



SCENARIOSPECIFIEKE CONTRASTVERBETERING VOOR SATELLIETCOREGISTRATIE

EEN EVALUATIE VAN MATCHANYTHING



1 JUNI 2026
MERLIJN DUMAREY



SCENARIOSPECIFIEKE CONTRASTVERBETERING VOOR SATELLIETCOREGISTRATIE

EEN EVALUATIE VAN MATCHANYTHING



1 JUNI 2026
MERLIJN DUMAREY

Voorwoord

Eerst en vooral wens ik dr. Devlieger Ines, mijn lector en begeleider voor het vak “Bachelorproef Toegepaste Informatica – AI” oprecht te bedanken.

Daarnaast wens ik ook mijn familie en naaste omgeving te bedanken voor hun uitvoerige steun, inspiratie en geduld.

Merlijn Dumarey, Gistel, 1 juni 2026

Abstract

Mogelijk kan het toevoegen van een contrastverhogende preprocessingstap een gunstige invloed hebben op de prestaties van MatchAnything. Een algemeen bekende vorm van contrastverhoging is CLAHE. CLAHE is een wiskundige techniek die in afbeeldingen het contrast kan verhogen zonder ruis te vergroten. CLAHE steunt op twee parameters: cliplimiet en tegelrastergrootte. Beide hebben een invloed op het uiteindelijke resultaat van de contrastverhoging.

De hypothese dat per categorie een optimale CLAHE-instelling gevonden kan worden die, in vergelijking met matching-inferenties zonder CLAHE, resulteert in meer matches met een lage fout en minder gefaalde inferenties, kan dankzij dit onderzoek deels bevestigd worden.

Dit onderzoek levert voor negen van de dertig afbeeldingscategorieën een CLAHE-instelling die een significante prestatieverbetering tot resultaat heeft ten opzichte van een inferentie zonder CLAHE. Van deze negen categorieën tonen er zeven een significante verbetering in zowel de ratio van het aantal matches met een fout lager dan 50 pixels, als in het slagingspercentage van de inferentie.

Inhoudsopgave

Voorwoord	1
Abstract	3
Figurenlijst.....	7
Lijst van Codefragmenten.....	14
Tabellenlijst	15
Afkortingen	16
Glossarium	17
Introductie	21
Context	21
Probleemstelling.....	22
Doelstelling.....	22
1 Theoretische Achtergrond	23
1.1 Coregistratie en Satellietdata	23
1.2 Feature Matching & Modellen	26
1.2.1 Features	26
1.2.2 Feature Description	26
1.2.3 Feature Detection.....	26
1.2.4 Feature Matching.....	26
1.2.5 LoFTR.....	26
1.2.6 ELoFTR.....	27
1.2.7 MatchAnything	28
1.3 Beeldverbetering: CLAHE.....	28
1.4 RANSAC als Outlier Filter	30
2 Methodologie	30
2.1 Hypothese.....	30
2.2 Dataset en Proxybenadering.....	31
2.3 Data Pipeline en Synthetische Transformaties	32
2.4 Evaluatie: Scoring zonder RANSAC	34
2.5 Modelconfiguratie en Hardware	35
2.6 Experimenteel Ontwerp	36

Scenariospecifieke Contrastverbetering voor Satellietcoregistratie

2.6.1	Random Search Verkenning	36
2.6.2	Analyse van de Random Search & Parameterselectie	39
2.6.3	Gerichte Grid Search en Validatie	42
3	Resultaten en Analyse	43
3.1	Fase 1: Random Search Verkenning.....	43
3.2	Fase 2: Data-Analyse en Parameterselectie	44
3.2.1	Algemene Correlaties en Moeilijkheidsgraad per Categorie.....	44
3.2.2	Kwantitatieve Analyse: Slagingspercentage en Matchvolume	45
3.2.3	Kwalitatieve Analyse: Nauwkeurigheid en Confidence Scores	46
3.2.4	Parameterselectie voor de Grid Search.....	47
3.3	Fase 3: Gerichte Grid Search en Validatie	47
3.4	Analyse Grid Search	48
3.5	Inliers Ratio, Mean Abs Inliers & Failure Rate.....	49
3.6	Statistische Validatie	50
	Conclusie	53
	Aanvaarding van de Hypothese	53
	Mogelijke Verklaring.....	54
	Bronnenlijst	58
	Appendix 1: Tabellen	61
	Appendix 2: Afbeeldingen	66

Figurenlijst

Figuur 1: Luchtfoto van de Burj Khalifa [5].	22
Figuur 2: Luchtfoto van de Burj Khalifa [6].	22
Figuur 3: Luchtfoto uit 2022 met drie duidelijk herkenbare punten A, B en C. De originele foto in deze figuur is afkomstig van Google Earth [7].	24
Figuur 4: Luchtfoto uit 2025 met drie duidelijk herkenbare punten A, B en C. De originele foto in deze figuur is afkomstig van Google Earth [8].	24
Figuur 5: Luchtfoto's na coregistratie op basis van punten A, B en C. Een zichtbare verandering is omkaderd. Beide foto's komen origineel van Google Earth [7], [8].	24
Figuur 6: In deze afbeelding wordt de trapeziumvormige perspectiefverandering bij opname van satellietfoto's geïllustreerd [10].	25
Figuur 7: Overzicht van de LoFTR-methode, zoals te zien in de LoFTR-paper [13].	27
Figuur 8: Overzicht van de eLoFTR-methode, zoals te zien in de eLoFTR-paper [14].	28
Figuur 9: Vergelijkende illustratie van het toepassen van CLAHE op een luchtfoto met een cliplimiet gelijk aan 2.0 tegenover met een cliplimiet gelijk aan 20.0. De foto die gebruikt wordt van dit plot komt origineel van Google Earth [8].	29
Figuur 10: Vergelijkende illustratie van het toepassen van CLAHE op een luchtfoto met een tegelrastergrootte gelijk aan 2x2 tegenover met een tegelrastergrootte gelijk aan 32x32. De foto die gebruikt wordt in dit plot komt origineel van Google Earth [8].	29
Figuur 11: Voorbeeld van een afbeelding uit de AID-dataset met categorie Bridge.	32
Figuur 12: Voorbeeld van een afbeelding uit de AID-dataset met categorie Stadium.	32
Figuur 13: Voorbeeld van een afbeelding uit de AID-dataset met categorie Mountain. .	32
Figuur 14: Voorbeeld van een afbeelding uit de AID-dataset met categorie Desert.	32
Figuur 15: Illustratie van ongewenste zwarte randen bij uitzoomeffect na inwaartse transformatie.	33
Figuur 16: Schematische voorstelling van de data pipeline. De originele afbeelding (1) wordt volgens een trapeziumvormige transformatie vergroot. Deze afbeelding (3) vormt samen met een gaussiaanse ruis (4) een nieuwe afbeelding (5).	34
Figuur 17: Schematische voorstelling van hoe na het voltooiën van de data-pipeline (5) het afbeeldingenpaar wordt aangeleverd aan het AI-model (6). Na toepassing van CLAHE (7) wordt ook dit paar aangeleverd aan het AI-model (8).	38
Figuur 18: Voorbeeld van een afbeelding uit de AID-dataset met categorie Playground.	44
Figuur 19: Voorbeeld van een afbeelding uit de AID-dataset met categorie Forest.	44
Figuur 20: Voorbeeld van een afbeelding uit de AID-dataset met categorie Church.	45
Figuur 21: Voorbeeld van een afbeelding uit de AID-dataset met categorie Desert.	45
Figuur 22: Voorbeeld van een afbeelding uit de AID-dataset met categorie Pond.	46
Figuur 23: Voorbeeld van een afbeelding uit de AID-dataset met categorie Desert.	46
Figuur 24: Voorbeeld van een afbeelding uit de AID-dataset met categorie DenseResidential.	46

Figuur 25: Voorbeeld van een afbeelding uit de AID-dataset met categorie Airport.	57
Figuur 26: Voorbeeld van een afbeelding uit de AID-dataset met categorie Church.	57
Figuur 27: Voorbeeld van een afbeelding uit de AID-dataset met categorie DenseResidential.	57
Figuur 28: Voorbeeld van een afbeelding uit de AID-dataset met categorie Forest.	57
Figuur 29: Voorbeeld van een afbeelding uit de AID-dataset met categorie Mountain. .	57
Figuur 30: Deze figuur illustreert dat tot een gemiddelde fout van 50 pixels de overlay bruikbaar is.	66
Figuur 31: Dit is de densiteitsplot van de geometrische fout in functie van de confidence score voor alle matches uit de random search. De Spearman-correlatiecoëfficiënt (-0.249) voor alle matches is zichtbaar in de titel. Er is een omgekeerd evenredig verband tussen de geometrische fout en de confidence score van de matches. Het zwaartepunt van de densiteitsplot ligt bij hoge confidence scores en lage geometrische fouten.	67
Figuur 32: De Spearman-correlatiecoëfficiënt van de categorie Forest (-0.319) is zichtbaar in de titel. Het zwaartepunt van de densiteitsplot ligt bij lage confidence scores. Dit wijst erop dat Forest algemeen genomen een moeilijke categorie is voor MatchAnything om te matchen.	68
Figuur 33: De densiteitsplot voor categorie Playground.	68
Figuur 34: De densiteitsplot voor categorie Church.	68
Figuur 35: De densiteitsplot voor categorie Desert.	68
Figuur 36: De densiteitsplot voor het verband tussen het eerste kwartiel van de fouten en het aantal matches in een matching-inferentie.	68
Figuur 37: De densiteitsplot voor het verband tussen de mediaan van de fouten en het aantal matches in een matching-inferentie.	68
Figuur 38: De histogrammen en cumulatieve plots van de random search voor het aantal matches per matching-inferentie, gegeven de categorie Pond. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is.	69
Figuur 39: De histogrammen en cumulatieve plots van de random search voor het aantal matches per matching-inferentie, gegeven de categorie Desert. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is.	70
Figuur 40: De histogrammen en cumulatieve plots van de random search voor de geometrische fout per match, gegeven de categorie Pond. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de	

Figurenlijst

histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.	71
Figuur 41: De histogrammen en cumulatieve plots van de random search voor de geometrische fout per match, gegeven de categorie Desert. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.	72
Figuur 42: De histogrammen en cumulatieve plots van de random search voor de confidence score per match, gegeven de categorie Desert. Op de x-as staat de grootte van de confidence score van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.	73
Figuur 43: De histogrammen en cumulatieve plots van de random search voor de confidence score per match, gegeven de categorie Pond. Op de x-as staat de grootte van de confidence score van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.	74
Figuur 44: De histogrammen en cumulatieve plots van de random search voor de confidence score per match, gegeven de categorie DenseResidential. Op de x-as staat de grootte van de confidence score van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.	75
Figuur 45: De histogrammen en cumulatieve plots van de random search voor het aantal matches per matching-inferentie, gegeven de categorie Airport. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is.	76
Figuur 46: De histogrammen en cumulatieve plots van de grid search voor het aantal matches per matching-inferentie, gegeven de categorie Airport. Op de x-as staat het	

aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is. 76

Figuur 47: De histogrammen en cumulatieve plots van de random search voor de geometrische fout per match, gegeven de categorie Airport. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn. 77

Figuur 48: De histogrammen en cumulatieve plots van de grid search voor de geometrische fout per match, gegeven de categorie Airport. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn. 77

Figuur 49: De histogrammen en cumulatieve plots van de random search voor het aantal matches per matching-inferentie, gegeven de categorie DenseResidential. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is. 78

Figuur 50: De histogrammen en cumulatieve plots van de grid search voor het aantal matches per matching-inferentie, gegeven de categorie DenseResidential. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is. 78

Figuur 51: De histogrammen en cumulatieve plots van de random search voor de geometrische fout per match, gegeven de categorie DenseResidential. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn. 79

Figurenlijst

- Figuur 52: De histogrammen en cumulatieve plots van de grid search voor de geometrische fout per match, gegeven de categorie DenseResidential. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn. 79
- Figuur 53: De histogrammen en cumulatieve plots van de grid search voor het aantal matches per matching-inferentie, gegeven de categorie Pond. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is. 80
- Figuur 54: De histogrammen en cumulatieve plots van de grid search voor de geometrische fout per match, gegeven de categorie Pond. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn. 80
- Figuur 55: De histogrammen en cumulatieve plots van de random search voor het aantal matches per matching-inferentie, gegeven de categorie Beach. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is. 81
- Figuur 56: Het histogram en de cumulatieve plot van de grid search voor het aantal matches per matching-inferentie, gegeven de categorie Beach. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is. 81
- Figuur 57: De histogrammen en cumulatieve plots van de random search voor de geometrische fout per match, gegeven de categorie Beach. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de

betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.	82
Figuur 58: Het histogram en de cumulatieve plot van de grid search voor de geometrische fout per match, gegeven de categorie Beach. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.	82
Figuur 59: Deze figuur toont bij een matching-inferentie van de categorie Pond de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd geen CLAHE toegepast.	83
Figuur 60: Deze figuur toont bij een matching-inferentie van de categorie Pond de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd CLAHE toegepast.	83
Figuur 61: Deze figuur toont bij een matching-inferentie van de categorie Desert de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd geen CLAHE toegepast.	84
Figuur 62: Deze figuur toont bij een matching-inferentie van de categorie Desert de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd CLAHE toegepast.	84
Figuur 63: Deze figuur toont bij een matching-inferentie van de categorie Beach de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd geen CLAHE toegepast.	85
Figuur 64: Deze figuur toont bij een matching-inferentie van de categorie Beach de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd CLAHE toegepast.	85
Figuur 65: Deze figuur toont bij een matching-inferentie van de categorie Bridge de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De	

Lijst van Codefragmenten

originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd geen CLAHE toegepast.	86
Figuur 66: Deze figuur toont bij een matching-inferentie van de categorie Bridge de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd CLAHE toegepast.	86
Figuur 67: Deze figuur toont bij een matching-inferentie van de categorie Farmland de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd geen CLAHE toegepast.	87
Figuur 68: Deze figuur toont bij een matching-inferentie van de categorie Farmland de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd CLAHE toegepast.	87

Lijst van Codefragmenten

Codefragment 1: Deze code illustreert de kerngedachte van de pipeline.....	34
Codefragment 2: Deze code illustreert hoe de fout van een match absoluut bepaald wordt.	35
Codefragment 3: Dit illustreert hoe in Python-code het model geconfigureerd wordt. .	36
Codefragment 4: Deze code illustreert hoe de random search wordt geïmplementeerd.	39
Codefragment 5: Deze code illustreert hoe de grid search wordt geïmplementeerd. ...	43
Codefragment 6: Deze code illustreert de bepaling van inliers en gefaalde inferenties.	50
Codefragment 7: Deze code illustreert hoe de McNemar-test wordt opgesteld.	51
Codefragment 8: Om de randen van de getransformeerde afbeelding virtueel voor te stellen op de originele afbeelding, wordt de inverse perspectiefmatrix berekend. Dit wordt gebruikt in afbeeldingen Figuur 59 tot en met Figuur 68.	56

De gebruikte code is beschikbaar op GitHub [1].

Tabellenlijst

Tabel 1: Beschrijving van de kolommen van Tabel 3.	49
Tabel 2: Beste CLAHE-instellingen per AID-datasetcategorie.....	54
Tabel 3: Resultaat van de berekeningen omtrent Inliers Ratio, Mean Abs Inliers & Failure Rate per categorie, per CLAHE-instelling.	61
Tabel 4: Resultaten van de statistische validatie per categorie, per CLAHE-instelling. .	63

Afkortingen

Opmerking: Bij het opstellen en structureren van deze lijst is ter ondersteuning gebruikgemaakt van Google AI Studio [2].

AHE	Adaptive Histogram Equalization
AI	Artificial Intelligence (Artificiële Intelligentie)
AID	Aerial Image Dataset
CLAHE	Contrast Limited Adaptive Histogram Equalization
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CT	Computed Tomography (gebruikt als CT-scan)
ELoFTR	Efficient LoFTR
FP16	16-bit floating-point (precisieformaat)
FP32	32-bit floating-point (precisieformaat)
GPS	Global Positioning System
GPU	Graphics Processing Unit
GR-RANSAC	Geographic Relationship RANSAC
HE	Histogram Equalization
JSONL	JavaScript Object Notation Lines (bestandsformaat)
LoFTR	Detector-Free Local Feature Matching with Transformers
MAGSAC	Marginalizing Sample Consensus
MRI	Magnetic Resonance Imaging (gebruikt als MRI-scan)
ORB	Oriented FAST and Rotated BRIEF
PROSAC	Progressive Sample Consensus
RANSAC	Random Sample Consensus
RGB	Red, Green, Blue
RoMa	Robust Matching
RoPE	Rotary Position Embedding
SAR	Synthetic Aperture Radar
SIFT	Scale-Invariant Feature Transform
USAC	Universal Framework for Random Sample Consensus

Glossarium

Opmerking: Bij het opstellen en structureren van deze lijst is ter ondersteuning gebruikgemaakt van Google AI Studio [2].

AHE (Adaptive Histogram Equalization)	Een wiskundige methode en uitbreiding op HE om contrast in een afbeelding te verbeteren. In tegenstelling tot HE kan AHE beter overweg met lokale verschillen in lichtintensiteitswaarden.
AID-dataset (Aerial Image Dataset)	Een dataset bestaande uit tienduizend satellietafbeeldingen met een resolutie van 600x600 pixels, verdeeld over dertig verschillende afbeeldingscategorieën. Oorspronkelijk ontworpen voor classificatie, maar in dit onderzoek gebruikt voor het nabootsen van coregistratie.
CLAHE (Contrast Limited Adaptive Histogram Equalization)	Een algoritme voor beeldverbetering (uitbreiding op AHE) dat in afbeeldingen lokaal het contrast kan verhogen. Om te voorkomen dat aanwezige ruis ook wordt uitvergroot, maakt het gebruik van twee parameters: cliplimiet en tegelrastergrootte.
Cliplimiet	Een van de twee parameters van CLAHE. Deze limiet beperkt de extreme vergroting van lokale contrasten om te voorkomen dat beeldruis wordt versterkt.
Coregistratie	Het proces waarbij twee of meer afbeeldingen exact op elkaar worden afgestemd. Hierbij worden statische elementen tussen de opnames gematcht, waarna een afbeelding wordt vervormd (gewarped) zodat de elementen geografisch op dezelfde locatie liggen.
Dense / Semi-dense model	Modellen voor feature matching. Een <i>dense model</i> (zoals RoMa) is ontworpen om alle mogelijke overeenkomsten tussen afbeeldingen te schatten (trager, robuuster). Een <i>semi-dense model</i> (zoals LoFTR of eLoFTR) gebruikt een 'coarse-to-fine'-strategie die geoptimaliseerd is voor efficiëntie.
ELoFTR (efficient LoFTR)	De efficiëntere en 2,5 keer snellere opvolger van LoFTR. Dit model gebruikt een RepVGG-backbone voor feature-extractie en een two-stage fine-level matching (correlatie eerst op pixelniveau, daarna op sub-pixelniveau).

Feature description	Een set van karakteristieken of features die iets nuttigs zeggen over een afbeelding. Vaak wordt dit weergegeven in een behapbare vorm (array of lijst van numerieke waarden) die fungeert als een 'digitale vingerafdruk' voor een specifiek punt (keypoint).
Feature detector (Feature detection)	Een wiskundig algoritme (zoals SIFT of ORB) dat een discreet aantal punten uit een afbeelding haalt die het als belangrijk beschouwt.
Feature matching	De essentie van coregistratie. Hierbij worden features (keypoints) van twee verschillende afbeeldingen met elkaar verbonden door, op basis van hun descriptions, de dichtste buur te vinden.
Features	Een numerieke manier om specifieke kenmerken van afbeeldingen voor te stellen (bijv. m.b.v. een Convolutional Neural Network). Er is een onderscheid tussen <i>low-level features</i> (contouren, randen, kleuren) en <i>high-level features</i> (gecompliceerde vormen, zoals een gezicht).
Geometrische transformatiematrix	Een matrix die wiskundig de verplaatsing of vervorming tussen twee afbeeldingen beschrijft. In dit document gesimuleerd door de vier hoekpunten van een afbeelding trapeziumvormig en naar buiten toe te verplaatsen.
GR-RANSAC (Geographic Relationship RANSAC)	Een specifieke, moderne variant van het RANSAC-algoritme gericht op geografische toepassingen.
HE (Histogram Equalization)	Een basismethode voor contrastverbetering. Het verhoogt het contrast globaal door de lichtintensiteitswaarden (het histogram) uniform over een complete afbeelding te verspreiden.
Inliers	Matches tussen twee afbeeldingen die correct zijn (vaak algoritmisch bepaald door RANSAC). In het kader van dit experiment worden inliers deterministisch gedefinieerd als matches met een zeer lage fout (een ruimtelijke afwijking van maximaal 50 pixels ten opzichte van de wiskundige grondwaarheid).
Keypoints	Een discreet aantal punten in een afbeelding die door een feature detector (of detector-free algoritme) als belangrijk of onderscheidend worden beschouwd.

Glossarium

LoFTR (Detector-Free Local Feature Matching with Transformers)	Een algoritme dat, in plaats van een klassieke feature detector, direct vertrekt vanuit feature extractie en via <i>self-attention</i> en <i>cross-attention</i> verbanden zoekt op zowel globaal als sub-pixel niveau.
MatchAnything	Een grootschalig pretrained AI-model voor universele cross-modale afbeeldingmatching, gelanceerd in 2025. Het presteert goed op complexe datasets en steunt op architecturen zoals eLoFTR en RoMa.
Outliers	Matches tussen twee afbeeldingen die niet correct of onlogisch zijn. Deze worden traditioneel weggefilterd door algoritmes zoals RANSAC.
Parallax(fenomeen)	Een optisch fenomeen bij grote hoogteverschillen waarbij een statisch, niet-bewegend object (zoals de top van de Burj Khalifa) zich op foto's sterk lijkt te verplaatsen doordat de afbeeldingen vanuit verschillende hoeken zijn genomen.
PROSAC (Progressive Sample Consensus)	Een moderne variant op RANSAC die gebaseerd is op probabiliteit (waarschijnlijkheid) en bijvoorbeeld rekening kan houden met de confidence score van matches.
RANSAC (Random Sample Consensus)	Een algoritme met een willekeurig (gerandomiseerd) element dat wordt gebruikt om outliers (foutieve matches) weg te filteren en een transformatiematrix te schatten aan de hand van logische matches (inliers).
RoMa	Een dense AI-model ontworpen voor feature matching, nauwkeuriger maar doorgaans zwaarder en trager dan semi-dense alternatieven.
Spearman-correlatie	Een statistische methode om de samenhang tussen twee variabelen te meten. In tegenstelling tot de Pearson-correlatie is Spearman geschikt voor monotoon stijgende of dalende relaties en niet louter voor lineaire verbanden.
Tegelrastergrootte	De tweede parameter van het CLAHE-algoritme. Deze bepaalt de grootte van het raster (de 'tegels') waarbinnen het algoritme de contrastverhoging lokaal berekent.
Veranderingsdetectie	Het observeren en zoeken van veranderingen tussen twee of meerdere momenten in computervisie. Vereist vaak een eerste stap van coregistratie om de afbeeldingen correct over elkaar heen te leggen.

Warping / Gewarped

Het wiskundig vervormen van een afbeelding op basis van gevonden matches, zodat de inhoud exact overeenkomt met een referentieafbeelding qua schaal, hoek en perspectief.

Introductie

Context

Veranderingsdetectie is het observeren van veranderingen tussen twee of meerdere momenten. Binnen computervisie, als het gaat om het vergelijken van afbeeldingen, is veranderingsdetectie een uitdagend onderwerp. Dit omdat er enerzijds grote veranderingen kunnen zijn tussen twee afbeeldingen, anderzijds omdat de locaties op de afbeeldingen niet helemaal overeenkomen. Indien de locaties op twee afbeeldingen niet exact overeenkomen, dient eerst een coregistratiestap gedaan te worden. Hierbij worden elementen die verondersteld zijn niet te wijzigen of verplaatsen tussen de tijdstippen van opname van de afbeeldingen gematcht. Op basis van deze matches kan een van de afbeeldingen gewarped (vervormd) worden, zodat deze objecten op dezelfde locatie liggen. Als alle elementen die verondersteld worden niet te kunnen wijzigen of verplaatsen exact op dezelfde locatie liggen in beide afbeeldingen, en er zijn voldoende van deze elementen, kan men ervan uitgaan dat ook de overige elementen van de afbeeldingen op de correcte locatie liggen. Het matchen van vaste elementen en een gepaste warping vinden heet coregistratie [3]. Coregistratie kan binnen hetzelfde afbeeldingstype zijn, maar heeft ook toepassingen in het warpen van afbeeldingen die een ander type hebben. Een voorbeeld hiervan is het combineren van een MRI-scan en een CT-scan binnen een medische context. Een voorbeeld van coregistratie binnen eenzelfde afbeeldingstype is coregistratie van visuele satellietafbeeldingen waarvan de afbeeldingen genomen zijn op verschillende momenten. Dit is een schoolvoorbeeld van coregistratie voor veranderingsdetectie.

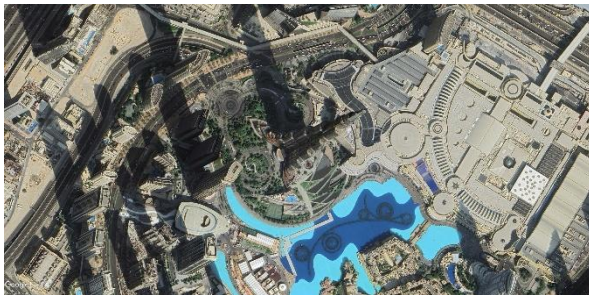
MatchAnything behaalt state-of-the-artresultaten op cross-modale benchmarks [4]. Daarom is er een vermoeden dat het MatchAnything-eLoFTR-model op het matchen van verschillende satellietafbeeldingen ook goed zal scoren. Het MatchAnything-model is een pretrained AI-model dat steunt op de architectuur van eLoFTR, dat gefinetuned is op zowel cross-modulaire afbeeldingstypes, als synthetisch gewarpte afbeeldingen. MatchAnything neemt twee afbeeldingen als input en levert als output de locaties van de matchende elementen (keypoints), samen met een confidence score per match. Op het eerste gezicht lijkt coregistratie van satellietafbeeldingen op basis van matches gegenereerd door MatchAnything een geschikt stramien.

Mogelijk kan het toevoegen van een contrastverhogende preprocessingstap een gunstige invloed hebben op de prestaties van MatchAnything. Een algemeen bekende vorm van contrastverhoging is CLAHE. CLAHE is een wiskundige techniek die in afbeeldingen het contrast kan verhogen zonder ruis te vergroten. CLAHE steunt op twee parameters: cliplimiet en tegelrastergrootte. Beide hebben een invloed op het uiteindelijke resultaat van de contrastverhoging.

Probleemstelling

Er zijn echter tal van complexiteiten aan coregistratie van satellietdata. Zo is er ten eerste de invloed van hemellichamen zoals de zon op het visuele uitzicht van de satellietafbeeldingen. Maar er zijn ook minder voor de hand liggende complexiteiten, zoals perspectieffervorming door de hoek die de camera van de satelliet maakt met het aardoppervlak en het parallaxfenomeen dat optreedt bij grote hoogteverschillen. Bepaalde elementen kunnen hierdoor onverwacht verborgen zijn achter andere. Parallax zorgt er ook voor dat een warping zeer complex kan worden, willen alle statische elementen gematcht worden.

Vergelijkt men twee satellietafbeeldingen van de Burj Khalifa (zie Figuur 1 en Figuur 2), dan is een grote parallax zichtbaar. De statische elementen op de grond verplaatsen nagenoeg niet, maar een element van de top van de Burj Khalifa lijkt enorm veel verplaatst. De fysieke locatie van de top van het gebouw is op beide foto's exact dezelfde, toch lijkt erop de foto's een enorme verplaatsing te zijn gebeurd.



Figuur 1: Luchtfoto van de Burj Khalifa [5].



Figuur 2: Luchtfoto van de Burj Khalifa [6].

De vraag rijst of MatchAnything hiermee overweg kan. Na een pilotonderzoek blijkt dat MatchAnything het moeilijk heeft om valide matches te vinden bij satellietafbeeldingen waar grote parallax of perspectieffervorming aanwezig is.

Doordat het niet eenvoudig is om manueel een reële warping te berekenen op basis van twee satellietafbeeldingen, kan deze niet gebruikt worden om te bepalen of een matching goed dan wel slecht verliep. Het manueel controleren van alle matches zou arbeidsintensief zijn en is niet werkbaar op grote schaal.

Doelstelling

Uit het pilotonderzoek bleek dat voor sommige gevallen door contrastverhogende technieken toe te passen op de afbeeldingen voordat de matching-inferentie gebeurt een beter resultaat behaald kan worden.

Om dit te testen wordt een experiment opgezet waarbij men kan vaststellen wanneer een matching goed verlopen is. Omdat datasets van satellietafbeeldingen vaak geen grondwaarheid hebben over wiskundige transformaties tussen twee afbeeldingen, wordt in het experiment een proxyaanpak gebruikt. Door een dataset van satellietafbeeldingen (de AID-dataset) te nemen en daar gecontroleerde synthetische

transformaties en ruis op toe te passen, simuleren we het scenario waarin twee verschillende satellieten op hetzelfde moment een foto nemen van dezelfde locatie. Dit zorgt ervoor dat we zelf de grondwaarheid van de transformatie kunnen bepalen en weten. Dit is een vereenvoudigde benadering van echte scenario's die noodzakelijk is om een experiment te kunnen voltooien.

Doordat we zelf de transformaties bepalen, kunnen we de correctheid van een match aftoetsen op deze transformatie. Dit zorgt ervoor dat we niet afhankelijk zijn van algoritmische methoden om de transformatie te schatten, zoals RANSAC. Deze methoden introduceren namelijk hun eigen bias.

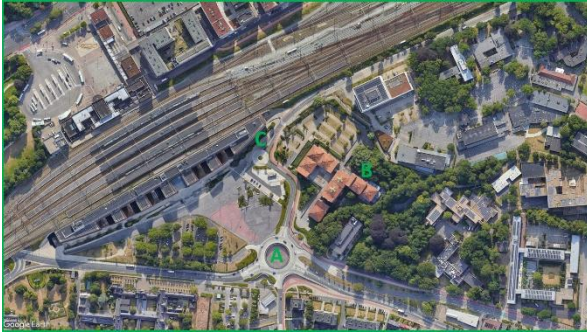
Het doel van deze thesis is tweeledig. Kan CLAHE een verbetering in het resultaat van MatchAnything-inferentie teweegbrengen? En kan op basis van de categorie van een locatie bepaald worden welke CLAHE-instelling het best gebruikt wordt?

