# Project 1 Applied Programming – Histogram Stretching

## De opdracht

Het maken van een grafische toepassing in C# die histogram stretchen toepast op een afbeelding.  
De afbeelding wordt door een gebruiker ingeladen waarvan vervolgens een histogram gemaakt word. Aan de hand van het gegenereerde histogram wordt deze afbeelding gestretcht en wordt de bekomen afbeelding met het bijbehorende histogram ook getoond.

## Het onderzoek

Het onderzoek begint bij het opzoeken van de formule voor histogram stretching. Deze gebruikt de waarde van een individueel component uit kleurmodellen. Bij het RGB-kleurmodel is dit R (red), G (green) of B (blue). In het CMYK-kleurmodel is dit C (cyan), M (magenta), Y (yellow) of K (key of zwart).

Buiten RGB moet er nog een tweede kleurenmodel ondersteund zijn in de toepassing. Er is gekozen voor CMYK. Dit is een kleurenmodel dat vaak gebruikt wordt bij het uitprinten van documenten waar RGB niet toepasbaar is. Om CMYK te bekomen moet RGB met een formule omgezet worden.

In de toepassing wordt de afbeelding aan de hand van een bitmap in de grafische toepassing weergegeven. Een bitmap werkt enkel met het RGB-kleurenmodel. De uitgewerkte CMYK-waarden moeten aan de hand van een formule terug naar RGB omgezet worden. Zo kan de uitgewerkte afbeelding aan de gebruiker getoond worden. (CMYK to RGB conversion | color conversion, n.d.) (RGB to CMYK conversion | color conversion, n.d.)

