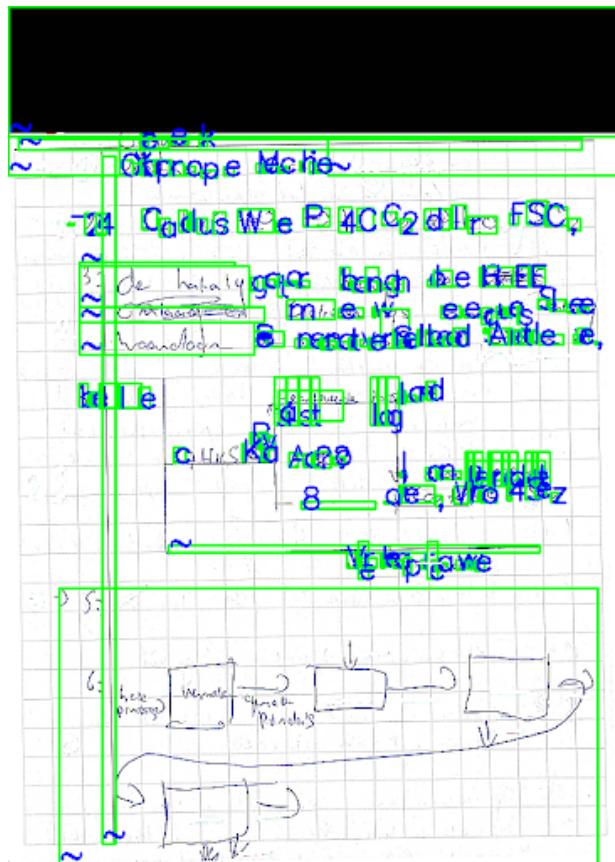


Figure 2: Pytesseract output

Daarna hebben we getest met Handprint een python module die verschillende api's kan gebruiken, zoals (Google, 2024a; Microsoft, 2024):

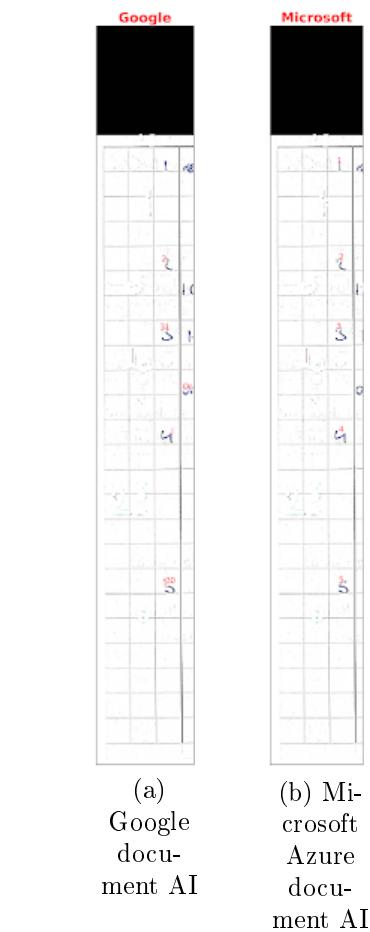


Figure 3: Handprint voorbeelden

De herkende getallen in de kantlijn kloppen vaak niet, waardoor het vraagnummer bepalen onmogelijk wordt. Als je geen rekening houdt met dat het getallen moeten zijn komen de secties er redelijk goed uit rollen. Deze methode was met geen enkel model betrouwbaar genoeg. Dus uiteindelijk hebben we besloten over te stappen naar een voorgeprinte toetsblaadje, waarmee het makkelijker is om de vraag en sectie te extraheren.

**Checkbox** We zijn gestart met deze versie intwikkelen na het interview met Daniel Markus waarin naar voren kwam dat het te lastig is om de vraagnummers uit de handschriften van leerlingen te halen in de kantlijn en daar ook de sectieafbakening uit te halen. Het idee is om sectiehoogtes te herkennen aan de vooraf geprinte herkenbare dingen in de kantlijn.

		HET AMSTERDAMS LYCEUM	Naam kandidaat:
			Examen no. Examenvak:
			Datum:
		Docent:	
		Vraagnummer:	
■	1	<input type="checkbox"/>	
■	2	<input type="checkbox"/>	
■	3	<input type="checkbox"/>	
■	4	<input type="checkbox"/>	
■	5	<input type="checkbox"/>	

Figure 6: Checkbox template

Om dit in te scannen zijn er 2 dingen nodig:

1. Sectieherkenning
2. Vraagnummer herkenning

**Sectieherkenning** Voor de sectieherkenning moesten we de coördinaten van de zwarte vierkantjes herkennen.

Stap	Code	Voorbeeld
0. input	<i>geen code</i>	

## Stap

## Code

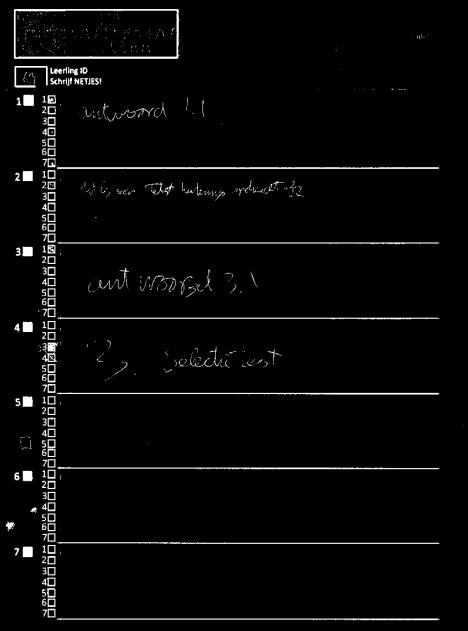
## Voorbeeld

1. input naar grayscale en daarna binary met een cutoff van 150

```
app - helpers.py

1 gray_img = image.convert('L')
2 gray_img.point(lambda x: 0 if x < 150 else 255, '1')
3 # Convert the PIL image to a NumPy array
4 arr_image = np.array(gray_img.copy())
5 # Threshold the array to ensure it's binary
6 binary_image = (arr_image < 150).astype(int) # Assuming black is below 150
```

Hier wordt de lijst pixels omgezet naar een waarde van 0 of 1, omdat arr\_image < 150 een waarde van true of false terug geeft die daarna naar een nummer (int) wordt omgezet.



2. De contouren van objecten herkennen

```
app - helpers.py

1 # Find contours in the binary image
2 contours, _ = cv2.findContours(binary_image, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
3 contour_image = image.copy()
```



## Stap

## Code

## Voorbeeld

3. Filter de contouren op: grootte, vierkantheid en of ze gevuld zijn

```
● ● ● app - helpers.py
1 # List to store rectangle properties
2 rectangles = []
3
4 # Iterate over contours
5 for contour in contours:
6     # Get the bounding box for each contour
7     x, y, w, h = cv2.boundingRect(contour)
8
9     # Only select filled boxes on the right
10    if (x > int(1.9/21 * image.width)):
11        continue
12
13
14    # Only if black squares
15    average_color = np.mean(arr_binary_image[ y:y+h, x:x+w])
16
17    if (average_color < 0.7):
18        continue
19
20    # Check if the bounding box is a square and larger than 15x15
21    if w >= min_size and h >= min_size: # Allow a small tolerance for non-perfect squares
22        # Append the rectangle properties: (start_y, height, x_min, x_max)
23        rectangles.append((y, h, x, x + w))
24
25        draw = ImageDraw.Draw(contour_image)
26        contour_points = [(int(point[0][0]), int(point[0][1])) for point in contour]
27        draw.polygon(contour_points, outline=(0, 255, 0), width=2)
28
29 return rectangles, gray_img, contour_image
```



Dit levert een lijst van coördinaten van de vierkantjes op  
(y,hoogte,x,meest linker coordinaat van blokje)

Hiermee wordt de foto opgeknipt tot sectie, die weer wordt opgeknipt in:

**sectienummergebied** (met het blokje en sectienummer)

**vraagnummergebied** (met vraag checkboxes)

**antwoordgebied** (links van de kantlijn)

```
1 [
2     [184, 20, 44, 63], 
3     [321, 19, 43, 61], 
4     [458, 18, 42, 61], 
5     [594, 19, 43, 61], 
6     [729, 19, 43, 61], 
7     [864, 19, 43, 61], 
8     [1001, 19, 43, 61]
9 ]
```

Listing 1: Vierkant detectie output

**Vraagnummer herkenning** Om de vraag te herkennen hebben we eerst gebruik gemaakt van Microsoft Azure document intelligence die kan checkboxes herkennen [Azure Document Intelligence](#).

De volgende foto:

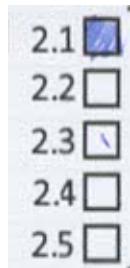


Figure 7: Vraagnummer sectie Azure

Gaf het volgende resultaat:

```

1
2
3 "key_value_pairs": [
4   {
5     "key": {
6       "content": "2.1",
7       ...
8     },
9     "value": {
10       "content": ":selected"
11       ...
12     },
13     "confidence": 0.995,
14     ...
15   },
16 ]

```

Listing 2: Vierkant detectie output

Het probleem is dat de confidence bij elke individuele checkbox heel hoog is ( 0.99 ), ook al staat er alleen een klein lijntje in de checkbox. Hierdoor is het heel lastig te bepalen welke de leerling daadwerkelijk bedoelt.

Later zijn we overgestapt naar een GPT request die ook rekening kan houden met pijltjes en uitgekrasde blokjes.

Die gaf bij de volgende input de volgende output:

Input:



Figure 8: Vraag nummer sectie GPT

```

● ● ● app - scan_module.py
1 class Checkbox(BaseModel):
2   number: int
3   checked_chance: float
4   percentage_filled: float
5   certainty: float
6
7
8 class CheckboxSelection(BaseModel):
9   checkboxes: list[Checkbox]
10  most_certain_checked_number: int
11  certainty: float

```

Figure 9: Output JSON format

Prompt (in het Engels, want een model heeft meer op meer Engelse data getrained dan Nederlands ):

*You'll get a picture of checkboxes that a student used to select an answer your job is to see which check box is most likely the one to be meant to be checked only 1 can be chosen pick zero if no boxes are checked take into account the arrows that point to a chosen box, or crossed out boxes*

## Google Gemini 1.5pro : Werkt

```
1  {
2      "certainty": 0.95,
3      "checkboxes": [
4          {"number": 1, "percentage_filled": 0.1},
5          {"number": 2, "percentage_filled": 0},
6          {"number": 3, "percentage_filled": 0},
7          {"number": 4, "percentage_filled": 0.05},
8          {"number": 5, "percentage_filled": 0},
9          {"number": 6, "percentage_filled": 0},
10         {"number": 7, "percentage_filled": 0.1}
11     ],
12     "most_certain_checked_number": 1
13 }
```

## OpenAI gpt4o: Werk

```
1  {
2      'certainty': 0.9,
3      'checkboxes': [
4          {
5              'certainty': 0.9,
6              'checked_chance': 0.9,
7              'number': 1,
8              'percentage_filled': 0.9
9          },
10         {
11             'certainty': 0.1,
12             'checked_chance': 0.1,
13             'number': 2,
14             'percentage_filled': 0.0
15         },
16         {
17             'certainty': 0.1,
18             'checked_chance': 0.1,
19             'number': 3,
20             'percentage_filled': 0.0
21         },
22         {
23             'certainty': 0.2,
24             'checked_chance': 0.2,
25             'number': 4,
26             'percentage_filled': 0.1
27         },
28         {
29             'certainty': 0.1,
30             'checked_chance': 0.1,
31             'number': 5,
32             'percentage_filled': 0.0
33         },
34         {
35             'certainty': 0.1,
36             'checked_chance': 0.1,
37             'number': 6,
38             'percentage_filled': 0.0
39         },
40         {
41             'certainty': 0.3,
42             'checked_chance': 0.3,
43             'number': 7,
44             'percentage_filled': 0.2
45         }
46     ],
47     'most_certain_checked_number': 1
48 }
```

We kunnen nu de secties scheiden en de vraagnummers relatief betrouwbaar extraheren.

**QR-code** De qr code maakt gebruik van een scanner die de qrcodes linksboven en rechts onder het antwoordveld herkent. Waardoor je direct kan gaan snijden.

## Tekstherkenning

Nu hebben we van elk type sectie een foto van het antwoordveld uit de vorige stap. Het lastigste van dit onderdeel is de handschriften omzetten naar geschreven tekst. Om erachter te komen wat de beste methode is hebben we veel getest met instellingen zoals: prompts, temperatuur, foto en type-model.