1 Theoretische Achtergrond

1.1 Coregistratie en Satellietdata

Er zijn diverse redenen waarom men veranderingen over de tijd wil detecteren tussen twee afbeeldingen. Veranderingen over tijd detecteren is een toepassing van het zoeken van verschillen tussen twee momenten. Voor mensen is dit relatief eenvoudig, doordat mensen goed patronen of herkenbare punten kunnen herkennen in beide afbeeldingen. Door deze punten met elkaar te verbinden, ontstaat een relatie tussen beide afbeeldingen. Eenmaal de herkenbare punten uit beide afbeeldingen in relatie met elkaar staan, kunnen mensen de afbeeldingen denkbeeldig op elkaar leggen, zodanig dat de meeste herkenbare punten geografisch dicht bij elkaar liggen. Mensen kunnen dit vrij eenvoudig, ook als de opnamehoek van beide afbeeldingen verschillend is. Het op elkaar leggen van twee afbeeldingen met een andere opnamehoek vereist een geometrische transformatie van een van de afbeeldingen. Eenmaal de transformatie gedaan is, en de overeenkomstige herkenbare punten van de afbeeldingen op elkaar liggen, kunnen mensen goed opmerken welke zaken er verschillend zijn tussen beide afbeeldingen. Dit wordt geïllustreerd met een voorbeeld in Figuur 3, Figuur 4 en Figuur 5. Op Figuur 5 is te zien dat op een parking (Figuur 3) een nieuw gebouw geplaatst werd (Figuur 4).

Scenariospecifieke Contrastverbetering voor Satellietcoregistratie



Figuur 3: Luchtfoto uit 2022 met drie duidelijk herkenbare punten A, B en C. De originele foto in deze figuur is afkomstig van Google Earth [7].



Figuur 4: Luchtfoto uit 2025 met drie duidelijk herkenbare punten A, B en C. De originele foto in deze figuur is afkomstig van Google Earth [8].



Figuur 5: Luchtfoto's na coregistratie op basis van punten A, B en C. Een zichtbare verandering is omkaderd. Beide foto's komen origineel van Google Earth [7], [8].

Een computer heeft heel wat meer moeite met het detecteren van verschillen tussen afbeeldingen. De eerste fase van het vinden van een geometrische transformatie die ervoor zorgt dat twee afbeeldingen gealigneerd kunnen worden, is hierin zeer belangrijk. Zonder beide afbeeldingen goed te aligneren met behulp van een transformatie lijkt het voor een computer alsof alles verschillend is. Dit proces van zoeken naar een transformatie en deze toepassen (warpen) heet coregistratie. Om een effectieve veranderingsdetectie te kunnen doen, dient eerst een succesvolle coregistratie te gebeuren.

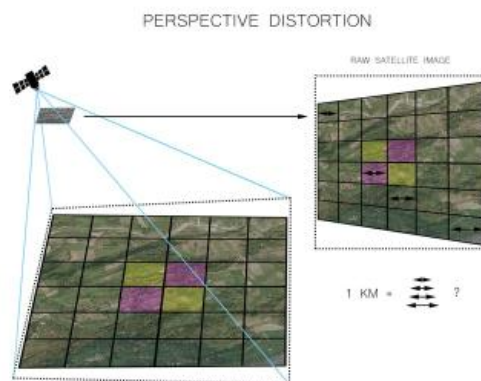
Om de coregistratie van verschillende satellietafbeeldingen te voltooien, zijn er een aantal complexiteiten. Ten eerste is er het verschil in tijdstip. Wat zichtbaar is op een satellietafbeelding is niet enkel afhankelijk van op welke dag de afbeelding genomen is. Ook binnen een dagcyclus zijn er variaties te merken door de stand van de zon. Dit heeft een effect op de vorm en richting van de schaduw van objecten.

Het zicht van de aarde wijzigt vanuit de ruimte continu, dit door natuurlijke of menselijke veranderingen. Seizoenen of klimaatveranderingen hebben een invloed op de kleur van vegetatie, de hoeveelheid sneeuw op het aardoppervlak, enzovoort. Menselijke factoren zoals stedelijke uitbreidingen, landbouw en het aanleggen van wegennetwerken hebben ook een grote invloed op het uitzicht van satellietdata. Bovengenoemde factoren hebben een relatief trage invloed op de satellietafbeeldingen. Het verplaatsen van een voertuig, de beweging van vee in een weide, de straatverlichting

Theoretische Achtergrond

aansteken, aanwezigheid van bewolking hebben allemaal een onmiddellijk effect op satellietdata. Het is belangrijk bij coregistratie dat de matches toegepast worden op de elementen die statisch zijn, en dus niet bewegen over een bepaalde tijd.

Zelfs als een satellietafbeelding op exact hetzelfde moment opgenomen is door twee verschillende satellieten, zal er toch steeds een verschil te zien zijn tussen beide afbeeldingen. Dit is doordat de twee satellieten onmogelijk op dezelfde plek in de ruimte kunnen zijn. Het gevolg hiervan is een trapeziumvormige perspectieffervorming. Dit wordt geïllustreerd in Figuur 6. Met behulp van technieken als orthorectificatie kan deze perspectieffervorming deels weggewerkt worden [9], maar deze vereist nauwkeurige hoogtedata en is niet perfect.



Figuur 6: In deze afbeelding wordt de trapeziumvormige perspectieverandering bij opname van satellietfoto's geïllustreerd [10].

Bij grote hoogteverschillen kan er ook parallax optreden. Dit is een optisch fenomeen dat zich voordoet wanneer een statisch object, dat dus niet bewegen kan, zich toch op verschillende locaties lijkt te bevinden, gezien vanuit verschillende standpunten. Een voorbeeld hiervan is wanneer hoge gebouwen of standbeelden lijken te hellen in een bepaalde richting, weg van het punt waarvan de luchtfoto is genomen. Dit is duidelijk zichtbaar op de afbeeldingen van de Burj Khalifa (zie Figuur 1 en Figuur 2), waarbij de basis van de toren vrijwel niet beweegt, maar de top zich enorm ver lijkt te verplaatsen.

Daarnaast heeft gps-locatie ook een bepaalde nauwkeurigheid, ook hierdoor kunnen kleine variaties ontstaan op de afbeeldingen. Ook kan er ruis optreden in de afbeeldingen, zeker wanneer het SAR-afbeeldingen zou betreffen.

Met deze complexiteiten dient rekening gehouden te worden wanneer veranderingsdetectie van satellietafbeeldingen gedaan wordt.

1.2 Feature Matching & Modellen

De theoretische kaders voor feature extraction en matching in deze sectie zijn gebaseerd op de methodiek zoals uiteengezet in de Hugging Face computervisie cursus [11], aangevuld met de oorspronkelijke literatuur van de betreffende algoritmen.

1.2.1 Features

Features zijn een manier om specifieke kenmerken van afbeeldingen voor te stellen. Een toepassing is CNN (Convolutional Neural Network), waar met behulp van verschillende filters features uit afbeeldingen gehaald worden. Features van afbeeldingen kunnen opgedeeld worden in verschillende niveaus, low-level en high-level. Voorbeelden van low-level features zijn contouren, randen, hoeken en kleuren. High-level features bouwen voort op low-level features en zijn gecompliceerder. Een voorbeeld van een high-level feature is een menselijk gezicht [12]. Belangrijk is dat de features op zichzelf enkel numerieke waarden zijn die voor mensen bijna onmogelijk te begrijpen zijn. Het feature zegt welke waarde erop welke plaats aanwezig is [11].

1.2.2 Feature Description

Een goede feature description is een set van karakteristieken of features die iets nuttigs zeggen over een afbeelding. Dit dient in een behapbare vorm weergegeven te worden, vaak is dit een array of een lijst van numerieke waarden. Als het ware krijgen punten op een afbeelding een digitale vingerafdruk [11].

1.2.3 Feature Detection

Een feature detector is een wiskundig algoritme dat een discreet aantal punten uit een afbeelding haalt die het als belangrijk beschouwt. Deze punten worden keypoints genoemd. Bij elk punt kan een description berekend worden. Voorbeelden van feature detectors zijn SIFT en ORB [11].

1.2.4 Feature Matching

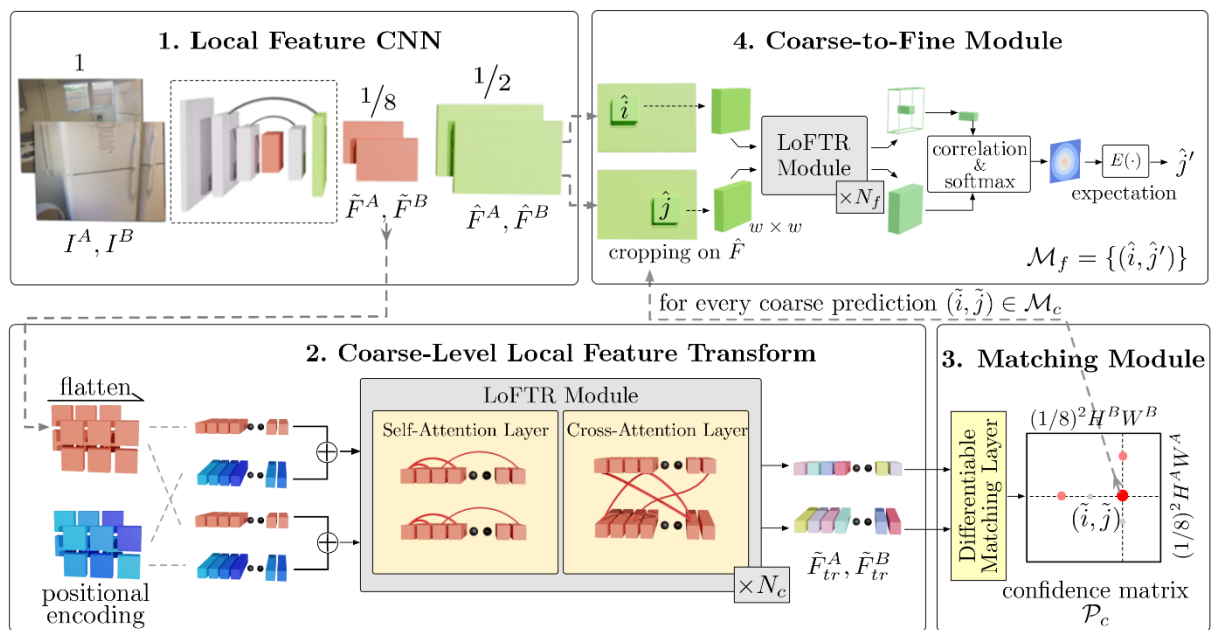
Het matchen van features is de essentie van coregistratie. Als op twee afbeeldingen een feature detection werd gedaan, kan aan de hand van de descriptions van elk keypoint een dichtste buur gevonden worden. Door deze keypoints van beide afbeeldingen met elkaar te connecteren, krijg je verbindingen tussen matchende features. Dit is hoe detectorgebaseerde feature matching werkt [11].

1.2.5 LoFTR

In plaats van detectorgebaseerde feature matching kan ook een detector-free algoritme zoals LoFTR (Detector-Free Local Feature Matching with Transformers) gebruikt worden [13]. Deze technieken gebruiken geen feature detector. In plaats van te vertrekken vanuit feature detection om gebruikmakend van feature description tot feature matching te komen, vertrekt LoFTR vanuit feature extractie. Nadien wordt zowel self-attention als

cross-attention toegepast. Met self-attention wordt een context gecreëerd binnen de afbeelding zelf. Met cross-attention worden verbanden gevonden tussen afbeeldingen. Eerst wordt dan op basis van de ruwe, high-level, features gematcht tussen de afbeeldingen, nadien wordt dit bijgesteld tot op pixelniveau aan de hand van de fijne, low-level, features. Dit wordt schematisch voorgesteld op Figuur 7.

Alternatieve detector-free feature matchers zijn eLoFTR en RoMa. RoMa is een dense model, terwijl (e)LoFTR een semi-dense model genoemd wordt. Een semi-dense model gebruikt een coarse-to-fine-strategie geoptimaliseerd voor efficiëntie, in tegenstelling tot een dense model dat ontworpen is om alle mogelijke overeenkomsten tussen afbeeldingen te schatten. Dense modellen zijn over het algemeen trager door de robuuste architectuur dan semi-dense modellen [14].

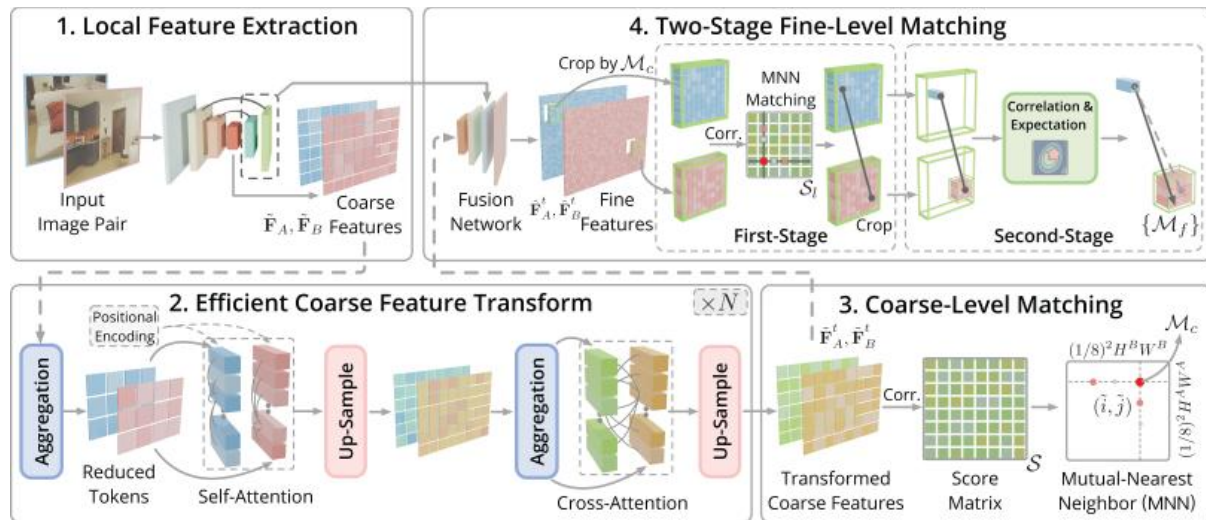


Figuur 7: Overzicht van de LoFTR-methode, zoals te zien in de LoFTR-paper [13].

1.2.6 ELoFTR

ELoFTR (ook wel efficient LoFTR) is de opvolger van LoFTR, die beter presteert [14]. LoFTR maakt gebruik van een multi-branch ResNet-architectuur, wat standaard is in veel detector-free matchers. Omdat dit eerder zwaar is, implementeert eLoFTR een efficiënte RepVGG-backbone voor feature-extractie. Daarnaast gebruikt eLoFTR een eigen aggregated attention module die eerst saillante tokens zal filteren. Doordat er dus minder tokens zijn dan LoFTR zou hebben, kan gebruikgemaakt worden van Vanilla Attention gecombineerd met RoPE. Daarnaast maakt eLoFTR in tegenstelling tot LoFTR gebruik van two-stage fine-level matching. Voor fijne features wordt eerst op pixelniveau een correlatie bepaald, nadien op sub-pixelniveau. Dit wordt schematisch voorgesteld

in Figuur 8. Bovenstaande, gecombineerd met nog wat andere optimalisaties, zorgt ervoor dat eLoFTR 2,5 keer sneller is dan LoFTR.



Figuur 8: Overzicht van de eLoFTR-methode, zoals te zien in de eLoFTR-paper [14].

1.2.7 MatchAnything

MatchAnything is een pretrained training framework, toegespitst op universele cross-modale afbeeldingmatching. In 2025 introduceerde MatchAnything voor zowel eLoFTR als RoMa een grootschalig voorgetraind model [15], [16]. Deze modellen zijn getraind op afbeeldingen getrokken vanuit verschillende hoeken, opeenvolgende frames van videofragmenten en synthetisch gewarpte afbeeldingen. Ook werd er gebruikgemaakt van synthetisch gegenereerde cross-modale afbeeldingen in de training. Op bepaalde datasets, die voornamelijk gericht waren op cross-modaliteit of extreme opnamehoekverschillen, scoort dit model beter dan het eLoFTR-model of het RoMa-model, die niet getraind werden door MatchAnything. Ook presteren het MatchAnything-eLoFTR-model en het MatchAnything-RoMa-model beter dan detectorgebaseerde alternatieven zoals SIFT.

Het MatchAnything-RoMa-model is nauwkeuriger dan het MatchAnything-eLoFTR-model, beide zijn toegankelijk via een Hugging Face Space [17]. Wil men het model lokaal runnen, dan kon men in december 2025 enkel een beroep doen op het MatchAnything-eLoFTR-model, dat beschikbaar is op Hugging Face [15]. Het model lokaal runnen heeft diverse voordelen. Zo is er geen internetconnectie vereist en blijft de data privé, wat een groot voordeel is voor veel bedrijven.

1.3 Beeldverbetering: CLAHE

CLAHE (Contrast Limited Adaptive Histogram Equalization) is een algoritme dat een uitbreiding is op AHE (Adaptive Histogram Equalization), wat op zijn beurt een uitbreiding is op HE (Histogram Equalization) [18], [19]. Allemaal zijn het methodes om contrast in een afbeelding te verbeteren. HE zal de verdeling van de verschillende

Theoretische Achtergrond

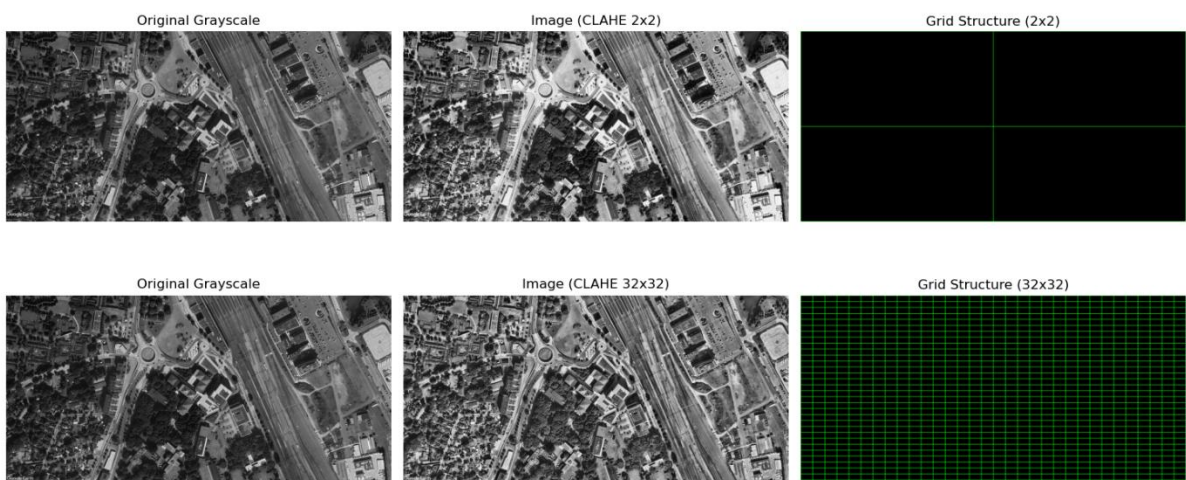
lichtintensiteitswaarden, voorgesteld in een histogram, zodanig over de complete afbeelding verspreiden dat deze uniform wordt. Het contrast wordt globaal dus verhoogd. In tegenstelling tot HE kan AHE beter omgaan met lokale verschillen in lichtintensiteitswaarden. Er wordt bij AHE gebruikgemaakt van een bepaalde tegelgrootte, dit is een parameter, om lokaal het contrast te verhogen. CLAHE zal naast een tegelrastergrootte ook een cliplimiet gebruiken om de contrastverbetering te beperken. De cliplimiet kan voorkomen dat ruis ook vergroot wordt. CLAHE heeft dus twee parameters, cliplimiet en tegelrastergrootte. Beide hebben een effect op het uiteindelijke contrast van een afbeelding [20]. Het effect van beide parameters wordt visueel gedemonstreerd in Figuur 9 en Figuur 10.

CLAHE: Clip Limit Comparison (Grid: 8x8)



Figuur 9: Vergelijkende illustratie van het toepassen van CLAHE op een luchtfoto met een cliplimiet gelijk aan 2.0 tegenover met een cliplimiet gelijk aan 20.0. De foto die gebruikt wordt van dit plot komt origineel van Google Earth [8].

CLAHE: Tile Grid Size Comparison



Figuur 10: Vergelijkende illustratie van het toepassen van CLAHE op een luchtfoto met een tegelrastergrootte gelijk aan 2x2 tegenover met een tegelrastergrootte gelijk aan 32x32. De foto die gebruikt wordt in dit plot komt origineel van Google Earth [8].

1.4 RANSAC als Outlier Filter

RANSAC (Random Sample Consensus) is een techniek die gebruikt kan worden om outliers weg te filteren [21]. Uit feature matching komen heel wat matches, waarvan een aantal niet correct zijn. Het is niet per definitie zo dat op basis van de confidence score van de matches gezegd kan worden of ze correct en logisch zijn. De transformatiematrix berekenen die de warping van de coregistratie moet voltooien, vereist een set van logische en correcte matches. Met behulp van RANSAC kan op een gerandomiseerde wijze een schatting gemaakt worden van een aantal matches die een logisch verband houden.

Er zijn tegenwoordig talloze varianten die beter presteren dan RANSAC. USAC_MAGSAC is algemeen beter en moderner dan RANSAC. RANSAC werd gepubliceerd in 1981, USAC in 2012 en MAGSAC in 2019 [22], [23], [24]. Ook zijn er varianten die specifiek op een bepaald toepassingsgebied beter zijn. Voorbeelden zijn PROSAC (gebaseerd op probabiliteit) of GR-RANSAC (Geographic Relationship RANSAC) [25], [26].

Hoewel RANSAC-methoden vaak gebruikt worden om foutieve matches weg te filteren, en nadien een goede transformatie te kunnen schatten, introduceert het door zijn willekeurige aard een bias. Bij het kwantitatief vergelijken van resultaten door het toepassen van preprocessingtechnieken zoals CLAHE kan deze willekeur de resultaten sterk kleuren en beïnvloeden. Daarom wordt in het experiment van deze paper geen gebruikgemaakt van RANSAC-methoden om outliers te filteren.

2 Methodologie

Opmerking: Bij het schrijven van de software voor dit experiment is Google AI Studio [2] ingezet ter ondersteuning voor het verfijnen, optimaliseren en debuggen van de Python-code. De conceptuele logica is eigen werk.

2.1 Hypothese

De hypothese is dat het toepassen van een bepaalde CLAHE-instelling op paren van satellietafbeeldingen, die getransformeerd werden en waaraan ruis werd toegevoegd, kan leiden tot betere prestaties van het MatchAnything-model. Dit wordt afgetoetst op de verschillende categorieën binnen de AID-dataset [27]. Verwacht wordt dat per afbeeldingscategorie een optimale CLAHE-instelling bepaald kan worden die leidt tot meer matches met een lagere fout, en leidt tot meer succesvolle MatchAnything-inferenties.

Een MatchAnything-inferentie wordt als gefaald beschouwd als er geen enkele match gevonden wordt tussen twee afbeeldingen. De fout van een match wordt als laag beschouwd wanneer de ruimtelijke afwijking maximaal 50 pixels bedraagt.

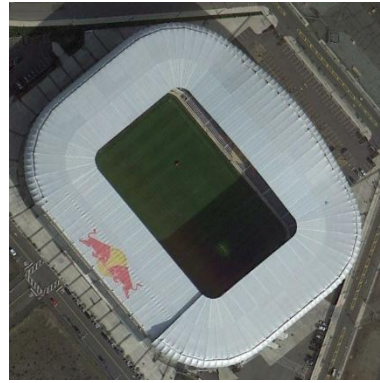
2.2 Dataset en Proxybenadering

Het selecteren van een geschikte dataset vormde een uitdaging. Idealiter zou er gebruikgemaakt worden van een dataset met satellietafbeeldingen van diverse afbeeldingscategorieën van locaties, opgenomen op verschillende tijdstippen, waarvan de geometrische transformatie tussen de beelden exact gekend is. Een mogelijke manier om aan zo'n dataset te komen is om op Google Earth afbeeldingen te nemen van verschillende historische datums. Een grote beperking is dat hier de transformatie ongekend blijft en manueel zou moeten bepaald worden. Daarnaast introduceert dit, in combinatie met de nauwkeurigheid van gps-locaties [28], een foutenmarge waardoor de afbeeldingsalignering niet tot op de pixel nauwkeurig is. Dit zou voor een bias zorgen in de dataset. Bovendien zou het gebruik van Google Earth als bron voor een grootschalige dataset onwerkbaar zijn. Bij het gebruik van Google Earth zouden ook de afbeeldingscategorieën handmatig bepaald moeten worden.

Om deze redenen werd voor een proxybenadering met behulp van de AID (Aerial Image Dataset) gekozen [27]. Deze dataset is eigenlijk ontworpen voor classificatie. De AID-dataset, publiek beschikbaar via Kaggle [29], bevat in totaal tienduizend satellietafbeeldingen met een resolutie van 600x600 pixels, verdeeld over dertig verschillende afbeeldingscategorieën. Wanneer er in het vervolg van deze tekst verwezen wordt naar een afbeeldingscategorie zal dit korter aangeduid worden met "categorie". Bij wijze van illustratie wordt in Figuur 11, Figuur 12, Figuur 13 en Figuur 14 een afbeelding uit enkele categorieën getoond.



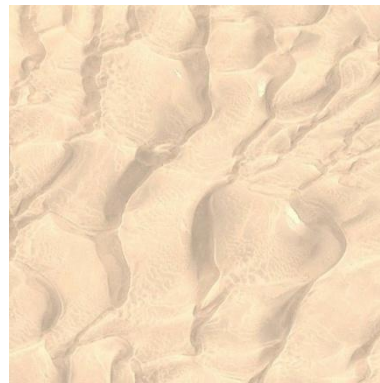
Figuur 11: Voorbeeld van een afbeelding uit de AID-dataset met categorie Bridge.



Figuur 12: Voorbeeld van een afbeelding uit de AID-dataset met categorie Stadium.



Figuur 13: Voorbeeld van een afbeelding uit de AID-dataset met categorie Mountain.



Figuur 14: Voorbeeld van een afbeelding uit de AID-dataset met categorie Desert.

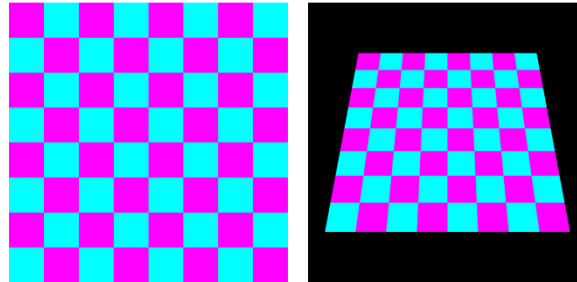
We creëren de proxydataset door kopieën van deze afbeeldingen een gecontroleerde transformatie te laten ondergaan. Hiermee bereiken we iets dat neigt naar perspectiefvervorming. Nadien wordt ook een gaussiaanse ruis toegevoegd aan de getransformeerde afbeelding. Deze ruis zorgt voor een verschil tussen de twee afbeeldingen. Het parallaxfenomeen ontbreekt hier wel in. Het enorme voordeel van deze proxybenadering is dat het matchen van twee afbeeldingen, genomen door verschillende satellieten, nagebootst wordt, terwijl de wiskundige grondwaarheidstransformatie exact gekend is.

2.3 Data Pipeline en Synthetische Transformaties

Met behulp van opencv-python wordt elke afbeelding ingelezen. Deze wordt herschaald naar een resolutie van 832x832, dit omdat het MatchAnything-model deze inputgrootte verwacht [30].

Vervolgens wordt een geometrische transformatiematrix gegenereerd. Deze transformatiematrix komt tot stand door een pseudowillekeurige verplaatsing van de vier hoekpunten van de afbeelding. De hoekpunten worden uitsluitend naar buiten toe,

weg van het centrum van de afbeelding, verplaatst. Dit voorkomt een uitzoomeffect, wat ervoor zou zorgen dat er artificiële zwarte randen gecreëerd worden. Dit zou een bias kunnen vormen in de data, omdat het AI-model de oriëntatie van deze randen als een feature zou kunnen gebruiken om de transformatiematrix te bepalen. Het ontstaan van deze randen wordt weergegeven in Figuur 15.



Figuur 15: Illustratie van ongewenste zwarte randen bij uitzoomeffect na inwaartse transformatie.

Om de reële perspectieffervorming na te bootsen, wordt de transformatie beperkt tot enkel trapeziumvormige transformaties. Er wordt willekeurig gekozen of de parallelle zijden van het trapezium horizontaal of verticaal liggen. Elk van de vier hoekpunten wordt willekeurig en onafhankelijk van elkaar verplaatst. De verplaatsing van de hoekpunten wordt beperkt van minimaal 5% tot maximaal 25% van de afbeeldingsresolutie.

Op basis van deze vier verplaatste punten wordt een transformatiematrix berekend en toegepast op de originele afbeelding. Op deze getransformeerde (gewarpte) afbeelding wordt ook nog een gaussiaanse ruis toegevoegd met een gemiddelde van 0 en een standaardafwijking van 30. Dit is een relatief grote hoeveelheid ruis, doelbewust gekozen om het realistische scenario dat twee verschillende afbeeldingen gematcht worden te simuleren.

