Meetrapport

Tijdmeting

Hogeschool Utrecht 8-4-2017

Luke Roovers & Luuk Steeman

# Doel

Dit experiment moet gaan uitwijzen welke van de twee kernels sneller over een Image heen kan worden gehaald. De twee kernels, Sobel & Prewitt, zijn vergelijkbaar in opzet maar zijn de verschillen ook van invloed op de snelheid van het programma? Dit leidt tot de onderzoeksvraag:

“Welke kernel werkt sneller onder dezelfde omstandigheden”   
  
Door het beantwoorden van deze vraag wordt er hopelijk een duidelijker beeld geschetst over de werking van de beide kernels in gezichtsherkenning.

# Hypothese

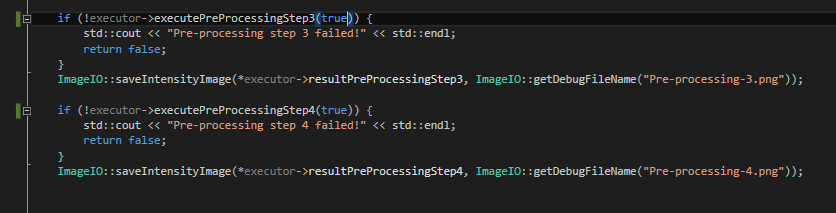
Bij het testen van beide kernels worden een aantal resultaten verwacht. Ten eerste zou de Prewitt een kleine snelheidswinst moeten opleveren, dit omdat er in theorie minder rekenkracht vereist is voor het Prewitt kernel. Hoe onze implementatie zich verhoudt tot de standaard is lastig, aangezien er gekozen is voor in principe dezelfde methode die geïmplementeerd is maar de standaard edge detection waarschijnlijk een stuk gestroomlijnder is dan deze eerste poging tot edge detection.

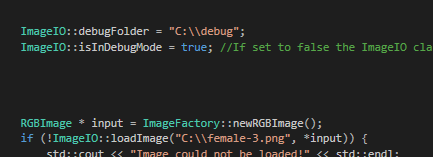
# Werkwijze

Eerste de meting zelf. Er wordt gemeten hoe lang elke complete cycle van de software duurt. Dit is dus de complete facial recognition software. De clock timer staat dan ook aan het begin van de main en wordt stopgezet aan het eind van de main. Deze meting wordt 20 keer per plaatje uitgevoerd op dezelfde pc.   
  
De input bestaat uit twee plaatjes van de test set. Hierbij is gekozen voor female-1 & female-3, hierbij is niet alleen gelet op het feit dat ondergetekende er de hele tijd tegen aan moet kijken, ook had Prewitt moeite met het herkennen van sommige van de testsets aangezien de rest van thresholding gebaseerd is op het Sobel kernel.

Alle tests zijn uitgevoerd op een systeem waar zo min mogelijk achtergrond applicaties werden uitgevoerd. De specs van het systeem kunnen worden teruggevonden op de volgende link:  
https://tweakers.net/pricewatch/345048/lenovo-ideapad-y510p-00061/specificaties/

Het experiment

  
In de main worden de beide studentenimplementaties op “true” gezet.

  
In de main wordt het pad naar het plaatje ingevoerd en een folder gekozen waar het resultaat van de facial recognition in wordt opgeslagen. Deze stap zal herhaald moeten worden met het andere test plaatje wanneer nodig.  
  
  
Vervolgens werd het project gecleaned, gecompiled en vervolgens zonder debugging als release versie uitgevoerd. De resultaten werden bijgehouden in een excel sheet.