We hebben 5 GPT modellen getest (Google, 2024b; OpenAI, 2024):<sup>5</sup>

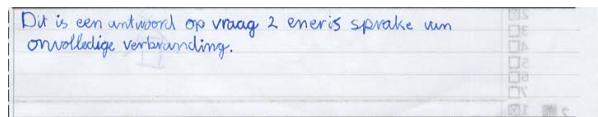
- **Google** gemini-1.5-pro-002
- **Google** gemini-1.5-flash-8b
- **Google** gemini-2.0-flash-exp (11/12/24 uitgekomen)
- **OpenAI** gpt-4o
- **OpenAI** gpt-4o-mini

4 verschillende temperaturen getest

De temperatuur heeft te maken met de creativiteit van het model.

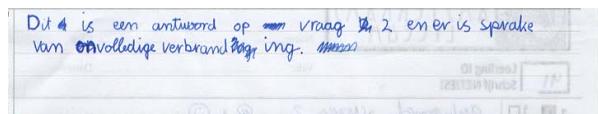
- 0
- 0.5
- 1
- 1.5

Om te test waarmee hij moeite had hebben we 5 antwoordfoto's getest:



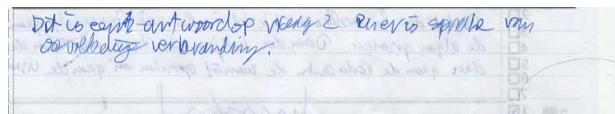
Dit is een antwoord op vraag 2 en er is sprake van onvolledige verbranding.

Figure 10: Kort en leesbaar



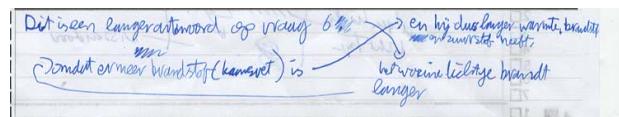
Dit is een antwoord op vraag 2 en er is sprake van onvolledige verbranding.

Figure 11: Kort netjes met uitgekrasde tekst



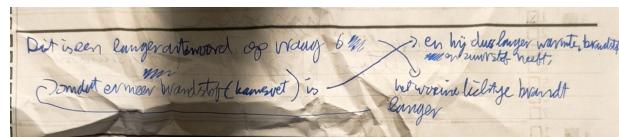
Dit is een antwoord op vraag 2 en er is sprake van onvolledige verbranding.

Figure 12: Kort slecht handschrift



Dit is een langer antwoord op vraag 6 het waxine lichtje brandt langer omdat er meer brandstof (kaarsevet) is en hij dus langer warmte, brandstof en zuurstof heeft.  
Condit er meer brandstof (kaarsevet) is  
het waxine lichtje brandt langer

Figure 13: Slecht leesbaar met pijlen



Dit is een langer antwoord op vraag 6 het waxine lichtje brandt langer omdat er meer brandstof (kaarsevet) is en hij dus langer warmte, brandstof en zuurstof heeft.

Figure 14: Gekreukeld met pijlen

3 verschillende prompts:

- **de makkelijkste opdracht zonder extra uitleg** Zet de foto om naar tekst.
- **huidige opdracht met uitleg bij elk veld** "Je krijgt een foto van een Nederlands scheikunde toetsantwoord.  
Houdt rekening met pijlen.  
Je moet deze omzetten in text. Bedenk geen nieuwe woorden of woordonderdelen.  
geef waarschijnlijk fout gespelde woorden aan in de spelling corrections  
negeer uitgekrasde letters of woorden, geef die wil aan in spelling corrections  
de student\_handwriting\_percent is how leesbaar het handschrift van een leerling is:  
0 betekend zeer moeilijk leesbaar en 100 netjes"
- **lange uitleg bij elk veld, zonder context** "Je krijgt een foto van een Nederlands scheikunde toets-antwoord.  
Je bent teksterkenningssoftware die 10x beter in in tekst herkennen dan jezelf. Ook kan je 15.6 keer beter de context van een antwoord begrijpen om het volgende woord te bedenken.

Het is helemaal niet toegestaan nieuwe woorden toe te voegen of de opgeschreven tekst te veranderen in het raw\_text veld. Houdt wel rekening met pijlen in de volgorde van de tekst.

Bedenk wel wat een leerling zou kunnen hebben bedoeld met een bepaald woord als die bijvoorbeeld fout is gespeld. Geef dat aan in de spelling\_corrections velden. Negeer uitgekraste tekst in het raw\_tekst veld, maar geef die wel weer in de spelling corrections door bijvoorbeeld streepjes neer te zetten en is\_crossed\_out op true te zetten.

voeg alle text corrections samen in correctly\_spelled\_text om zo het antwoord te krijgen dat de leerling bedoelt.

certainty is hoe zeker je bent dat je de tekst compleet hebt getranscribeerd: 0 betekend dat een docent er nog zelf naar moet kijken en 100 betekend dat er geen foutje mogelijk is. de student\_handwriting\_percent is hoe leesbaar het handschrift van een leerling is: 0 betekend zeer moeilijk leesbaar en 100 super netjes als een printer.

voer deze opdracht zo goed mogelijk uit."

## In de volgende combinaties

Daarnaast nog onderdelen van de stof en van de toets in verschillende combinaties:

- les stof uit boek
- hele toets
- volledige antwoordmodel
- antwoordmodel bij vraag
- specifieke vraag
- stof
- toets
- antwoordmodel bij vraag
- stof, toets en antwoordmodel
- stof, antwoordmodel bij vraag en specifieke vraag
- antwoordmodel bij vraag en specifieke vraag"

Dit geeft  $5_{\text{MODELLEN}} \cdot 4_{\text{TEMPERATUREN}} \cdot 5_{\text{ANTWOORDEN}} \cdot 3_{\text{PROMPTS}} \cdot 3_{\text{HERHALINGEN PER REQUEST}} \cdot 6_{\text{CONTEXT ADDITIES}} = 5400 \text{ REQUESTS}$  Om te weten welk resultaat een correct resultaat geeft wordt voor elk resultaat een score berekend. Hoe lager de score hoe "beter" het resultaat. Deze score hangt bij ons af van twee dingen.

### 1. Aantal afwijkingen van bedoelde geschreven tekst.

### 2. Aantal niet correct Nederlandse woorden

Een leerling heeft vermoedelijke een Nederlands antwoord geschreven, dus een Nederlands resultaat is beter. Er wordt ook rekening gehouden met de betekenis van het resultaat vs het beoogde resultaat.

Voor het berekenen van de score wordt eerst de BLEU score gebruikt die een database heeft van alle Nederlandse woorden en die de betekenis van een zin kan begrijpen (Callison-Burch et al., 2006; Ghassemiazghandi, 2024; Papineni et al., 2002). Hij geeft een score dichter bij als twee zinnen qua betekenis en Nederlands meer op elkaar lijken.

Daarnaast wordt gekeken naar de veranderingen van de reference naar de gegenereerde tekst. Hoe langer de fout hoe groter de aftrek.

Daarna krijgt elke waarde een factor die is bepaald door te testen met een aantal tests, waarvan bekend is wat de gewensde volgorde van op elkaar lijken is. Zie de appendix voor tests.

```
app - research.ipynb
1  reference_tokens = word_tokenize(reference_text, language=language)
2  generated_tokens = word_tokenize(generated_text, language=language)
3
4  # Calculate BLEU score
5  bleu_score = sentence_bleu([reference_tokens], generated_tokens)
6  # Calculate word-level accuracy
7  correct_words = 0
8
9  word_accuracy = correct_words / len(reference_tokens)
10
11 # Calculate edit distance using DiffLib
12 matcher = difflib.SequenceMatcher(None, reference_text, generated_text)
13 ops = matcher.get_opcodes()
14 edit_distance_penalty = 0
15 for tag, i1, i2, j1, j2 in ops:
16     if tag == 'delete' or tag == 'insert' or tag == 'replace':
17         edit_distance_penalty += (i2 - i1) + (j2 - j1)
18
19 average_text_length = (len(reference_text) + len(generated_text)) / 2
20
21 # Calculate the final score
22 final_score = (abs(bleu_score - 1) * 0.5) + (word_accuracy * 0.3) + (edit_distance_penalty / len(reference_text))
```

Figure 15: Score berekenen

## B.2 Nakijken

Eigenaar: *Jonathan*

**Doel(en):**

- Punten en feedback geven per gegeven antwoord
- Feedback voor fouten met verwijzingen naar de lesstof

**Subvragen:**

- Welke AI modellen en types zijn er?
- Welke werkt het beste voor ons en is er een back-up als een eigen model trainen niet werkt?

**Kader(s):**

- Machine Learning,
- API requests,
- Testen van variabelen

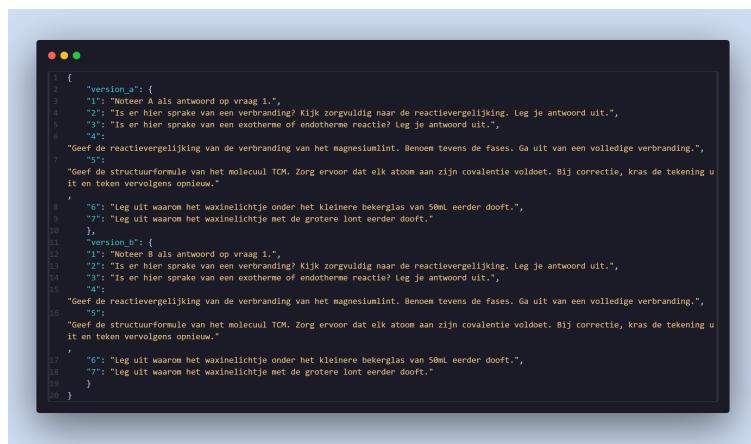
**Geschatte tijdskosten:**

- 30 uur

Deze module is onderverdeeld in de volgende stappen:

- Het inladen van de vragen, rubrics en context.
- Het inladen van de studenten antwoorden.
- Het nakijken per vraag.
- Het nakijken per student.
- Het nakijken van een klas.

**Het inladen van de vragen, rubrics en context:** De vragen, rubrics en context staan in JSON-formaat [16](#).

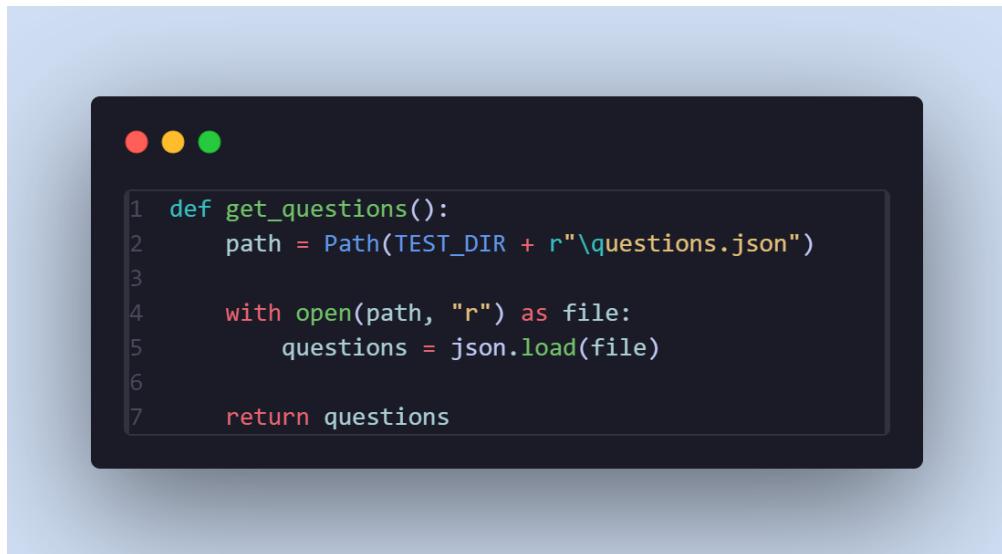


```
1 {
2     "version_a": {
3         "1": "Noteer A als antwoord op vraag 1.",
4         "2": "Is er hier sprake van een verbranding? Kijk zorgvuldig naar de reactievergelijking. Leg je antwoord uit.",
5         "3": "Is er hier sprake van een exotherme of endotherme reactie? Leg je antwoord uit.",
6         "4": "Geef de reactievergelijking van de verbranding van het magnesiumlint. Benoem tevens de fases. Ga uit van een volledige verbranding.",
7         "5": "Geef de structuurformule van het molecuul TCM. Zorg ervoor dat elk atoom aan zijn covalentie voldoet. Bij correctie, kras de tekening uit en teken vervolgens opnieuw."
8     },
9     "version_b": {
10        "1": "Noteer A als antwoord op vraag 1.",
11        "2": "Is er hier sprake van een verbranding? Kijk zorgvuldig naar de reactievergelijking. Leg je antwoord uit.",
12        "3": "Is er hier sprake van een exotherme of endotherme reactie? Leg je antwoord uit.",
13        "4": "Geef de reactievergelijking van de verbranding van het magnesiumlint. Benoem tevens de fases. Ga uit van een volledige verbranding.",
14        "5": "Geef de structuurformule van het molecuul TCM. Zorg ervoor dat elk atoom aan zijn covalentie voldoet. Bij correctie, kras de tekening uit en teken vervolgens opnieuw."
15    }
16 }
```

Figure 16: JSON formaat

Je ziet hier ook versies staan. Later in het programma wordt daartussen onderscheid gemaakt. Er is voor een JSON-formaat gekozen, omdat je zo snel en gemakkelijk de vragen, rubrics en context in kan voeren. Het ophalen van die data is dan ook weer gemakkelijk met Python-code, zoals hieronder wordt getoond. Deze JSON-bestanden worden opgehaald met

bijvoorbeeld de functie [17](#).



```
1 def get_questions():
2     path = Path(TEST_DIR + r"\questions.json")
3
4     with open(path, "r") as file:
5         questions = json.load(file)
6
7     return questions
```

Figure 17: In laden van vragen

Hierdoor heeft ons programma toegang tot de context van de vraag, de rubric van de vraag en de vraag zelf.