Hoe dit in Python-code geïmplementeerd wordt is zichtbaar in Codefragment 1. Een schematische voorstelling van de data-pipeline is te vinden in Figuur 16. Codefragment 1: Deze code illustreert de kerngedachte van de pipeline.

```

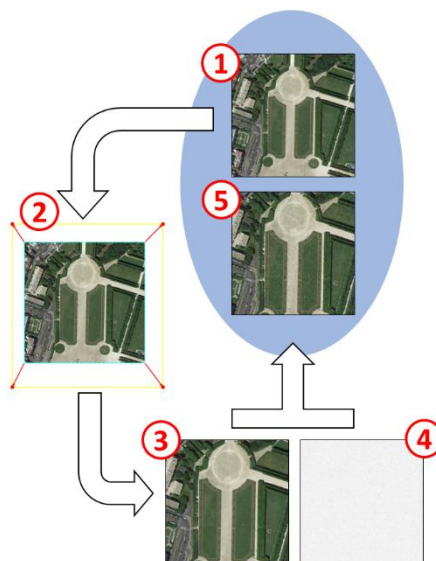
is_horizontal = random.choice([True, False])
def r():
    return random.uniform(0.05, 0.25)

if is_horizontal:
    dy_top = -r()
    dy_bot = r()
    pts_dest = np.array([
        [-r() * w, dy_top * h],
        [w + r() * w, dy_top * h],
        [w + r() * w, h + dy_bot * h],
        [-r() * w, h + dy_bot * h]
    ], dtype=np.float32)
    ...

noise = np.random.normal(mean, std, (h, w)).astype(np.float32)
noisy_image = np.clip(image.astype(np.float32) + noise, 0, 255)

```

Codefragment 1: Deze code illustreert de kerngedachte van de pipeline.



Figuur 16: Schematische voorstelling van de data pipeline. De originele afbeelding (1) wordt volgens een trapeziumvormige transformatie vergroot. Deze afbeelding (3) vormt samen met een gaussische ruis (4) een nieuwe afbeelding (5).

2.4 Evaluatie: Scoring zonder RANSAC

Een manier om te bepalen hoe goed een feature matching gebeurt, is het schatten van een transformatiematrix op basis van RANSAC-inliers. Deze inliers worden bepaald op algoritmische wijze, waar willekeurig by design aanwezig is [22]. Hierbij wordt de geschatte transformatiematrix vergeleken met de originele transformatiematrix. Dit kan op een algebraïsche manier, of door een aantal controlepunten (zoals de hoekpunten van de afbeelding) te transformeren met de geschatte matrix. De euclidische afstand tussen hoekpunten na de originele transformatie enerzijds, en de hoekpunten na de post-RANSAC geschatte transformatie anderzijds, vormt dan een maatstaf voor de fout.

RANSAC wordt vaak toegepast na het uitvoeren van feature matching in computervisie [31]. Hoewel het gebruik van RANSAC op het eerste gezicht een logische manier is om te evalueren of de ene matching beter is dan de andere, wordt hiermee een ongewenste bias geïntroduceerd. RANSAC, en zijn moderne variaties, steunen op een willekeurige initiële selectie van punten, een random sampling. Door het toepassen van RANSAC wordt een extra willekeurigheid geïntroduceerd, waardoor de resultaten van de matching niet meer louter van de MatchAnything-inferentie zouden afhangen. Om het effect van een optimale CLAHE-instelling betrouwbaar en kwantitatief te meten, is deze willekeurigheid onaanvaardbaar.

Daarom wordt een deterministische scoringsmethode gebruikt die enkel steunt op de beschikbare gegevens na de MatchAnything-inferentie. Aangezien de originele transformatiematrix gekend is (zie sectie 2.3 voor meer context) kan voor elk gevonden keypoint uit de originele afbeelding exact berekend worden waar op de getransformeerde afbeelding het matchende keypoint zich zou moeten bevinden, de verwachte locatie. MatchAnything levert als output de voorspelde locatie van dit keypoint. De euclidische afstand tussen de verwachte locatie van het keypoint en de voorspelde locatie van het keypoint vormt voor elke individuele match een absolute waarde voor de fout. Hoe dit in Python geïmplementeerd kan worden, is geïllustreerd in Codefragment 2.

```
def calculate_match_errors(kpts0, kpts1, true_matrix):
    pts0 = np.array(kpts0, dtype=np.float32).reshape(-1, 1, 2)
    pts1 = np.array(kpts1, dtype=np.float32).reshape(-1, 1, 2)
    m = np.array(true_matrix, dtype=np.float32)

    projected_pts0 = cv2.perspectiveTransform(pts0, m)
    return np.linalg.norm(projected_pts0 - pts1, axis=-1).flatten()
```

Codefragment 2: Deze code illustreert hoe de fout van een match absoluut bepaald wordt.

Figuur 30, te vinden op pagina 66, toont visueel aan dat een coregistratie waarbij de gemiddelde fout van de hoekpunten, meer dan 100 pixels bedraagt, tot een onbetrouwbare coregistratie leidt. Deze figuur toont ook aan dat een coregistratie waarbij de gemiddelde fout van de hoekpunten, minder dan 50 pixels bedraagt, tot een betekenisvolle en bruikbare coregistratie leidt.

2.5 Modelconfiguratie en Hardware

Voor de lokale uitvoering van het model wordt de documentatie die te vinden is op Hugging Face gevolgd [30], waarbij er gebruikgemaakt wordt van het Hugging Face Transformers-framework. De preprocessor wordt aangeboden door Hugging Face. Aan de configuratie van deze preprocessor wordt niets veranderd. De preprocessor zal de inputafbeeldingen omzetten naar grijswaarden en herschalen naar een resolutie van 832x832. Het herschalen naar een resolutie van 832x832 werd al door de pipeline, uitgelegd in sectie 2.3, toegepast wanneer de data getransformeerd werd.

Het AI-model van de Hugging Face-repository heeft als standaardconfiguratie een coarse matching threshold-instelling van 20%. Om een maximaal aantal matches te bereiken, wordt deze drempelwaarde verlaagd naar 1%. Ook wanneer de matching gebeurt, wordt een drempelwaarde van 1% ingesteld in de post-process keypoint-matching-functie. Dit is omdat er niet per definitie vanuit gegaan kan worden dat matches met een hoge confidence score automatisch een lage fout impliceren, en vice versa. Een hogere drempelwaarde zou ertoe kunnen leiden dat een perfecte match die een lage confidence score heeft niet opgeslagen wordt in de resultaten. Het logische gevolg van deze lage drempelwaarde is dat er aanzienlijk veel meer foutieve matches behouden blijven. Dit nadeel weegt echter niet op tegen het risico om schaarse, uiterst correcte matches te verliezen.

Om de inferentie met het AI-model efficiënt te laten verlopen, wordt gekozen om gebruik te maken van Kaggle Notebooks. Dit platform biedt gratis de mogelijkheid om tot dertig uur per week (waarvan maximaal twaalf uur aaneengesloten) gebruik te maken van cloudgebaseerde GPU-inferentie [32]. Als hardware-accelerator is gekozen voor de duale NVIDIA T4 (T4 x2)-setup, aangezien deze beter geoptimaliseerd is voor AI-inferentietaken dan het P100-alternatief [33]. Voor fasen waar er geen AI-inferentie benodigd is, zoals data-analyse, kan gebruikgemaakt worden van de standaard CPU in Kaggle. Om Out-of-Memory-fouten te voorkomen, wordt het model bovendien omgezet van 32-bit floating-point (FP32) naar 16-bit floating-point (FP16) precisie. Dit wordt geïllustreerd in Codefragment 3.

```
self.config = AutoConfig.from_pretrained(
    "zju-community/matchanything_eloctr",
    trust_remote_code=True
)
self.config.coarse_matching_threshold = 0.01

self.model = AutoModelForKeypointMatching.from_pretrained(
    "zju-community/matchanything_eloctr",
    config=self.config,
    trust_remote_code=True,
    torch_dtype=torch.float16
).to(self.device)
self.model.eval()

self.processor = AutoImageProcessor.from_pretrained("zju-
community/matchanything_eloctr")
```

Codefragment 3: Dit illustreert hoe in Python-code het model geconfigureerd wordt.

2.6 Experimenteel Ontwerp

2.6.1 Random Search Verkenning

Een eerste executieve fase in het experiment omvat een random search. Het doel hiervan is een idee te krijgen van welke CLAHE-parameters het gunstigste effect zouden kunnen hebben op de modelprestaties voor elke individuele categorie.

Methodologie

Er wordt gebruikgemaakt van een round-robin-sampler. Deze zorgt ervoor dat per iteratie de categorieën willekeurig gerangschikt worden, waarna er per categorie een willekeurige afbeelding gekozen wordt. In een volgende iteratie kunnen alle reeds gebruikte afbeeldingen per categorie niet meer gekozen worden, totdat alle afbeeldingen uit die specifieke categorie eenmaal aan bod zijn geweest. Pas daarna is de volledige set van afbeeldingen van die categorie weer volledig beschikbaar. Hierdoor wordt gegarandeerd dat elke categorie even vaak gebruikt wordt en er geen onbalans in de evaluatie ontstaat.

Om het experiment strikt reproduceerbaar te houden, wordt gebruikgemaakt van een vastgestelde random seed. Deze random seed wordt gebruikt als instelling voor de volgende Python-modules: random, numpy, cv2 en torch.

Voor deze exploratieve fase is een set van potentiële CLAHE-parameters gedefinieerd. Voor de cliplimiet is gekozen voor de waarden 2.0, 5.0, 10.0, 20.0 en 40.0. Voor tegelrastergrootte is gekozen voor 4x4, 8x8, 16x16, 32x32 en 64x64. Deze waarden zijn doelbewust zo gekozen, omdat ze enerzijds beperkt in aantal zijn voor de haalbaarheid van de rekentijd, maar anderzijds voldoende divers in bereik zijn om een trend te kunnen waarnemen.

Per iteratie van het experiment wordt een nieuwe afbeelding uit de round-robin sampler genomen. Deze afbeelding ondergaat een willekeurige trapeziumvormige transformatie, waarbij de transformatiematrix wiskundig bepaald wordt op basis van de hoekpuntverplaatsingen. Op de getransformeerde afbeelding wordt een gaussiaanse ruis toegevoegd met een standaardafwijking van 30. Nadien wordt eenmaal een (baseline) matching-inferentie uitgevoerd zonder CLAHE toe te passen. Aan de matcher worden hiervoor de originele en de getransformeerde afbeelding met ruis gevoed. De output van deze baseline-inferentie bevat een lijst keypoint-coördinaten in de originele afbeelding, de bijbehorende keypoint-coördinaten in de getransformeerde afbeelding, en een lijst met bijhorende confidence scores.

Vervolgens wordt op exact hetzelfde paar van originele en getransformeerde afbeeldingen CLAHE toegepast. Hierbij wordt willekeurig een combinatie van de vooraf gedefinieerde CLAHE-parameters gekozen. Na deze beeldverbetering vindt een tweede matching-inferentie plaats. Dit wordt schematisch voorgesteld in Figuur 17. De output hiervan heeft dezelfde structuur als de baseline-output. Voor beide inferenties wordt een drempelwaarde (threshold) van 1% gehanteerd.

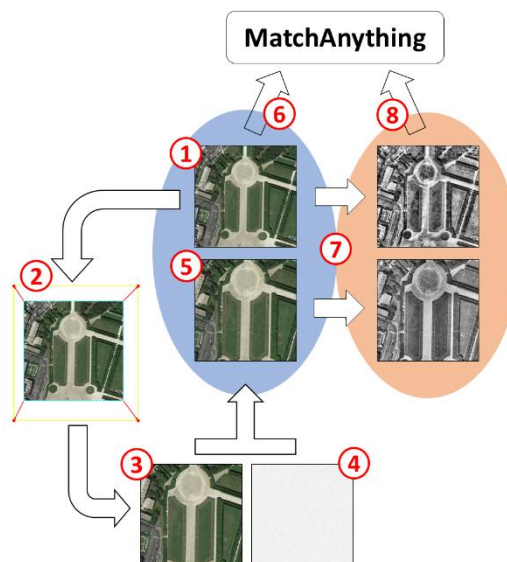
Aan het einde van het proces voor het verwerken van elke afbeelding wordt een nieuwe dataregel (lijn) weggeschreven naar een JSONL-bestand met de volgende attributen:

- Een timestamp
- De naam van de afbeelding
- De afbeeldingscategorie

Scenariospecifieke Contrastverbetering voor Satellietcoregistratie

- De looptijd van de iteratie
- De geometrische transformatiematrix
- De standaardafwijking van de gaussiaanse ruis
- De cliplimiet van de CLAHE-instelling
- De tegelrastergrootte van de CLAHE-instelling
- De lijst van keypoints op de originele afbeelding (zonder CLAHE)
- De lijst van voorspelde keypoints op de getransformeerde afbeelding (zonder CLAHE)
- De lijst van confidence scores van de matches (zonder CLAHE)
- De lijst van keypoints op de originele afbeelding (met CLAHE)
- De lijst van voorspelde keypoints op de getransformeerde afbeelding (met CLAHE)
- De lijst van confidence scores van de matches (met CLAHE)

Door telkens direct een JSONL-regel toe te voegen, is het outputbestand resistent tegen plots falen van het script. Omdat de executie via Kaggle Notebooks loopt, zal het script na de strikte limiet van twaalf uur aaneengesloten inferentie abrupt afgebroken worden. Dankzij het JSONL-formaat gaat er in dat geval echter geen verzamelde data verloren. Hoe dit in Python geïmplementeerd wordt, is zichtbaar in Codefragment 4.



Figuur 17: Schematische voorstelling van hoe na het voltooien van de data-pipeline (5) het afbeeldingenpaar wordt aangeleverd aan het AI-model (6). Na toepassing van CLAHE (7) wordt ook dit paar aangeleverd aan het AI-model (8)

```

engine = MatcherEngine(coarse_threshold=0.01)
image_generator = round_robin_image_sampler(args.data_dir)

...

matrix = generate_outward_trapezoid_matrix(w, h)
warped_img = transform_image(img, matrix)
warped_noisy = add_random_noise(warped_img, mean=0.0, std=noise_std)
results_raw = engine.match(img, warped_noisy)

clip = random.choice(clip_limits)
grid = random.choice(grid_sizes)
img_clahe = apply_clahe_rgb(img, clip, grid)
warped_noisy_clahe = apply_clahe_rgb(warped_noisy, clip, grid)
results_clahe = engine.match(img_clahe, warped_noisy_clahe)

record = {
    "timestamp": datetime.now().isoformat(),
    "image_name": os.path.basename(img_path),
    "class_name": class_name,
    "processing_time_sec": processing_time,
    "transformation": {
        "type": "outward_trapezoid",
        "matrix": matrix.tolist()
    },
    "noise_std": noise_std,
    "clahe_settings": {
        "clip_limit": clip,
        "grid_size": grid
    },
    "matches_raw": { ... },
    "matches_clahe": { ... }
}

f.write(json.dumps(record) + "\n")
f.flush()

```

Codefragment 4: Deze code illustreert hoe de random search wordt geïmplementeerd.

2.6.2 Analyse van de Random Search & Parameterselectie

Na afloop van de random search wordt voor elke iteratie de ruimtelijke fout (euclidische afstand) van de gevonden matches berekend (zie sectie 2.4). Omdat de fout van de match hierbij louter berekend wordt ten opzichte van de wiskundige grondwaarheid, is deze onafhankelijk van de door het model gegeven confidence score.

Hoewel het JSONL-formaat ideaal was voor persistente dataopslag, is het niet geoptimaliseerd voor een grootschalige data-analyse. Daarom wordt het JSONL-bestand in een eerste stap omgevormd tot Parquet-bestanden. Elke lijn in het originele JSONL-bestand bevat zowel data van matches met als zonder CLAHE. Belangrijk is dat deze in het Parquet-bestand gesplitst worden, waarbij de baseline-matches (zonder CLAHE) toch een referentie behouden naar de CLAHE-instelling van die specifieke iteratie. Dit voorkomt een bias. Het vergelijken van de geaggregeerde resultaten van alle baseline-matches met resultaten van slechts een specifieke CLAHE-instelling zou immers een scheefgetrokken beeld geven.

Er worden twee gescheiden Parquet-bestanden opgebouwd. Een eerste bestand aggregereert statistieken op afbeeldingsniveau. Hierin worden per iteratie de volgende gegevens bijgehouden:

- De categorie van de afbeelding
- De geteste CLAHE-instelling, met een addendum dat aanduidt of CLAHE werd toegepast of niet (“CLAHE” of “RAW”)
- Het totaal aantal gevonden matches
- De minimale, maximale en gemiddelde foutgrootte van de matches binnen de afbeelding
- De drie kwartielwaarden van de foutgroottes van de matches binnen de afbeelding

Een tweede Parquet-bestand houdt statistieken bij op match-niveau. Om Out-of-Memory-fouten te voorkomen tijdens de data-analyse in Kaggle, wordt hierbij een downsampling toegepast waarbij een op de vijf matches behouden blijft. In dit bestand worden per individuele match de volgende gegevens bijgehouden:

- De categorie van de afbeelding waartoe de match behoort
- De CLAHE-instelling die gebruikt werd bij de iteratie, met een addendum dat aanduidt of CLAHE werd toegepast of niet (“CLAHE” of “RAW”)
- De confidence score van de match
- De foutgrootte van de match

Er worden twee zaken onderzocht in deze analyse:

1. Zijn er correlaties tussen:
 - a. het aantal matches in een afbeelding;
 - b. de grootte van de fout van de matches;
 - c. de confidence scores van de matches?
2. Zijn er bepaalde CLAHE-instellingen die voor een gegeven categorie betere matches als resultaat hebben?

Om de correlaties te bepalen, wordt gebruikgemaakt van de Spearman-correlatie. In tegenstelling tot bijvoorbeeld de Pearson-correlatie, die een lineaire relatie veronderstelt, is Spearman geschikt voor monotoon stijgende of dalende relaties. De waarden hiervan liggen tussen -1 en 1, waar nul wijst op het ontbreken van enige correlatie. Naast de Spearman-correlatiecoëfficiënt wordt ook een plot weergegeven waarin de densiteit van de matches visueel geïnspecteerd kan worden.

Voor het selecteren van de best presterende CLAHE-instellingen per categorie worden densiteitsplots, histogrammen en cumulatieve plots gegenereerd. Deze visualisaties laten toe om de verdeling van foutgroottes en het totale aantal geslaagde versus gefaalde inferenties per instelling in kaart te brengen. Het inzichtelijk maken van de

gefaalde matching-inferenties (waarbij geen enkele match werd gevonden) is hierbij cruciaal om een survivorship bias te voorkomen. Een CLAHE-instelling wordt als “interessant genoeg” beschouwd om op te nemen in een grid search indien de plots, ten opzichte van de baseline, een daling in het aantal gefaalde inferenties weergeven, gecombineerd met een toename van het aantal matches met een lage fout en idealiter ook een afname van het aantal matches met een zeer hoge fout. In de cumulatieve plots vertaalt zich dit in een steile helling die snel afvlakt.

Dit zal in hoofdstuk 3 met de correlaties en densiteitsplots alsook met plots voor de kwaliteit van de matchings verduidelijkt worden. Deze plots zullen de verdeling van het aantal matches per afbeelding, per categorie en per CLAHE-instelling tonen. Dit zowel in een histogram (y-as links) als een cumulatieve plot (y-as rechts). Ook is af te leiden voor hoeveel afbeeldingen MatchAnything geen enkele match kon vinden. In deze plots wordt, met bijvoorbeeld $N=(104/178)$, aangegeven wat de verhouding is van het aantal geslaagde matching-inferenties (104) ten opzichte van het totaal aantal matching-inferenties (178). Hieruit kan ook het aantal gefaalde matching-inferenties bepaald worden ($178 - 104$). Het aantal gefaalde matching-inferenties is interessant om bewust te zijn van eventuele survivorship bias, in omstandigheden waar het aantal gefaalde matching-inferenties zeer hoog is. In een ideaal scenario is het aantal afbeeldingen waar geen enkele match voor gevonden werd zeer laag, en zijn er veel afbeeldingen waar veel matches voor gevonden worden.

Deze plots zeggen iets over het aantal matches per afbeelding, niets over de kwaliteit van de matches. Andere plots in hoofdstuk 3 tonen het verband tussen het aantal matches met een bepaalde fout per categorie, per CLAHE-instelling. Hier is een steile helling die vlug afvlakt wenselijk. Dit betekent dat er veel matches zijn met een lage fout, en weinig met een hoge fout.

Eenzelfde analyse kan gedaan worden op de confidence scores. Deze plots werden niet gebruikt als criteria om de grid search-parameters te selecteren. Dit kan wel interessant zijn, voor een volgend onderzoek waarin een RANSAC-methode die rekening houdt met de score (bijvoorbeeld PROSAC) gebruikt wordt.

Op basis van:

1. De plots van het verband tussen de verdeling van de grootte van de fouten en de CLAHE-instelling
2. het verband tussen het aantal gefaalde matching-inferenties met toepassing van de CLAHE-instelling en het aantal gefaalde matching-inferenties zonder toepassing van de CLAHE-instelling

werd voor elke categorie een aantal potentiële CLAHE-instellingen geselecteerd.

Er kan van de initiële veronderstelling uitgegaan worden dat deze op basis van de plots van de random search geselecteerde instellingen, geschikte kandidaten kunnen zijn om beter te presteren dan de ruwe matching-inferentie (zonder CLAHE).

2.6.3 Gerichte Grid Search en Validatie

Voor de grid search wordt hetzelfde stramien als voor de random search doorlopen, mits enkele aanpassingen. Er wordt gebruikgemaakt van dezelfde round-robin-sampler. De afbeeldingen ondergaan dezelfde willekeurige trapeziumvormige transformatie en krijgen op dezelfde wijze ruis toegevoegd. Deze transformatie en ruis zijn per iteratie constant voor elke afbeelding. Nu wordt echter per iteratie naast de ruwe matching (zonder CLAHE), ook elke CLAHE-instelling (uit de selectie voor bijbehorende categorie) toegepast per afbeelding. Dit wordt in Kaggle uitgevoerd met T4 x2 GPU als accelerator.

Voor het overige worden dezelfde instellingen gebruikt als in de random search voor de generatie van het grid search JSON-bestand.

Omdat nu voor elke afbeelding per iteratie een matching-inferentie zonder CLAHE en nadien een matching-inferentie met elke CLAHE-instelling gedaan wordt, kan, in tegenstelling tot bij de random search, wel elke CLAHE-instelling vergeleken worden met alle ruwe matching-inferenties per categorie. Hoe dit in Python-code geïmplementeerd wordt is te zien in Codefragment 5.

```

engine = MatcherEngine(coarse_threshold=0.01)
image_generator = round_robin_image_sampler(args.data_dir)

...

matrix = generate_outward_trapezoid_matrix(w, h)
warped_img = transform_image(img, matrix)
warped_noisy = add_random_noise(warped_img, mean=0.0, std=noise_std)
res_raw = engine.match(img, warped_noisy)

clahe_results = []
for setting in settings_to_test:
    clip = setting['clip']
    grid = setting['grid']

    img_c = apply_clahe_rgb(img, clip, grid)
    warped_c = apply_clahe_rgb(warped_noisy, clip, grid)

    res_c = engine.match(img_c, warped_c)

    clahe_results.append({
        "clip_limit": clip,
        "grid_size": list(grid),
        "keypoints0": get_tensor_as_list(res_c["keypoints0"]),
        "keypoints1": get_tensor_as_list(res_c["keypoints1"]),
        "confidence": get_tensor_as_list(res_c["matching_scores"])
    })

record = {
    "timestamp": datetime.now().isoformat(),
    "image_name": os.path.basename(img_path),
    "class_name": class_name,
    "processing_time_sec": processing_time,
    "transformation": {
        "type": "outward_trapezoid",
        "matrix": matrix.tolist()
    },
    "noise_std": noise_std,
    "matches_raw": { ... },
    "matches_clahe": clahe_results
}

f.write(json.dumps(record) + "\n")
f.flush()

```

Codefragment 5: Deze code illustreert hoe de grid search wordt geïmplementeerd.

3 Resultaten en Analyse

3.1 Fase 1: Random Search Verkenning

De random search, uit sectie 2.6.1, werd uitgevoerd in Kaggle Notebooks. Deze werd na twaalf uur afgebroken, nadat er meer dan 55.000 iteraties voltooid waren. Dit is ruwweg vijf maal de volledige dataset, al bevat niet elke categorie exact evenveel afbeeldingen.

Aangezien er voor zowel de cliplimiet als de tegelrastergrootte vijf opties waren gedefinieerd (wat resulteert in 25 unieke parametercombinaties), verspreid over dertig categorieën, komt dit ruwweg neer op een zeventigtal iteraties per categorie, per CLAHE-instelling. Dit vormt een voldoende representatieve steekproef om een

betrouwbare selectie van potentieel prestatieverhogende CLAHE-instellingen te kunnen maken voor de gerichte grid search uit sectie 2.6.3, zoals verder beschreven aan de hand van fase 3 in sectie 3.3.

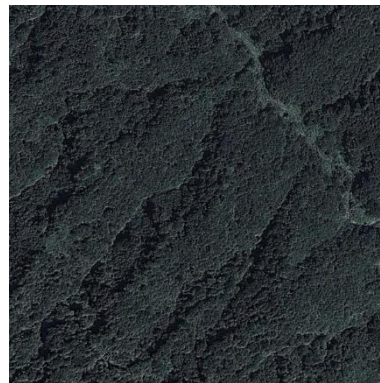
3.2 Fase 2: Data-Analyse en Parametersselectie

3.2.1 Algemene Correlaties en Moeilijkheidsgraad per Categorie

Uit de data-analyse blijkt een matige, negatieve correlatie (Spearman's $\rho = -0.249$) tussen de confidence score en de ruimtelijke fout van de matches over de volledige dataset (zie Figuur 31 op pagina 67). De correlatie is het kleinst voor de categorie *Playground* ($\rho = -0.124$) en het grootst voor de categorie *Forest* ($\rho = -0.319$) zoals geïllustreerd in Figuur 32 en Figuur 33, te vinden op pagina 68. In de titels van de correlatieplots wordt naast de Spearman-correlatie aangegeven hoeveel matches voorgesteld worden, bijvoorbeeld $N=58,528,422$. Afbeeldingen Figuur 18 en Figuur 19 hieronder zijn ter illustratie van deze categorieën.



Figuur 18: Voorbeeld van een afbeelding uit de AID-dataset met categorie *Playground*.



Figuur 19: Voorbeeld van een afbeelding uit de AID-dataset met categorie *Forest*.

Uit Figuur 34 en Figuur 35 op pagina 68 is ook af te leiden dat bepaalde categorieën tot veel meer matches leiden dan andere. Zo is te zien dat in de categorie *Church* het grootste aantal matches gevonden werd (2.425.315). In de categorie *Desert* werd het kleinste aantal matches gevonden (723.899). Dit suggereert dat woestijnlandschappen aanzienlijk moeilijker te matchen zijn voor het MatchAnything-model dan kerken. Figuur 20 en Figuur 21 zijn toegevoegd ter illustratie van deze categorieën.



Figuur 20: Voorbeeld van een afbeelding uit de AID-dataset met categorie Church.



Figuur 21: Voorbeeld van een afbeelding uit de AID-dataset met categorie Desert.

Algemeen gezien, dit wil zeggen over alle categorieën en CLAHE-instellingen heen, is er ook een matige correlatie tussen het aantal matches in een afbeelding en het eerste kwartiel en (in iets mindere mate) de mediaan van de fouten van de matches binnen die afbeeldingen (zie Figuur 36 en Figuur 37 op pagina 68). Het hebben van veel matches met een lage score is gewenst, omdat dan correcter en efficiënter een transformatiematrix bepaald zou kunnen worden. Omdat in veel gevallen RANSAC volgt na het bepalen van de matches om inliers te filteren, zijn veel matches met een lage fout gewenst. Hoe lager de mediaan of het eerste kwartiel, hoe groter de kans dat bijvoorbeeld een RANSAC-algoritme voldoende correcte matches heeft om tot een goede consensus te komen.

3.2.2 Kwantitatieve Analyse: Slagingspercentage en Matchvolume

Naast de correlaties en densiteitsplots is de kwaliteit van de matchings belangrijk. De plots over de kwaliteit van de matchings, zoals hierna besproken, tonen de verdeling van het aantal matches per afbeelding, per categorie, per CLAHE-instelling. Dit wordt zowel in een histogram (y-as links) als een cumulatieve plot (y-as rechts) weergegeven. In een ideaal scenario is het aantal afbeeldingen waar geen enkele match voor gevonden werd, met andere woorden het aantal gefaalde matches, zeer laag, en zijn er veel afbeeldingen waarvoor er veel matches gevonden worden.

Bij de categorie *Pond* is hier iets opvallend waarneembaar. Het toepassen van CLAHE leidt hier bijna altijd tot minder gefaalde matching-inferenties, maar zorgt wel voor een algemene daling van het aantal matches per afbeelding. Dit zie je in Figuur 38 op pagina 69 aan het aantal geslaagde inferenties en het zwaartepunt van het histogram. De piek van het aantal matches per inferentie van de inferenties waar geen CLAHE werd toegepast (blauwe histogram) ligt meer naar rechts dan de piek waar wel CLAHE op werd toegepast (oranje histogram).

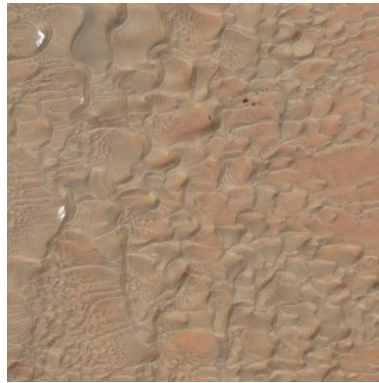
Voor de categorie *Desert* leidt het toepassen van CLAHE dan weer tot een algemene verhoging van het aantal gefaalde matching-inferenties. Dit is te zien in Figuur 39 op pagina 70, aangezien de cumulatieve plot bij nul matches per inferentie hoger is voor

inferenties waar CLAHE werd toegepast (oranje curve) dan waar geen CLAHE werd toegepast (blauwe curve).

Voor de categorie *DenseResidential* is het resultaat zeer afhankelijk van de CLAHE-instelling. Het merendeel van de CLAHE-instellingen leidt tot gelijkaardige resultaten, met of zonder CLAHE. Echter, voor de CLAHE-instelling met cliplimiet 2.0 en tegelrastergrootte 4x4 lijkt het erop dat er minder gefaalde matching-inferenties zijn bij met CLAHE dan zonder. Bij de CLAHE-instelling met cliplimiet 20.0 en tegelrastergrootte 64x64 lijkt het er juist op dat er meer gefaalde matching-inferenties zijn met CLAHE dan zonder CLAHE (zie Figuur 49 op pagina 78). Figuur 22, Figuur 23 en Figuur 24 zijn hieronder ter illustratie van deze categorieën toegevoegd.



Figuur 22: Voorbeeld van een afbeelding uit de AID-dataset met categorie *Pond*.



Figuur 23: Voorbeeld van een afbeelding uit de AID-dataset met categorie *Desert*.



Figuur 24: Voorbeeld van een afbeelding uit de AID-dataset met categorie *DenseResidential*.