|  |  |  |  |
| --- | --- | --- | --- |
| **Meting nummer** | **Tijd ( in ms ) Sobel Kernel** | **Tijd ( in ms ) Prewitt Kernel** | **Tijd ( in ms ) default** |
| **1** | **97** | **89** | **90** |
| **2** | **92** | **79** | **83** |
| **3** | **93** | **86** | **83** |
| **4** | **91** | **90** | **85** |
| **5** | **92** | **93** | **82** |
| **6** | **87** | **91** | **93** |
| **7** | **93** | **88** | **98** |
| **8** | **92** | **90** | **87** |
| **9** | **110** | **95** | **93** |
| **10** | **78** | **88** | **120** |
| **11** | **96** | **90** | **83** |
| **12** | **93** | **89** | **93** |
| **13** | **97** | **91** | **83** |
| **14** | **77** | **95** | **85** |
| **15** | **91** | **88** | **94** |
| **16** | **90** | **98** | **81** |
| **17** | **90** | **95** | **85** |
| **18** | **88** | **93** | **82** |
| **19** | **92** | **98** | **84** |
| **20** | **90** | **91** | **94** |
| **Gemiddeld** | **91,45** | **90,85** | **88,9** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Meting nummer** | **Tijd ( in ms ) Sobel Kernel** | **Tijd ( in ms ) Prewitt Kernel** | **Tijd ( in ms ) default** |
| **1** | **80** | **77** | **86** |
| **2** | **116** | **91** | **78** |
| **3** | **108** | **88** | **77** |
| **4** | **86** | **93** | **83** |
| **5** | **88** | **87** | **82** |
| **6** | **87** | **87** | **71** |
| **7** | **93** | **85** | **80** |
| **8** | **87** | **84** | **87** |
| **9** | **92** | **95** | **82** |
| **10** | **93** | **90** | **83** |
| **11** | **81** | **85** | **78** |
| **12** | **90** | **81** | **80** |
| **13** | **70** | **87** | **80** |
| **14** | **88** | **83** | **80** |
| **15** | **87** | **70** | **76** |
| **16** | **91** | **81** | **75** |
| **17** | **84** | **93** | **76** |
| **18** | **86** | **84** | **86** |
| **19** | **83** | **88** | **80** |
| **20** | **93** | **85** | **80** |
| **Gemiddeld** | **89,15** | **85,7** | **80** |

# Resultaten

De bovenste tabel is het female-1 image met een ingestelde threshold van 80 hoog en 40 laag. De onderste is de meting van female-3 met een ingestelde threshold van 90 hoog en 50 laag. Opvallende zaken zijn het verschil in tijd tussen Sobel en Prewitt. Het grote verschil in tijd tussen de female-1 en female-3 meting en het feit dat default eigenlijk altijd sneller is.

1 Waardes van de drie verschillende metingen

In de grafiek hierboven is te zien dat er soms ineens wat uitschieters naar boven zijn. Dit kan te maken hebben met hardware technische redenen zoals het OS wat bepaalde taken uitvoert op de achtergrond of andere vertragende factoren. We nemen dit wel mee in de metingen omdat deze vertragingen niet exclusief zijn tot onze testomgeving.

# Verwerking

Met de resultaten gemeten kan er terug gekoppeld worden naar de hypothese. We kunnen zien dat er inderdaad een snelheidsverschil bestaat tussen de Prewitt en Sobel kernels. De Prewitt kernel is in beide situaties sneller gebleken. Is dit altijd het geval? Waarschijnlijk niet. Zoals eerder aangegeven werkte het Prewitt filter soms helemaal niet tijdens het testen, omdat de thresholding niet goed aansloot op dit filter. Sobel lijkt in dit geval een veel consistenter resultaat te vertonen.   
  
Dat gedachte dat de default implementatie sneller is , is correct gebleken. De code die geschreven is voor deze meting loopt nog niet zo lekker als de default. Deze uitkomt is verder niet heel belangrijk voor het verschil tussen Prewitt en Sobel, maar wel als er gekozen moet worden tussen de default en de nieuwe code.

# Conclusie

Concluderend kunnen we stellen dat de hoofdvraag “Welke van de twee kernels werkt sneller” beantwoord is door middel van de meting. Prewitt heeft hier een kleine winst behaald. Helaas is het dus niet met zekerheid te zeggen of dit altijd het geval zal zijn en of er niet op andere vlakken ingeleverd word.

Evaluatie   
  
Het proces van implementatie plan naar deze meting liep redelijk soepel. De code zelf was een kleine worsteling op sommige aspecten, maar het resultaat van de code is van een acceptabel niveau om mee te werken. Bij de volgende meting zou het fijn zijn om deze code om te zetten in klassen en verbeteringen aan t brengen zodat er daadwerkelijk snelheidswinst word behaald ten opzichte van de default implementatie.   
  
Tijdens het meten liepen we tegen een aantal problemen aan. De Prewitt kernel herkende niet alle test images, wat zonde is. Ook zou de manier van testen geautomatiseerd kunnen worden, om op die manier veel meer testen uit te voeren en een accurater gemiddelde te krijgen.