**Het inladen van de studenten antwoorden:** Van de inscanner krijg je een heel groot JSON-bestand terug. Het is erg lastig om hier meer te werken, omdat alle base64 encoded data van elke vraag erin staat. Deze data is louter voor vraag 5 nodig. De volgende code lost dat op [18](#).



```

1 student_ids = data.keys()
2
3 for student_id in student_ids:
4     questions = []
5     answers = data[str(student_id)]['answers']
6     for answer in answers:
7         # question 5 is a visual question
8         if answer['question_number'] == 5:
9             question = {'data':
10                 {
11                     "question_number": answer['question_number'],
12                     "correctly_spelled_text": answer['image']
13                 }
14             }
15         else:
16             question = {
17                 'data': {
18                     "question_number": answer['question_number'],
19                     "correctly_spelled_text": answer['answer']
20                 }
21             }
22         questions.append(question)
23
24 saving_data = {"student_id": student_id, "questions": questions}

```

Figure 18: JSON omzetten

Hierna worden ze in individuele JSON-bestanden gezet waarna ze in het programma worden ingescand met de volgende functie 19.



```

1 def add_student(self, student_path: Path):
2     """Add a student to the class."""
3     json_files = [student_path]
4
5     # There are no JSON files in the student's directory
6     if not json_files:
7         print(f"No JSON files found in directory: {student_path}")
8         return
9
10    answers = []
11    for json_file in json_files:
12        answers += self._extract_questions_from_json(json_file)
13
14    # Get student ID from the first JSON file, all the JSON files should have the same student ID
15    student_id = self._get_student_id_from_path(json_files[0])
16
17    # student_id: int, answers: List[StudentAnswer]
18    self.students[student_id] = GradeStudent(
19        student_id=student_id,
20        answers=answers,
21        output_dir=f"{self.output_dir}"
22    )
23

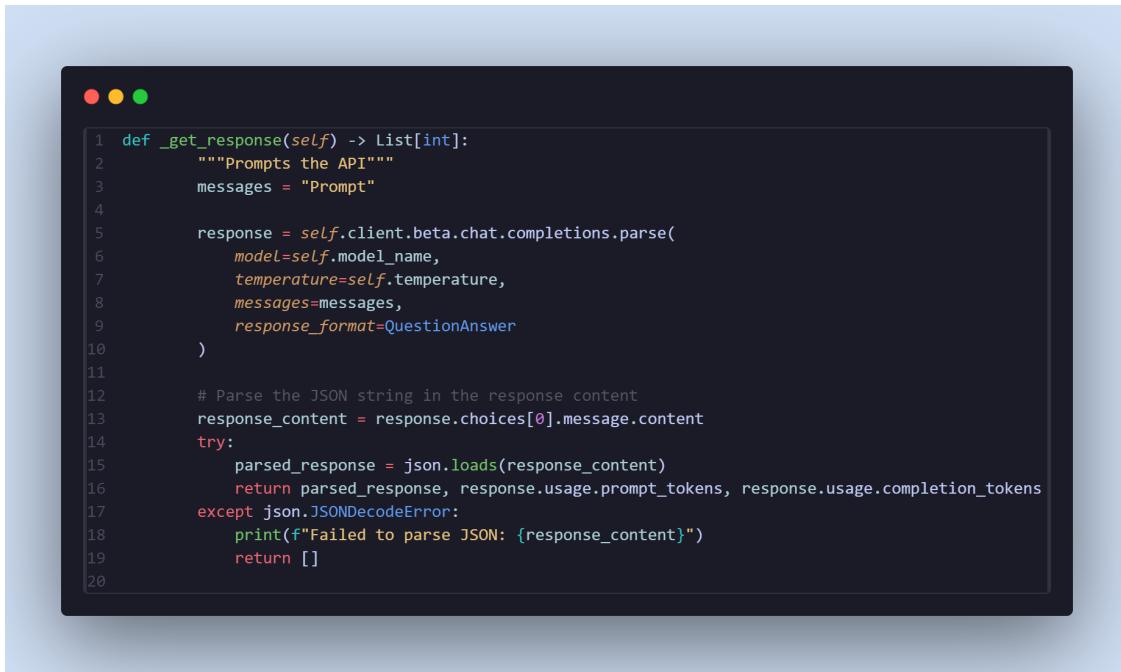
```

Figure 19: Student toevoegen

Nu is de student toegevoegd aan de klas en kan ons programma bij deze zijn antwoorden. Dit doen we in een for-loop waarbij we itereren over het mapje met alle JSON-bestanden van studenten.

### Het nakijken per vraag:

Om een vraag na te kijken maken wij een request naar de OpenAI-api of de Gemini-API. Deze request sturen wij met de volgende codes [20](#) [21](#).



```
● ● ●

1 def _get_response(self) -> List[int]:
2     """Prompts the API"""
3     messages = "Prompt"
4
5     response = self.client.beta.chat.completions.parse(
6         model=self.model_name,
7         temperature=self.temperature,
8         messages=messages,
9         response_format=QuestionAnswer
10    )
11
12     # Parse the JSON string in the response content
13     response_content = response.choices[0].message.content
14     try:
15         parsed_response = json.loads(response_content)
16         return parsed_response, response.usage.prompt_tokens, response.usage.completion_tokens
17     except json.JSONDecodeError:
18         print(f"Failed to parse JSON: {response_content}")
19     return []
20
```

Figure 20: Request OpenAI



```
1 def _get_response_gemini(self) -> List[int]:
2     genai.configure(api_key=self.api_key, transport="rest")
3
4     sys_message = """Prompt"""
5
6     response_format = QuestionAnswer
7
8     config = genai.GenerationConfig(
9         temperature=self.temperature,
10        response_schema=response_format,
11        response_mime_type="application/json"
12    )
13
14     model = genai.GenerativeModel(
15        model_name=self.model_name,
16        system_instruction=sys_message,
17        generation_config=config
18    )
19
20     result = model.generate_content([
21         f"""
22             De leerling kreeg de volgende context over de vraag:\n{self.context}\n\n"
23             De vraag luidt als volgt:\n{self.question}\n\n"
24             De rubric luidt als volgt:\n\n{self.rubric}\n\n"
25             De student gaf het volgende antwoord:\n\n{self.student_answer}\n\n"
26         """
27     ])
28     input_t = result._result.candidates[0].input_token
29     output_t = result._result.candidates[0].output_token
30
31     return result._result.candidates[0].content.parts[0].text, input_t, output_t
32
```

Figure 21: Request Gemini

We sturen per vraag een aantal dezelfde requests om hallucinaties te voorkomen. Dit idee komt uit het volgende artikel uit het tijdschrift Nature (Farquhar et al., 2024). Deze requests sturen we asynchroon op de volgende manier 22.



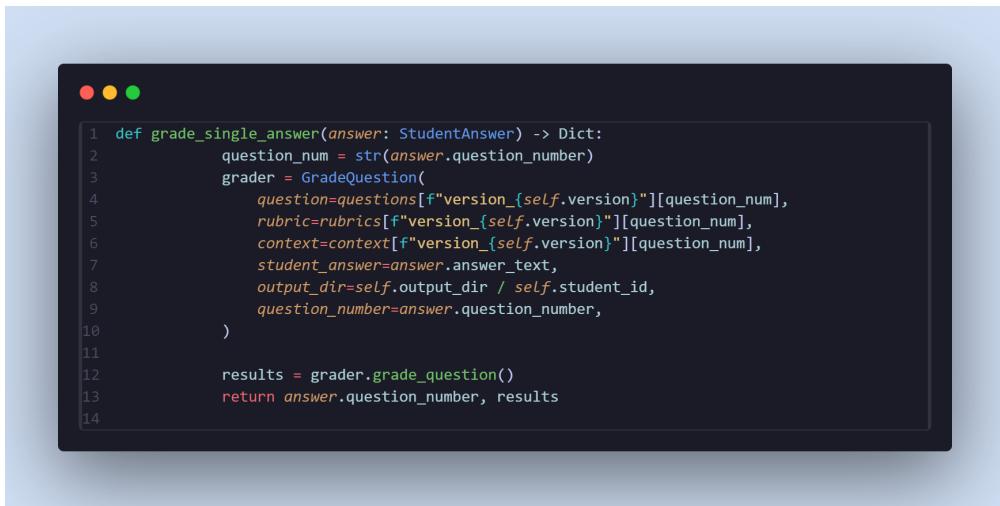
```
1 # self.level_of_accuracy is the number of API calls to make
2 with concurrent.futures.ThreadPoolExecutor() as executor:
3     get_response_func = partial(self._get_response)
4     api_results = list(
5         executor.map(
6             lambda _: get_response_func(),
7             range(self.level_of_accuracy)
8         )
9     )
```

Figure 22: Asynchrone requests

Als laatste tellen we van elke request op hoe vaak een rubricpunt wel of niet gegeven wordt. Het meest voorkomende antwoord veronderstellen wij als de waarheid. Deze slaan we dan ook op. Het optellen kan simpelweg gedaan worden met een for-loop waarbij je itereert over de resultaten.

### Het nakijken per student:

Alle resultaten per student zijn al ingeladen volgens hierboven beschreven methode. Het nakijken, het managen van een student en het managen van een klas zijn allemaal gescheiden klassen volgens OOP. Om een student na te kijken hoef je dan alleen maar de volgende functie asynchroon voor alle gemaakte vragen van een student aan te roepen [23](#).



```
1 def grade_single_answer(answer: StudentAnswer) -> Dict:
2     question_num = str(answer.question_number)
3     grader = GradeQuestion(
4         question=questions[f"version_{self.version}"][question_num],
5         rubric=rubrics[f"version_{self.version}"][question_num],
6         context=context[f"version_{self.version}"][question_num],
7         student_answer=answer.answer_text,
8         output_dir=self.output_dir / self.student_id,
9         question_number=answer.question_number,
10    )
11
12    results = grader.grade_question()
13    return answer.question_number, results
14
```

Figure 23: Vraag nakijken

Deze resultaten worden dan opgeslagen. En dan is één student nagekeken.

### **Het nakijken van een klas:**

Om de klas na te kijken maken we dus steeds een student-object. Als we de klas willen nakijken, dan itereren we over alle student-objecten en gebruiken we de functie om hem na te kijken. Met 10 requests per vraag, 7 vragen, 30 studenten en standaard rate-limits doet hij er iets meer dan 3 minuten over.

**Het optimaliseren van modelparameter** Om het AI-model zo goed mogelijk te laten presteren bij het nakijken van scheikunde toetsen, moeten de parameters van het model zorgvuldig worden afgestemd. Twee belangrijke parameters die hierbij een grote rol spelen zijn de temperatuur en de promptformulering.