3.2.3 Kwalitatieve Analyse: Nauwkeurigheid en Confidence Scores

Voor de categorie *Pond* lijken CLAHE-instellingen zoals (5.0, 4x4), (10.0, 8x8) of (10.0, 4x4) het beter te doen met CLAHE dan zonder. Dit omdat er meer matches zijn met een fout lager dan 50 pixels voor deze CLAHE-instellingen met CLAHE ten opzichte van de ruwe baseline (zie Figuur 40 op pagina 71).

Voor de categorie *Desert* lijkt het eerdergenoemde vermoeden bevestigd. Als CLAHE wordt toegepast, zijn er niet enkel meer gefaalde matching-inferenties en minder matches per afbeelding. Er zijn ook minder matches met een lage fout. De “minst slechte” CLAHE-instelling lijkt hier een met een cliplimiet van 2.0 en een tegelrastergrootte van 8x8, omdat deze het hoogste aantal lage fouten heeft en een steile daling maakt in de cumulatieve plot (zie Figuur 41 op pagina 72).

Voor de categorie *DenseResidential* lijkt er enkel voor de CLAHE-instellingen (2.0, 4x4) of (2.0, 8x8) een (lichte) verhoging te zijn in het aantal matches met een lage fout (zie Figuur 51 op pagina 79).

Eenzelfde analyse kan gedaan worden op de confidence scores. Deze plots werden echter niet gebruikt als criterium om de grid search-parameters te selecteren en zijn hier louter ter illustratie toegevoegd (zie Figuur 42, Figuur 43 en Figuur 44 op pagina's 73

tot en met 75). Echter kunnen deze plots interessant zijn voor verder onderzoek indien men een RANSAC-methode die rekening houdt met de score (bijvoorbeeld PROSAC) wenst te gebruiken.

3.2.4 Parameterselectie voor de Grid Search

Op basis van deze analyse werd voor elke categorie een selectie van minimaal één tot maximaal acht veelbelovende CLAHE-instellingen per categorie geselecteerd voor de volgende fase. Deze veelbelovende CLAHE-instellingen per categorie kunnen teruggevonden worden in Tabel 3, die verder behandeld wordt in sectie 3.5.

3.3 Fase 3: Gerichte Grid Search en Validatie

De grid search werd uitgevoerd op Kaggle in een notebook (zie sectie 2.6.3). Deze werd na meer dan acht uur en ruim 26.000 verwerkte afbeeldingen (875 afbeeldingen per categorie) automatisch stopgezet, omdat het maximum van 20GB opslag bereikt was voor deze notebook. Dit komt overeen met ongeveer 2,5 keer de complete dataset. Door het gebruik van het JSONL-formaat als bestandstype voor de output is deze persistent na het beëindigen van de notebook.

Hierbij vallen er een aantal zaken op.

Voor enkele categorieën, waarvan *Airport* een voorbeeld is, geeft de grid search niet het beoogde resultaat. Waar in de random search het aantal gefaalde matching-inferenties lager was voor CLAHE dan voor de baseline (zie Figuur 45 op pagina 76), is dit bij de grid search omgekeerd (zie Figuur 46 op pagina 76). Ook het aantal matches met een lage score voor deze categorie is in de grid search in tegenstelling tot in de random search lager voor matching-inferenties met CLAHE dan zonder CLAHE (zie Figuur 47 en Figuur 48 op pagina 77).

Voor de categorie *DenseResidential* is er in de grid search enkel een vermindering van het aantal gefaalde matching-inferenties ten opzichte van de ruwe matchings na toepassing van de CLAHE-instelling (2.0, 8x8). Voor de CLAHE-instellingen (2.0, 4x4) en (40.0, 4x4) leidt, in tegenstelling tot bij de random search, een toepassing van deze CLAHE-instellingen tot meer gefaalde matching-inferenties (zie Figuur 49 en Figuur 50 op pagina 78). Voor deze specifieke instellingen zijn er tevens opvallend minder matches met een kleine fout na toepassing van de CLAHE-instellingen ten opzichte van de ruwe matches. Voor de CLAHE-instelling (2.0, 8x8) is er een gelijkaardig aantal matches met een kleine fout voor matching-inferentie met of zonder CLAHE (zie Figuur 51 en Figuur 52 op pagina 79).

Voor de categorie *Pond* wordt duidelijk weerspiegeld wat de random search deed vermoeden. Voor elke CLAHE-instelling die onderzocht werd in de grid search (zie Figuur 53 op pagina 80) zijn er minder matching-inferenties die falen als CLAHE wordt toegepast ten opzichte van zonder CLAHE. Wel zorgt de toepassing van CLAHE voor een

meer uniforme verdeling van het aantal matches per afbeelding. Als deze matches beter zijn hoeft dit geen probleem te zijn. Een plot van de foutgroottes per CLAHE-instelling leert dat CLAHE-instelling (10.0, 8x8) meer matches vindt met een zeer lage score dan zonder CLAHE. Er zijn tevens minder matches met een hoge fout (zie Figuur 54 op pagina 80).

Voor de categorie *Beach* zijn er minder gefaalde matching-inferenties indien de CLAHE-instelling (5.0, 16x16) wordt toegepast dan zonder CLAHE (zie Figuur 55 en Figuur 56 op pagina 81). Hier komt echter als neveneffect dat de verdeling, indien alle gefaalde matching-inferenties achterwege gelaten worden, veranderde van linksscheef naar rechtsscheef. Dit vertaalt zich dan ook in de verdeling van de grootte van de fouten. Er zijn opvallend minder matches met CLAHE dan zonder, zo ook in het aantal met een lage fout (zie Figuur 57 en Figuur 58 op pagina 82).

3.4 Analyse Grid Search

Op basis van de grid search uit sectie 3.3 (waarbij steeds een trapeziumvormige transformatie en gaussiaanse ruis werden toegepast) is voor de categorie *Pond* een duidelijke verbetering te merken wanneer een CLAHE-instelling met een cliplimiet van 10.0 en een tegelrastergrootte van 8x8 wordt gebruikt (zie Figuur 52 op pagina 79). Er is een groter aantal matches waarvan de fout lager is dan 50 pixels, en een lager aantal matches waarvan de fout hoger is dan 150 pixels in vergelijking met dezelfde afbeeldingen zonder CLAHE. Ook zijn er minder matching-inferenties waarbij er geen enkele match gevonden kon worden. Een kanttekening is dat er over het algemeen wel minder matches per afbeelding gevonden worden met deze CLAHE-instelling, maar de gevonden matches zijn kwalitatief beter. De resultaten van de grid search uit sectie 3.3 lijken ook positief voor de categorieën *DenseResidential* (zie Figuur 50 en Figuur 52 op pagina 78), *Industrial*, *MediumResidential*, *Resort* en *Square*.

Een belangrijke opmerking is dat grafieken genoemd in sectie 3.3 enkel de verdeling van het aantal matches in vergelijking met hun fout tonen, binnen elke categorie. Er kan hierdoor iets gezegd worden over hoeveel matches binnen een categorie een fout hebben die lager is dan 50 pixels. Veel goede matches per categorie is positief voor een goed presterend MatchAnything-model. Dit zegt echter nog niets over het gemiddelde aantal matches met een lage fout per individuele afbeelding. Als de toepassing van een specifieke CLAHE-instelling ervoor zou zorgen dat het aantal gefaalde matching-inferenties daalt, impliceert dit dat het aantal geslaagde inferenties zou stijgen. Afhankelijk van de kwaliteit van de matches binnen deze “nieuwe geslaagde” inferenties, kan dit een vertekend beeld geven van het gemiddeld aantal matches van alle geslaagde matching-inferenties. Om deze invloed van CLAHE correct te bestuderen, wordt in de volgende sectie gekeken naar harde statistieken: Inliers Ratio, Mean Abs Inliers & Failure Rate.

3.5 Inliers Ratio, Mean Abs Inliers & Failure Rate

Berekeningen omtrent Inliers Ratio, Mean Abs Inliers & Failure Rate worden weergegeven in Tabel 3, te vinden op pagina 61. Ter verduidelijking van de kolommen van Tabel 3 is Tabel 1 bijgevoegd.

Tabel 1: Beschrijving van de kolommen van Tabel 3.

Category	De betreffende AID-afbeeldingscategorie
CLAHE	De geteste CLAHE-instelling (cliplimiet & tegelrastergrootte)
Inlier Ratio % RAW	De relatieve hoeveelheid matches met een fout lager dan 50 pixels, ten opzichte van alle matches binnen de categorie, voor de ruwe baseline-inferenties (zonder CLAHE).
Inlier Ratio % CLAHE	De relatieve hoeveelheid matches met een fout lager dan 50 pixels, ten opzichte van alle matches binnen de categorie, voor inferenties met de betreffende CLAHE-instelling.
Mean Abs Inliers RAW	De gemiddelde absolute hoeveelheid matches met een fout lager dan 50 pixels per matching-inferentie (inclusief gefaalde matching-inferenties) binnen de categorie, voor de ruwe baseline-inferenties (zonder CLAHE).
Mean Abs Inliers CLAHE	De gemiddelde absolute hoeveelheid matches met een fout lager dan 50 pixels per matching-inferentie (inclusief gefaalde matching-inferenties) binnen de categorie, met de betreffende CLAHE-instelling.
Failure Rate % RAW	De relatieve hoeveelheid matching-inferenties die geen enkele match vonden ten opzichte van het totaal aantal matching-inferenties binnen de categorie, voor de ruwe baseline-inferenties (zonder CLAHE).
Failure Rate % CLAHE	De relatieve hoeveelheid matching-inferenties die geen enkele match vinden ten opzichte van het totaal aantal matching-inferenties binnen de categorie, met de betreffende CLAHE-instelling.

Codefragment 6 verduidelijkt hoe de inliers en gefaalde inferenties kwantitatief bepaald werden met behulp van Pandas-aggregatie.

```
df_match['is_inlier'] = df_match['Error'] <= threshold
inlier_stats = df_match.groupby(['Category', 'Setting'], observed=False).agg(
    total_matches=('Error', 'count'),
    total_inliers=('is_inlier', 'sum')
).reset_index()

df_img['is_failed'] = df_img['Amount'] == 0
img_stats = df_img.groupby(['Category', 'Setting'], observed=False).agg(
    total_images=('Amount', 'count'),
    failed_images=('is_failed', 'sum')
).reset_index()
```

Codefragment 6: Deze code illustreert de bepaling van inliers en gefaalde inferenties.

De resultaten uit Tabel 3, te vinden op pagina 61, leren dat de categorie *Pond* ook hier echt wel baat heeft bij het toevoegen van de CLAHE-instelling met cliplimiet 10.0 en tegelrastergrootte 8x8. Naast een stijging in de procentuele inlier ratio, stijgt ook het gemiddelde absolute aantal inliers na toevoeging van de CLAHE-instelling. Ook daalt het aantal gefaalde matching-inferenties drastisch ten opzichte van de baseline-inferenties.

Voor de categorieën *BareLand*, *Beach*, *Bridge*, *Desert*, *Farmland*, *Meadow* en *Playground* stijgt de procentuele inlier ratio (licht) en daalt het aantal gefaalde matching-inferenties na toevoeging van een specifieke CLAHE-instelling. Echter, voor deze categorieën daalt ook het gemiddelde absolute aantal inliers na toevoeging van deze CLAHE-instelling. Dit betekent dat er in absolute zin minder matches gevonden worden, maar dat de relatieve kwaliteit van deze behouden matches hoger ligt.

Voor de categorieën *MediumResidential* en *Square* stijgt het gemiddelde absolute aantal inliers (licht) en daalt het aantal gefaalde matching-inferenties na toevoeging van een bepaalde CLAHE-instelling. Echter, voor deze categorieën daalt de procentuele inlier ratio na toevoeging van deze CLAHE-instelling. Dit betekent dat naast een daling van het aantal gefaalde matching-inferenties, de absolute kwantiteit van de matches stijgt (zowel voor matches met lage als hoge fouten), maar dat de kwaliteit van matches van afbeeldingen binnen deze categorieën proportioneel afneemt.

3.6 Statistische Validatie

Om te kunnen bepalen of de genoemde verbeteringen door bepaalde CLAHE-instellingen significant zijn, moeten er statistische testen uitgevoerd worden. Om de numerieke waarden “inlier ratio” en “mean abs inliers” te toetsen, wordt gebruikgemaakt van de Wilcoxon signed-rank test [34]. Dit is te verantwoorden omdat het om gepaarde, continue, ordinale data gaat die niet-parametrisch is. Om de “failure rate” te toetsen wordt de McNemar-test gebruikt [35]. Dit is te verantwoorden omdat het om gepaarde, dichotome data gaat die niet-parametrisch is. Python-implementatie van de McNemar-test is te zien in Codefragment 7.

```

if not raw_failed and not clahe_failed:
    dp["a"] += 1
elif raw_failed and not clahe_failed:
    dp["b"] += 1
elif not raw_failed and clahe_failed:
    dp["c"] += 1
elif raw_failed and clahe_failed:
    dp["d"] += 1

table = [[dp["a"], dp["b"]],
         [dp["c"], dp["d"]]]

exact_calc = True if (b + c) < 25 else False
mc_res = mcnemar(table, exact=exact_calc, correction=True)

```

Codefragment 7: Deze code illustreert hoe de McNemar-test wordt opgesteld.

De resultaten van deze statistische testen kunnen gevonden worden in Tabel 4, te vinden op pagina 63. Hierbij wordt een significantieniveau van $p < 0,05$ gehanteerd.

Uit deze resultaten blijkt dat de categorie *Pond (10.0, 8x8)* significant voordeel heeft op het vlak van kwaliteit (inlier ratio) en falingspercentage (failure rate) bij matching-inferentie met toepassing van CLAHE. Daarbovenop heeft het geen significant nadeel ten opzichte van matching-inferentie zonder toepassing van CLAHE op het vlak van het absolute aantal inliers.

Daarnaast blijkt dat de categorieën *BareLand (2.0, 8x8)*, *Beach (5.0, 16x16)*, *Bridge (40.0, 32x32)*, *Desert (2.0, 8x8)*, *Farmland (40.0, 16x16)* en *Meadow (2.0, 16x16)* significant voordeel hebben op het vlak van kwaliteit en falingspercentage bij matching-inferentie met toepassing van CLAHE. Daartegenover staat dan wel dat deze, in tegenstelling tot de categorie *Pond (10.0, 8x8)*, een significant nadeel hebben ten opzichte van matching-inferentie zonder toepassing van CLAHE op het vlak van het absolute aantal inliers.

De categorie *MediumResidential (2.0, 4x4)* heeft een significant voordeel op het vlak van falingspercentage bij matching-inferentie met toepassing van CLAHE. Daarbovenop heeft deze categorie geen significant nadeel ten opzichte van matching-inferentie zonder toepassing van CLAHE op het vlak van kwaliteit en het absolute aantal inliers.

De categorie *River (20.0, 4x4)* heeft een significant voordeel op het vlak van falingspercentage bij matching-inferentie met toepassing van CLAHE. Daarbovenop heeft deze categorie geen significant nadeel ten opzichte van matching-inferentie zonder toepassing van CLAHE op het vlak van kwaliteit. Deze categorie heeft wel een significant nadeel ten opzichte van matching-inferentie zonder toepassing van CLAHE op het vlak van absolute aantal inliers.

Voor alle overige onderzochte categorieën (*Airport, BaseballField, Center, Church, Commercial, DenseResidential, Forest, Industrial, Mountain, Park, Parking, Playground, Port, RailwayStation, Resort, School, SparseResidential, Square, Stadium, StorageTanks* en *Viaduct*) levert de toevoeging van CLAHE, althans voor de instellingen die in de grid

Scenariospecifieke Contrastverbetering voor Satellietcoregistratie

search opgenomen werden, geen significant voordeel op voor minstens een van de drie evaluatievlakken (kwaliteit, het absolute aantal inliers of falingspercentage).

Integendeel, vaak resulteert de toepassing van CLAHE voor deze categorieën juist in een significant nadeel.

Conclusie

Aanvaarding van de Hypothese

De hypothese dat per categorie een optimale CLAHE-instelling gevonden kan worden die, in vergelijking met matching-inferenties zonder CLAHE, resulteert in meer matches met een lage fout en minder gefaalde inferenties, kan deels bevestigd worden.

Voor de categorie *Pond* is het duidelijk dat de hypothese unaniem aanvaard kan worden. Voor diverse categorieën (*BareLand*, *Beach*, enzovoort) geldt dat de hypothese kan worden aanvaard wat betreft het verhogen van de ratio van matches met een lage fout en het verlagen van de falingskans van de inferentie. Voor de categorieën *MediumResidential* en *River* kan de hypothese enkel aanvaard worden voor het verlagen van de falingskans van de inferentie.

Scenariospecifieke Contrastverbetering voor Satellietcoregistratie

Tabel 2 geeft weer welke CLAHE-instellingen het best gebruikt worden per categorie. Categorieën waar de hypothese voor verworpen wordt, zijn aangeduid met “/”.

Tabel 2: Beste CLAHE-instellingen per AID-datasetcategorie.

Categorie	clipLimit	tileGridSize
<i>Airport</i>	/	/
<i>BareLand</i>	2.0	8x8
<i>BaseballField</i>	/	/
<i>Beach</i>	5.0	16x16
<i>Bridge</i>	40.0	32x32
<i>Center</i>	/	/
<i>Church</i>	/	/
<i>Commercial</i>	/	/
<i>DenseResidential</i>	/	/
<i>Desert</i>	2.0	8x8
<i>Farmland</i>	40.0	16x16
<i>Forest</i>	/	/
<i>Industrial</i>	/	/
<i>Meadow</i>	2.0	16x16
<i>MediumResidential</i>	2.0	4x4
<i>Mountain</i>	/	/
<i>Park</i>	/	/
<i>Parking</i>	/	/
<i>Playground</i>	/	/
<i>Pond</i>	10.0	8x8
<i>Port</i>	/	/
<i>RailwayStation</i>	/	/
<i>Resort</i>	/	/
<i>River</i>	20.0	4x4
<i>School</i>	/	/
<i>SparseResidential</i>	/	/
<i>Square</i>	/	/
<i>Stadium</i>	/	/
<i>StorageTanks</i>	/	/
<i>Viaduct</i>	/	/

Er kan dus besloten worden dat voor negen van de dertig categorieën een CLAHE-instelling gevonden kan worden die een significante prestatieverbetering tot resultaat heeft ten opzichte van een inferentie zonder CLAHE. Van deze negen categorieën tonen er zeven een significante verbetering in zowel de ratio van het aantal matches met een fout lager dan 50 pixels, als in het slagingspercentage van de inferentie. Hierbij dient men echter wel in het achterhoofd te houden dat het absolute aantal matches na de toepassing van CLAHE mogelijk kan dalen.

Mogelijke Verklaring

Een hoge cliplimiet (≥ 10.0) zorgt ervoor dat CLAHE de lokale contrasten extreem zal vergroten, met het risico dat de ruis ook versterkt wordt. Een lage cliplimiet (< 10.0) zorgt ervoor dat CLAHE contrast subtieler zal vergroten, zonder dat de ruis ook versterkt

Conclusie

wordt. Een grote tegelrastergrootte ($> 8 \times 8$) zorgt voor veel kleine tegels, waardoor het contrast zeer lokaal bepaald wordt. Een kleine tegelrastergrootte ($< 8 \times 8$) zorgt voor een klein aantal grote tegels, waardoor het contrast eerder globaal bepaald wordt.

Voor de categorie *Pond* is, met de instelling (10.0, 8×8), een hoge cliplimiet gewenst. Dit is wellicht omdat een vijver een groot wateroppervlak is met weinig tot geen structuur. Door de toegevoegde ruis kan het zijn dat de weinige structuur die het water origineel had, verkeerd gematcht wordt met vegetatie. Door toevoeging van de hoge cliplimiet wordt de willekeurige structuur op het wateroppervlak extremer, waardoor het meer verschilt van zaken die effectief structuur hebben, zoals vegetatie. Er worden minder grote foute matches gemaakt tussen water en vegetatie (zie Figuur 59 en Figuur 60 op pagina 83).

Voor de categorieën *Desert* en *BareLand* is, met de instelling (2.0, 8×8), een lage cliplimiet optimaal. Doordat het zanderige oppervlak al bestaat uit “natuurlijke ruis”, zijn er niet veel harde lijnen zichtbaar. Door hier een lage cliplimiet te nemen, wordt het contrast van de structuur verhoogd, zonder valse structuren, geïntroduceerd door de gaussiaanse ruis, op te dringen (zie Figuur 61 en Figuur 62 op pagina 84).

Voor de categorie *Beach* is, met de CLAHE-instelling (5.0, 16×16), een cliplimiet gekozen die tussen de 10.0 van de *Pond*-categorie en de 2.0 van de zanderige categorieën in ligt. Dit is logisch omdat de *Beach*-categorie bestaat uit strand en zee. Een grote tegelrastergrootte zorgt ervoor dat de lokale elementen zoals golven extra contrast krijgen, wat resulteert in minder foutieve matches. (zie Figuur 63 en Figuur 64 op pagina 85).

Bij de categorie *Bridge*, met de CLAHE-instelling (40.0, 32×32), is er een combinatie van water en wegdek zichtbaar. Het water heeft wel reliëf, terwijl het wegdek dat niet heeft. Hierdoor is het kiezen van een grote cliplimiet interessant. Hierdoor zal de natuurlijke golving van het water vergroot worden, terwijl op het wegdek enkel willekeurige ruis (zonder structuur) vergroot zal worden. Het aantal grote foute matches tussen water en wegdek vermindert hierdoor. Een hoge tegelrastergrootte heeft als effect dat scherpe randen lokaal vergroot worden. Een gelijkaardige redenering kan gemaakt worden voor de categorie *Farmland*, met de CLAHE-instellingen (40.0, 16×16). Ook hier zijn duidelijk afgelijnde vlakken zichtbaar, met verschillende samenstellingen, gewasafhankelijk, per vlak. Door het toepassen van de CLAHE worden er minder grote fouten tussen verschillende vlakken gevonden (zie Figuur 65, Figuur 66, Figuur 67 en Figuur 68 op pagina 86 en 87).

Voor de categorie *River*, met de CLAHE-instelling (20.0, 4×4), en de categorie *MediumResidential*, met de CLAHE-instelling (2.0, 4×4), is er enkel een significante daling van het aantal gefaalde matching-inferenties waarneembaar. Door grote tegels te kiezen, wordt hierop globale schaal het contrast verhoogd zonder in te zoomen op

lokale details. Voor *River* heeft een grote cliplimiet het voordeel dat het contrast van eventuele onderliggende structuren van het water vergroot kunnen worden. Voor *MediumResidential* zorgt een lage cliplimiet voor subtiele contrastverhoging, zonder gaussiaanse ruis op menselijke structuren te vergroten.

Codefragment 8: Om de randen van de getransformeerde afbeelding virtueel voor te stellen op de originele afbeelding, wordt de inverse perspectiefmatrix berekend. Dit wordt gebruikt in afbeeldingen Figuur 59 tot en met Figuur 68. Codefragment 8 verduidelijkt hoe de inhoud van de afbeeldingen Figuur 59 tot en met Figuur 68 tot stand kwam.

```
m_float = matrix.astype(np.float32)
m_inv = np.linalg.inv(m_float)
corners_img1 = np.array([[0, 0], [w2, 0], [w2, h2], [0, h2]],
dtype=np.float32).reshape(-1, 1, 2)
proj_1_in_0 = cv2.perspectiveTransform(corners_img1, m_inv).reshape(-1, 2)
poly_1_in_0 = np.vstack((proj_1_in_0, proj_1_in_0[0]))

ax.plot(poly_1_in_0[:, 0], poly_1_in_0[:, 1], color='cyan', linewidth=2.5, linestyle='--',
label='Transformed Canvas Bounds')
```

Codefragment 8: Om de randen van de getransformeerde afbeelding virtueel voor te stellen op de originele afbeelding, wordt de inverse perspectiefmatrix berekend. Dit wordt gebruikt in afbeeldingen Figuur 59 tot en met Figuur 68.

Andere categorieën kunnen onderverdeeld worden in twee groepen. Enerzijds zijn er de menselijke categorieën zoals *Airport*, *Church* en *DenseResidential*. Anderzijds de natuurlijke categorieën zoals *Forest* of *Mountain*. Figuur 25, Figuur 26, Figuur 27, Figuur 28 en Figuur 29 werden ter illustratie toegevoegd.



Figuur 25: Voorbeeld van een afbeelding uit de AID-dataset met categorie *Airport*.



Figuur 26: Voorbeeld van een afbeelding uit de AID-dataset met categorie *Church*.



Figuur 27: Voorbeeld van een afbeelding uit de AID-dataset met categorie *DenseResidential*.



Figuur 28: Voorbeeld van een afbeelding uit de AID-dataset met categorie *Forest*.



Figuur 29: Voorbeeld van een afbeelding uit de AID-dataset met categorie *Mountain*.

De menselijke categorieën zijn vaak zeer repetitief of bestaan net uit zeer veel egale vlakken zoals beton. De gaussiaanse ruis die in het experiment werd toegevoegd, bracht reliëf aan op egale vlakken, waar niet-bestaande structuren op werden gecreëerd.

De natuurlijke categorieën die we bekeken hebben zijn *Mountain* en *Forest*. Berglandschappen hebben, zeker boven de boomgrens, een structuur die lijkt op die van asfalt. Grote egale vlakken met scherpe randen. Door hier nog eens gaussiaanse ruis aan toe te voegen, krijg je hier hetzelfde effect als voor de menselijke categorieën. Berglandschappen onder de boomgrens lijken dan weer op boslandschappen. Categorie *Forest* heeft weinig onderliggende structuur, in tegenstelling tot bijvoorbeeld *Desert*. In woestijnlandschappen zorgt wind voor een duidelijk patroon in de zandkorrels. Bij boslandschappen zijn het allemaal bomen, vaak van dezelfde soort, met kleine bladeren. Hierdoor kan enkel ruis verhoogd worden met CLAHE, omdat onderliggende structuren zo goed als onbestaand zijn. Een kanttekening bij de categorieën *Forest* en *Mountain* is dat door de resultaten van de random search, bij de grid search slechts een beperkt aantal CLAHE-instellingen afgetoetst werd.

Bronnenlijst

- [1] “MerlijnDumarey/satellite-matching-clahe-eval: Source code for my AI-Engineering thesis: An evaluation of CLAHE preprocessing for cross-temporal satellite coregistration with MatchAnything.” Accessed: May 30, 2026. [Online]. Available: <https://github.com/MerlijnDumarey/satellite-matching-clahe-eval>
- [2] *Google AI Studio (Gemini 3.1 Pro)*. (2026). Google. [Online]. Available: <https://aistudio.google.com/>
- [3] “What is coregistration in remote sensing? | by Elise Colin | Medium.” Accessed: May 05, 2026. [Online]. Available: <https://elisecolin.medium.com/what-is-coregistration-in-remote-sensing-7d76d48d337d>
- [4] X. He *et al.*, “MatchAnything: Universal Cross-Modality Image Matching with Large-Scale Pre-Training,” Jan. 13, 2025, *arXiv*: arXiv:2501.07556. doi: 10.48550/arXiv.2501.07556.
- [5] Google, “Luchtfoto van de Burj Khalifa, 25°11’49.91”N 55°16’27.76”O,” Nov. 04, 2025.
- [6] Google, “Luchtfoto van de Burj Khalifa, 25°11’49.91”N 55°16’27.76”O,” Jun. 06, 2025.
- [7] Google, “Luchtfoto van Campus Brugge Station (Howest), 51°11’40.83”N 3°12’58.19”O,” Jul. 19, 2022.
- [8] Google, “Luchtfoto van Campus Brugge Station (Howest), 51°11’40.83”N 3°12’58.19”O,” Aug. 23, 2025.
- [9] “Why Orthorectification is Key for Real-World Terrain Mapping and How To Use It,” Geoawesome. Accessed: May 26, 2026. [Online]. Available: <https://geoawesome.com/eo-hub/why-orthorectification-is-key-for-real-world-terrain-mapping-and-how-to-use-it/>
- [10] Intermap Technologies, *Perspective Distortion*. Accessed: May 30, 2026. [Online]. Available: <https://www.intermap.com/hs-fs/hubfs/image4-1.png?width=999&name=image4-1.png>
- [11] “Hugging Face Learn - Computer Vision.” Accessed: Apr. 04, 2026. [Online]. Available: <https://huggingface.co/learn/computer-vision-course>
- [12] G. Nanos, “Computer Vision: Differences Between Low-Level and High-Level Features | Baeldung on Computer Science.” Accessed: May 05, 2026. [Online]. Available: <https://www.baeldung.com/cs/cv-low-vs-high-level-features>
- [13] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, “LoFTR: Detector-Free Local Feature Matching with Transformers,” Apr. 01, 2021, *arXiv*: arXiv:2104.00680. doi: 10.48550/arXiv.2104.00680.
- [14] Y. Wang, X. He, S. Peng, D. Tan, and X. Zhou, “Efficient LoFTR: Semi-Dense Local Feature Matching with Sparse-Like Speed,” Mar. 11, 2024, *arXiv*: arXiv:2403.04765. doi: 10.48550/arXiv.2403.04765.
- [15] “zju-community/matchanything_eliftr at main.” Accessed: Mar. 21, 2026. [Online]. Available: https://huggingface.co/zju-community/matchanything_eliftr/tree/main
- [16] “MatchAnything.” Accessed: Mar. 21, 2026. [Online]. Available: <https://zju3dv.github.io/MatchAnything/>
- [17] “MatchAnything - a Hugging Face Space by LittleFrog.” Accessed: Mar. 21, 2026. [Online]. Available: <https://huggingface.co/spaces/LittleFrog/MatchAnything>

- [18] “Histogram equalization,” *Wikipedia*. May 11, 2026. Accessed: May 30, 2026. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Histogram_equalization&oldid=1353585859
- [19] A. Mishra, “Contrast Limited Adaptive Histogram Equalization (CLAHE) Approach for Enhancement of the Microstructures of Friction Stir Welded Joints,” Aug. 15, 2021, *arXiv*: arXiv:2109.00886. doi: 10.48550/arXiv.2109.00886.
- [20] “OpenCV: Histograms - 2: Histogram Equalization.” Accessed: May 05, 2026. [Online]. Available: https://docs.opencv.org/4.x/d5/daf/tutorial_py_histogram_equalization.html
- [21] R. Ebker, “Random Sample Consensus Explained | Baeldung on Computer Science.” Accessed: May 05, 2026. [Online]. Available: <https://www.baeldung.com/cs/ransac>
- [22] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981, doi: 10.1145/358669.358692.
- [23] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J.-M. Frahm, “USAC: A Universal Framework for Random Sample Consensus,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 2022–2038, Aug. 2013, doi: 10.1109/TPAMI.2012.257.
- [24] D. Barath, J. Matas, and J. Noskova, “MAGSAC: Marginalizing Sample Consensus,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 10189–10197. doi: 10.1109/CVPR.2019.01044.
- [25] O. Chum and J. Matas, “Matching with PROSAC — Progressive Sample Consensus,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, San Diego, CA, USA: IEEE, 2005, pp. 220–226. doi: 10.1109/CVPR.2005.221.
- [26] A. Elashry, B. Sluis, and C. Toth, “IMPROVING RANSAC FEATURE MATCHING BASED ON GEOMETRIC RELATION,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLIII-B2-2021, pp. 321–327, Jun. 2021, doi: 10.5194/isprs-archives-XLIII-B2-2021-321-2021.
- [27] “AID: A Benchmark Dataset for Performance Evaluation of Aerial Scene Classification.” Accessed: May 08, 2026. [Online]. Available: <https://captain-whu.github.io/AID/>
- [28] “GPS.gov: GPS Accuracy.” Accessed: May 31, 2026. [Online]. Available: <https://archive.gps.gov/systems/gps/performance/accuracy/>
- [29] “AID: A scene classification dataset.” Accessed: May 08, 2026. [Online]. Available: <https://www.kaggle.com/datasets/jiayuanchengala/aid-scene-classification-datasets>
- [30] “zju-community/matchanything_eloctr · Hugging Face.” Accessed: Mar. 21, 2026. [Online]. Available: https://huggingface.co/zju-community/matchanything_eloctr
- [31] Deepanraj, “RANSAC,” Medium. Accessed: May 31, 2026. [Online]. Available: <https://medium.com/@717821i208/ransac-3187c732ac5d>
- [32] “Notebooks Documentation.” Accessed: May 31, 2026. [Online]. Available: <https://www.kaggle.com/docs/notebooks>

- [33] A. Rukshar, “LinkedIn: Kaggle Accelerators: A Comparison.” Accessed: May 08, 2026. [Online]. Available: <https://www.linkedin.com/pulse/kaggle-accelerators-comparison-rukshar-alam-ki9bc/>
- [34] “Wilcoxon Signed Rank Test: Definition, How to Run, SPSS - Statistics How To.” Accessed: May 28, 2026. [Online]. Available: <https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/wilcoxon-signed-rank-test/>
- [35] “McNemar Test Definition, Examples, Calculation - Statistics How To.” Accessed: May 28, 2026. [Online]. Available: <https://www.statisticshowto.com/mcnemar-test/>

Appendix 1: Tabellen

Appendix 1: Tabellen

Tabel 3: Resultaat van de berekeningen omtrent Inliers Ratio, Mean Abs Inliers & Failure Rate per categorie, per CLAHE-instelling.