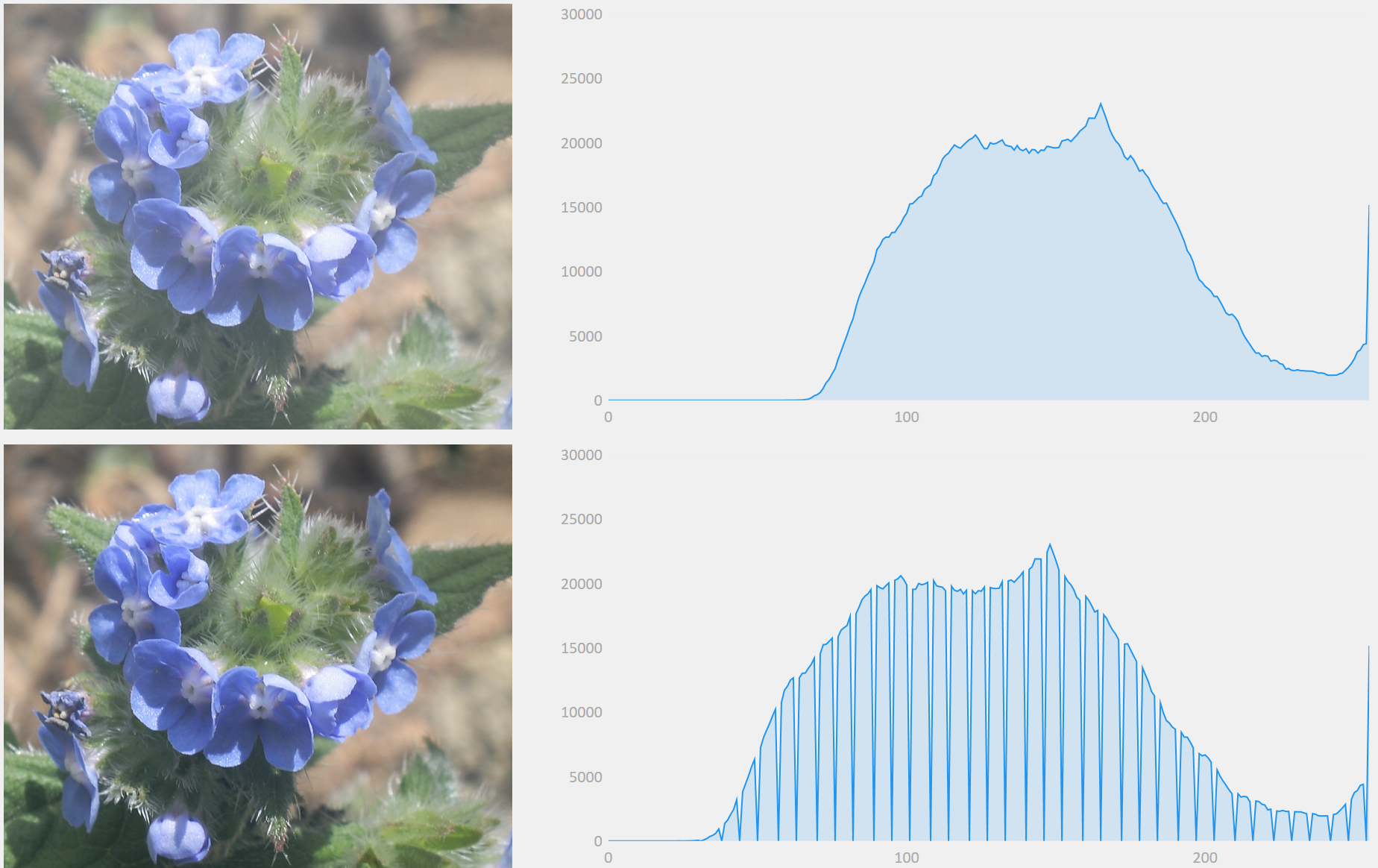
## De praktische uitwerking

Deze formule wordt in de toepassing op twee manieren uitgewerkt. Voor elk component van elke pixel wordt de formule uitgevoerd.

### Methode 1

Bij de eerste manier worden alle waarden van elk component van elke pixel bij elkaar opgeteld om de histogram te maken. De formule