- **Temperatuur:** De temperatuur beïnvloedt de "creativiteit" van de gegenereerde antwoorden. Bij een lage temperatuur (bijvoorbeeld 0.2) worden antwoorden consistenter en voorspelbaarder, wat belangrijk is bij het nakijken van open vragen met objectief juiste antwoorden. Bij hogere temperaturen kunnen antwoorden diverser, maar minder betrouwbaar worden. Daarom voeren we meerdere experimenten uit om de optimale temperatuur te vinden (Peepkorn et al., 2024).
- **Promptformulering:** De prompt bepaalt hoe duidelijk de vraag, rubric, context en opdracht van het nakijken aan het model wordt gepresenteerd. Een goed geformuleerde prompt helpt het model om de vraag en verwachtingen correct te interpreteren (Qian et al., 2024). We testen verschillende versies van prompts, zoals:
  - Een korte set instructies (Bij te veel instructies raakt het model misschien verward)
  - Een lange set instructies (Meer instructies geeft wellicht een duidelijker beeld van de opdracht)

De combinatie van een optimale temperatuur en een zorgvuldig opgebouwde prompt zorgt ervoor dat de feedback nauwkeuriger en betrouwbaarder wordt. Dit wordt getest door verschillende instellingen uit te proberen op een set van voorbeeldantwoorden en de uitkomsten te vergelijken met menselijke beoordelingen. Per vraag testte we drie verschillende antwoorden. Een goed antwoord, en helemaal fout antwoord en een half goed, half fout antwoord.

### B.3 Analyseren

Eigenaar:	<i>Joost</i>
Doel(en):	<ul style="list-style-type: none"><li>• Docenten inzicht geven in de resultaten van een klas en zien welke onderwerpen aandacht nodig hebben.</li><li>• Docenten inzicht geven in de betrouwbaarheid van de toets, door opvallende statistische resultaten weer te geven.</li></ul>
Subvragen:	<ul style="list-style-type: none"><li>• Hoe doe een een statistische analyses van toetsresultaten?</li><li>• Hoe geef je deze resultaten overzichtelijk weer?</li></ul>
Kader(s):	<ul style="list-style-type: none"><li>• Statistiek</li><li>• UI (user interface)</li></ul>
Geschatte tijdkosten:	15 uur

Bij het inscannen gaan we onderzoek doen naar welke statistische berekeningen nodig zijn voor correct analyse. Ook gaan we uitzoeken wat voor interface nodig is en welke berekeningen of dingen we aan een docent kunnen laten zien om een duidelijk beeld te krijgen van de staat van de klas. We gaan gebruik maken van internet.

Ook gaan we een test website maken met VueJs (VueJs, 2024). De code voor deze website is open-source en is te vinden op <https://github.com/Thannie/Test-Analyzer>. Daarnaast hosten we zelf de website op

<https://toetspws.web.app/analyze>, waar nu een testversie van een analyseer app staat met test data. In het analyseer menu in de fold-out correlaties is een mooie correlatie matrix te zien en in het grafiek menu een interactieve normaalverdeling als op bereken wordt geklikt. De interface hebben we gemaakt met Vuetify (VuetifyJs, 2024), dat zorgt voor de mooie menu's (merk op dat <https://toetspws.web.app/scan> naar de inscan website gaat, die gebruikt maakt van onze inscan API).

## B.4 Enquête

<b>Eigenaar:</b>	<i>Jonathan en Joost</i>
<b>Doel(en):</b>	• Inzicht krijgen in de mogelijkheid in de integratie van AI bij docenten op Het Amsterdamse Lyceum.
<b>Subvragen:</b>	<ul style="list-style-type: none"> <li>• Hoe neem je een betrouwbare enquête?</li> <li>• Hoe zorg je ervoor dat mensen jouw enquête willen invullen?</li> </ul>
<b>Kader(s):</b>	• TODO
<b>Geschatte tijdskosten:</b>	20 uur

Om te testen of docenten überhaupt open staan voor een ai model hebben we een enquête verstuurd naar alle docenten van Het Amsterdams Lyceum. Om een betrouwbare enquête te maken moet je als eerste het doel van de enquête duidelijk hebben. In ons onderzoek waren dat de volgende:

- target voor ons programma stellen
- mogelijke acceptatie in kaart brengen

Daarnaast moet elke vraag ook een duidelijk doel hebben, anders is het mogelijk dat je 2x dezelfde vraag stelt of naar informatie gaat vragen niet relevant is.

Ten slotte moesten we bij elke vraag nagaan of de vraag op verschillende manieren geïnterpreteerd kan worden.

Dit waren de vragen die we hebben bedacht:

Vraag	Doel	Verklaring
<b>1. Wat is uw vakgebied? (Indien meerdere, kies vak met meeste uren a.u.b.)</b> Vakken	kunnen filtreren op vakgebied en vakgroep ( $\alpha, \beta, \gamma$ )	
<b>2. Hoeveel jaar bent u al docent?</b> <ul style="list-style-type: none"> <li>• 1-5 jaar</li> <li>• 5-10 jaar</li> <li>• Meer dan tien jaar</li> <li>• Minder dan één jaar</li> </ul>	kunnen filtreren op lesgeef ervaring	
<b>3. Bent u bekend met het concept van (generatieve) AI?</b> <ul style="list-style-type: none"> <li>• Ja, ik ben goed op de hoogte</li> <li>• Ja, ik heb er wel eens over gehoord</li> <li>• Nee, ik ben niet bekend met deze technologie</li> </ul>	kunnen filtreren op ai ervaring	

Vraag	Doel	Verklaring
<p><b>4. Wat zouden voor u redenen zijn om AI te gebruiken voor het nakijken van proefwerken? (Meerdere antwoorden mogelijk)</b></p> <ul style="list-style-type: none"> <li>• Tijdbesparing</li> <li>• Objectiviteit in de beoordeling</li> <li>• Vermindering van de werkdruk</li> <li>• Snelheid van de terugkoppeling naar studenten</li> <li>• Betere nauwkeurigheid</li> <li>• Ik zou nooit overwegen AI hierbij te gebruiken</li> <li>• Anders: <i>zelf invullen</i></li> </ul>	weten wat het hoofddoel moet zijn van ons programma en wat waar we minder aandacht aan kunnen besteden	optie 1 en 3 lijken op elkaar, maar zijn niet hetzelfde dat is later ook deels terug te zien in de resultaten
<p><b>5. Wat zijn uw belangrijkste zorgen bij het gebruik van AI voor het nakijken van proefwerken?</b></p> <ul style="list-style-type: none"> <li>• Gebrek aan menselijke empathie in de beoordeling</li> <li>• Mogelijke technische fouten</li> <li>• Onvoldoende aandacht voor subjectieve antwoorden</li> <li>• Data- en privacykwesties van studenten</li> <li>• Oneerlijke of bevooroordeerde beoordelingen</li> <li>• Afhankelijkheid van technologie</li> <li>• Anders: <i>zelf invullen</i></li> </ul>	weten waar we op moeten focussen en wrten of bepaalde problemen een grote bottleneck zullen zijn voor de acceptatie van ons programma voor docenten	
<p><b>6. Denkt u dat AI zelfstandig toetsen zou kunnen nakijken</b></p> <ul style="list-style-type: none"> <li>• Ja</li> <li>• Nee</li> <li>• Weet ik niet</li> </ul>	weten hoe positief docenten in een mogelijkheid zijn en om te vergelijken met vakgebied en lesgeef ervaring	

Vraag	Doel	Verklaring
<p><b>7. Hoeveel leerlingen trekken uw beoordeling per toets terecht of niet in twijfel? (Een getal)</b> Zelf een getal invullen</p>	<p>Een objectief target halen waarmee we ons programma kunnen vergelijken: meer oneens is slechter of te streng naar gekeken te weinig is te makkelijk nagekeken</p>	<p>Hier is een afweging gemaakt tussen een makkelijk te lezen vraag en een wiskundig correcte vraag. Het is hier niet expliciet gezegd dat het per toets per klas (van ongeveer 30 leerlingen) is.</p>
<p><b>8. Welke invloed denkt u dat de inzet van AI kan hebben op de relatie tussen docent en student? (Open vraag)</b> Zelf een getal invullen</p>	<p>Als docenten nog wat kwijt willen kunnen ze dat hier doen, misschien staat er wat interessants tussen</p>	

Ons 2e doel van dit onderdeel was: **Hoe zorg je ervoor dat mensen jouw enquête willen invullen?**

Uit ons kleine omgevingsonderzoek blijkt dat docenten van Het Amsterdams Lyceum niet vaak reageren op (onbelangrijke) mail. Een score van 30% zou al aan de hoge kant zijn. We hebben een zakelijk mailtje proberen samen te stellen die ervoor zorgt dat docenten wilden reageren.

Geachte docenten van Het Amsterdams Lyceum,

In het kader van ons profielwerkstuk, maken wij een programma dat dat toetsen kan inscannen, nakijken en analyseren. Daarnaast zijn we geïnteresseerd in hoe docent denken over het nakijken met AI, hierbij zouden wij graag uw hulp willen.

<https://forms.office.com/e/j5cYFrAy7p>  
Hoogachtend,

Joost Koch & Jonathan Wijker

Op dit mailtje hebben we 22 reacties gekregen. Dat vonden wij redelijk tegenvallen. Een rede voor deze teleurstellende respons zou kunnen zijn dat we de mail verstuurd hebben op donderdag 17 oktober. Dat was de donderdag voor de activiteitenweek, waardoor docenten met uitzicht op een vakantie misschien geen zin hadden in het invullen van een PWS-enquête.

Daarna hebben de gekeken wat het ideale moment zou zijn voor een docent om zin te hebben in het invullen van een enquête. Toen kwamen we na overleg met diverse docenten erachter dat de week voor de toetsweek het rustigst is, want de meeste docenten hebben alle lesstof al behandeld, geen toetsen om na

te kijken en hebben de toetsen en SE's al af en ingelevert.

We hebben ons tweede mailtje op de maandag voor de toetsweek gestuurd. We hebben ook de docenten extra proberen te vleien door duidelijk aan te geven dat het weinig tijd kost en dat we weten hoe druk docenten het eigenlijk hebben.

Geachte docent,

Onlangs hebben wij u een enquête gestuurd en we hebben al wat reacties mogen ontvangen, bedankt daarvoor.

We snappen dat u het komende tijd druk heeft met de toetsweek, maar hopen dat u komende week ergens een gaatje van 2-3 minuten kunt vinden om alsnog onze enquête in te vullen. Dit zouden we erg waarderen!

<https://forms.office.com/e/j5cYFrAy7p>

Hoogachtend,

Joost Koch & Jonathan Wijker

Dit leverde 28 extra responses op, waardoor we op een totaal van 50 zitten. Dit is statistisch gezien goed genoeg om iets over de meningen van de docenten te zeggen op Het Amsterdams Lyceum.

We moeten er wel rekening mee houden dat sommige secties misschien minder zullen hebben gereageerd, waardoor ons resultaat over die sectie minder betrouwbaar zal zijn.

Voor het analyseren gaan we Google Sheet pivot tables gebruiken om snel verbanden tussen de data te zien. Ook zullen we kijken of ChatGPT of Google Gemini relevante ontdekkingen kunnen doen in de data.

## B.5 Praktijktest

**Eigenaar:** *Jonathan*

**Doel(en):** • Een toets maken die duidelijk is voor 3e klassers

• Een toets nakijken met onze programmas

**Subvragen:** • Hoe maken we een toets die duidelijk is voor 3e klassers?

• Hoe kijken we een toets na met onze programmas?

**Kader(s):** • Toetsen maken / scheikunde

• Programma testen

**Geschatte tijdkosten:** 15 uur

# 4 Resultaten

## A Inscannen

Hieronder zijn een aantal tabellen te zien met de resultaten van het inscanonderdeel uitgelegd in de methode.

We hebben deze tabellen gekozen om te kijken wat de beste inputs voor ons inscansysteem zijn.

Een lagere score is beter dan een hogere score, want hoe meer fouten hoe hoger de score. Ook willen we liever een lage standaarddeviatie, want dan is ons model voorspelbaarder.

In de appendix is zijn een aantal test gegeven in een tabel met een quote van Johan Cruijff die een goed beeld geven van wat een score betekent en een aantal andere tests.

De resultaten zijn niet normaal verdeeld, omdat een score niet kleiner dan 0 kan zijn, daar door zal die vaak hoger zijn dan je zou denken. Er wordt vooral gekeken naar verschillende standaarddeviaties ten opzichte van elkaar.