Category	CLAHE	Inlier	Inlier	Mean	Mean	Failure	Failure
		Ratio	Ratio	Abs	Abs	Rate	Rate %
		%	%	Inliers	Inliers	%	CLAHE
		RAW	CLAHE	RAW	CLAHE	RAW	
Airport	C:2,0 G:4x4	31,30	30,44	204,80	169,83	42,06	48,91
	C:2,0 G:8x8	31,30	29,30	204,80	169,93	42,06	46,86
	C:20,0 G:8x8	31,30	29,77	204,80	156,37	42,06	48,00
	C:10,0 G:8x8	31,30	28,36	204,80	150,54	42,06	47,31
BareLand	C:2,0 G:8x8	20,17	21,09	105,44	77,40	38,51	32,00
BaseballField	C:2,0 G:8x8	23,17	21,73	128,15	107,06	48,69	50,74
	C:2,0 G:4x4	23,17	21,02	128,15	110,92	48,69	47,89
	C:2,0 G:32x32	23,17	19,88	128,15	100,89	48,69	47,31
Beach	C:5,0 G:16x16	19,19	21,95	97,07	57,55	36,69	22,51
Bridge	C:40,0 G:32x32	19,16	20,04	101,73	69,99	44,57	35,09
	C:10,0 G:4x4	19,16	21,19	101,73	89,25	44,57	39,54
	C:20,0 G:4x4	19,16	20,61	101,73	91,37	44,57	36,00
	C:20,0 G:8x8	19,16	21,08	101,73	84,42	44,57	38,06
	C:5,0 G:16x16	19,16	21,56	101,73	84,00	44,57	39,43
Center	C:10,0 G:32x32	31,37	25,90	190,20	140,60	46,63	46,63
	C:10,0 G:4x4	31,37	29,28	190,20	173,01	46,63	45,94
Church	C:2,0 G:4x4	39,89	38,75	269,76	257,25	42,40	42,97
	C:2,0 G:8x8	39,89	37,79	269,76	246,38	42,40	43,89
	C:40,0 G:16x16	39,89	34,73	269,76	222,47	42,40	43,54
	C:40,0 G:4x4	39,89	36,57	269,76	234,60	42,40	44,46
Commercial	C:40,0 G:4x4	40,24	37,14	261,21	224,24	44,91	47,09
	C:5,0 G:4x4	40,24	37,40	261,21	238,86	44,91	44,46
	C:2,0 G:32x32	40,24	35,87	261,21	228,23	44,91	44,91
DenseResidential	C:2,0 G:4x4	46,67	44,90	321,72	303,15	41,60	42,63
	C:2,0 G:8x8	46,67	44,22	321,72	321,17	41,60	37,94
	C:40,0 G:4x4	46,67	43,00	321,72	270,51	41,60	45,94
Desert	C:2,0 G:8x8	14,58	19,08	47,69	29,93	25,49	14,06
Farmland	C:40,0 G:16x16	20,40	21,41	110,95	66,45	45,60	32,69
	C:2,0 G:4x4	20,40	18,93	110,95	85,16	45,60	44,46
Forest	C:20,0 G:16x16	25,26	18,14	164,18	63,35	40,00	25,26
Industrial	C:20,0 G:8x8	40,19	36,10	272,12	226,25	42,40	45,26
	C:2,0 G:8x8	40,19	38,40	272,12	261,44	42,40	41,60
	C:40,0 G:4x4	40,19	37,02	272,12	238,68	42,40	44,23
	C:5,0 G:32x32	40,19	36,12	272,12	215,63	42,40	46,86
	C:40,0 G:8x8	40,19	37,02	272,12	243,45	42,40	42,63
Meadow	C:2,0 G:16x16	10,15	10,19	38,03	16,37	29,37	14,06
MediumResidential	C:10,0 G:4x4	39,09	38,67	241,92	234,86	46,74	44,34
	C:2,0 G:16x16	39,09	36,20	241,92	226,64	46,74	44,11
	C:2,0 G:4x4	39,09	37,21	241,92	247,96	46,74	41,03
	C:5,0 G:16x16	39,09	36,48	241,92	225,77	46,74	42,17
Mountain	C:2,0 G:8x8	37,33	32,67	251,39	174,30	41,94	46,86
Park	C:10,0 G:16x16	36,83	32,07	232,01	171,99	45,60	47,43
	C:10,0 G:4x4	36,83	34,65	232,01	187,74	45,60	48,69
	C:5,0 G:16x16	36,83	34,14	232,01	195,34	45,60	44,69
Parking	C:5,0 G:16x16	39,04	33,15	259,80	202,65	42,74	45,03

Scenariospecifieke Contrastverbetering voor Satellietcoreregistratie

	C:2,0 G:32x32	39,04	33,40	259,80	225,39	42,74	40,57
<i>Playground</i>	C:10,0 G:4x4	22,28	22,29	127,67	96,46	45,03	44,57
	C:2,0 G:32x32	22,28	19,79	127,67	97,08	45,03	44,69
<i>Pond</i>	C:10,0 G:4x4	17,38	24,29	83,67	84,17	45,60	31,54
	C:10,0 G:8x8	17,38	25,49	83,67	89,03	45,60	26,40
	C:5,0 G:4x4	17,38	23,59	83,67	81,79	45,60	35,09
	C:40,0 G:4x4	17,38	24,94	83,67	82,76	45,60	29,14
<i>Port</i>	C:2,0 G:32x32	25,13	23,16	155,63	126,68	42,29	44,69
	C:10,0 G:8x8	25,13	24,64	155,63	124,08	42,29	40,23
	C:40,0 G:4x4	25,13	24,68	155,63	113,78	42,29	42,97
	C:5,0 G:4x4	25,13	24,08	155,63	120,62	42,29	45,37
<i>RailwayStation</i>	C:10,0 G:4x4	34,35	31,35	217,88	195,54	45,37	42,74
	C:2,0 G:8x8	34,35	31,91	217,88	200,48	45,37	44,46
	C:5,0 G:16x16	34,35	30,38	217,88	180,96	45,37	43,54
<i>Resort</i>	C:2,0 G:4x4	39,07	37,82	246,66	242,21	45,49	44,00
	C:5,0 G:8x8	39,07	36,15	246,66	215,25	45,49	45,60
	C:20,0 G:8x8	39,07	37,26	246,66	228,61	45,49	43,89
	C:2,0 G:64x64	39,07	31,24	246,66	184,56	45,49	46,29
<i>River</i>	C:20,0 G:4x4	24,56	23,42	142,52	113,99	46,40	39,31
<i>School</i>	C:10,0 G:32x32	44,08	37,01	281,44	223,91	45,60	45,03
	C:10,0 G:4x4	44,08	41,06	281,44	258,02	45,60	44,91
	C:2,0 G:8x8	44,08	41,12	281,44	257,95	45,60	45,83
	C:2,0 G:4x4	44,08	41,12	281,44	268,45	45,60	43,66
	C:2,0 G:16x16	44,08	41,28	281,44	261,86	45,60	44,69
	C:40,0 G:16x16	44,08	38,05	281,44	224,48	45,60	47,31
	C:40,0 G:4x4	44,08	41,28	281,44	266,50	45,60	43,54
	C:40,0 G:8x8	44,08	41,30	281,44	261,25	45,60	44,34
<i>SparseResidential</i>	C:10,0 G:4x4	39,95	35,26	275,36	222,91	41,49	44,57
	C:10,0 G:10x10	39,95	34,03	275,36	212,05	41,49	45,14
	C:10,0 G:8x8	39,95	34,93	275,36	231,89	41,49	41,14
	C:20,0 G:8x8	39,95	35,09	275,36	208,21	41,49	47,54
	C:40,0 G:16x16	39,95	32,12	275,36	189,47	41,49	46,63
<i>Square</i>	C:2,0 G:16x16	37,84	36,22	232,70	234,66	47,14	43,02
	C:40,0 G:16x16	37,84	33,72	232,70	196,39	47,14	44,97
	C:5,0 G:4x4	37,84	36,01	232,70	219,47	47,14	45,88
	C:2,0 G:8x8	37,84	35,88	232,70	219,07	47,14	46,57
<i>Stadium</i>	C:2,0 G:4x4	29,83	27,84	171,99	156,93	48,46	48,57
	C:20,0 G:4x4	29,83	27,60	171,99	144,35	48,46	50,97
	C:2,0 G:16x16	29,83	27,41	171,99	161,54	48,46	45,83
<i>StorageTanks</i>	C:10,0 G:8x8	33,96	30,38	214,74	172,95	45,26	48,11
	C:2,0 G:4x4	33,96	31,94	214,74	191,32	45,26	46,86
	C:2,0 G:32x32	33,96	29,72	214,74	180,39	45,26	45,94
<i>Viaduct</i>	C:2,0 G:4x4	31,72	30,11	209,42	181,43	42,97	46,97
	C:2,0 G:8x8	31,72	30,33	209,42	184,70	42,97	46,29
	C:20,0 G:4x4	31,72	29,64	209,42	168,15	42,97	48,11
	C:2,0 G:32x32	31,72	28,18	209,42	182,65	42,97	41,94

Appendix 1: Tabellen

Tabel 4: Resultaten van de statistische validatie per categorie, per CLAHE-instelling.

Category	CLAHE	N	Wilcoxon P Ratio	Ratio Significant (p<0,05)	Ratio Winner	Wilcoxon P Abs	Abs Significant (p<0,05)	Abs Winner	McNemar P Fail	Fail Significant (p<0,05)	Fail Winner
Airport	C:10,0 G:8x8	875	9,46E-06	Yes	RAW	1,85E-10	Yes	RAW	3,23E-02	Yes	RAW
	C:2,0 G:4x4	875	1,06E-05	Yes	RAW	2,29E-07	Yes	RAW	3,65E-03	Yes	RAW
	C:2,0 G:8x8	875	1,06E-05	Yes	RAW	2,24E-07	Yes	RAW	4,60E-02	Yes	RAW
	C:20,0 G:8x8	875	1,22E-06	Yes	RAW	7,82E-12	Yes	RAW	1,46E-02	Yes	RAW
BareLand	C:2,0 G:8x8	875	5,68E-06	Yes	CLAHE	1,54E-12	Yes	RAW	2,72E-03	Yes	CLAHE
BaseballField	C:2,0 G:32x32	875	3,54E-02	Yes	RAW	4,49E-05	Yes	RAW	6,01E-01	No	-
	C:2,0 G:4x4	875	1,39E-01	No	-	4,83E-03	Yes	RAW	7,69E-01	No	-
	C:2,0 G:8x8	875	4,44E-02	Yes	RAW	9,79E-04	Yes	RAW	4,23E-01	No	-
Beach	C:5,0 G:16x16	875	5,90E-16	Yes	CLAHE	5,33E-24	Yes	RAW	5,53E-11	Yes	CLAHE
Bridge	C:10,0 G:4x4	875	5,84E-08	Yes	CLAHE	2,68E-03	Yes	RAW	3,16E-02	Yes	CLAHE
	C:20,0 G:4x4	875	5,78E-11	Yes	CLAHE	3,46E-02	Yes	RAW	3,31E-04	Yes	CLAHE
	C:20,0 G:8x8	875	3,99E-10	Yes	CLAHE	7,08E-04	Yes	RAW	5,74E-03	Yes	CLAHE
	C:40,0 G:32x32	875	3,05E-11	Yes	CLAHE	7,58E-09	Yes	RAW	6,96E-05	Yes	CLAHE
	C:5,0 G:16x16	875	2,29E-09	Yes	CLAHE	6,42E-04	Yes	RAW	2,88E-02	Yes	CLAHE
Center	C:10,0 G:32x32	875	1,89E-04	Yes	RAW	3,22E-09	Yes	RAW	9,61E-01	No	-
	C:10,0 G:4x4	875	5,25E-02	No	-	5,25E-03	Yes	RAW	8,12E-01	No	-
Church	C:2,0 G:4x4	875	2,88E-03	Yes	RAW	1,46E-03	Yes	RAW	8,47E-01	No	-
	C:2,0 G:8x8	875	4,16E-04	Yes	RAW	1,61E-04	Yes	RAW	5,57E-01	No	-
	C:40,0 G:16x16	875	3,76E-06	Yes	RAW	3,14E-07	Yes	RAW	6,62E-01	No	-
	C:40,0 G:4x4	875	6,05E-06	Yes	RAW	1,04E-06	Yes	RAW	3,99E-01	No	-
Commercial	C:2,0 G:32x32	875	1,05E-03	Yes	RAW	3,56E-04	Yes	RAW	9,62E-01	No	-
	C:40,0 G:4x4	875	2,11E-04	Yes	RAW	2,20E-05	Yes	RAW	3,88E-01	No	-
	C:5,0 G:4x4	875	3,46E-03	Yes	RAW	9,03E-04	Yes	RAW	8,83E-01	No	-
DenseResidential	C:2,0 G:4x4	875	1,41E-03	Yes	RAW	1,22E-03	Yes	RAW	6,97E-01	No	-
	C:2,0 G:8x8	875	5,03E-02	No	-	3,58E-02	Yes	RAW	1,22E-01	No	-
	C:40,0 G:4x4	875	6,23E-07	Yes	RAW	1,84E-07	Yes	RAW	6,97E-02	No	-
Desert	C:2,0 G:8x8	875	1,21E-15	Yes	CLAHE	6,36E-36	Yes	RAW	5,24E-10	Yes	CLAHE
Farmland	C:2,0 G:4x4	875	1,43E-01	No	-	1,31E-06	Yes	RAW	6,67E-01	No	-
	C:40,0 G:16x16	875	2,13E-06	Yes	CLAHE	2,67E-18	Yes	RAW	4,80E-08	Yes	CLAHE
Forest	C:20,0 G:16x16	875	4,15E-03	Yes	RAW	8,91E-41	Yes	RAW	6,87E-11	Yes	CLAHE

Scenariospecifieke Contrastverbetering voor Satellietcoreregistratie

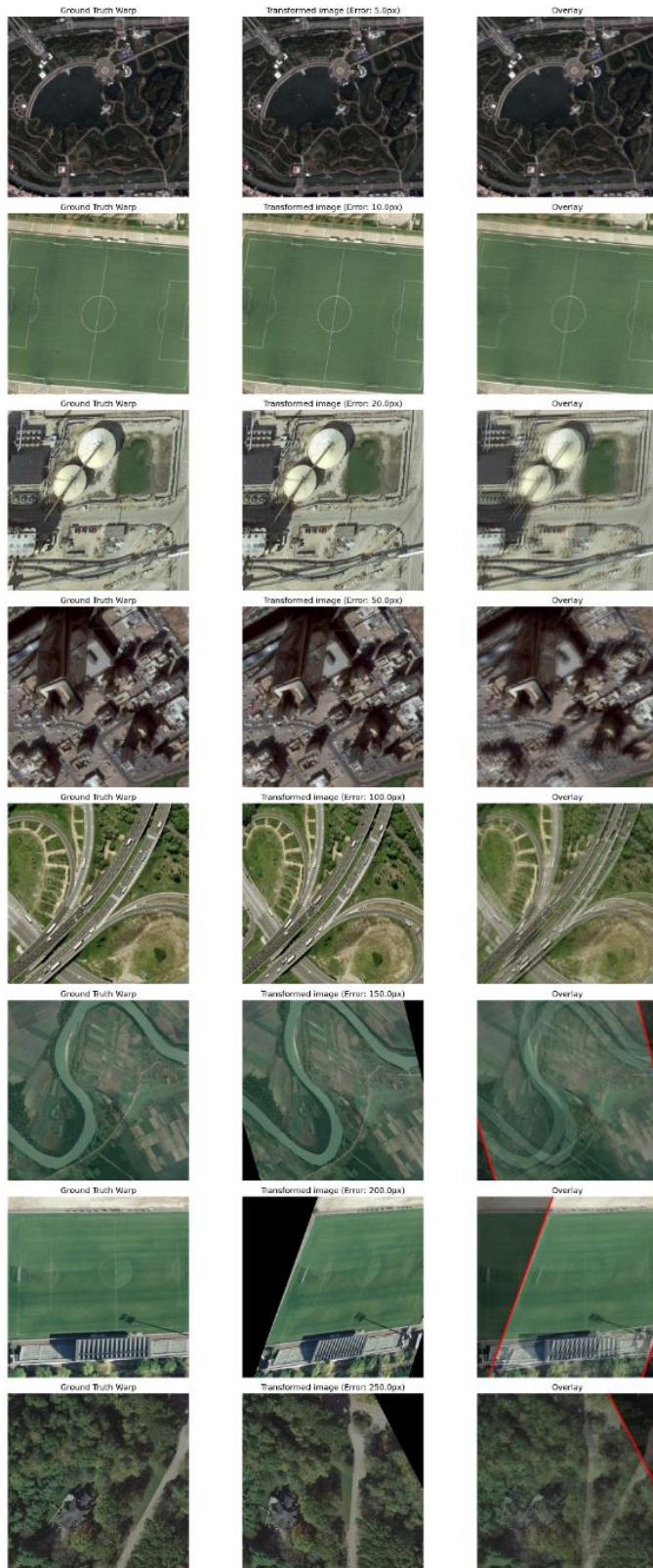
<i>Industrial</i>	C:2,0 G:8x8	875	7,15E-03	Yes	RAW	3,00E-03	Yes	RAW	7,66E-01	No	-
	C:20,0 G:8x8	875	8,91E-06	Yes	RAW	5,62E-07	Yes	RAW	2,45E-01	No	-
	C:40,0 G:4x4	875	1,56E-04	Yes	RAW	2,28E-05	Yes	RAW	4,64E-01	No	-
	C:40,0 G:8x8	875	1,25E-03	Yes	RAW	1,43E-04	Yes	RAW	9,62E-01	No	-
	C:5,0 G:32x32	875	7,44E-08	Yes	RAW	9,81E-10	Yes	RAW	6,78E-02	No	-
<i>Meadow</i>	C:2,0 G:16x16	875	6,76E-17	Yes	CLAHE	9,37E-35	Yes	RAW	5,72E-15	Yes	CLAHE
<i>MediumResidential</i>	C:10,0 G:4x4	875	4,84E-01	No	-	4,45E-02	Yes	RAW	3,49E-01	No	-
	C:2,0 G:16x16	875	2,07E-01	No	-	3,71E-02	Yes	RAW	2,93E-01	No	-
	C:2,0 G:4x4	875	8,98E-01	No	-	5,44E-01	No	-	1,79E-02	Yes	CLAHE
	C:5,0 G:16x16	875	3,25E-01	No	-	1,16E-02	Yes	RAW	6,24E-02	No	-
<i>Mountain</i>	C:2,0 G:8x8	875	2,23E-10	Yes	RAW	3,41E-14	Yes	RAW	4,70E-02	Yes	RAW
<i>Park</i>	C:10,0 G:16x16	875	1,46E-04	Yes	RAW	4,81E-09	Yes	RAW	4,78E-01	No	-
	C:10,0 G:4x4	875	8,55E-04	Yes	RAW	7,02E-07	Yes	RAW	2,20E-01	No	-
	C:5,0 G:16x16	875	1,03E-02	Yes	RAW	3,35E-06	Yes	RAW	7,34E-01	No	-
<i>Parking</i>	C:2,0 G:32x32	875	7,54E-04	Yes	RAW	1,53E-04	Yes	RAW	3,81E-01	No	-
	C:5,0 G:16x16	875	8,13E-08	Yes	RAW	1,80E-09	Yes	RAW	3,50E-01	No	-
<i>Playground</i>	C:10,0 G:4x4	875	4,74E-01	No	-	2,88E-08	Yes	RAW	8,86E-01	No	-
	C:2,0 G:32x32	875	1,84E-02	Yes	RAW	8,71E-08	Yes	RAW	9,23E-01	No	-
<i>Pond</i>	C:10,0 G:4x4	875	4,65E-37	Yes	CLAHE	9,90E-01	No	-	2,52E-09	Yes	CLAHE
	C:10,0 G:8x8	875	2,02E-50	Yes	CLAHE	5,03E-01	No	-	9,69E-17	Yes	CLAHE
	C:40,0 G:4x4	875	8,68E-41	Yes	CLAHE	5,58E-01	No	-	3,00E-12	Yes	CLAHE
	C:5,0 G:4x4	875	4,20E-31	Yes	CLAHE	6,33E-01	No	-	6,63E-06	Yes	CLAHE
<i>Port</i>	C:10,0 G:8x8	875	3,08E-01	No	-	8,61E-06	Yes	RAW	4,00E-01	No	-
	C:2,0 G:32x32	875	1,87E-03	Yes	RAW	6,25E-07	Yes	RAW	3,33E-01	No	-
	C:40,0 G:4x4	875	7,21E-01	No	-	3,28E-08	Yes	RAW	8,12E-01	No	-
	C:5,0 G:4x4	875	1,09E-01	No	-	1,36E-07	Yes	RAW	2,02E-01	No	-
<i>RailwayStation</i>	C:10,0 G:4x4	875	9,36E-02	No	-	3,22E-03	Yes	RAW	2,80E-01	No	-
	C:2,0 G:8x8	875	2,89E-02	Yes	RAW	7,85E-03	Yes	RAW	7,35E-01	No	-
	C:5,0 G:16x16	875	8,25E-03	Yes	RAW	1,83E-05	Yes	RAW	4,77E-01	No	-
<i>Resort</i>	C:2,0 G:4x4	875	2,84E-01	No	-	1,23E-01	No	-	5,69E-01	No	-
	C:2,0 G:64x64	875	2,60E-09	Yes	RAW	2,65E-11	Yes	RAW	7,64E-01	No	-
	C:20,0 G:8x8	875	1,76E-01	No	-	6,90E-03	Yes	RAW	5,42E-01	No	-
	C:5,0 G:8x8	875	4,09E-02	Yes	RAW	6,42E-04	Yes	RAW	1,00E+00	No	-
<i>River</i>	C:20,0 G:4x4	875	4,78E-01	No	-	5,12E-07	Yes	RAW	2,78E-03	Yes	CLAHE
<i>School</i>	C:10,0 G:32x32	875	6,57E-06	Yes	RAW	6,72E-08	Yes	RAW	8,44E-01	No	-

Appendix 1: Tabellen

	C:10,0 G:4x4	875	1,55E-02	Yes	RAW	4,16E-03	Yes	RAW	8,14E-01	No	-
	C:2,0 G:16x16	875	1,29E-02	Yes	RAW	5,63E-03	Yes	RAW	7,40E-01	No	-
	C:2,0 G:4x4	875	3,59E-02	Yes	RAW	1,80E-02	Yes	RAW	4,40E-01	No	-
	C:2,0 G:8x8	875	2,22E-03	Yes	RAW	8,93E-04	Yes	RAW	9,61E-01	No	-
	C:40,0 G:16x16	875	1,51E-05	Yes	RAW	4,74E-07	Yes	RAW	5,06E-01	No	-
	C:40,0 G:4x4	875	4,01E-02	Yes	RAW	1,12E-02	Yes	RAW	4,19E-01	No	-
	C:40,0 G:8x8	875	1,83E-02	Yes	RAW	3,94E-03	Yes	RAW	6,35E-01	No	-
<i>SparseResidential</i>	C:10,0 G:10x10	875	7,46E-10	Yes	RAW	1,73E-11	Yes	RAW	1,22E-01	No	-
	C:10,0 G:4x4	875	9,18E-08	Yes	RAW	8,14E-09	Yes	RAW	1,99E-01	No	-
	C:10,0 G:8x8	875	1,19E-05	Yes	RAW	3,26E-07	Yes	RAW	9,23E-01	No	-
	C:20,0 G:8x8	875	1,27E-10	Yes	RAW	3,55E-12	Yes	RAW	1,23E-02	Yes	RAW
	C:40,0 G:16x16	875	5,46E-13	Yes	RAW	1,84E-15	Yes	RAW	3,28E-02	Yes	RAW
<i>Square</i>	C:2,0 G:16x16	874	2,69E-01	No	-	1,27E-01	No	-	8,62E-02	No	-
	C:2,0 G:8x8	874	6,03E-02	No	-	2,50E-02	Yes	RAW	8,49E-01	No	-
	C:40,0 G:16x16	874	3,27E-03	Yes	RAW	6,70E-06	Yes	RAW	3,69E-01	No	-
	C:5,0 G:4x4	874	9,85E-02	No	-	2,95E-02	Yes	RAW	6,39E-01	No	-
<i>Stadium</i>	C:2,0 G:16x16	875	1,89E-01	No	-	5,98E-02	No	-	2,96E-01	No	-
	C:2,0 G:4x4	875	5,29E-02	No	-	2,03E-02	Yes	RAW	1,00E+00	No	-
	C:20,0 G:4x4	875	1,90E-03	Yes	RAW	8,03E-05	Yes	RAW	3,09E-01	No	-
<i>StorageTanks</i>	C:10,0 G:8x8	875	1,08E-04	Yes	RAW	1,38E-06	Yes	RAW	2,59E-01	No	-
	C:2,0 G:32x32	875	1,00E-03	Yes	RAW	1,06E-04	Yes	RAW	8,14E-01	No	-
	C:2,0 G:4x4	875	5,02E-03	Yes	RAW	1,61E-03	Yes	RAW	5,40E-01	No	-
<i>Viaduct</i>	C:2,0 G:32x32	875	1,27E-04	Yes	RAW	6,19E-06	Yes	RAW	6,92E-01	No	-
	C:2,0 G:4x4	875	7,70E-05	Yes	RAW	1,43E-05	Yes	RAW	9,35E-02	No	-
	C:2,0 G:8x8	875	1,11E-03	Yes	RAW	1,95E-04	Yes	RAW	1,84E-01	No	-
	C:20,0 G:4x4	875	2,34E-05	Yes	RAW	1,74E-07	Yes	RAW	3,70E-02	Yes	RAW

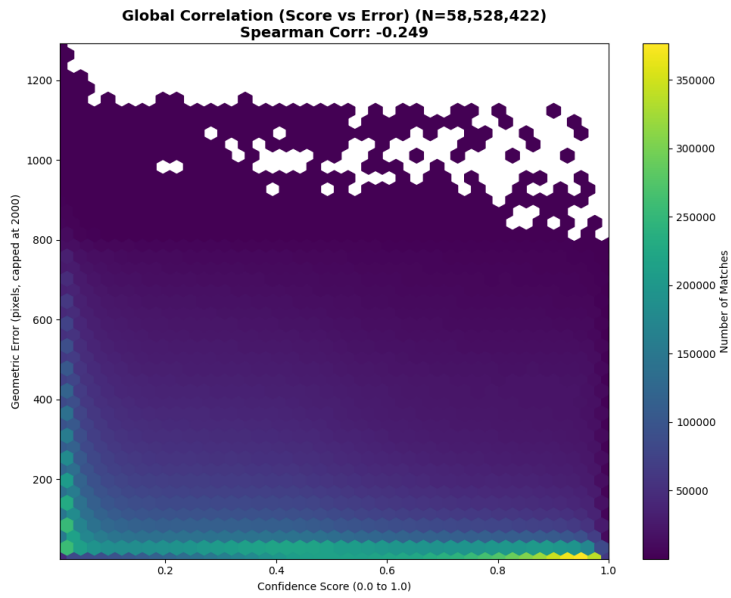
Appendix 2: Afbeeldingen

Visualizing Geometric Error Magnitudes



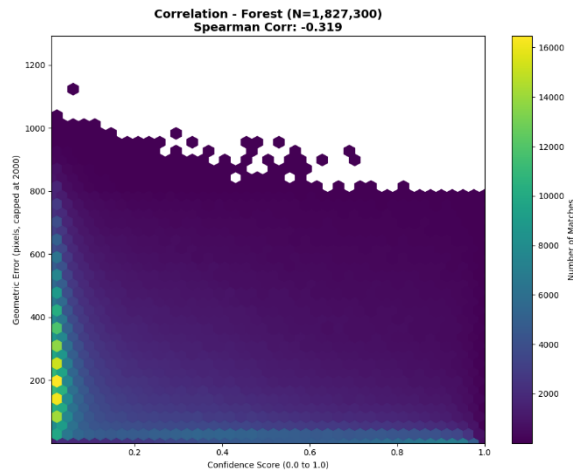
Figuur 30: Deze figuur illustreert dat tot een gemiddelde fout van 50 pixels de overlay bruikbaar is.