gebruikt dan de waarde van de component samen met de maximum en minimum mogelijk waardes voor de component, en de maximum en minimum waarden van de histogram.

RGB:

CMYK:

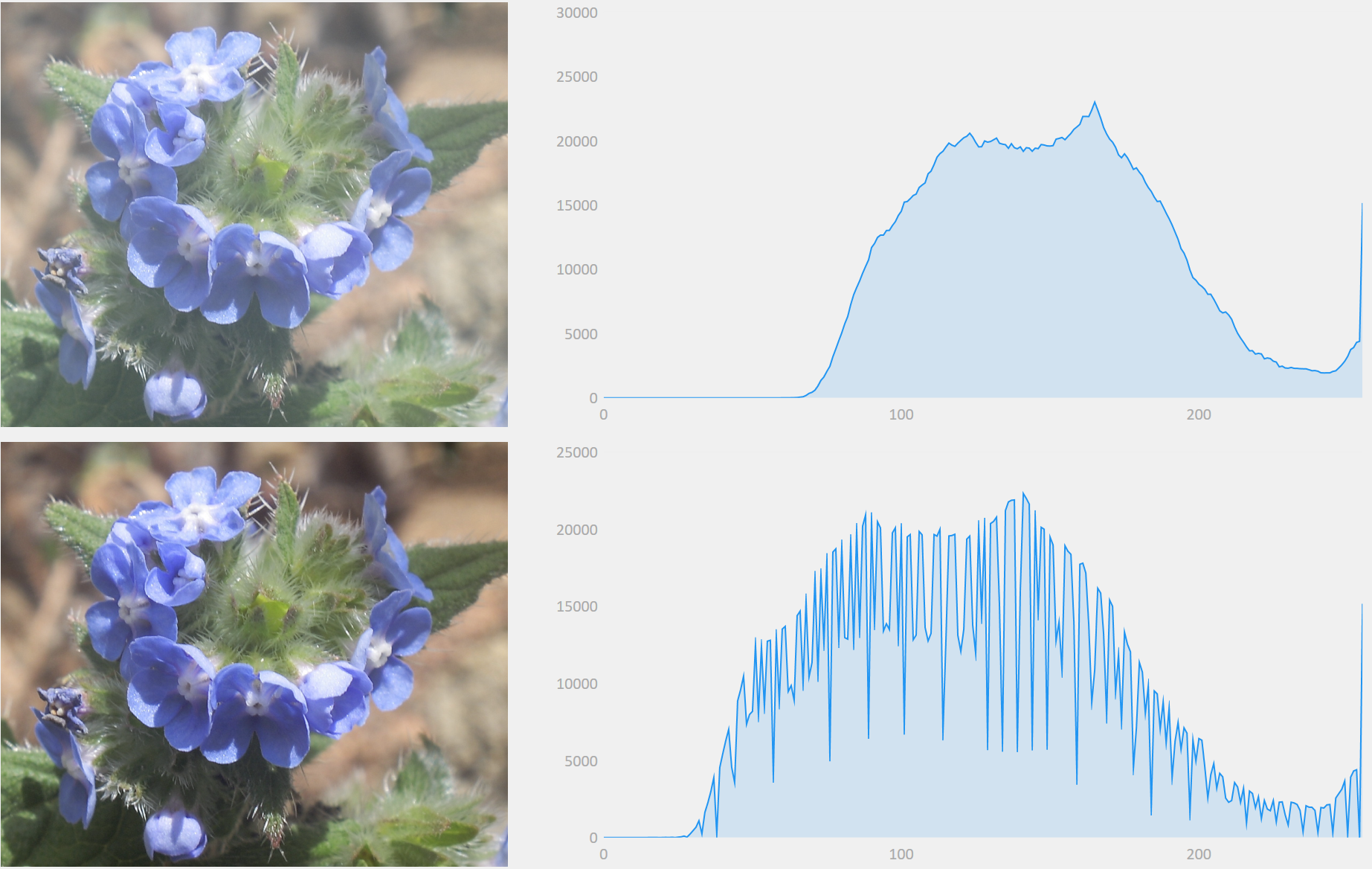
Bij methode 1 is het CMYK model voor kleurafbeeldingen net beter. Het zorgt voor een hoger contrast dan RGB maar dit is amper zichtbaar.