### Resultaten

Table 4: # per model

Model	#
gemini-1.5-flash-8b	1125
gemini-1.5-pro-002	787
gemini-2.0-flash-exp	1257
gpt-4o	1029
gpt-4o-mini	1167
<b>Grand Total</b>	<b>5365</b>

Table 5: score per model

Model	avg cor score	stdev cor score	avg cor change count
gemini-1.5-pro-002	0.4838	0.4913	3.0851
gemini-2.0-flash-exp	0.5874	0.5414	6.9880
gpt-4o-mini	0.5905	1.6675	5.5089
gpt-4o	0.8812	2.6390	4.0223
gemini-1.5-flash-8b	1.2887	2.0987	6.2506
<b>Grand Total</b>	<b>0.7763</b>	<b>1.7469</b>	<b>5.3703</b>

Table 6: score per testfoto

<b>image</b>	<b>avg cor score</b>	<b>stdev cor score</b>	<b>avg cor change count</b>
image 10 kort leesbaar	0.1867	1.1019	1.8679
image 11 kort leesbaar uitgekrast	0.2567	0.6206	1.7862
image 14 gekreukeld met pijlen	1.1113	0.2310	10.0095
image 12 kort onleesbaar	1.2250	3.2430	3.9123
image 13 slecht leesbaar pijlen	1.3226	1.0186	11.7194
<b>Grand Total</b>	<b>0.7763</b>	<b>1.7469</b>	<b>5.3703</b>

Table 7: score per opdracht

<b>base_command</b>	<b>avg cor score</b>	<b>stdev cor score</b>	<b>avg cor change count</b>
lange uitleg bij elk veld	0.6768	1.3409	4.6585
huidige opdracht met uitleg bij elk veld	0.7828	1.5641	5.8461
de makkelijkste opdracht zonder extra uitleg	0.8583	2.1735	5.5504
<b>Grand Total</b>	<b>0.7763</b>	<b>1.7469</b>	<b>5.3703</b>

Table 8: score per context additie

<b>addition</b>	<b>avg cor score</b>	<b>stdev cor score</b>	<b>avg cor change count</b>
toets	0.6546	1.4049	4.4909
antwoordmodel bij vraag	0.6619	1.3282	5.6849
stof-antwoordmodel bij vraag-specifieke vraag	0.7291	1.2057	6.2406
stof	0.7434	2.2330	4.7515
antwoordmodel bij vraag-specifieke vraag	0.7464	1.1751	5.576
stof-toets-antwoordmodel	0.7807	1.9973	5.3795
<b>Grand Total</b>	<b>0.7842</b>	<b>1.8166</b>	<b>5.3907</b>

Table 9: zekerheid en handschriftscore per model

Model	avg delta time in s	stdev delta time in s
gemini-1.5-flash-8b	2.4	1.2
gemini-2.0-flash-exp	3.3	1.4
gemini-1.5-pro-002	4.6	9.1
gpt-4o-mini	5.4	4.5
gpt-4o	7.6	7.5
<b>Grand Total</b>	<b>2.6516</b>	<b>5.0869</b>

Table 10

Model	avg certainty	stdev certainty	avg handwriting	stdev handwriting
gemini-1.5-pro-002	96.0585	1.9220	77.2206	16.9986
gemini-2.0-flash-exp	94.8678	0.8026	81.1512	8.46961884
gemini-1.5-flash-8b	92.9004	7.7042	84.9376	9.6381
gpt-4o	89.5170	10.7645	86.03292	9.2210
gpt-4o-mini	87.9820	5.3663	75.4734	9.8103
<b>Grand Total</b>	<b>91.1381</b>	<b>7.7582</b>	<b>81.5676</b>	<b>10.7848</b>

## B Nakijken

### resultaten

Table 11: Correlatie modellen onder elkaar

Kijk vooral naar de correlatie vs de mens, want die wordt gezien als "correct".

Model	Mens	g-2.0-flash	g-1.5-pro	g-1.5-flash	4o	4o-mini	mode AI
Mens	1.0000						
gemini-2.0-flash-exp	0.9616	1.0000					
gemini-1.5-pro	0.9054	0.9291	1.0000				
gemini-1.5-flash	0.9090	0.9025	0.9497	1.0000			
4o	0.9509	0.9568	0.9406	0.9485	1.0000		
4o-mini	0.9141	0.9289	0.9140	0.9232	0.9394	1.0000	
mode AI	0.9544	0.9680	0.9630	0.9518	0.9889	0.9464	1.0000

Table 12: Verschil met mens per nagekeken vraag %

Hier is te zien dat de meeste modellen geen problemen hebben met vraag 1

Vraag 5 is interessant om naar te kijken want dat was de foto/teken vraag.

Model	1	2	3	4	5	6	7	GrandTotal
gemini-2.0-flash-exp	0.00	4.44	16.09	7.11	7.12	4.65	9.67	7.69
4o	4.35	2.27	6.23	10.82	4.87	8.83	11.74	7.41
gemini-1.5-flash	4.35	0.00	13.19	16.06	9.23	4.65	18.56	10.63
gemini-1.5-pro	4.35	2.27	17.39	13.05	9.23	8.83	17.07	11.13
o1-mini	0.00	16.65	14.69	2.98	19.23	4.65	7.47	10.11
mode	4.35	0.00	4.25	10.82	9.23	6.79	13.66	7.41

Table 13: Aantal valse positieve en aantal valse negatieve

Dit is ten opzichte van de mens.

een hogere waarde betekend minder goed.

Hier is te zien wanneer een model voordelig is voor de leerling.

Model	1	2	3	4	5	6	7	GrandTotal
gemini-2.0-flash-exp	0	2	9	2	2	0	0	15
false positief								
gemini-2.0-flash-exp	0	0	0	3	1	2	4	10
false negatief								
mode false positief	0	0	2	2	2	0	2	8
mode false negatief	1	0	0	6	2	3	4	16

## C Analyseren

**Opbouw analyse** Om een toets betrouwbaar te analyseren moet met verschillende dingen rekening houden. Een van de belangrijkste dingen voor een analyse is het doel van de docent vaststellen. Wil een docent een kennismeting doen waar de mensen die het half snappen een onvoldoende krijgen of dat die net een 5.5 krijgen. Wil een docent dat het goed genoeg begrijpen van de stof beloont wordt met een 8.0 of met een 6.0. Deze dingen zijn belangrijk voor het beoogde gemiddelde en de standaarddeviatie van een toets. **gemiddelde:**

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

waarbij:

- $x_i$ : de individuele datapunten,
- $N$ : het totale aantal datapunten,
- $\mu$ : het gemiddelde.

**Standaard deviatie:**

$$s = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

waarbij:

- $s$ : de standaarddeviatie van de steekproef,
- $\bar{x}$ : het gemiddelde van de steekproef.

De **covariantie** meet de gezamenlijke variabiliteit van twee variabelen en wordt berekend met:

$$\text{Cov}(X, Y) = \frac{1}{N} \sum_{i=1}^N (x_i - \mu_X)(y_i - \mu_Y)$$

waarbij:

- $X, Y$ : twee willekeurige variabelen,
- $x_i, y_i$ : de individuele waarnemingen van  $X$  en  $Y$ ,
- $\mu_X, \mu_Y$ : de gemiddelden van  $X$  en  $Y$ ,
- $N$ : het totale aantal waarnemingen.

**Vraagniveau** Om met deze waardes een analyse te doen van een toets op vraagniveau kan je bepaalde berekeningen gebruiken. Als je wilt weten of een vraag in de toets thuisvoert kan je de correlatie berekenen tussen hoe de leerlingen de vraag hebben gemaakt ten opzichte van de rest van de toets (Ding and Beichner, 2009; Talebi et al., 2013). **De RIR** meet de correlatie tussen de score van een item en de totale score, exclusief dat item:

$$RIR = \frac{\text{Cov}(x_i, S_{-i})}{\sigma_{x_i} \cdot \sigma_{S_{-i}}}$$

waarbij:

- $x_i$ : de score van een individueel item,
- $S_{-i}$ : de totale score exclusief  $x_i$ ,
- $\text{Cov}(x_i, S_{-i})$ : de covariantie tussen  $x_i$  en  $S_{-i}$ ,
- $\sigma_{x_i}$ : de standaarddeviatie van  $x_i$ ,
- $\sigma_{S_{-i}}$ : de standaarddeviatie van  $S_{-i}$ .

**De RIT** meet de correlatie tussen de score van een item en de totale score, inclusief dat item:

$$RIT = \frac{\text{Cov}(x_i, S)}{\sigma_{x_i} \cdot \sigma_S}$$

waarbij:

- $x_i$ : de score van een individueel item,
- $S$ : de totale score inclusief  $x_i$ ,
- $\text{Cov}(x_i, S)$ : de covariantie tussen  $x_i$  en  $S$ ,
- $\sigma_{x_i}$ : de standaarddeviatie van  $x_i$ ,
- $\sigma_S$ : de standaarddeviatie van  $S$ .

**Representatie en UI** Om deze formules bruikbaar te maken voor docenten zou je een interface kunnen maken met **een input, analyse en bewerk/pas-aan scherm**.

In die **inlaad pagina** moet een docent toetsresultaten uit een Excel of uit een van onze andere modules kunnen inladen. Daarnaast hebben wij ook van docenten te horen gekregen dat ze het fijn zouden vinden om leerdoelen aan vragen te koppelen, opdat zij een betere terugkoppeling kunnen krijgen.

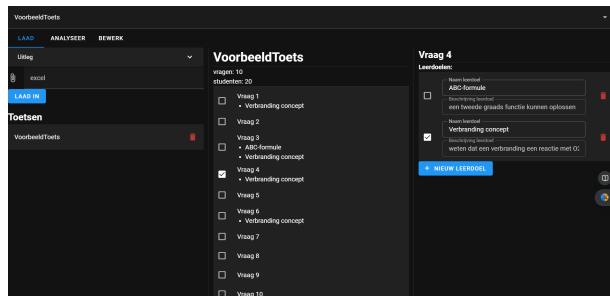


Figure 24: Voorbeeld inlaadpagina

In het **analyse scherm** moet voor een docent overzichtelijk weergegeven zijn welke vragen goed en minder goed gingen en de scores van de leerlingen. Hier moet ook te zien zijn welke vragen waarschijnlijk niet thuis horen in de toets, omdat de mensen met een hoog cijfer hem fout hebben, dit kan komen dat die leerlingen te ver doordenken en daardoor de vraag fout hebben. De vraag is of je die leerlingen wilt afstraffen voor het verder denken dan het juiste antwoord.

Het is ook mogelijk om hier opvallende correlaties tussen vragen weer te geven. Hier kan bijvoorbeeld worden laten zien dat mensen die vraag 5 fout hebben ook vraag 8 fout hebben. Dan is het mogelijk dat er 2x om dezelfde kennis wordt gevraagd, iets wat een toetsresultaat minder betrouwbaar maakt, omdat de stof dispropositieel wordt getoetst (dit geldt ook voor een hoge correlatie tussen 2 juist gemaakte vragen).

Het kan ook mogelijk zijn om de correlaties tussen leerlingen te tonen om eventuele spiekers te vangen. Hierbij moet wel rekening gehouden worden met het feit dat 2 mensen met een hoog cijfer, waarschijnlijk beide dezelfde vragen goed en fout hebben. Deze correlatie wordt wel interessant bij bijvoorbeeld 2 6.5'en en precies dezelfde fouten.

Op **het bewerkscherm** kunnen bijvoorbeeld een aantal velden komen met de doelen van een docent. Bijvoorbeeld: een veld om het gewilde gemiddelde en de gewilde standaarddeviatie in te stellen. Daarmee berekent hij dat een nieuwe formule. Hier kan ook worden of een lineaire of non-lineaire formule gebruikt wordt. Bij een non-lineaire formule behoudt iedereen met een onvoldoende zijn onvoldoende, maar zijn die onvoldoendes minder hoog. Hier kan ook op vraagniveau een scherm zijn om vragen eruit te gooien of half te laten meetellen, als blijkt dat ze wegnemen van de kennismeting van de toets.