Appendix 2: Afbeeldingen

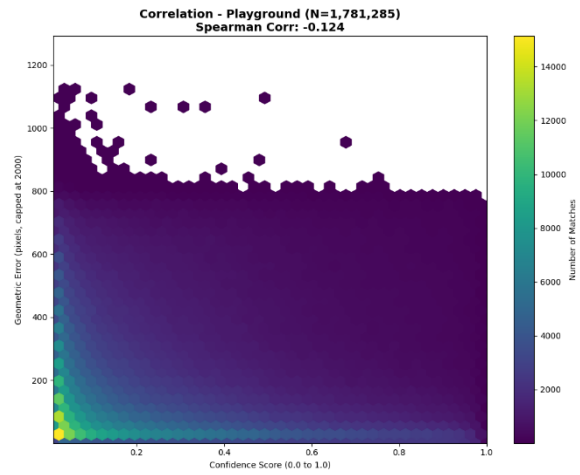


Figuur 31: Dit is de densiteitsplot van de geometrische fout in functie van de confidence score voor alle matches uit de random search. De Spearman-correlatiecoëfficiënt (-0.249) voor alle matches is zichtbaar in de titel. Er is een omgekeerd evenredig verband tussen de geometrische fout en de confidence score van de matches. Het zwaartepunt van de densiteitsplot ligt bij hoge confidence scores en lage geometrische fouten.

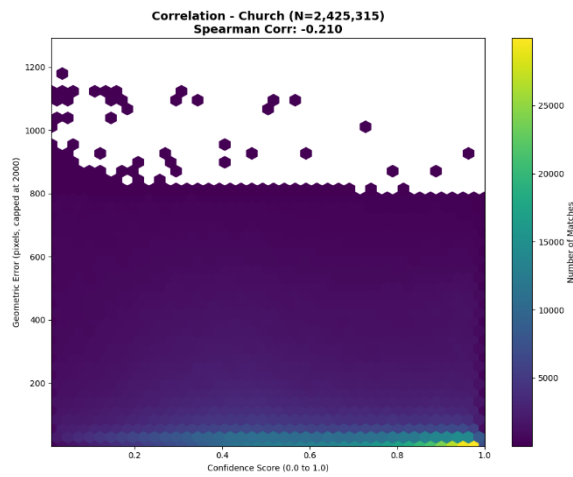
Scenariospecifieke Contrastverbetering voor Satellietcoregistratie



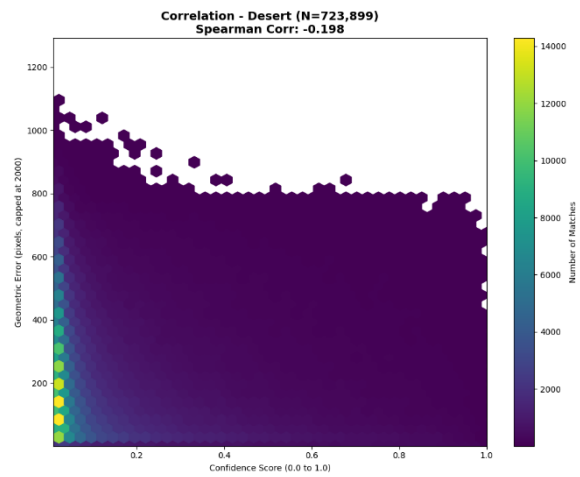
Figuur 32: De Spearman-correlatiecoëfficiënt van de categorie Forest (-0.319) is zichtbaar in de titel. Het zwaartepunt van de dichtheitsplot ligt bij lage confidence scores. Dit wijst erop dat Forest algemeen genomen een moeilijke categorie is voor MatchAnything om te matchen.



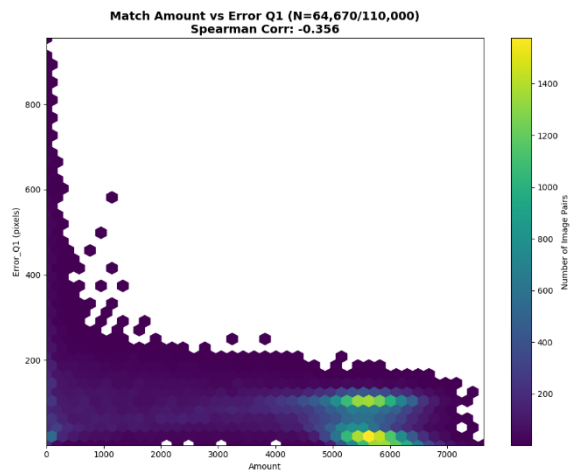
Figuur 33: De dichtheitsplot voor categorie Playground



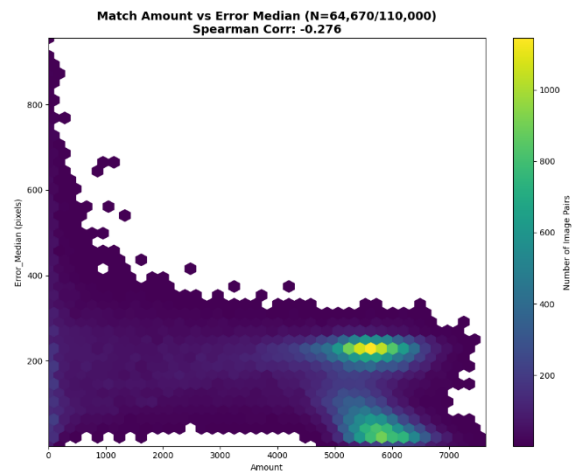
Figuur 34: De dichtheitsplot voor categorie Church.



Figuur 35: De dichtheitsplot voor categorie Desert



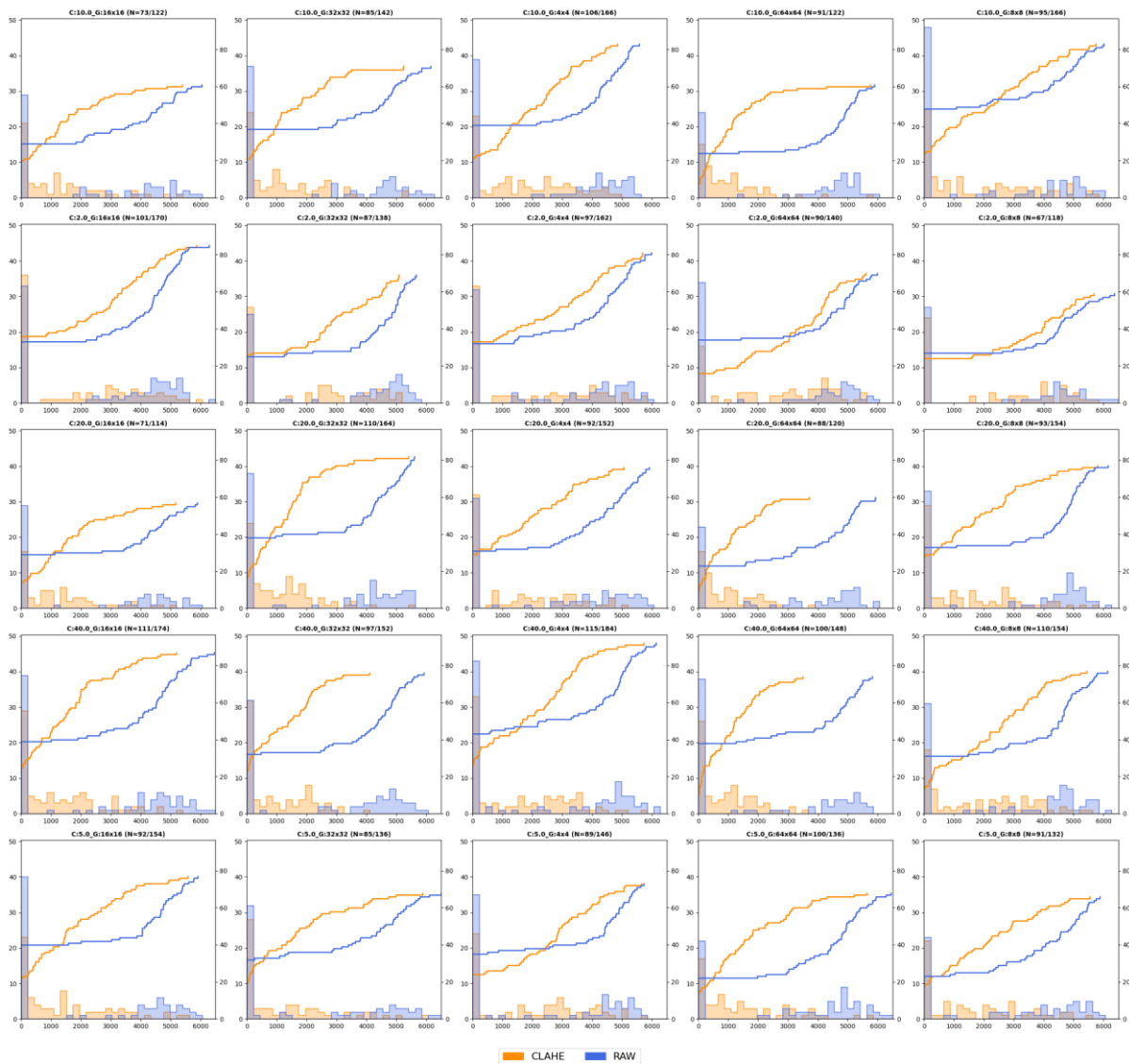
Figuur 36: De dichtheitsplot voor het verband tussen het eerste kwartiel van de fouten en het aantal matches in een matching-inferentie.



Figuur 37: De dichtheitsplot voor het verband tussen de mediaan van de fouten en het aantal matches in een matching-inferentie.

Appendix 2: Afbeeldingen

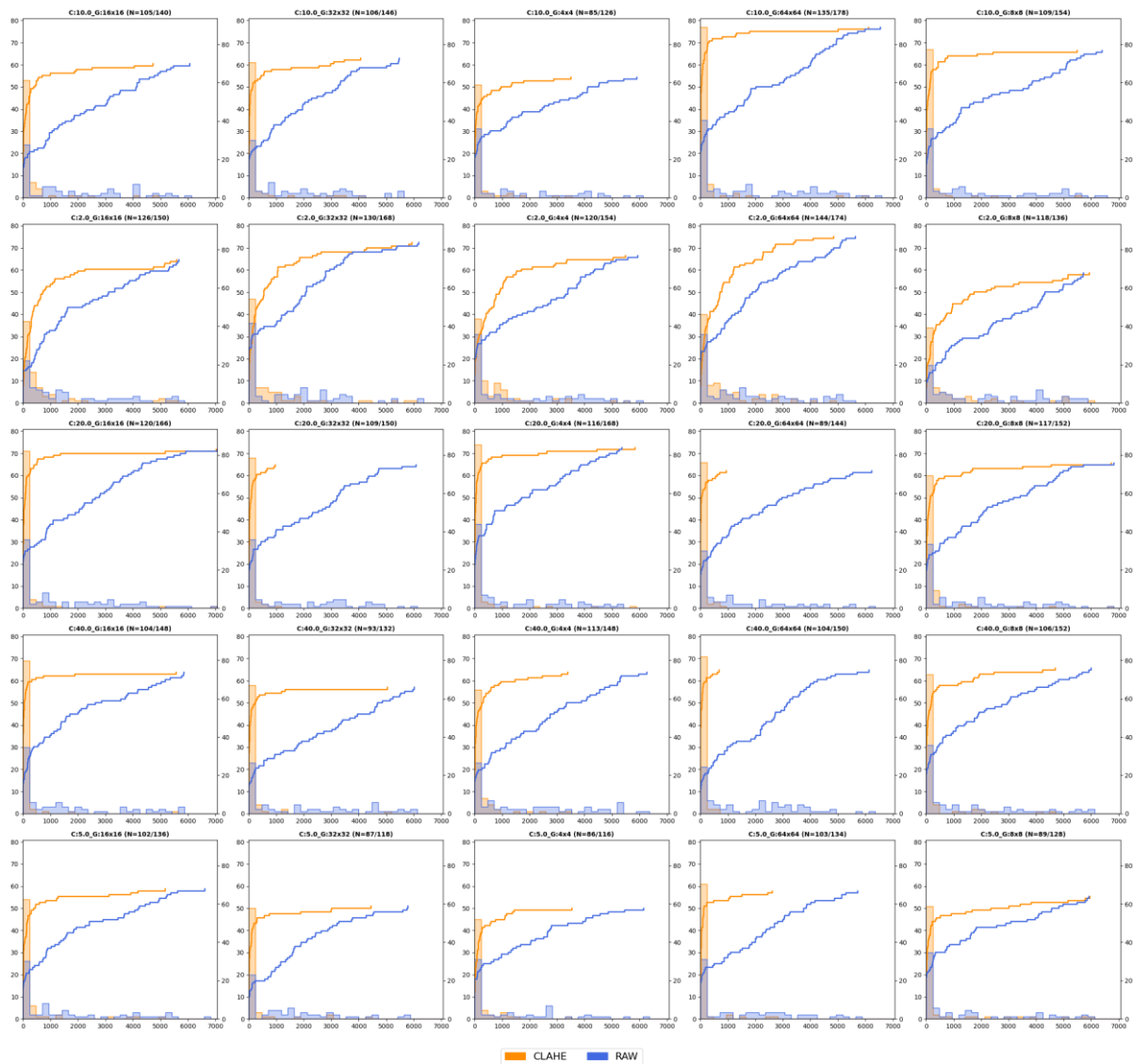
Match Amount Paired Breakdown: Pond



Figuur 38: De histogrammen en cumulatieve plots van de random search voor het aantal matches per matching-inferentie, gegeven de categorie Pond. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is.

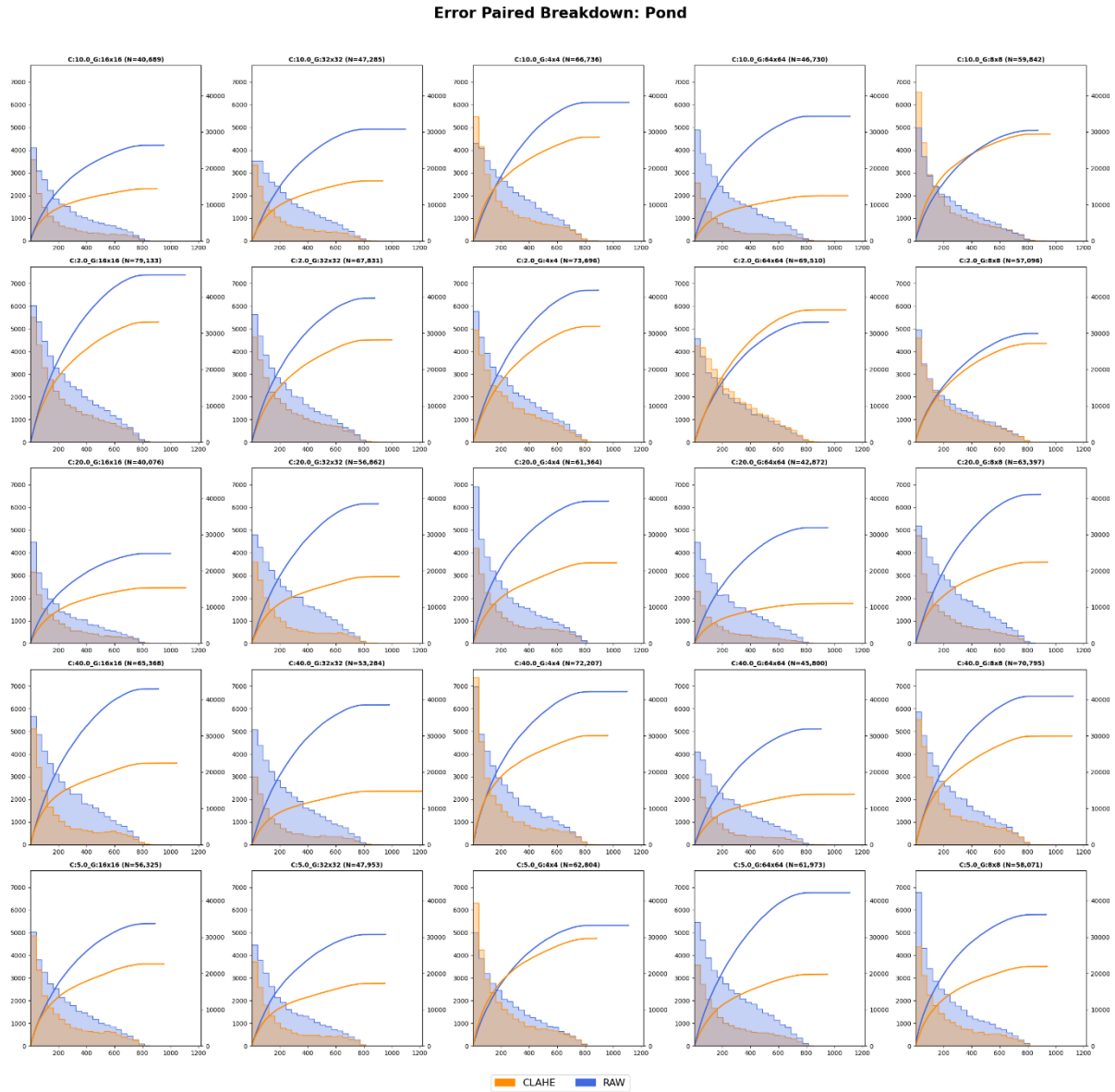
Scenariospecifieke Contrastverbetering voor Satellietcoregistratie

Match Amount Paired Breakdown: Desert



Figuur 39: De histogrammen en cumulatieve plots van de random search voor het aantal matches per matching-inferentie, gegeven de categorie Desert. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is.

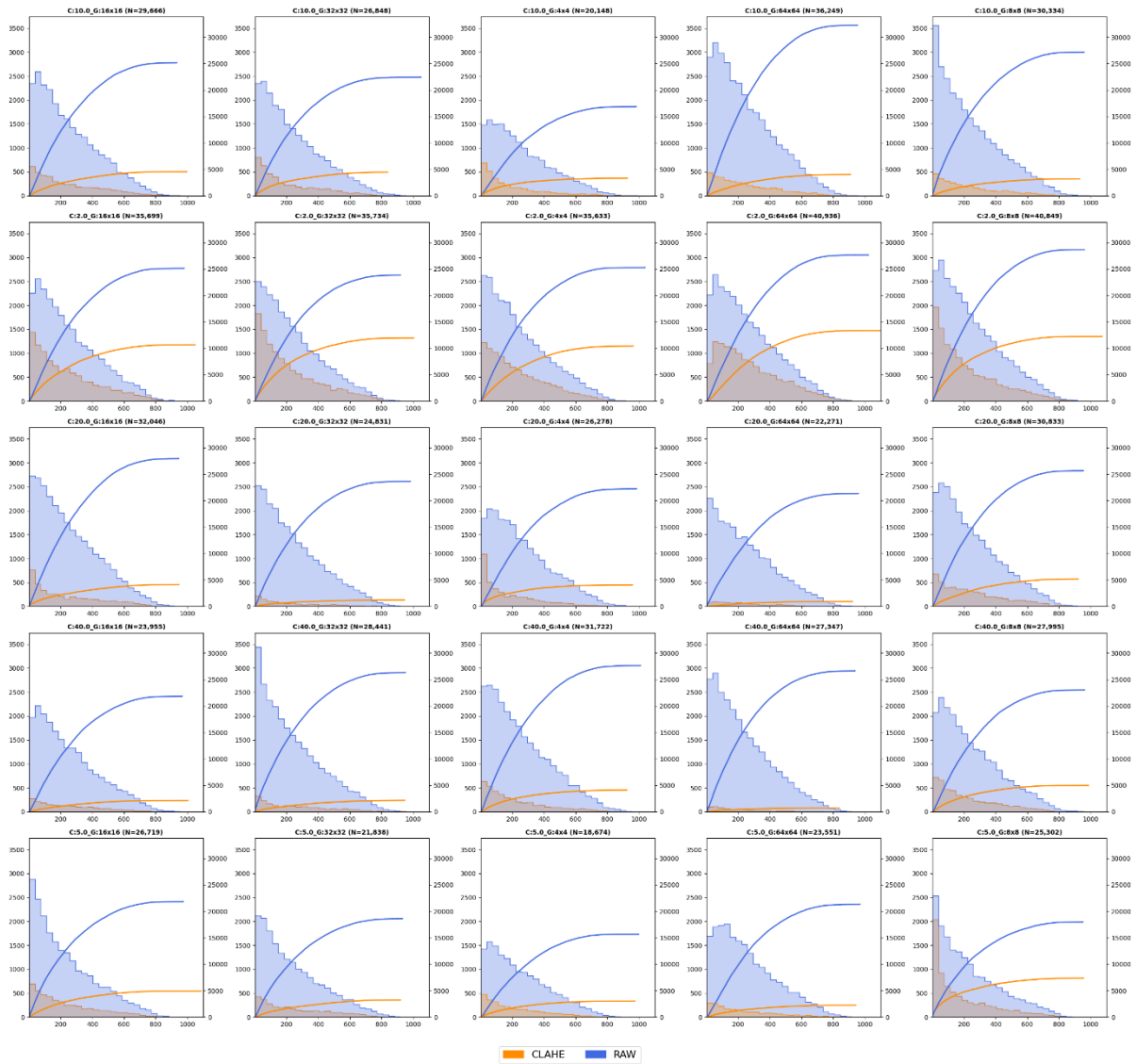
Appendix 2: Afbeeldingen



Figuur 40: De histogrammen en cumulatieve plots van de random search voor de geometrische fout per match, gegeven de categorie Pond. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.

Scenariospecifieke Contrastverbetering voor Satellietcoregistratie

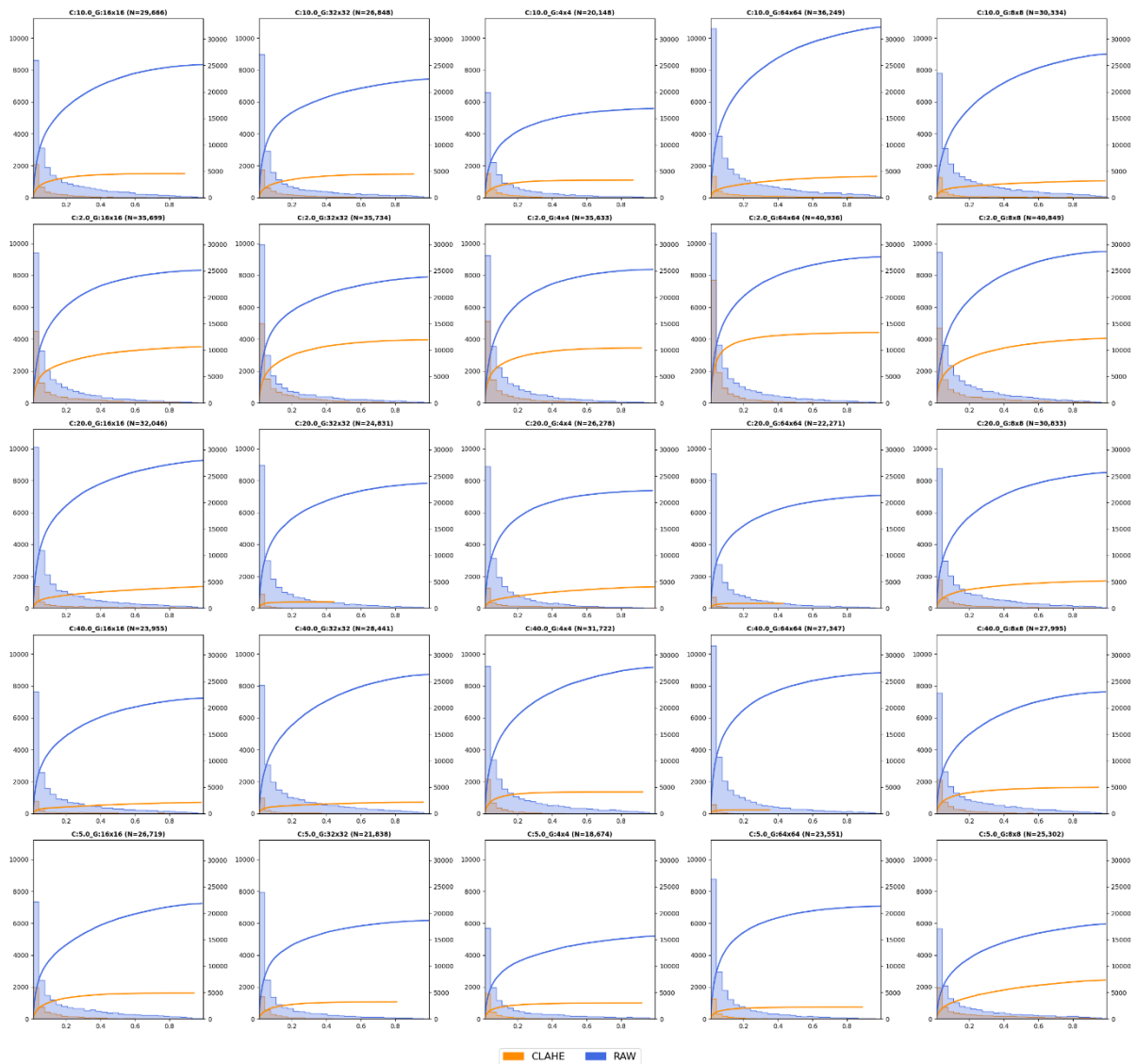
Error Paired Breakdown: Desert



Figuur 41: De histogrammen en cumulatieve plots van de random search voor de geometrische fout per match, gegeven de categorie Desert. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.

Appendix 2: Afbeeldingen

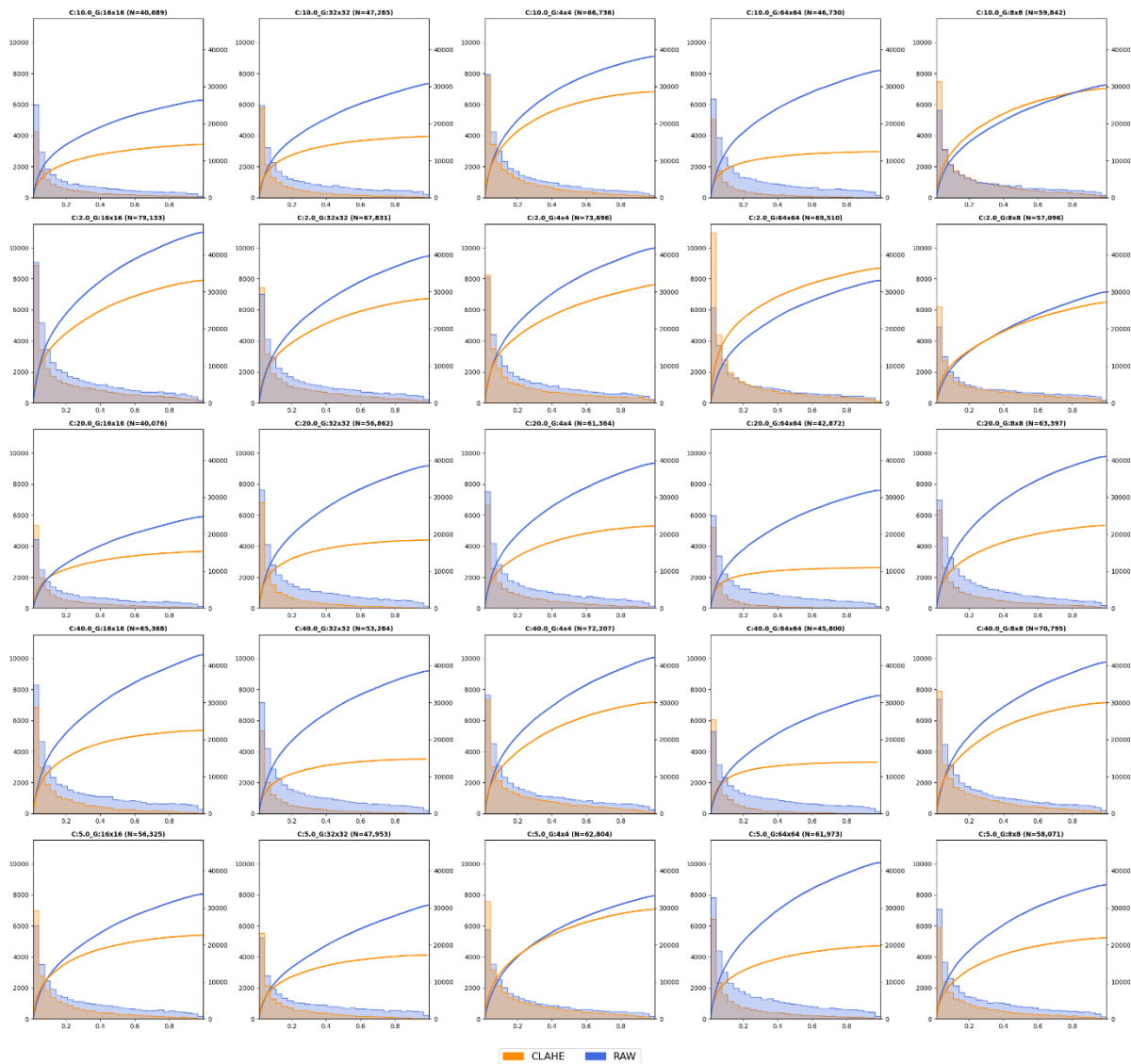
Score Paired Breakdown: Desert



Figuur 42: De histogrammen en cumulatieve plots van de random search voor de confidence score per match, gegeven de categorie Desert. Op de x-as staat de grootte van de confidence score van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.