Voor afbeeldingen in grijswaarden is RGB op zijn beurt beter. Bij CMYK word de afbeelding enkel verdonkerd, terwijl er bij RGB een duidelijke verbetering van kwaliteit is.

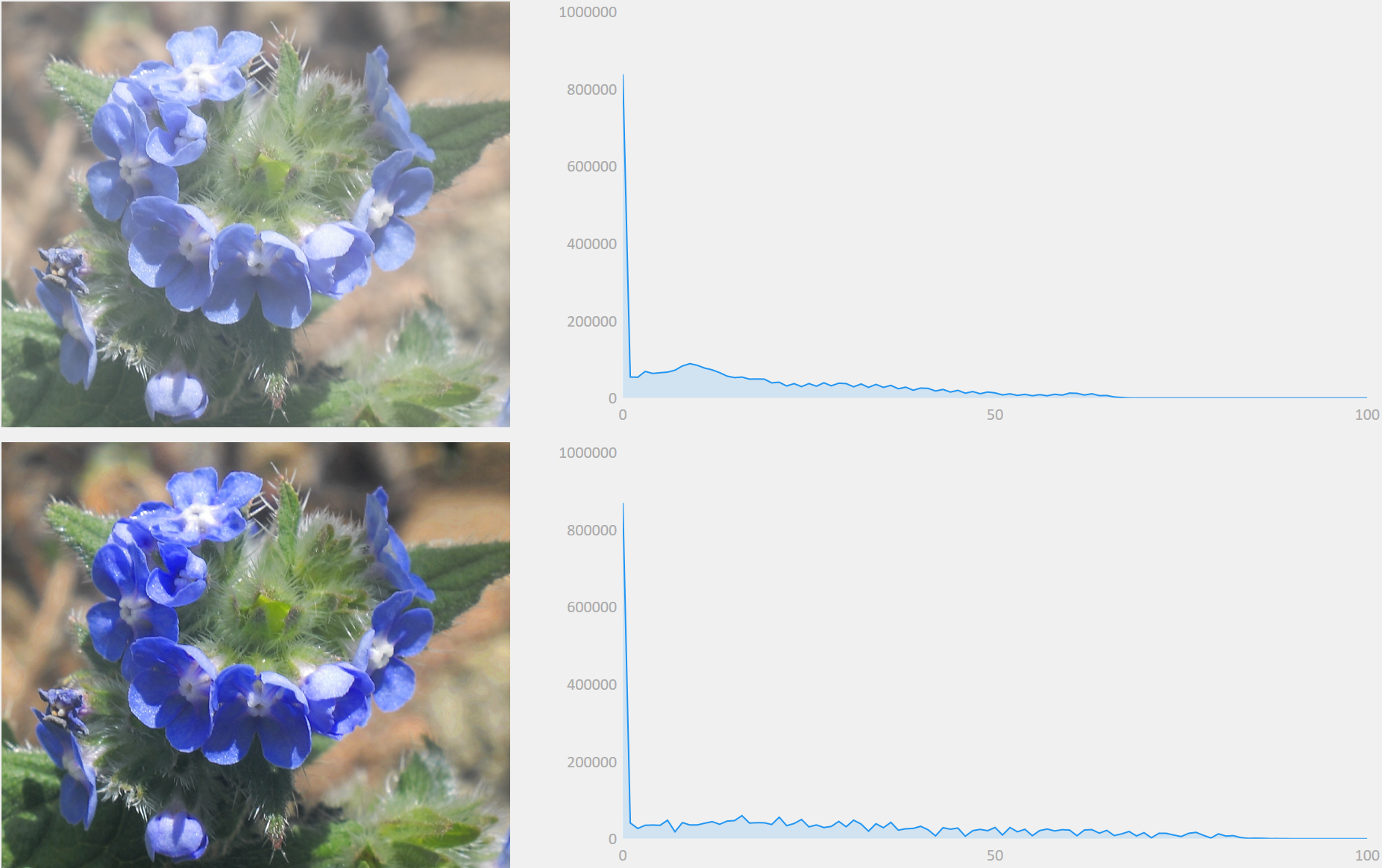
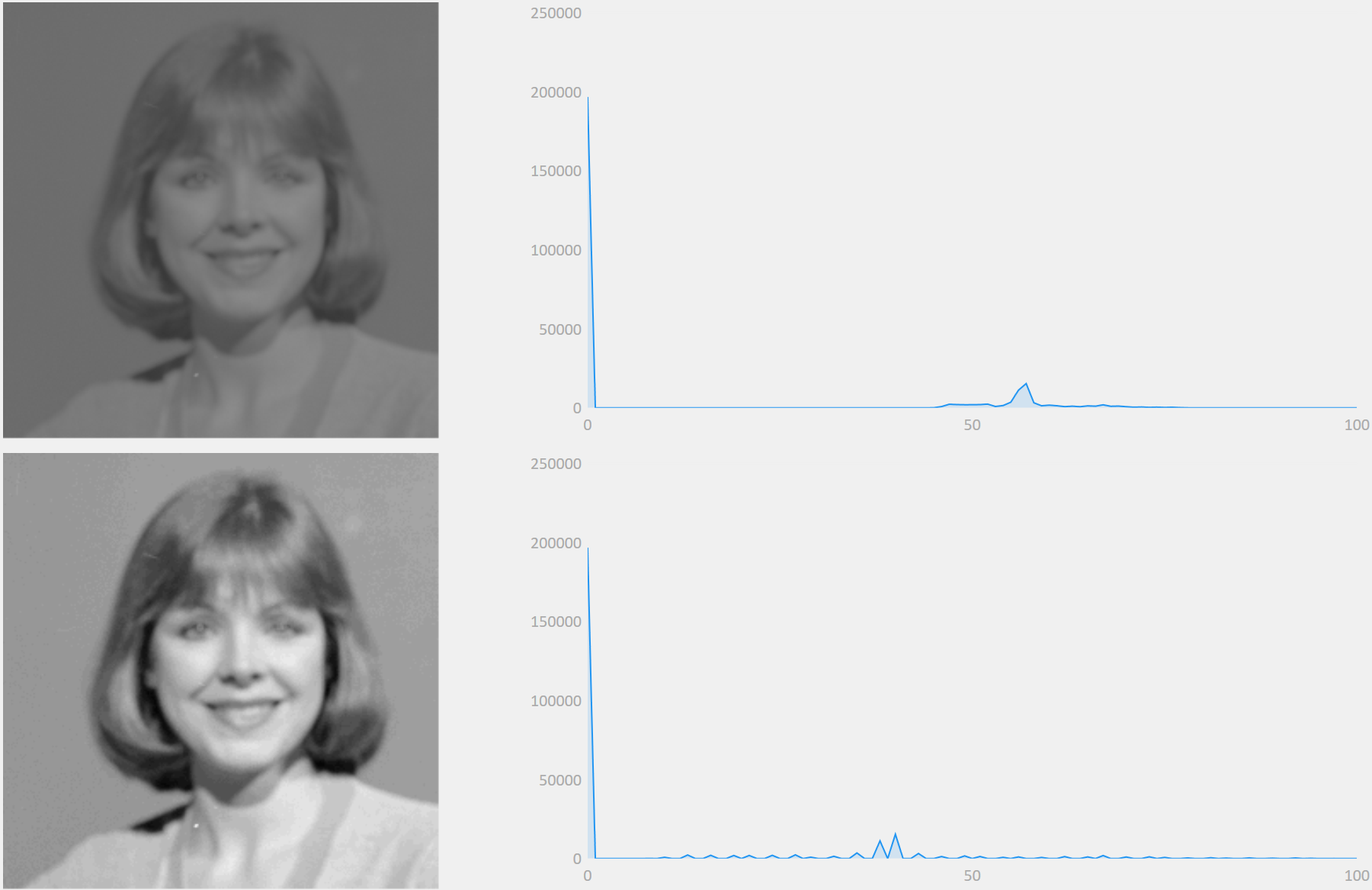
### Methode 2

Bij de tweede manier wordt van elk individueel component een eigen histogram gemaakt. De formule gebruikt dan de waarde van de component,de maximum en minimum mogelijke waarden en de maximum en minimum waarden van de histogram van die component.