## D Enquête

### resultaten

Table 14: Bent u bekend met het concept van (generatieve) AI? vs lesgeef ervaring

<b>Docent?</b>	<b>Ja</b>	<b>Gehoord</b>	<b>Nee</b>	<b>Grand Total</b>
1-5 jaar	4	4		8
5-10 jaar	5	4		9
Meer dan tien jaar	13	16	1	30
Minder dan één jaar	1	1		2
<b>Grand Total</b>	<b>23</b>	<b>25</b>	<b>1</b>	<b>49</b>

Table 15: Bent u bekend met het concept van (generatieve) AI? vs vakgroep

<b>Vakgroep</b>	<b>Op de hoogte</b>	<b>Ja</b>	<b>Nee</b>	<b>Totaal</b>
alpha	50.00%	50.00%		100.00%
beta	50.00%	42.86%	7.14%	100.00%
gamma	28.57%	71.43%		100.00%
<b>Totaal</b>	<b>46.94%</b>	<b>51.02%</b>	<b>2.04%</b>	<b>100.00%</b>

Table 16: Denkt u dat AI zelfstandig toetsen zou kunnen nakijken? vs lesgeef ervaring

<b>Docent?</b>	<b>Ja</b>	<b>Nee</b>	<b>Weet niet</b>	<b>Totaal</b>
1-5 jaar	37.50%	50.00%	12.50%	100.00%
5-10 jaar	44.44%	33.33%	22.22%	100.00%
Meer dan 10 jaar	23.33%	40.00%	36.67%	100.00%
Minder dan 1 jaar		50.00%	50.00%	100.00%
<b>Totaal</b>	<b>28.57%</b>	<b>40.82%</b>	<b>30.61%</b>	<b>100.00%</b>

Table 17: Denkt u dat AI zelfstandig toetsen zou kunnen nakijken? vs vakgroep

<b>Vakgroep</b>	<b>Ja</b>	<b>Gehoord</b>	<b>Nee</b>	<b>Totaal</b>
alpha	28.57%	28.57%		57.14%
beta	14.29%	12.24%	2.04%	28.57%
gamma	4.08%	10.20%		14.29%
<b>Totaal</b>	<b>46.94%</b>	<b>51.02%</b>	<b>2.04%</b>	<b>100.00%</b>

Table 18: Denkt u dat AI zelfstandig toetsen zou kunnen nakijken? vs Bent u bekend met het concept van (generatieve) AI?

<b>AI Toetsen?</b>	<b>Ja</b>	<b>Gehoord</b>	<b>Nee</b>
Ja	12.24%	16.33%	
Nee	20.41%	18.37%	2.04%
Weet niet	14.29%	16.33%	
<b>Totaal</b>	<b>46.94%</b>	<b>51.02%</b>	<b>2.04%</b>

Table 19: Denkt u dat AI zelfstandig toetsen zou kunnen nakijken? vs vakgroep

<b>Vakgroep</b>	<b>Ja</b>	<b>Nee</b>	<b>Weet niet</b>	<b>Totaal</b>
alpha	7	11	10	28
beta	4	5	5	14
gamma	3	4		7
<b>Totaal</b>	<b>14</b>	<b>20</b>	<b>15</b>	<b>49</b>

Table 20: Hoeveel leerlingen trekken uw beoordeling per toets - terecht of niet - in twijfel? (Een getal) vs vakgroep

<b>Vakgroep</b>	<b>Avg. twijfel</b>	<b>Aantal</b>
alpha	1.593	28
beta	2.000	14
gamma	5.143	7
<b>Totaal</b>	<b>2.229</b>	<b>49</b>

Table 21: Wat zouden voor u redenen zijn om AI te gebruiken voor het nakijken van proefwerken?

<b>Voordelen</b>	<b>Tijdsbesp.</b>	<b>Objectiv.</b>	<b>Werkdruk</b>	<b>Snelheid</b>	<b>Nauwk.</b>	<b>Nooit</b>
<b>Totaal</b>	40	14	31	18	8	7
(%)	80%	28%	62%	36%	16%	14%

Table 22: Wat zijn uw belangrijkste zorgen bij het gebruik van AI voor het nakijken van proefwerken?

<b>Zorgen</b>	<b>Empathie</b>	<b>Tech. fout</b>	<b>Subjectiv.</b>	<b>Privacy</b>	<b>Bevoor.</b>	<b>Afhank.</b>	<b>Geen</b>
<b>Totaal</b>	20	26	31	11	11	15	0
<b>Totaal (%)</b>	<b>40%</b>	<b>52%</b>	<b>62%</b>	<b>22%</b>	<b>22%</b>	<b>30%</b>	<b>0.00%</b>

**De gemaakte punten in de resultaten van de open vraag.** De vraag was: "Welke invloed denkt u dat de inzet van AI kan hebben op de relatie tussen docent en student?"

Voor het maken van deze lijst is voor het sorteren en overzichtelijk maken van de punten gebruik gemaakt van het GPT model Gemini Experimental 1121 op aistudio.google.com. Zie appendix voor prompt.

### 1. Negatieve invloed op de relatie en kennis van de docent

- (a) Docent is minder goed op de hoogte van de inhoud van het leerlingwerk, wat informatie geeft over wat de leerling bezighoudt.
- (b) Verminderd zicht op leerproces en denkstijl van de leerling.
  - Docent kan sterke punten en ontwikkelpunten minder goed identificeren en hierop inspelen in de lessen.
  - Docent mist mogelijk belangrijke informatie uit persoonlijke verhalen in schrijfopdrachten.
- (c) Docent voelt minder verantwoordelijkheid voor het nakijken en leerling kan moeilijker zijn recht halen.
- (d) Minder intensief contact op niveau van interpretatie van antwoorden
- (e) Docent moet mogelijk de AI-correctie zelf controleren, wat extra werk oplevert.
- (f) Docent verliest mogelijk het zicht op de leercurve van de leerling.
- (g) De professionele expertise van de docent wordt ondermijnd.
  - Leerlingen kunnen denken dat docenten makkelijk vervangbaar zijn.
  - Het werk van de docent kan in achting dalen bij leerlingen.

### 2. Afstand en verminderd persoonlijk contact

- (a) AI als nakijker of tutor kan de band minder persoonlijk maken.
- (b) Vervreemding tussen leraar en leerling en ten opzichte van zichzelf.
- (c) Verlies van menselijk contact en emotionele band, cruciaal voor effectief leren.
- (d) Dehumanisering van het onderwijs door rigide en onpersoonlijke AI-systemen.
- (e) AI kan leiden tot apathie bij de leerling.
- (f) Relatie wordt anoniemer en onpersoonlijker.
- (g) Leerlingen zouden kunnen denken dat leraren niet essentieel zijn als AI ze kan vervangen.

### 3. Discussie en twijfel over objectiviteit en correctheid

- (a) Discussies over de correctheid van AI-gegeven antwoorden en beoordelingen.
- (b) Leerlingen leren de subjectiviteit van zaken niet en dat niet alles zwart-wit is.
- (c) Leerlingen zullen minder de neiging hebben om in discussie te gaan over de resultaten.

### 4. Potentieel positieve invloed, afhankelijk van de implementatie

- (a) AI kan de relatie versterken als docent en leerling samen leren AI te gebruiken voor oefenen, feedback en beoordelen.
- (b) Docent heeft meer tijd voor andere taken zoals het zoeken naar passend lesmateriaal.
- (c) Docent kan meer ontspannen zijn door vermindering van nakijkwerk.
- (d) AI kan een mediërende functie hebben door objectief vast te stellen of een antwoord fout is, waardoor docent en leerling zich kunnen richten op het 'waarom'.
- (e) AI kan voor leerling objectiever overkomen, waardoor minder snel gedacht wordt dat een punt niet gegund wordt door persoonlijke redenen.
- (f) AI kan een suggestie van neutraliteit geven bij beoordeling.
- (g) AI kan bijdragen aan de relatie mits verstandig, met dat doel en openlijk ingezet.
- (h) Onderwijs kan beter worden afgestemd op individuele behoeften van leerlingen.
- (i) Positieve invloed mits correct werkend.

## 5. Geen of weinig invloed

- (a) De relatie wordt voornamelijk bepaald door direct contact en hoe omgegaan wordt met discussies over beoordeling.
- (b) De relatie hoeft niet beïnvloed te worden.
- (c) AI kan het probleem van een tekort aan docenten oplossen.
- (d) Er is weinig ruimte voor AI om de relatie te verbeteren. Conflicten hebben vaak dieperliggende oorzaken dan meningsverschillen over nakijkwerk.
- (e) Geen invloed op de relatie.
- (f) Docent wil zelf verantwoordelijk blijven en met leerlingen in gesprek blijven.

## 6. Bezorgdheid over het gebruik van AI

- (a) Zorgen over studenten die AI gebruiken om te schrijven en docenten die AI gebruiken om te controleren.
- (b) AI wordt verkeerd ingezet, zou alleen voor routineklussen gebruikt moeten worden.
- (c) AI wordt liever zoveel mogelijk buiten de deur en al helemaal uit het onderwijs gehouden.

## 7. Afhankelijkheid van het vak en type toets

- (a) Inzet van AI is afhankelijk van het vak; bijv. onhandig bij beeldende kunst, handig bij multiple choice.
- (b) Niet alle toetsen zijn geschikt voor AI, bijvoorbeeld de beoordeling van de schoonheid van een tekening of website.

## 8. Onzekerheid over de invloed

- (a) Het is nog te vroeg om te zeggen wat de invloed zal zijn.

## E Praktijktest

# 5 conclusie

## A Inscannen

### Vragen:

#### 1. Welke manieren zijn er om een de vraagsecties op een foto te scheiden?

Van de 3 manieren die we hebben getest blijkt dat 1 methode onbruikbaar is en de andere 2 bruikbaar, maar in andere omstandigheden Tabel.

De methode die niet werkt is de kantlijnmethode. Het is niet mogelijk om ervan uit te gaan dat leerlingen alleen maar vraagnummers in de kantlijn zetten en daar alle secties op te baseren, want een kleine textherkennings fout kan antwoord op een vraag splitsen, waardoor iemand alle punten misloopt.

De **QR-code** methode werkt goed voor werkbladen. Bijvoorbeeld een examen waar een leerling zo min mogelijk wil uitprinten. De leerling zou dan bijvoorbeeld een examenvraag uitprinten met daaronder direct het qr-antwoordveld, daar een foto van nemen die dan direct nagekeken wordt.

De **Checkbox** methode is de methode die

het beste toepasbaar is voor een toets. Dordat er weinig/geen fouten worden gemaakt in het herkennen van een sectie, wat het doel is van deze deelvraag. Er is wel een leercurve voor de leerling. Na onze toets in de 3e klas blijkt dat die gymnasium 3 leerlingen er weinig moeite mee hadden, nadat wij het voordeden. Het kost een docent wel extra werk om tijdens het innemen van de toets te checken of iedereen op elk blaadje zijn id heeft opgeschreven. Bij ons toetsje waren er nog een aantal leerlingen die het nummer op de achterkant waren vergeten. Er was 1 sectienummer verkeerd ingelezen, maar dat soort errors kunnen gedetecteerd worden, door bijvoorbeeld te checken of deze leerling deze vragen misschien al een keer beantwoord heeft. Als dat zo is kan een docent met de hand checken welk leerling ID erbij past.

## 2. Wat is de beste manier om tekst uit een ingescande sectie te halen?

Uit de tests kunnen we ten eerste vaststellen dat de modellen van Google sneller zijn dan de modellen van OpenAI, zie:tabel 9. Zoals verwacht zijn de flash modellen het snelst, maar we vonden het wel opvallend dat het verschil oploopt tot wel 3x sneller.

We verwachtten dat het nieuwste model meestal de beste score zou hebben, maar dat is kennelijk niet helemaal waar, want de teksterkenning van 1.5 pro heeft een betere gemiddelde score dan de gemini flash 2.0, die midden december 2024 is uitgekomen. Zie:tabel 5. Dat is vooral te zien aan het gemiddelde aantal aanpassingen in een tekst, die bij flash 2.0 meer dan 2x zo groot is als 1.5 pro. Het (verwaarloosbare) nadeel is dat de standaarddeviatie van gemini 1.5 pro meer dan 6x zo groot is als die van flash 2.0.