Scenariospecifieke Contrastverbetering voor Satellietcoregistratie

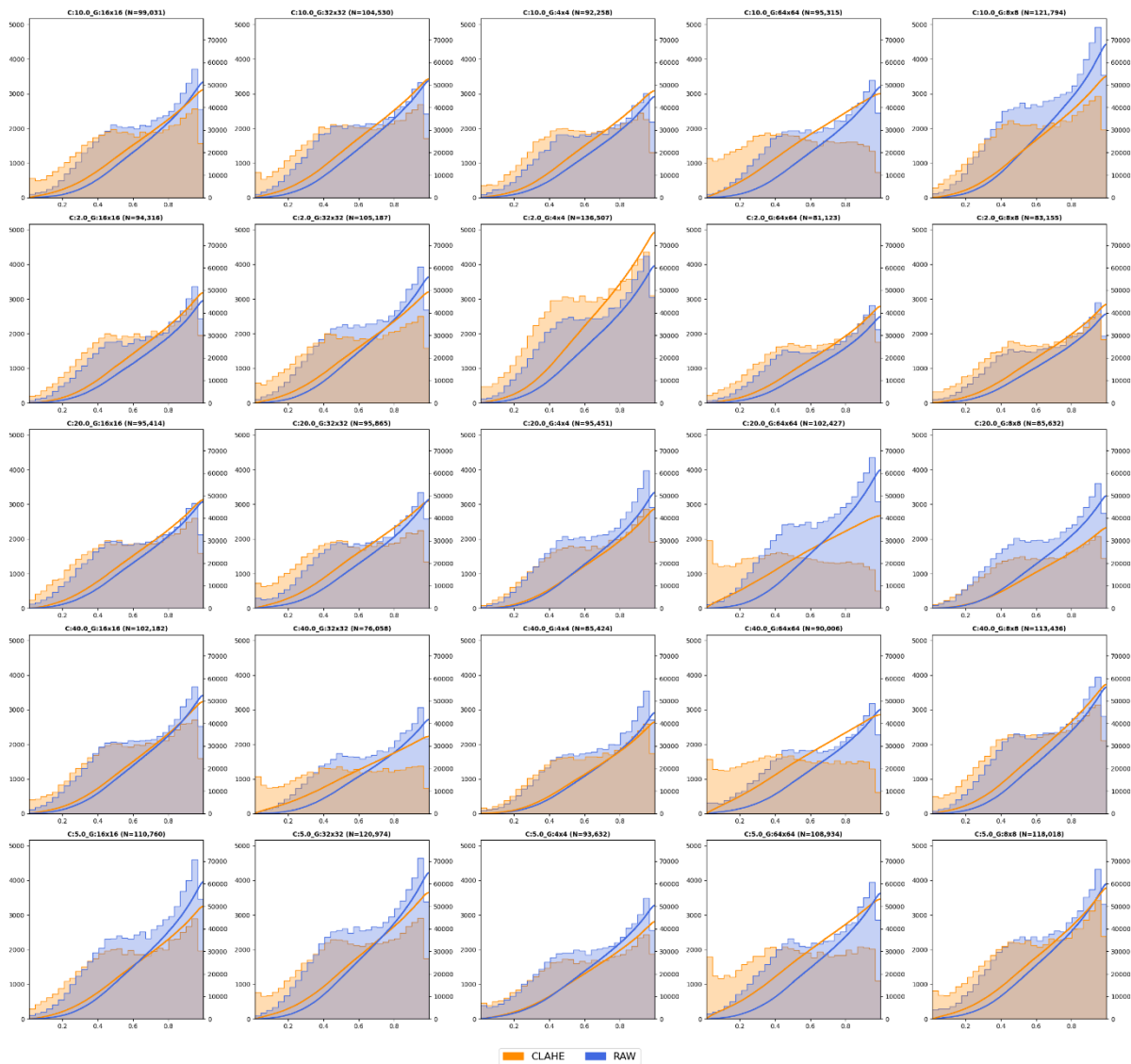
Score Paired Breakdown: Pond



Figuur 43: De histogrammen en cumulatieve plots van de random search voor de confidence score per match, gegeven de categorie Pond. Op de x-as staat de grootte van de confidence score van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.

Appendix 2: Afbeeldingen

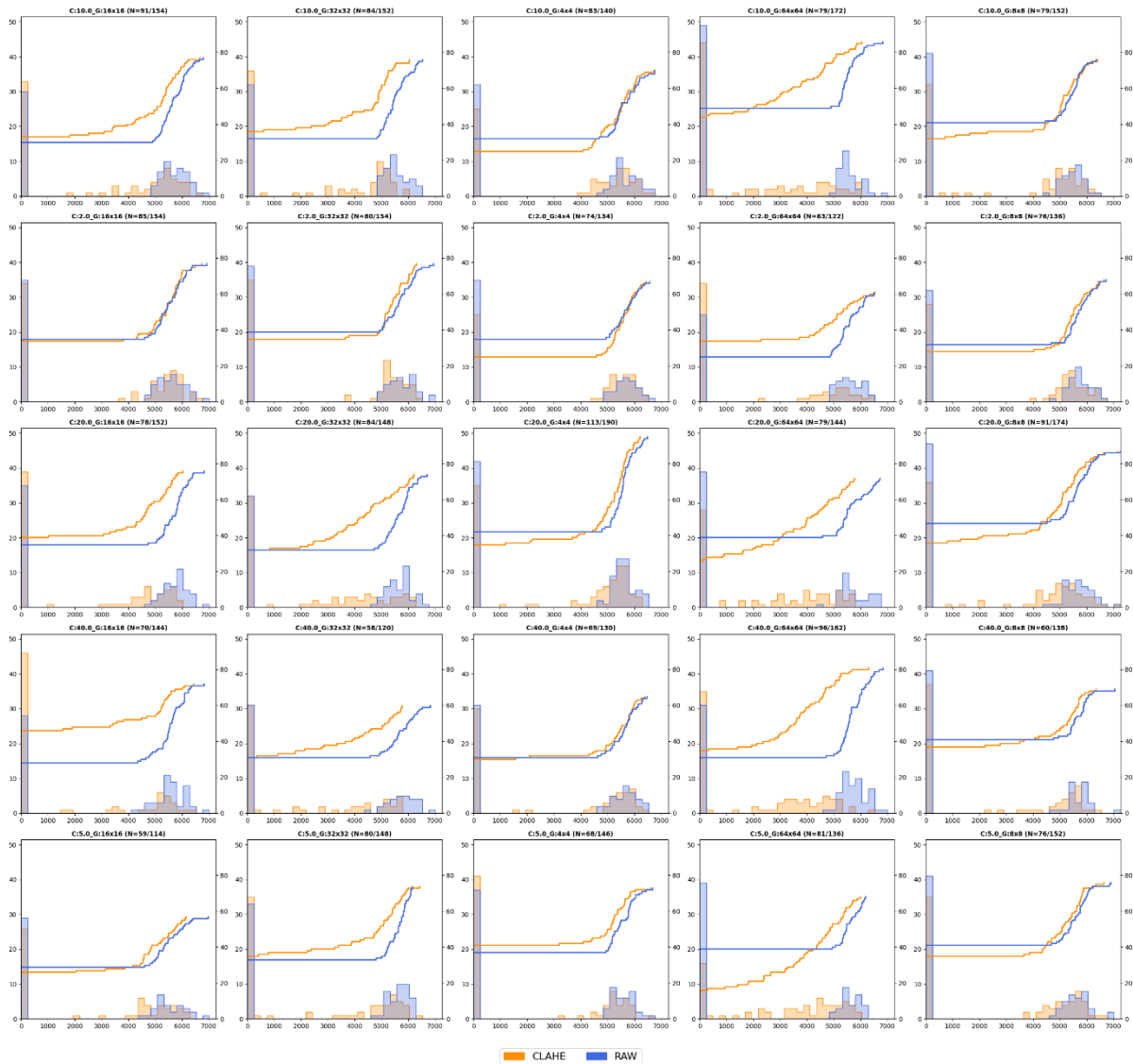
Score Paired Breakdown: DenseResidential



Figuur 44: De histogrammen en cumulatieve plots van de random search voor de confidence score per match, gegeven de categorie DenseResidential. Op de x-as staat de grootte van de confidence score van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.

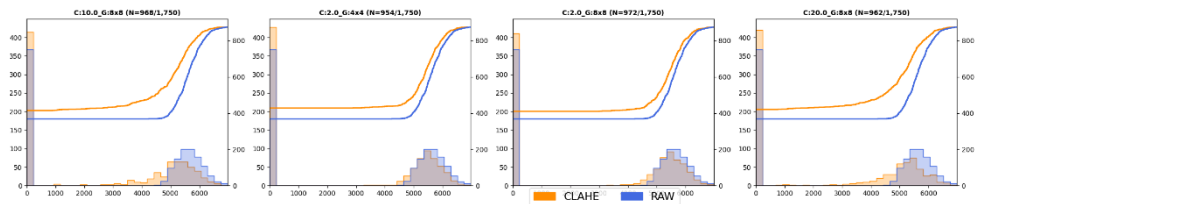
Scenariospecifieke Contrastverbetering voor Satellietcoregistratie

Match Amount Paired Breakdown: Airport



Figuur 45: De histogrammen en cumulatieve plots van de random search voor het aantal matches per matching-inferentie, gegeven de categorie Airport. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is.

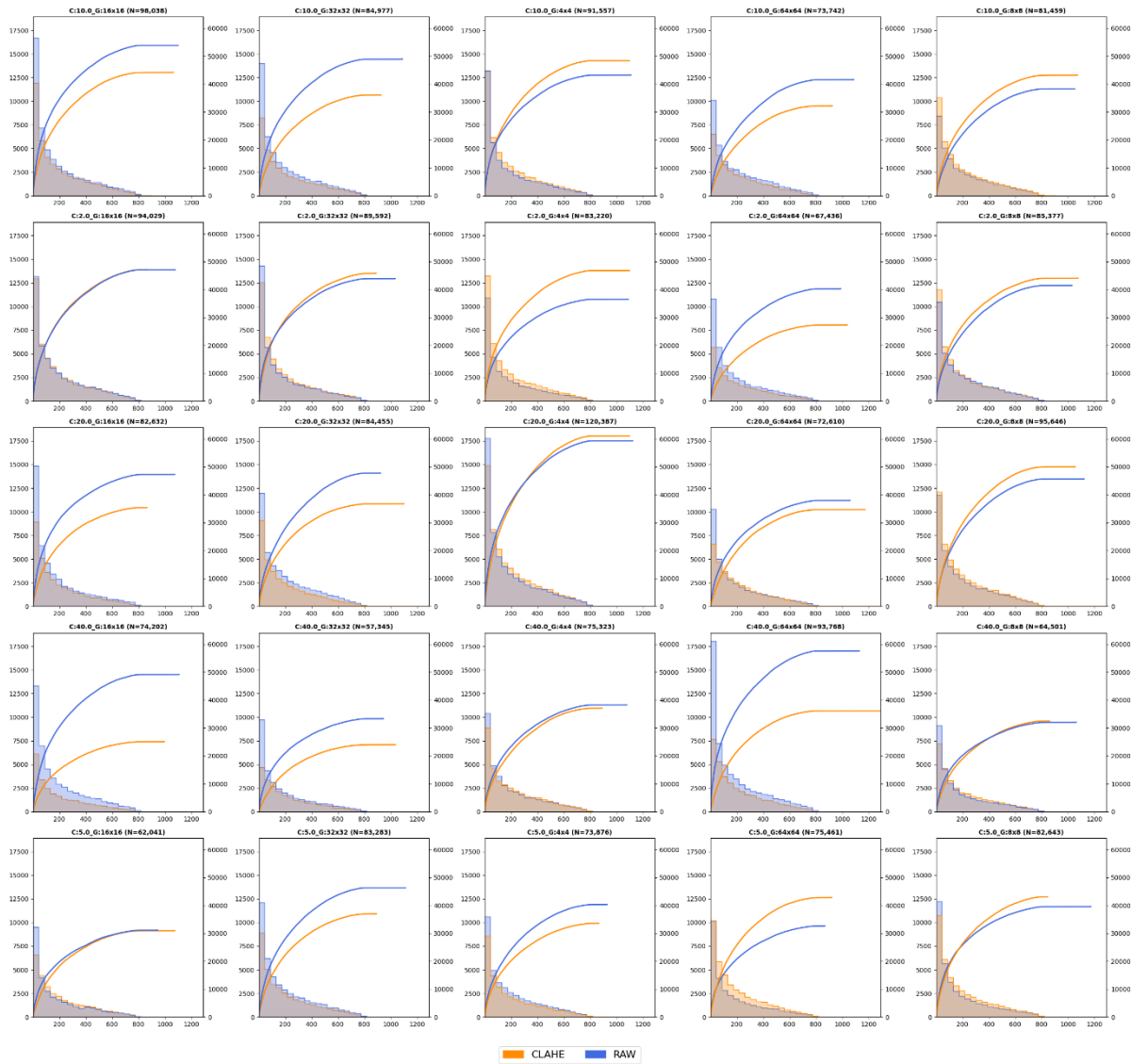
Match Amount Paired Breakdown: Airport



Figuur 46: De histogrammen en cumulatieve plots van de grid search voor het aantal matches per matching-inferentie, gegeven de categorie Airport. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is.

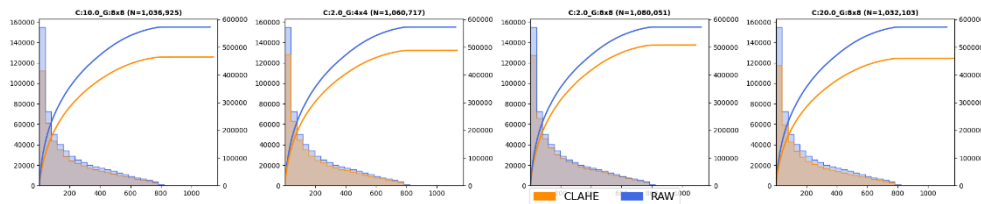
Appendix 2: Afbeeldingen

Error Paired Breakdown: Airport



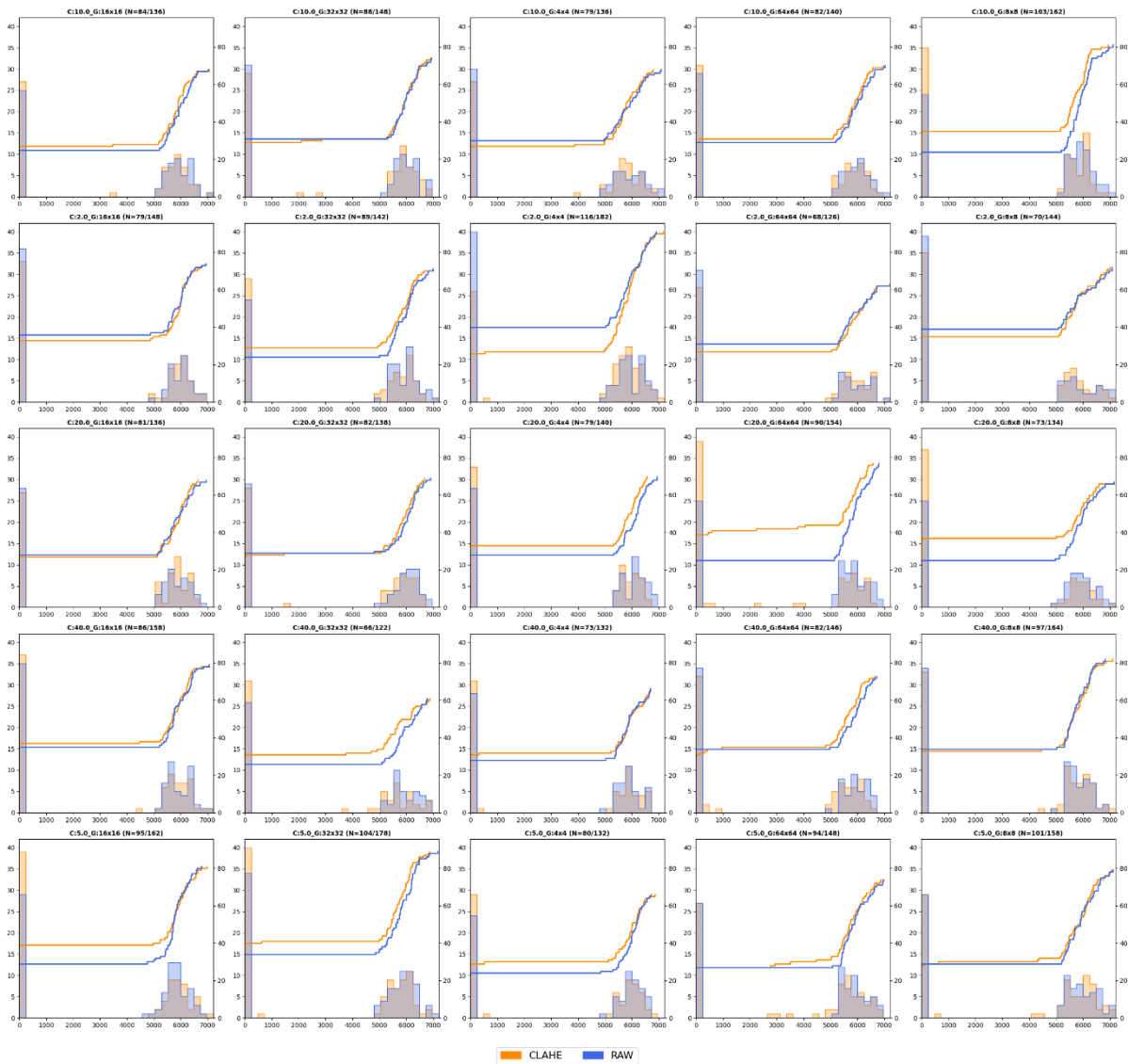
Figuur 47: De histogrammen en cumulatieve plots van de random search voor de geometrische fout per match, gegeven de categorie Airport. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.

Error Paired Breakdown: Airport



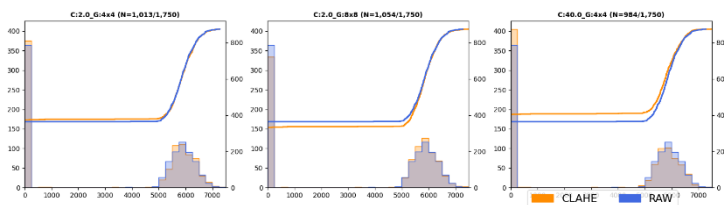
Figuur 48: De histogrammen en cumulatieve plots van de grid search voor de geometrische fout per match, gegeven de categorie Airport. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.

Match Amount Paired Breakdown: DenseResidential



Figuur 49: De histogrammen en cumulatieve plots van de random search voor het aantal matches per matching-inferentie, gegeven de categorie DenseResidential. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is.

Match Amount Paired Breakdown: DenseResidential

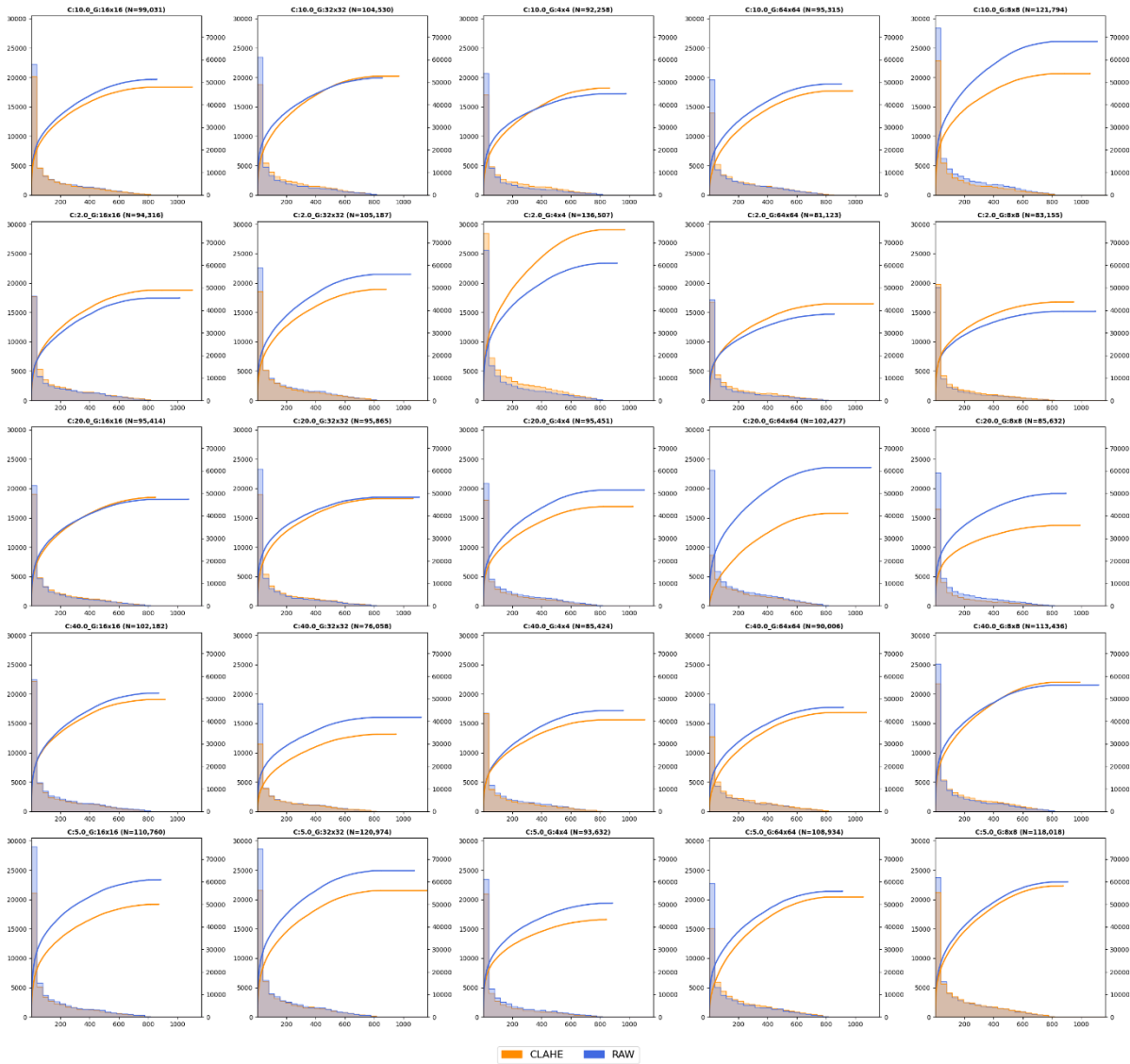


Figuur 50: De histogrammen en cumulatieve plots van de grid search voor het aantal matches per matching-inferentie, gegeven de categorie DenseResidential. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots

Appendix 2: Afbeeldingen

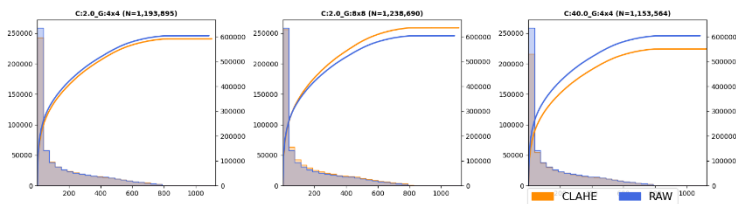
weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is.

Error Paired Breakdown: DenseResidential



Figuur 51: De histogrammen en cumulatieve plots van de random search voor de geometrische fout per match, gegeven de categorie DenseResidential. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.

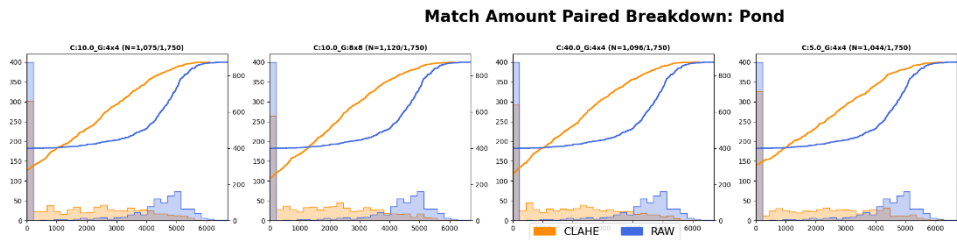
Error Paired Breakdown: DenseResidential



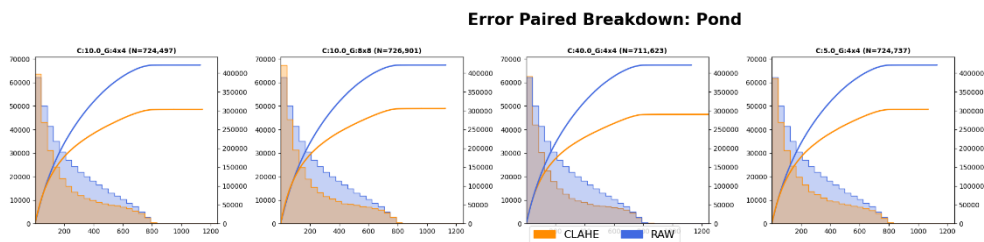
Figuur 52: De histogrammen en cumulatieve plots van de grid search voor de geometrische fout per match, gegeven de categorie DenseResidential. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt

Scenariospecifieke Contrastverbetering voor Satellietcoreregistratie

telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.



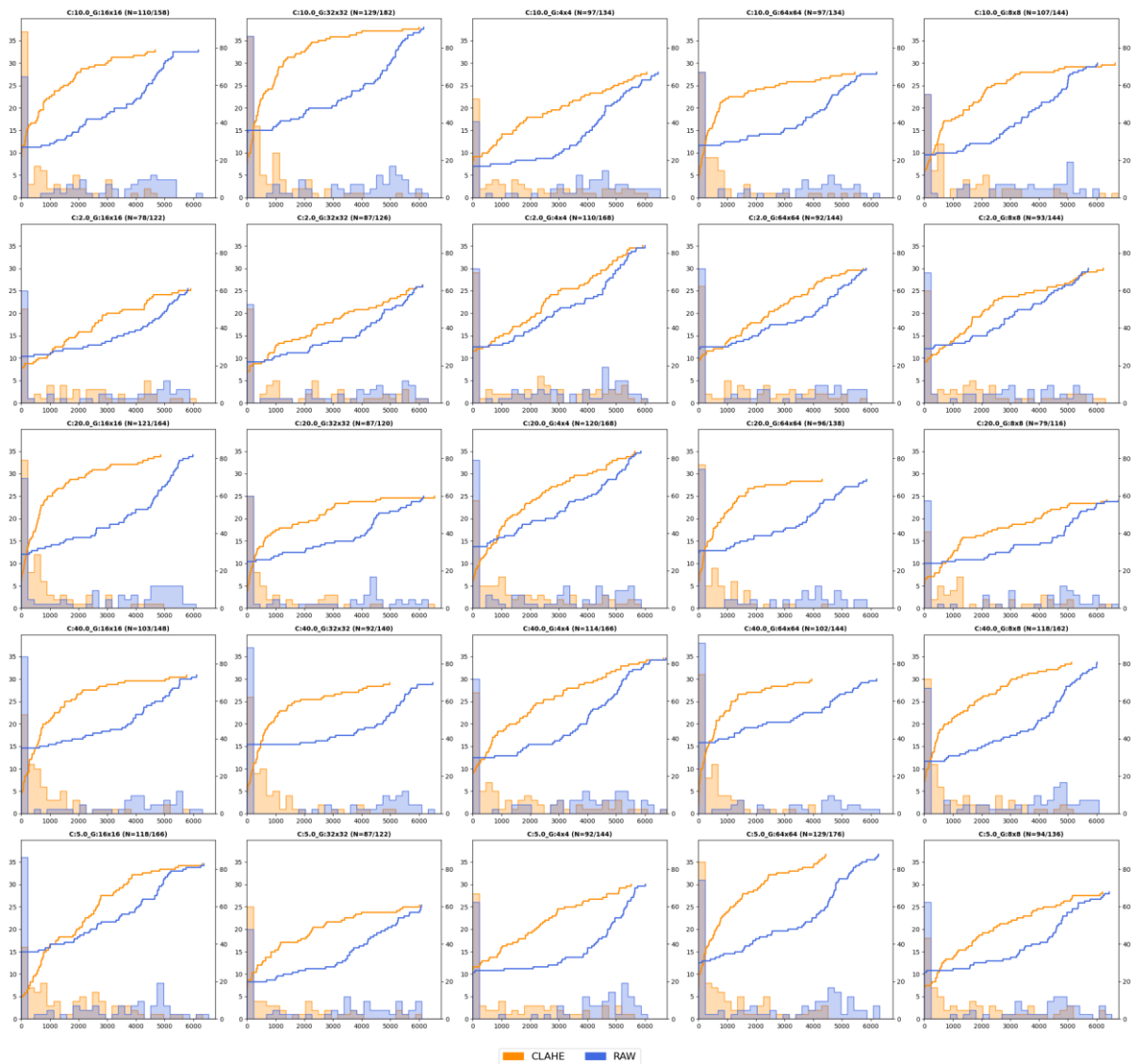
Figuur 53: De histogrammen en cumulatieve plots van de grid search voor het aantal matches per matching-inferentie, gegeven de categorie Pond. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is.



Figuur 54: De histogrammen en cumulatieve plots van de grid search voor de geometrische fout per match, gegeven de categorie Pond. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.

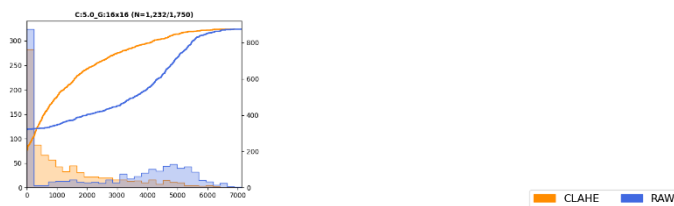
Appendix 2: Afbeeldingen

Match Amount Paired Breakdown: Beach



Figuur 55: De histogrammen en cumulatieve plots van de random search voor het aantal matches per matching-inferentie, gegeven de categorie Beach. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is.