RGB:

CMYK:

In vergelijking met methode 1, is methode 2 vooral voor CMYK beter. Bij RGB heeft de kleurenafbeelding een hoger contrast maar de grijswaarden afbeelding is hetzelfde als bij methode 1. Dit komt doordat bij RGB een afbeelding in grijswaarden in elke pixel de R-, G- en B-waarden gelijk zijn. Het individueel of tezamen stretchen van de componenten geeft als gevolg hetzelfde effect.

Bij CMYK is de grijswaarden afbeelding verbeterd in vergelijking met methode 1. Dit komt doordat het stretchen va de K-waarde enkel effect heeft in deze afbeelding. Door dit apart van de andere componenten te kunnen doen, is de kwaliteit van de afbeelding verbeterd.

De kleurenafbeelding bij CMYK heeft een zeer hoog contrast, dit heeft bij sommige afbeeldingen een positief effect:

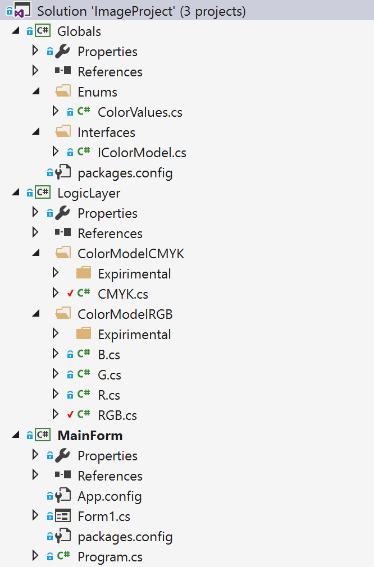
  

Maar bij andere afbeeldingen kan het ook een negatief effect hebben:



Dit komt doordat de verschillende kleurcomponenten zoals bij methode 1 geen rekening met elkaar houden. Elke component wordt op zijn eigen histogram gestretcht en zal dus elk op een andere manier versterkt worden. (Fischer, Perkins, Walker, & Wolfart, 2003)

### Code structuur

Het programma bestaat uit een grafische laag en een logische laag. In de logische laag zitten twee klassen CMYK en RGB, die zich houden aan de IColorModel interface. De IColorModel heeft 4 publieke properties: de ingelezen afbeelding, de gestretchte afbeelding, de grafiekwaarden van de ingelezen afbeelding en de grafiekwaarden van de gestretchte afbeelding.

Elke keer als er een afbeelding wordt ingelezen, word er een IColorModel object aangemaakt. Deze is RGB of CMYK beslissend door welk kleurmodel in de grafische toepassing geselecteerd is. In de constructor methode worden direct alle berekeningen gedaan.

Dan wordt in de grafische toepassing de afbeeldingen en de grafieken getoond aan de hand van de properties van het IColorModel object.

## Besluit

De toepassing doet het stretchen volgens de twee methodes en op twee kleurenmodellen. Dit zorgt voor meer inzicht in het effect van elke methode en wat het effect van stretchen van andere kleurenmodellen is. Er kan besloten worden dat elke methode voor kleuren afbeeldingen en afbeeldingen in grijswaarden andere effecten heeft. De CMYK bekomt met methode 2 afbeeldingen met hoger contrast wat voor mooie effecten kan zorgen, maar ook de afbeelding kan verslechten. RGB is voor afbeeldingen in grijswaarden beter en heeft met methode 2 het meest betrouwbare effect op kleurenafbeeldingen.

In de toepassing is de grafiek altijd een som van alle componenten. Dit zou beter weergegeven kunnen zijn als een “stacked area” grafiek, met de grafieken van individuele componenten op elkaar gestapeld. Dit zou een duidelijkere weergave geven van de waardes van elk componenten.

# Bibliography

*CMYK to RGB conversion | color conversion*. (n.d.). Retrieved from RapidTables: http://www.rapidtables.com/convert/color/cmyk-to-rgb.htm

Fischer, R., Perkins, S., Walker, A., & Wolfart, E. (2003). *Point Operations - Contrast Stretching*. Retrieved from HIPR2: https://homepages.inf.ed.ac.uk/rbf/HIPR2/stretch.htm

*RGB to CMYK conversion | color conversion*. (n.d.). Retrieved from RapidTables: http://www.rapidtables.com/convert/color/rgb-to-cmyk.htm