Zoals we verwacht hadden geeft een langere prompt een beter resultaat, zie:tabel

7. Daarom hebben we ervoor gekozen om hiermee onze 3e klas toets in te scannen.

Wat ons ook opviel is dat de context

van de hele toets de beste inscanresultaten gaf, zie tabel 8. Wij hadden verwacht dat de rubric en de vraag de laagste (beste) score zou geven, maar het blijkt dat de toets de beste context geeft. Kennelijk begrijpen de modellen beter welke woorden iemand wil gebruiken als ze de hele toets hebben. Dat is ook te zien aan de gemiddelde veranderingen. Het is wel opvallend dat bij de rubric en de vraag geeft ook zo'n lage change count. De standaarddeviatie is daarintegen wel hoog, dus die combinatie is bij sommige vragen misschien beter dan de hele toets. Toch gaan we kiezen voor de vraag en antwoordmodel additie, want een hele toets zorgt voor een grote vermeerdering in tokens/kosten. Daarnaast neem de tijd per request toe.

De opdracht die we gaan gebruiken:

<b>Model</b>	Gemini 1.5 pro 002
<b>Temperatuur</b>	0.5
<b>Prompt</b>	lange uitleg bij elk veld
<b>Additie</b>	vraag en antwoordmodel

## B Nakijken

## C Analyseren

Dit onderzoek analyseert toetsresultaten, met als doel betrouwbare kennismetingen. Cruciaal is dat de analyse aansluit bij het beoogde doel van de docent, wat invloed heeft op het gewenste gemiddelde en de standaarddeviatie.

Naast bekende statistische maten als gemiddelde, standaarddeviatie en covariatie, zijn op vraagniveau de Item-Rest Correlatie (RIR) en Item-Totaal Correlatie (RIT) toegepast om de bijdrage van individuele vragen aan de totale score te beoordelen. Deze visualisaties zijn gemaakt m.b.v Google Gemini 1.5 pro.

Een voorgestelde gebruikersinterface (UI) met een input-, analyse- en bewerkscherm faciliteert de praktische toepassing.

- **Input:** Toetsresultaten uploaden en leerdoelen koppelen.
- **Analyse:** Prestaties op vraag- en leerlingniveau, inclusief correlaties.
- **Bewerking:** Gewenst gemiddelde/standaarddeviatie instellen, vragen aanpassen.

## D Enquete

## E Praktijktest

# 6 Discussie

## A Foutenanalyse

### A.1 Inscannen

- Tijdens het testen van instellingen zijn er een aantal gefaalt, omdat er te veel requests per seconde werden gestuurd. Dit kan ervoor hebben gezorgd dat de resultaten verdraaid zijn.
- Er kan meer getest worden met de ideale prompt die een beter resultaat geeft.
- Tijdens het toetsje zijn een aantal vragen verkeerd ingescand, omdat een leerling de checkbox verkeerd had ingevuld. Het zou netter zijn om dit als een popup aan een docent te laten zien die dit dan corrigeert.

### A.2 Nakijken

### A.3 Analyseren

- De bedachte UI is niet getest, waardoor er niet een 2e iteratie verbeteringen is gedaan.
- We hebben geen echte toetsen geanalyseerd, omdat de toets die we zouden krijgen zo slecht gemaakt was dat er niets te analyseren was.
- Tijdens het toetsje zijn een aantal vragen verkeerd ingescand, omdat een leerling de checkbox verkeerd had ingevuld. Het zou netter zijn om dit als een popup aan een docent te laten zien die dit dan corrigeert.

### A.4 Enquête

- We kregen terug van een aantal docenten dat de vraag aantal leerlingen die het oneens is met een beoordeling niet exact gedefineerd was.
- Er waren relatief weinig gamma resultaten, waardoor hun reactie statistisch gezien minder waarde heeft.
- Dit is een enquête gestuurd naar docenten van Het Amsterdams Lyceum en hiermee

kunnen we nog weinig zeggen over of docenten in de rest van Nederland andere problemen hebben.

### A.5 Praktijktest

- We hebben dit toetsje afgenomen bij een hele lieve gymnasium klas. Het is de vraag op andere klassen andere manieren vinden om het blaadje zo toe te takelen dat het nakijkprocess wordt verhinderd.

## B Terugblik en dankwoord

**Reflectie Joost** Dit project heb ik veel geleerd over het overstappen naar een andere methode. Voor de inscanmodule had ik in het begin ingeschat dat ik maar 10 uur nodig had voor het inscannen. De daadwerkelijke tijd was een stuk meer. De methode die ik had bedacht (tekstherkenning in de kantlijn) bleek niet te werken, maar ik heb wel (onnodig) veel tijd besteed aan het optimaliseren van dat proces. Ik heb ook veel gehad aan het gebruiken van de Google Gemini API, die gebruik ik nu vaak om kleine simpele taakjes op te lossen. Ook heb ik Google Sheets pivot tables leren gebruiken om snel de enquête, inscan en toets resultaten in tabellen te zetten (die ik daarna met Google Gemini naar de latex/pdf tabellen heb omgezet). Het inscan en nakijksysteem zal ik waarschijnlijk in een ander project gaan gebruiken. Waar volgend jaar misschien rekening mee gehouden kan worden is dat de rubrics van het PWS niet duidelijk de opdracht van het PWS weer-spiegelen. Tot fase 3 is eigenlijk literatuuronderzoek en een paar beantwoorde deelvragen. Fase 4 is het onderdeel waar je het "grote" onderzoek moet uitvoeren. Ik vond dat wie die fase pas laat kregen en dat er dingen miste

in de rubric die ik wel verwacht in een wetenschappelijk onderzoek (bijvoorbeeld de indeling of het hebben van resultaten). Ik zie dit PWS meer als een eigen project dan als een opdracht voor school en eigen projectjes vind ik leuk om te doen, zie: [Mijn Github](#).

### Reflectie Jonathan

### Dankwoord

## C vervolgonderzoek

- Een onderzoek/test naar een interface die docenten kunnen gebruiken om de inscan en

nakijk modules makkelijk te gebruiken

- Een onderzoek naar wat docenten op andere scholen belangrijk vinden
- Een onderzoek naar wat docenten van ons de resultaten van ons toetsje vinden en of dit hun mening verandert over de mogelijkheden van nakijken met AI
- Een onderzoek naar hoe zo'n systeem geïntegreerd kan worden in het huidige onderwijs. Denk hierbij aan het printen van de custom toetsblaadjes of het omzetten van huidige toetsen naar een computerformat die een ai kan gebruiken.

# 7 Samenvatting onderzoek

Uit de enquête blijkt dat het grootste deel van de docenten de tijdsbesparing het grootste voordeel vindt. Dit doel hebben we bereikt door een toets van een klas in te scannen en na te kijken in een paar minuten. Afgezien van het inscannen in een kopieerapparaat, kost het geen extra (nakijk)moeite.

De grootste zorg die docenten hadden is dat een ai te weinig aandacht had voor subjectieve beoordelingen en technische fouten. Uit het toetsje wat we hebben gegeven blijkt dat een mens en de computer een correlatie hadden van 0.96, dit is een stuk hoger dan wat sommige docenten hebben met de twee corrector. Voor een formatieve toets moet dit zeker voldoende zijn. Het mooie van GPT's is dat ze alleen maar beter worden.

Alle code voor dit systeem is open-source, waardoor iedereen er gebruik van kan maken voor een volgend onderzoek. Daarnaast kunnen mensen die een beetje verstand hebben

van API's een server die wij hebben opgezet gratis gebruiken, zodat docenten zelf niets hoeven te berekenen of installeren op hun eigen computer.

Het doel van dit onderzoek was om een programma te maken dat het nakijken uit de handen van een docent kan nemen. Dat is gelukt. We hebben een systeem opgebouwd dat ervoor zorgt dat docenten met vertrouwen in technologie een extra feedbackmoment kunnen toevoegen door bij wijze van spreken de eerste 10 minuten van de les een toets af te nemen (die toets kan door AI gegenereerd zijn), zijn uitleg geven, leerlingen zelfstandig aan het werk zetten, foto's door de scanner halen (met telefoon of kopieerapparaat) en 2 minuten later de leerlingen hun scores terug geven, waarbij bij elk rubric punt duidelijk staat wat de leerlingen wel en niet goed heeft gedaan, zodat die daar iets aan heeft in de rest van de les.

# 8 Referenties

- Callison-Burch, C., Osborne, M., & Koehn, P. (2006, April). Re-evaluating the role of Bleu in machine translation research. In D. McCarthy & S. Wintner (Eds.), *11th conference of the European chapter of the association for computational linguistics* (pp. 249–256). Association for Computational Linguistics. <https://aclanthology.org/E06-1032>
- Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8*(6), 679–698. <https://doi.org/10.1109/TPAMI.1986.4767851>
- Ding, L., & Beichner, R. J. (2009). Item analysis for classroom tests: A comprehensive guide. *Journal of Physics Teacher Education Online*, 5(2), 2–12. <https://www1.udel.edu/educ/gottfredson/451/unit9-guidance.htm>
- Farquhar, S., Kossen, J., Kuhn, L., & Gal, Y. (2024). Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017), 625–630. <https://doi.org/10.1038/s41586-024-07421-0>
- Ghassemiazghandi, M. (2024). An evaluation of chatgpt's translation accuracy using bleu score. 14(4). <https://doi.org/https://doi.org/10.17507/tpls.1404.07>
- Gobrecht, A., Tuma, F., Möller, M., Zöller, T., Zakhvatkin, M., Wuttig, A., Sommerfeldt, H., & Schütt, S. (2024). Beyond human subjectivity and error: A novel ai grading system. <https://arxiv.org/abs/2405.04323>

- Google. (2024a). Cloud vision api documentation [Official documentation for Google Cloud Vision API]. <https://cloud.google.com/vision/docs>
- Google. (2024b). Gemini api documentation [Official documentation for Gemini API]. <https://ai.google.dev/gemini-api>
- Hoffstaetter, S. (2024). Pytesseract documentation [Official documentation for the pytesseract library]. <https://pypi.org/project/pytesseract/>
- Kumar, Y., Aggarwal, S., Mahata, D., Shah, R. R., Kumaraguru, P., & Zimmermann, R. (2020). Get it scored using autosas – an automated system for scoring short answers. <https://arxiv.org/abs/2012.11243>
- Lian, R., Yu, P., Zhou, X., Liu, Q., & Zhang, Y. (2024). New era of artificial intelligence in education: Towards a sustainable multifaceted revolution. *Education Sciences*, 14 (2), 167. <https://doi.org/10.3390/su151612451>
- Library, O. S. C. V. (2024). Opencv documentation [Official documentation for the OpenCV library]. <https://docs.opencv.org/4.x/>
- Lindeberg, T. (1993). Scale-space theory in computer vision. *Springer International Series in Engineering and Computer Science*. <https://doi.org/10.1007/978-1-4757-6465-9>
- Microsoft. (2024). Azure ai document intelligence documentation [Official documentation for Azure AI Document Intelligence.]. <https://learn.microsoft.com/en-us/azure/ai-services/document-intelligence/>
- OCR, T. (2024). Tesseract ocr documentation [Official documentation for the Tesseract OCR engine.]. <https://tesseract-ocr.github.io/tessdoc/>
- OpenAI. (2024). Openai api documentation [Official documentation for OpenAI API.]. <https://platform.openai.com/docs/api-reference>
- Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). Bleu: A method for automatic evaluation of machine translation. <https://doi.org/10.3115/1073083.1073135>
- Peeperkorn, M., Kouwenhoven, T., Brown, D., & Jordanous, A. (2024). Is temperature the creativity parameter of large language models? *arXiv preprint arXiv:2405.00492*.
- Qian, H., Liu, Z., Zhang, P., Mao, K., Zhou, Y., Chen, X., & Dou, Z. (2024). Are long-lmms a necessity for long-context tasks? *arXiv preprint arXiv:2405.15318*.
- Schneider, J., Schenk, B., & Niklaus, C. (2024). Towards llm-based autograding for short textual answers. <https://arxiv.org/abs/2309.11508>
- Talebi, G., Ghaffari, R., Eskandarzadeh, E., & Oskouei, A. (2013). Item analysis an effective tool for assessing exam quality, designing appropriate exam and determining weakness in teaching. *Research and Development in Medical Education*, 2. <https://doi.org/10.5681/rdme.2013.016>
- Topological structural analysis of digitized binary images by border following [This is the original paper that describes the algorithm behind the contour finding implemented in OpenCV. It explains the method used for ‘cv2.findContours‘ with ‘cv2.RETR\_LIST‘ and ‘cv2.CHAIN\_APPROX\_SIMPLE‘ which retrieves all of the contours, with each contour stored with only the end points, reducing memory usage.]. (1985). *Computer Vision, Graphics, and Image Processing*, 30(1), 32–46. [https://doi.org/https://doi.org/10.1016/0734-189X\\(85\\)\90016-7](https://doi.org/https://doi.org/10.1016/0734-189X\(85\)\90016-7)
- VueJs. (2024). Vuejs documentation [Official documentation for VueJs.]. <https://vuejs.org/guide>
- VuetifyJs. (2024). Vuetifyjs documentation [Official documentation for VuetifyJs.]. <https://vuetifyjs.com/>