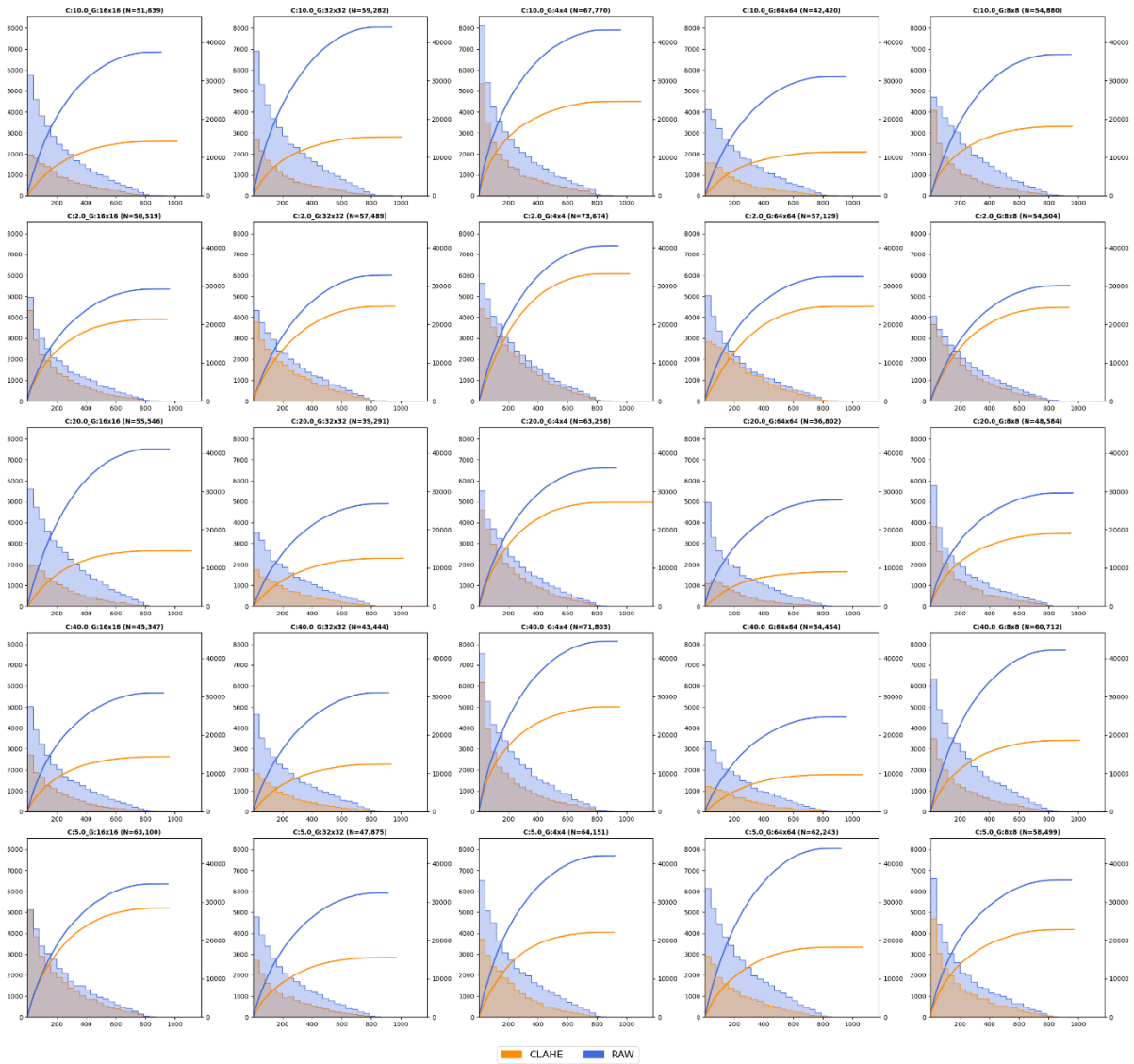
Match Amount Paired Breakdown: Beach



Figuur 56: Het histogram en de cumulatieve plot van de grid search voor het aantal matches per matching-inferentie, gegeven de categorie Beach. Op de x-as staat het aantal matches per matching-inferentie. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven wat de verhouding van het aantal geslaagde inferenties is.

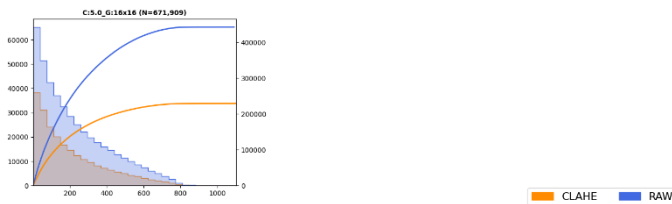
Scenariospecifieke Contrastverbetering voor Satellietcoregistratie

Error Paired Breakdown: Beach



Figuur 57: De histogrammen en cumulatieve plots van de random search voor de geometrische fout per match, gegeven de categorie Beach. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt telkens een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.

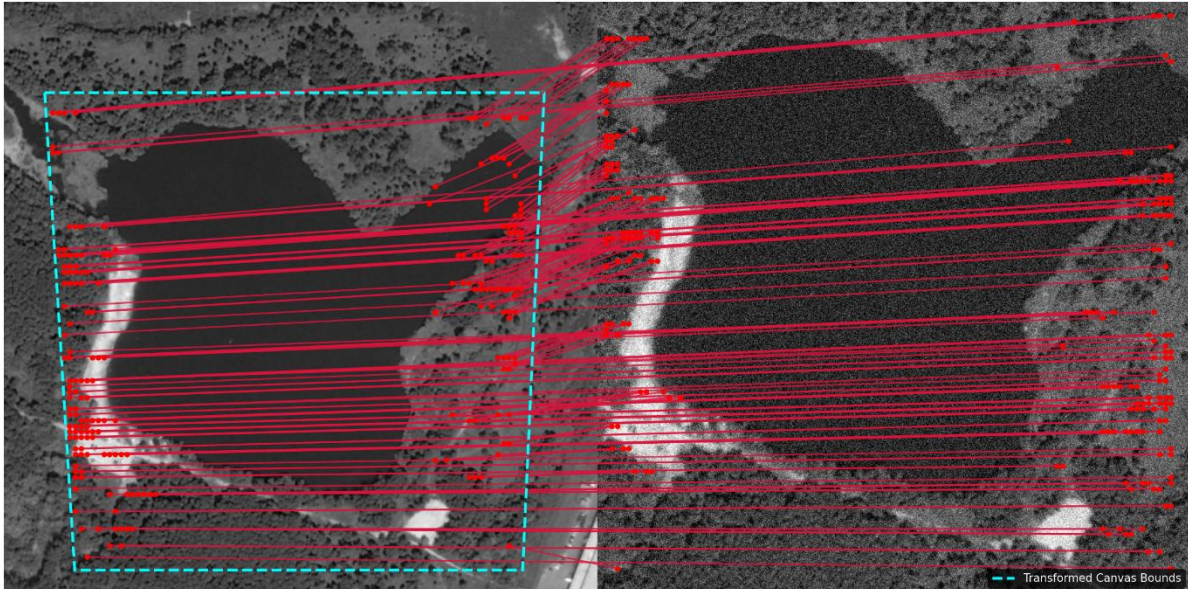
Error Paired Breakdown: Beach



Figuur 58: Het histogram en de cumulatieve plot van de grid search voor de geometrische fout per match, gegeven de categorie Beach. Op de x-as staat de grootte van de fout van de match weergegeven. De linker y-as geeft de waarde van de histogrammen weer. De rechter y-as geeft de waarde van de cumulatieve plots weer. Er wordt een onderscheid gemaakt tussen de inferenties met of zonder de betreffende CLAHE-instelling. In de titel wordt aangegeven hoeveel matches er in de betreffende CLAHE-instelling opgenomen zijn.

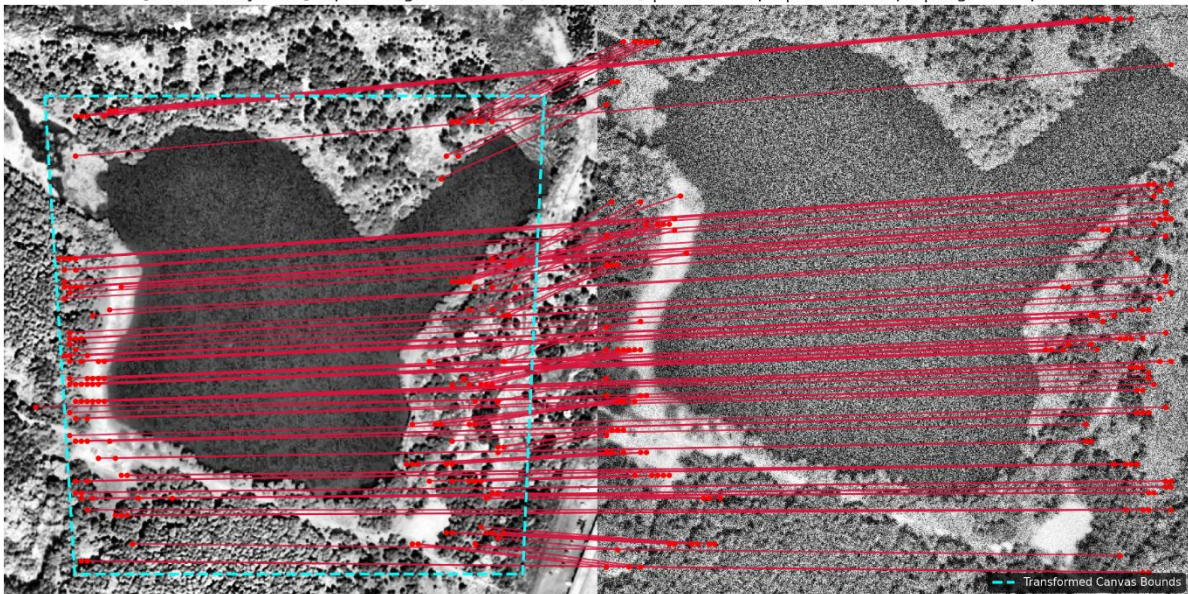
Appendix 2: Afbeeldingen

[Base Grayscale] Top 5% Highest Errors (194 matches) | Min: 629.8px | Max: 794.0px | Avg: 721.3px



Figuur 59: Deze figuur toont bij een matching-inferentie van de categorie Pond de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd geen CLAHE toegepast.

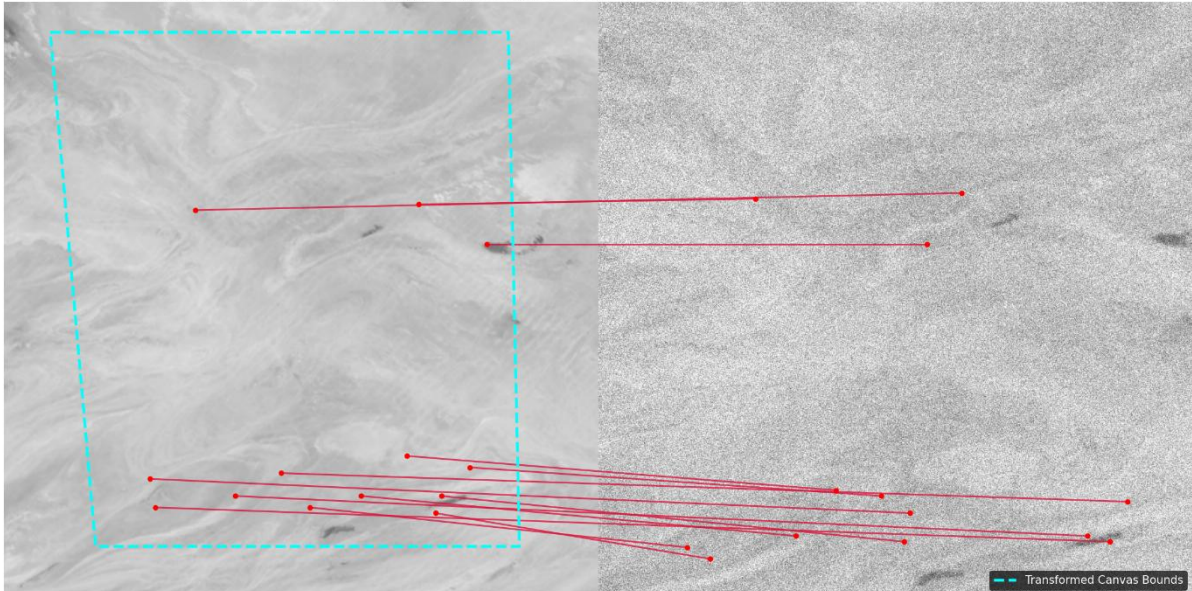
[CLAHE Grayscale] Top 5% Highest Errors (146 matches) | Min: 588.0px | Max: 807.8px | Avg: 684.6px



Figuur 60: Deze figuur toont bij een matching-inferentie van de categorie Pond de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd CLAHE toegepast.

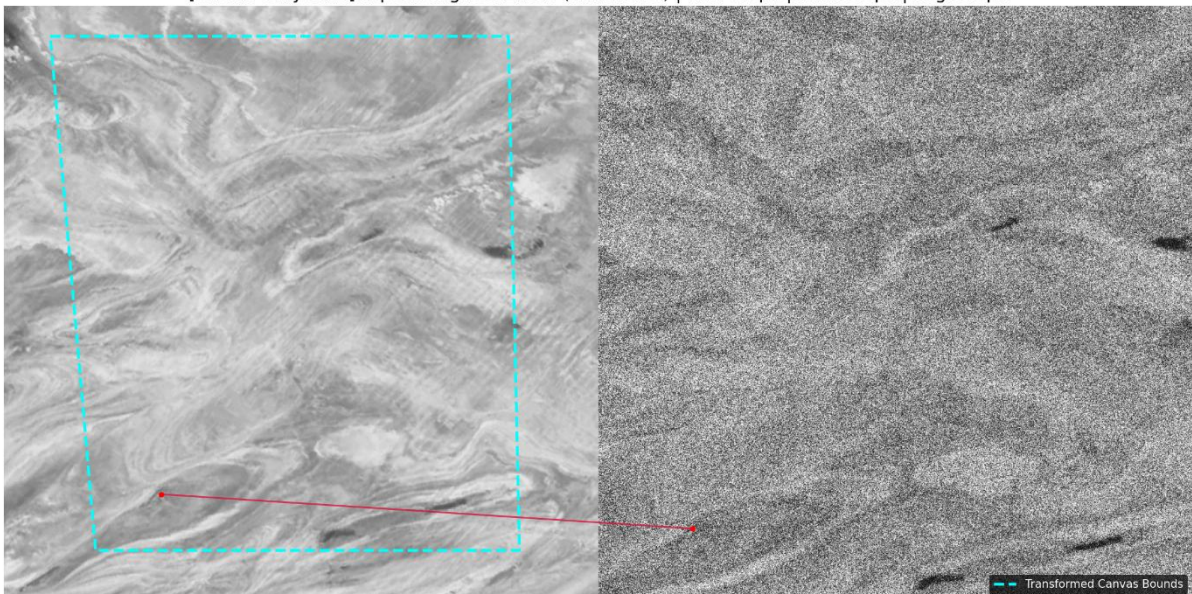
Scenariospecifieke Contrastverbetering voor Satellietcoregistratie

[Base Grayscale] Top 5% Highest Errors (13 matches) | Min: 250.0px | Max: 592.6px | Avg: 358.3px



Figuur 61: Deze figuur toont bij een matching-inferentie van de categorie Desert de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd geen CLAHE toegepast.

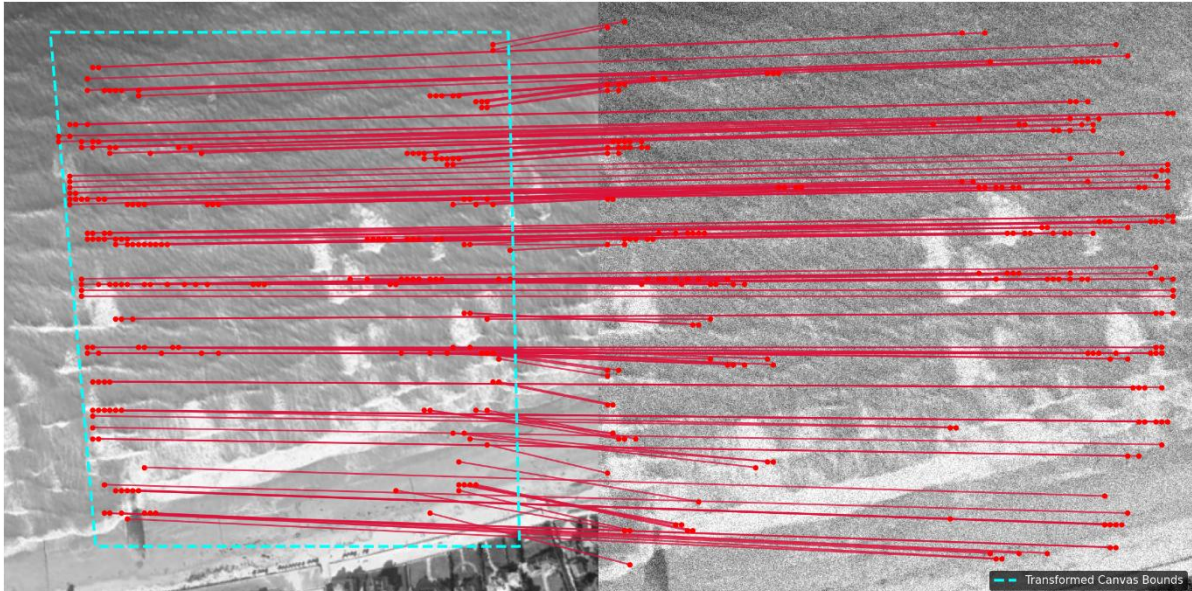
[CLAHE Grayscale] Top 5% Highest Errors (1 matches) | Min: 5.8px | Max: 5.8px | Avg: 5.8px



Figuur 62: Deze figuur toont bij een matching-inferentie van de categorie Desert de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd CLAHE toegepast.

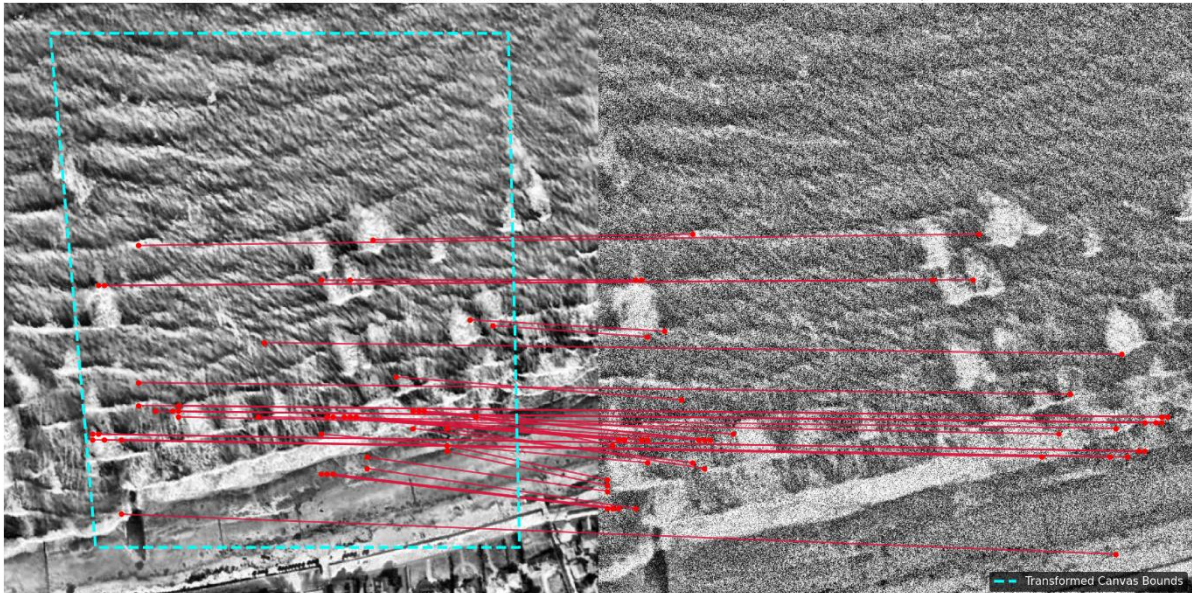
Appendix 2: Afbeeldingen

[Base Grayscale] Top 5% Highest Errors (214 matches) | Min: 430.6px | Max: 790.9px | Avg: 597.4px



Figuur 63: Deze figuur toont bij een matching-inferentie van de categorie Beach de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd geen CLAHE toegepast.

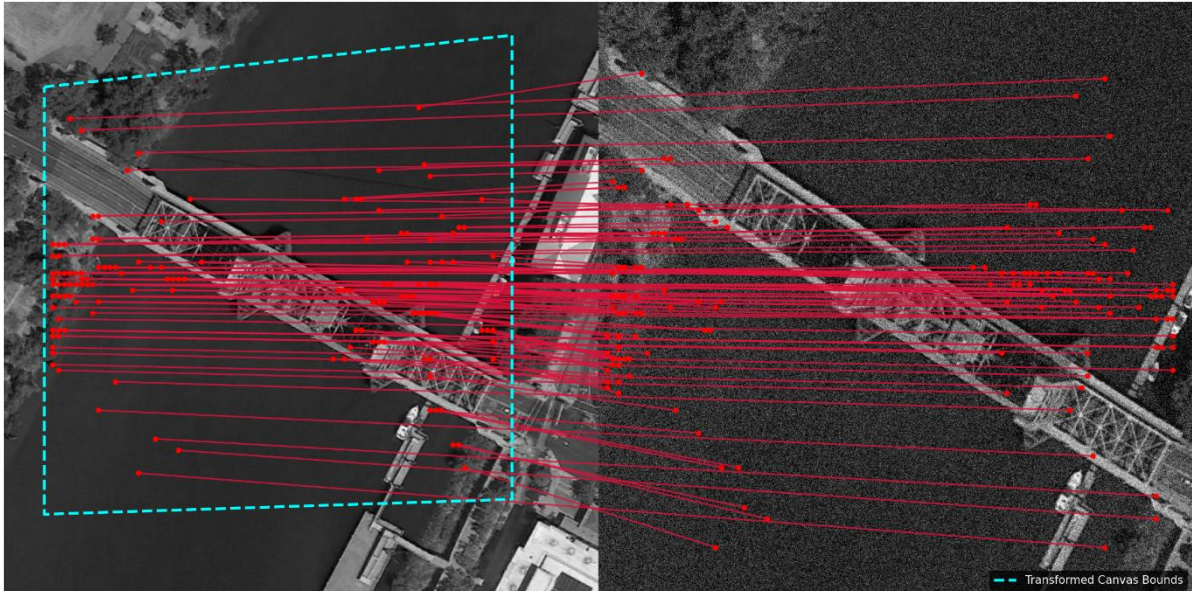
[CLAHE Grayscale] Top 5% Highest Errors (43 matches) | Min: 377.4px | Max: 742.7px | Avg: 534.0px



Figuur 64: Deze figuur toont bij een matching-inferentie van de categorie Beach de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd CLAHE toegepast.

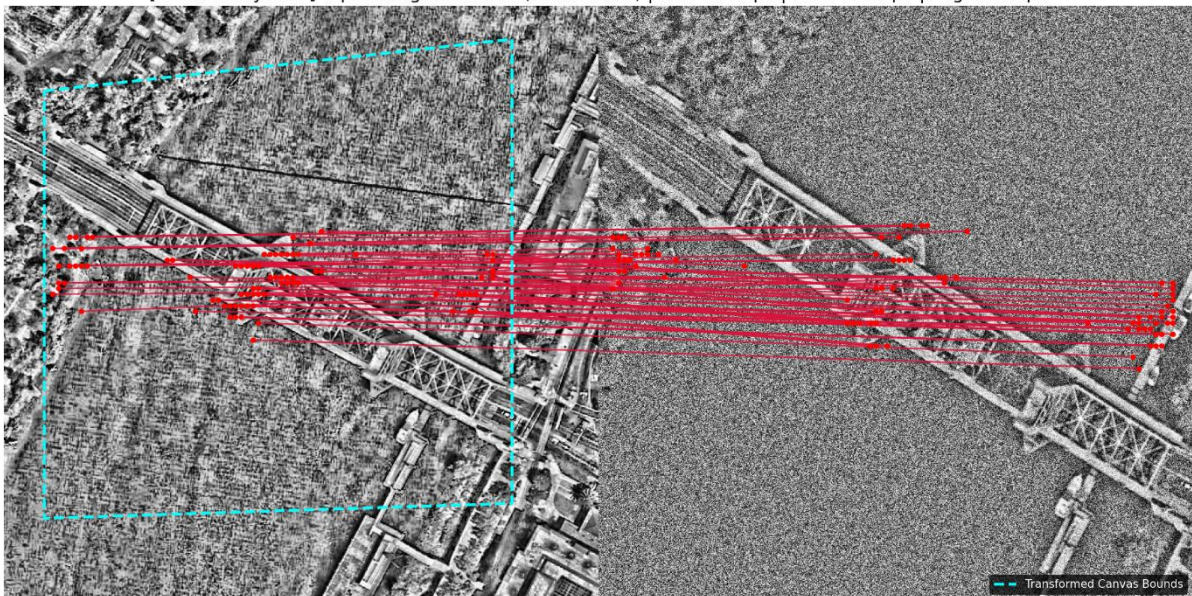
Scenariospecifieke Contrastverbetering voor Satellietcoregistratie

[Base Grayscale] Top 5% Highest Errors (133 matches) | Min: 506.5px | Max: 789.0px | Avg: 593.5px



Figuur 65: Deze figuur toont bij een matching-inferentie van de categorie Bridge de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd geen CLAHE toegepast.

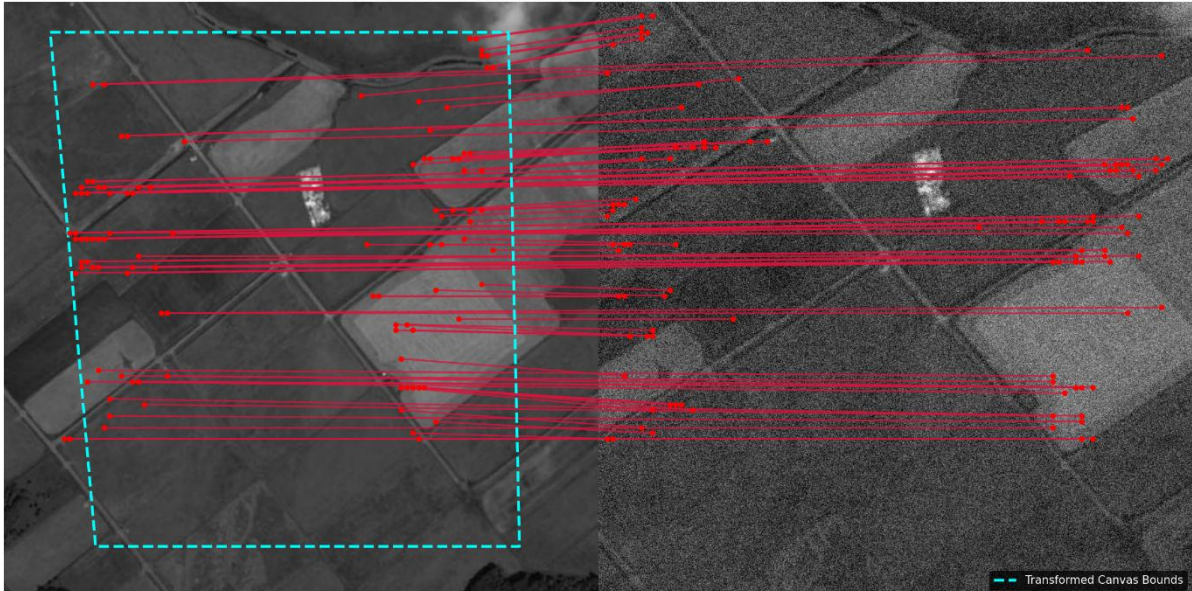
[CLAHE Grayscale] Top 5% Highest Errors (82 matches) | Min: 349.3px | Max: 734.3px | Avg: 406.3px



Figuur 66: Deze figuur toont bij een matching-inferentie van de categorie Bridge de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd CLAHE toegepast.

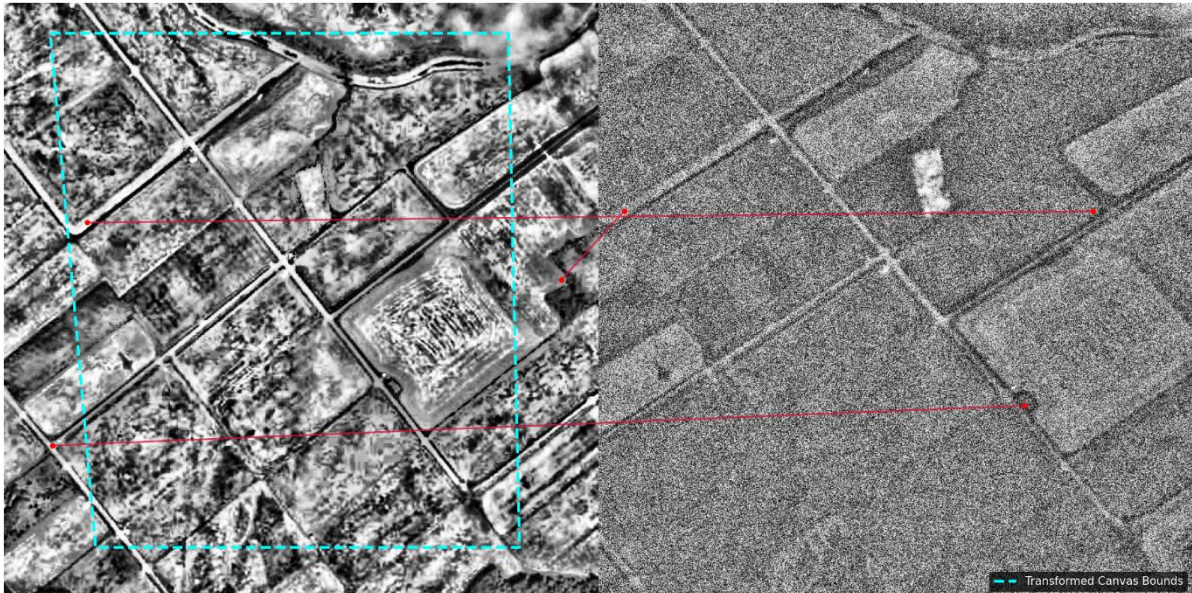
Appendix 2: Afbeeldingen

[Base Grayscale] Top 5% Highest Errors (102 matches) | Min: 518.9px | Max: 782.1px | Avg: 619.3px



Figuur 67: Deze figuur toont bij een matching-inferentie van de categorie Farmland de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd geen CLAHE toegepast.

[CLAHE Grayscale] Top 5% Highest Errors (3 matches) | Min: 654.8px | Max: 890.4px | Avg: 737.8px



Figuur 68: Deze figuur toont bij een matching-inferentie van de categorie Farmland de matches die behoren tot diegene met de 5 procent grootste geometrische fouten. De originele rand van de getransformeerde, rechtse, afbeelding is zichtbaar aan de blauwe omlijning in de originele, linkse, afbeelding. Bij deze matching-inferentie werd CLAHE toegepast.