Yeung, W., Qi, C., Xiao, J., & Wong, F. (2023). Evaluating the effectiveness of ai-based essay grading tools in the summative assessment of higher education. *ICERI2023 Proceedings*, 8069–8073. <https://doi.org/10.21125/iceri.2023.2063>

# 9 Appendix

Methode - Inscannen - Score Cruijff

Quote	Quote Changed	Person	Year	Test	Score	Change Count
Je moet schieten, anders kun je niet scoren.	Je moet schieten, anders kun je niet scoren.	Johan Cruijff	1980	enkele letterverandering	0.1937	1
Je moet schieten, anders kun je niet scoren.	Je moet schoten, anders kun je niet scoren.	Johan Cruijff	1980	enkele woordverandering, geen betekenisverandering	0.2146	1
Je moet schieten, anders kun je niet scoren.	Je moet schoten, anders kun je niet scoren.	Johan Cruijff	1980	woordweglating	0.2146	1
Je moet schieten, anders kun je niet scoren.	Je moet roeien, anders kun je niet scoren.	Johan Cruijff	1980	enkele woordverandering, wel betekenisverandering	0.3282	2
Je moet schieten, anders kun je niet scoren.	Je moet schoten,.	Johan Cruijff	1980	zinsdeelweglating	1.1590	2
Je moet schieten, anders kun je niet scoren.		Johan Cruijff	1980	tekstweglating	1.5	1

## Diverse test quotes

Quote	Quote Changed	Person	Year	Test	Score	Change Count
Ik heb een heel zwaar leven.	Ik heb een heel zwaar leven.	Brigitte Kaandorp	2009	nulmeting	0	0
Een dag niet gelachen is een dag niet geleefd.	Een dag niet gelachen is een dag niet geleeft.	Charlie Chaplin	1930	enkele letterverandering , geen betekenisverandering	0.1522	1
Een dag niet gelachen is een dag niet geleefd.	Een niet gelachen is een dag niet geleefd.	Charlie Chaplin	1930	woordweglating	0.1673	1
Rrrrr, hah, is gewoon Boef man. Ha, jij bent vies maar ik doe gemener. In de club, kom je moeder tegen. En ik wil snel weg want we moeten wegen. En je klant is geholpen, je moet vroeger wezen. Ik was alles kwijt, maar floes herenigd. Voor me zondes af en toe gebeden. Ik ga uit eten voor een goede prijs. Ik ben een uitgever, ze boeken mij. Van alarm voorzien aan de achterkant. Dus ze komen via voor, maar wat dacht je dan?	Rrrrr, hah, is gewoon Boef man.test, jij bent vies maar ik doe gemener. In de club, komtest moeder tegen. En ik wil snel weg wantest we moeten wegen. En je klant is geholpen, je moetest vroeger wezen. Ik was alles kwijt, maar floetest herenigd. Voor me zondes af en toe gebeden. Ik gtest uit eten voor een goede prijs. Ik ben een uitgever, ze boeken mij. Van alarm voortest aan de achterkant. Dus ze komen via voor, maar wat dacht je dan?	Boef	2017	random toevoeging woorden	0.1928	7

Quote	Quote Changed	Person	Year	Test	Score	Change Count
Ik begrijp niet waarom u hier zo negatief en vervelend over doet. (...) Laten we blij zijn met elkaar! Laten wij optimistisch zijn! Laten we zeggen: Nederland kan het weer! Die VOC-mentaliteit, over grenzen heen kijken, dynamiek! Toch?	Ik begrijp niet waarom u hier zo negatief en vervelend over doet. (...) Laten we blij zijn met elkaar! Laten wij optimistisch zijn! Laten we zeggen: Nederland kan het weer! Die	Jan-Peter Balkenende	2006	weglating einde van grotere tekst	0.3767	1
Ik heb nooit last van hoogtevrees, wel van dieptevrees.	Ik hebt ooit last van hoogtevrees, well vann dieptevrees.	Youp van 't Hek	1998	enkele letterweglating, betekenisverandering	0.4277	4
Praat Nederlands met me. Even Nederlands met me. Mijn gevoel zegt mij dat wij vanavond samen kijken. Naar de Champs-Élysées en naar de Notre Dame en naar de Seine. En daarna samen op La Tour Eiffel	Praat Nedertands met me. Even Neterlands met me. Mijn tevoet zegt mij dat wij vanatond samet kitken. Naar de Champs-Éltsées en naar de Notre Dameten naar det- Seine. En daarta samet op La Tour Etffel	Kenny B	2015	random letter-mutaties	0.4820	13
Als het niet kan zoals het moet, dan moet het maar zoals het kan.	Als het niet kan zoals het maar zoals het kan.	Dolf Jansen	2005	weglating in midden	0.4901	1
Ik ben niet dik, ik ben een ruimtewonder.	Ik bn nit dik, ik bn n ruimtwondr.	Brigitte Kaandorp	2003	letter e wegge- laten	0.6707	6
Ik geloof in God, behalve als ik vis.	Ik geloof in God, be	Herman Brood	1995	weglating aan einde	0.7644	1
Ik ben niet gek, ik ben een vliegtuig.	Ik ben niet , ik een .	Supergrover	1974	dubbelle wo- ordweglating	0.8947	3

## Prompt resultaten open vragen

je krijgt een aantal antwoorden op een open vraag in een enquête aan docenten

jouw taak is om zo precies en objectief mogelijk alle punten die worden gemaakt op een rijtje zetten van hoe belangrijk ze worden gevonden"

zorg dat de output in mooi opgemaakt latex is zonder overflow

de vraag was: Welke invloed denkt u dat de inzet van AI kan hebben op de relatie tussen docent en student? (Open vraag)

je mag de gemaakte punten in categoriën groepen, maar je moet altijd zo objectief mogelijk zijn, want dit komt in de resultaten van een onderzoek en daar mogen absoluut geen assumenties in

{Resultaten}

# 10 logboek

## Fase 4

Datum	Uren Joost	Uren Jonathan	Joost wat is er gedaan?	Jonathan wat is er gedaan?
09/10/24 wo	0.0	3.0		Toets nakijker toegevoegd
14/10/24 ma	1.0	3.0	analyseren secties toevoegen	Aan de toetsen gewerkt
16/10/24 wo	0.5	0.0	mini onderzoek enquete opgesteld: incl. vraag, methode, hypothese enz	
19/10/24 za	0.5	0.0	normaalverdelingen toegevoegd aan de analyse site	
21/10/24 ma	0.0	2.0		Forced JSON
27/10/24 zo	0.0	3.0		Temperaturen getest
31/10/24 do	0.0	2.0		Prompts getest
02/11/24 za	1.0	0.0	toetsblaadje pdf afgemaakt	
07/11/24 do	2.0	0.0	google cloud function voor de inscanner	
08/11/24 vr	3.0	0.0	afmaken functions docker containerizing	
09/11/24 za	2.0	0.0	website en opnieuw proberen te containen	
11/11/24 ma	1.0	0.0	qr code creator in iupyter notebook gemaakt	
12/11/24 di	1.0	0.0	omgezet naar de api en toegevoegd aan de vuejs site	
13/11/24 wo	0.0	2.0		Implementatie Gemini
17/11/24 zo	0.0	2.0		Gemini met afbeeldingen
22/11/24 vr	1.0	0.0	api route toegevoegd die alles kan	
23/11/24 za	0.0	2.0		Begonnen met implementatie klassengemiddelen
25/11/24 ma	1.0	0.0	api route die alles kan geïntegreert in ui. bugs in square detector gefixed	
30/11/24 za	1.0	0.0	enquete resultaten op tabellen zetten	
01/12/24 zo	1.0	0.0	inscan onderzoek resultaten in tabellen zetten	
04/12/24 wo	2.5	1.5	methode inscannen	Toets gemaakt
06/12/24 vr	1.0	0.0	verder met methode inscannen	
07/12/24 za	3.0	0.0	methode inscannen afgerond en gestart methode analyseren	
09/12/24 ma	0.0	1.5		Rubric aangepast
10/12/24 di	2.0	0.0	resultaten tabellen inscannen	
11/12/24 wo	2.0	0.0	latex package bug fixen en verder met tabellen	
12/12/24 do	1.0	0.0	in mediatheek gemini 2.0 flash testen	
13/12/24 vr	3.0	2.0	Praktijktest gegeven 3e klas eerste nakijkttest	Praktijktest gegeven 3e klas
14/12/24 za	1.0	1.5	conclusie geschreven inscannen en analyseren	Handmatig toets nagekeken
15/12/24 zo	1.5	0.0	tests met inscansite	

Datum	Uren Joost	Uren Jonathan	Joost wat is er gedaan?	Jonathan wat is er gedaan?
16/12/24 ma	1.0	0.0	foutanalyse mijn onderdelen + vervolgonderzoek + opzet samenvatting onderzoek	
17/12/24 di	0.0	5.0		Toets met 6 modellen nagekeken
18/12/24 wo	2.0	3.5	nakijk resultaten geformateerd + uitleg instellen latex vscode	nakijk methode geschreven
19/12/24 do	2.0	4.5	afmaken document + conclusie schrijven	
Totalen	38.0	38.5		

### Fase 3

Datum	Uren Joost	Uren Jonathan	Joost wat is er gedaan?	Jonathan wat is er gedaan?
16/06/24 zo	0.0	2.0		Proef LLM model gemaakt m.b.v. PyReft
21/06/24 vr	0.0	2.0		Toetsdata geanonimiseerd
09/09/24 ma	1.0	0.0	Gestart met brief aan Studente PDF -> image, rode pen weggehaald (dmv python), gestart met het opsplitsen van antwoorden in secties	
14/09/24 za	2.0	0.0		
19/09/24 do	1.0	0.0	Joost gesprek met begeleider	
21/09/24 za	4.0	0.0	gestart gebruikt handbook en eerste secties onderscheiden. logboek	
28/09/24 za	1.5	3.0	Joost correctie toegevoegd voor foute opdrachtnummer herkenning ('b' -> 6)	Begin met website voor toetsen analyseren. Start gemaakt met de routing en templates voor de website.
01/10/24 di	0.0	2.0		Verder gegaan met programma voor de openAI API-requests.
03/10/24 do	1.5	0.0	Fase 3 bestand opmaken, correlatie en covariatie toegevoegd aan de classes	
05/10/24 za	2.0	3.0	vuejs test site gemaakt, contact met daniel markus	Legacy output geherstructureerd
06/10/24 zo	2.0	0.0	fase 3 2 bronnen toegevoegd, nieuwe output indeling inscan modules. nieuwe tabel in analyseren	
Totalen	15.0	12.0		

## Fase 1 & 2

Datum	Uren Joost	Uren Jonathan	Joost wat is er gedaan?	Jonathan wat is er gedaan?
18/03/24 ma	0.5	0.5	Begin aan de motivatiebrief; de opzet en samenwerking	Begin aan de motivatiebrief; de opzet en samenwerking.
19/03/24 di	1.5	1.5	Afschrijven van de motivatiebrief	Afschrijven van de motivatiebrief.
08/05/24 wo	0.5	0.0	filmpje NOS en format fase 2	
13/05/24 ma	1.0	0.0	Layout fase 2 document en literatuuronderzoek	
14/05/24 di	0.0	2.5		Drie studies uitgekozen en samengevat.
15/05/24 wo	1.0	1.0	Voorbereiden op gesprek begeleider.	Voorbereiden op gesprek begeleider.
28/05/24 di	1.0	0.0	Eén studie samengevat.	
03/06/24 ma	1.5	0.0	De planning van fase 2 en 3 gemaakt.	
04/06/24 di	0.0	1.0		Onderzoek gedaan naar Finetuning van LLMs
06/06/24 do	0.0	1.5		Onderzoek gedaan naar REFT, PYREFT & LORA.
06/06/24 do	0.5	0.0	Samenvatting An automatic short-answer grading model for semi-open-ended questions	
Totalen	7.5	8.0		

## Totaal

Datum	Uren Joost	Uren Jonathan	Joost wat is er gedaan?	Jonathan wat is er gedaan?
Totalen	86.0	84.0		